Discriminant Learning for Face Recognition

by

Juwei Lu

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of The Edward S. Rogers Sr. Department of Electrical and
Computer Engineering
University of Toronto

Discriminant Learning for Face Recognition

Juwei Lu

Doctor of Philosophy, 2004

Graduate Department of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering

University of Toronto

Abstract

An issue of paramount importance in the development of a cost-effective *face recognition* (FR) system is the determination of low-dimensional, intrinsic face feature representation with enhanced discriminatory power. It is well-known that the distribution of face images, under a perceivable variation in viewpoint, illumination or facial expression, is highly non convex and complex. In addition, the number of available training samples is usually much smaller than the dimensionality of the sample space, resulting in the well documented "*small sample size*" (SSS) problem. It is therefore not surprising that traditional linear feature extraction techniques, such as Principal Component Analysis, often fail to provide reliable and robust solutions to FR problems under realistic application scenarios.

In this research, pattern recognition methods are integrated with emerging machine learning approaches, such as kernel and boosting methods, in an attempt to overcome the technical limitations of existing FR methods. To this end, a simple but cost-effective linear discriminant learning method is first introduced. The method is proven to be robust against the SSS problem. Next, the linear solution is integrated together with

Bayes classification theory, resulting in a more general quadratic discriminant learning method. The assumption behind both the linear and quadratic solutions is that face patterns under learning are subject to Gaussian distributions. To break through the limitation, a globally nonlinear discriminant learning algorithm was then developed by utilizing kernel machines to *kernelize* the proposed linear solution. In addition, two ensemble-based discriminant learning algorithms are introduced to address not only nonlinear but also large-scale FR problems often encountered in practice. The first one is based on the cluster analysis concept with a novel *separability criterion* instead of traditional *similarity criterion* employed in such methods as K-means. The second one is a novel boosting-based learning method developed by incorporating the proposed linear discriminant solution into an improved AdaBoost framework. Extensive experimentation using well-known data sets such as the ORL, UMIST and FERET databases was carried out to demonstrate the performance of all the methods presented in this thesis.

Dedication

To my wife, Julia Q. Gu.

Thanks for love, support, understanding and encouragement.

Acknowledgements

I wish to express my sincere gratitude to all the people who have assisted me during the years of my studies in University of Toronto. First, my most gratefulness goes to my advisor, coauthor and friend, Prof. K.N. Plataniotis, for his professional guidance, constant support and trust. Prof. Plataniotis is well experienced in the areas of image processing, pattern recognition and machine learning. I have obtained numerous valuable ideas from frequent discussions with him. None of the work included here could have happened without his advice. Looking back to my graduate career, it has been a great fortune to have Prof. Plataniotis as my supervisor. Also, I would like to thank my co-supervisor, Dean A. N. Venetsanopoulos, who has been very supportive in many respects over these years. From him, I have not only learned the knowledge of advanced image processing but also the leadership, which could be a huge fortune for my future career.

I would like to thank Prof. Dimitrios Hatzinakos, Prof. James MacLean, and Prof. Pas Pasupathy for their insightful comments and suggestions on my thesis work. I am also grateful to Prof. Nicolas D. Georganas for his kind help to serve as an external thesis appraiser. It was a privilege for me to have had each of them serve in my doctoral committee. In addition, I would like to thank all the members in the DSP group for their warm welcome and support. Special thank goes to my colleagues and friends, Ivan Icasella, Ryan Pacheco and Jessie Wang. I benefited a lot from their friendship and help. It has been really a great pleasure for me to study and work under the nice environment constructed and maintained by the efforts of all the members in the DSP labs.

Special thanks to my Master supervisor and friend, Dr Stan Z. Li for introducing me into the research areas of image processing and pattern recognition, and providing me with many valuable ideas.

Portions of the research in this dissertation use the FERET database of facial images collected under the FERET program [93]. I would like to thank the FERET Technical Agent, the U.S. National Institute of Standards and Technology (NIST) for providing the

FERET database. Also, I would like to thank Dr. Daniel Graham and Dr. Nigel Allinson for providing the UMIST face database, and thank AT&T Laboratories Cambridge for providing the ORL face database.

Finally, my sincere thanks go to all the members of my family. I share with them every piece of achievements that I have obtained.

Partial works in Chapter 3 have been published. Reprinted from Publication [74] with permission from IEEE.

Partial works in Chapter 4 have been published. Reprinted from Publication [75] with permission from Elsevier.

Partial works in Chapter 5 have been published. Reprinted from Publication [73] with permission from IEEE.

Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Description	Page #
AMLLM:	A Mixture of Locally Linear Models	55
AdaBoost:	Adaptive Boost	5
AdaBoost.M1:	AdaBoost's Multi-class Extention Version 1	103
AdaBoost.M2:	AdaBoost's Multi-class Extention Version 2	101
BCS:	Between-class Scatter	77
B-JD-LDA.A:	The algorithm of boosting JD-LDA with $A_t(p,q)$	109
B-JD-LDA.$\hat{A}$:	The algorithm of boosting JD-LDA with $\hat{A}_t(p,q)$	109
CRR:	Correct Recognition Rate	11
CER:	Classification Error Rate	16
CMD:	Cumulative Margin Distribution	102
D-LDA:	Direct Linear Discriminant Analysis	19
DF-LDA:	JD-LDA followed by F-LDA	31
D-QDA:	Direct QDA	41
D-WNC:	Direct WNC	46
FR:	Face Recognition	1
FERET:	Facial Recognition Technology	10
FRVT:	Face Recognition Vendor Tests	11
F-LDA:	Fractional-step LDA	18
GDA:	Generalized Discriminant Analysis	56
gClassifier:	The general classifier produced by the JD-LDA learner	104
HCF:	Hierarchical Classification Framework	79
HCF-Kmean:	HCF based on K-means	81
HCF-MSC:	HCF based on MSC	81
HTSP:	Hard to separate pairs	98

Acronym	Description	Page #
ICA:	Independent Component Analysis	3
JD-LDA:	Juwei's D-LDA	4
KPCA:	Kernel PCA	56
KDDA:	Kernel Direct Discriminant Analysis	5
KM:	Kernel Mahalanobis	56
KICA:	Kernel ICA	56
LDA:	Linear Discriminant Analysis	4
LDD:	Learning Difficulty Degree	91
LSFD:	Large-scale Face Databases	73
MAP:	Maximum A Posteriori	42
MFA:	Mixture of Factor Analyzers	74
MSC:	Maximally Separable Cluster	76
MLSes:	Mixtures of Linear Subspaces	83
NCC:	Nearest Center Classifier	99
NNC:	Nearest Neighbor Classifier	77
ORL:	Olivetti Research Laboratory	14
PCA:	Principle Component Analysis	9
PCDD:	Pairwise Class Discriminant Distribution	98
QDA:	Quadratic Discriminant Analysis	40
RBF:	Radial Basis Function	36
RDA:	Regularized Discriminant Analysis	41
RD-QDA:	Regularized Quadratic Discriminant Analysis	4
SSS:	Small Sample Size	3
SVM:	Support Vector Machines	3
SPR:	Statistical Pattern Recognition	23
STD:	Standard Deviations	48

Acronym	Description	Page #
S-JD-LDA:	The Stand-alone JD-LDA Method (without boosting)	114
UMIST:	University of Manchester Institute of Science and Technology	14
WNC:	Weighted Nearest Center	41
YD-LDA:	Yang's D-LDA	26

Important Notations

$\mathcal{G}$	the entire evaluation database.
$\mathcal{G}_1$, $\mathcal{G}_2$	two evaluation databases generalized from the FERET database.
$\mathcal{Z}$	the training set of face images, $\mathcal{Z} \subset \mathcal{G}$.
$\mathcal{Z}_i$	the training face images belonging to the ith class, $\mathcal{Z}_i \subset \mathcal{Z}$.
C, C_i	the number of classes and the number of training examples per class.
L	the averaged number of training examples per class, $L = \frac{1}{C}\sum_{i=1}^{C} L_i$.
N	the number of training examples in $\mathcal{Z}$.
$\mathcal{Q}$	the test data set, $\mathcal{Q} = \mathcal{G} - \mathcal{Z}$.
$\mathbf{z}_{ij}$, y_{ij}	the face image example and its label.
$\mathbb{Y}$	the label space, having $y_{ij} \in \mathbb{Y} = \{1, \cdots, C\}$.
$\mathbb{R}^J$, J	the J-dimensional real space, $J = \dim(\mathbf{z}_{ij})$.
R_n	the correct recognition rate for rank n.
$\bar{\mathbf{z}}_i$	the center of the ith class $\mathcal{Z}_i$, $\bar{\mathbf{z}}_i = \frac{1}{L_i}\sum_{j=1}^{L_i} \mathbf{z}_{ij}$.
$\mathbf{S}_{cov}$	the covariance matrix of the training set $\mathcal{Z}$.
$\mathbf{S}_b$, $\mathbf{S}_w$	the between- and within-class scatter matrices of the training set $\mathcal{Z}$.
Ψ, ψ_i	a set of M feature basis vectors, $\Psi = [\psi_1, \ldots, \psi_M]$.
$\mathbf{y}$	the feature representation of $\mathbf{z}$, $\mathbf{y} = \Psi^T \mathbf{z}$.
$\mathbb{R}^M$, M	the feature space, $\mathbf{y} \in \mathbb{R}^M$, and its dimensionality.
$\mathcal{L}(\cdot)$	the JD-LDA feature extractor, having $(\Psi, \{\bar{\mathbf{z}}_i\}_{i=1}^{C}) = \mathcal{L}(\mathcal{Z})$.
η	the regularization parameter in JD-LDA.
$w(d)$	the weighting function used in DF-LDA.
$\mathbf{H}$	the normalized complement space of null-space of $\mathbf{S}_b$.
$\hat{\Sigma}_i(\lambda, \gamma)$	the regularized sample covariance matrix estimate of class i.
(λ, γ)	a pair of regularization parameters in RD-QDA.
$\mathbb{F}$	the kernel space.

$\phi(\mathbf{z})$	the kernel mapping function, $\phi(\mathbf{z}) \in \mathbb{F}$.
$k(\mathbf{z}_i, \mathbf{z}_j)$	the kernel dot product function.
$\tilde{\mathbf{S}}_b$, $\tilde{\mathbf{S}}_w$	the between- and within-class scatter matrices in $\mathbb{F}$.
$\mathbf{K}$	the $N \times N$ kernel matrix.
$\nu(\phi(\mathbf{z}))$	the $N \times 1$ kernel vector of the sample $\mathbf{z}$.
Θ	the $M \times N$ matrix mapping $\nu(\phi(\mathbf{z}))$ to a low-dim. feature space.
S_t	the the total intra-cluster between-class scatter.
Ω_k	the k-th MSC.
h_t	the gClassifier produced by the JD-LDA learner at the tth iteration.
$D_t(\mathbf{z}_{ij})$	the sample distribution defined over $\mathcal{Z}$.
ϵ_t	the training error of h_t.
$\varrho_{h_f}(\mathbf{z}, y)$	the margin of example $(\mathbf{z}, y)$ obtained by the composite gClassifier h_f.
$P_{\mathcal{Z}}$	the cumulative margin distribution.
B	the set of all mislabels.
$\Upsilon(B)$	the mislabel distribution defined on B.
A_t, $\hat{A}_t$	the PCDD version 0 and 1.
$\hat{\mathbf{S}}_{b,t}$, $\hat{\mathbf{S}}_{w,t}$	the weighted variants of $\mathbf{S}_b$ and $\mathbf{S}_w$ used in B-JD-LDA.
$\hat{\epsilon}$	the pseudo-loss used in AdaBoost.M2.
$\hat{D}_t(\mathbf{z}_{ij})$	the pseudo sample distribution.
$\mathcal{R}_t$	the subset $\mathcal{R}_t \subset \mathcal{Z}$ generalized at tth iteration to train $\mathcal{L}(\mathcal{R}_t)$.
r	the number of training examples per class chosen to form $\mathcal{R}_t$.
$\rho_l(r)$	the LDD used to express the weakness of a learner.
$\rho_t(L)$	the LDD used to express the difficult extent of a learning task.
$\mathbf{R}(r)$	the generalization loss for model selection.
$\bar{e}$	the average CER.
T	the iteration number of boosting.

Chapter 1

Introduction

1.1 Motivation

Face recognition (FR) has a wide range of applications from biometric identity authentication to human-computer interaction. Table 1.1 lists some of these applications

Table 1.1: Some applications of face recognition [16, 133].

Areas	Examples of Applications
Biometrics	Drivers Licenses, Entitlement Programs, Smart Cards
	Immigration, National ID, Passports, Voter Registration
	Welfare Fraud, Airline Industry, Bank Industry
Information Security	Desktop Logon, Secure Trading Terminals
	Application Security, Database Security, File Encryption
	Intranet Security, Internet Access, Medical Records
Law Enforcement and Surveillance	Advanced Video Surveillance, Portal Control
	Postal-Event Analysis, Face Reconstruction from Remain
	Shoplifting and Suspect Tracking and Investigation
Access Control	Facility Access, Vehicular Access, PDA and Cell phone Access
Others	Human-Computer Interaction, Information Retrieval

During the past two decades, numerous FR algorithms have been proposed, and detailed surveys of the development in this area can be found in [16,20,35,102,119,122, 133]. Although the progress made has been encouraging, FR has also turned out to be a very difficult endeavor [119]. The key technical barriers are summarized below :

1. **Immense variability of 3D face object appearance**. Shape and reflectance are intrinsic properties of a face object, but a 2D image of the 3D face appearance is a function of several additional factors, including illumination, facial expression, pose of face object, and various imaging parameters such as aperture, exposure time, lens aberrations and sensor spectral response. All of these factors are confounded in the image data, so that "the variations between the images of the same face due to illumination and viewing direction are almost always larger than the image variations due to changes in face identity" [87]. For the reason, extracting the intrinsic information of the face objects from their respective images is a demanding discriminant task.

2. **Highly non convex and complex pattern distribution**. From the viewpoint of the so-called appearance-based learning [89,119], it intuitively can be imagined that existing in the high-dimensional real space, there is a vast convoluted face manifold, where all noise free face images lie. The manifold, which accounts for the immense variations of face pattern appearance, is commonly believed to be highly non convex and complex [9,119]. The issue of how to tackle the manifold is central to the appearance-based FR approach. During the past two decades, although significant research efforts have been made to address the issue, it has been turned out to be a very difficult task.

3. **High dimensionality *vs* small size of learning samples**. For the purpose of accurate face recognition, the acquired image samples should be of sufficient resolutions. For example, a canonical example used in FR tasks is an image of size

(112 × 92), which exists in a 10304-dimensional real space. Nevertheless, the number (L) of examples per class available for learning is usually much smaller than the dimensionality of the sample space, *e.g.* $L \leq 10$ in most cases. This produces the so-called *small sample size* (SSS) problem, which significantly degrades the performance of the feature extractors and the classifiers, particularly those operating in a supervised learning mode. In addition, the computational cost is very high so that some calculations are intractable to operate directly in the high-dimensional image space.

In summary, FR is far from being a solved problem and presents researchers with challenges and opportunities. Despite a difficult task, the rapid advancement in learning theories gives us much reason to be optimistic. For example, some machine learning techniques recently emerged, such as Independent Component Analysis (ICA) [5, 8, 49, 65, 101], Kernel Machines [3, 6], Support Vector Machine (SVM) [123] and Boosting [28], have been shown to be very powerful and promising in solving some pattern recognition problems that cannot be easily approached by traditional methods. In this thesis, we wish to explore the intersected area of traditional discriminant analysis methods and these lately advanced learning theories, and thereby develop some novel discriminant learning algorithms that can effectively merge their advantages to conquer the technical barriers encountered in FR tasks.

1.2 Contributions

The contributions of this research can be outlined by the tree shown in Fig.1.1, which is explained as follows:

- First, we proposed a novel regularized Fisher's discriminant criterion, which is particularly robust against the SSS problem compared to the traditional one used

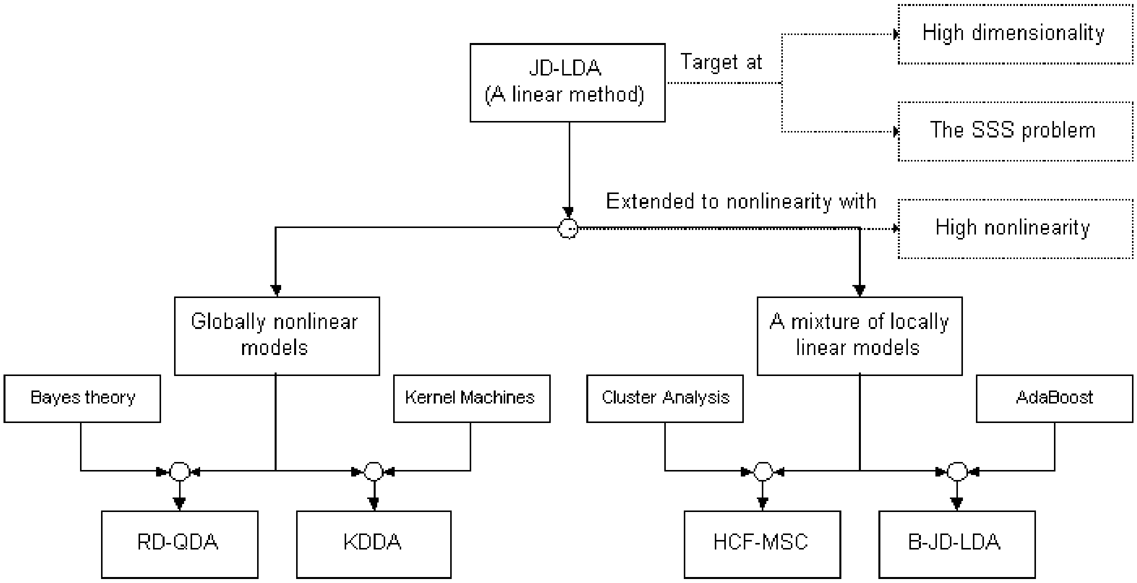

Figure 1.1: The contribution tree, targeted at conquering the three technical barriers in face recognition: high dimensionality of input images, small-size training sample, and highly non convex pattern distribution.

in *Linear Discriminant Analysis* (LDA). Based on the new criterion, a linear feature extraction method, called "JD-LDA" was then developed to effectively capture low-dimensional, intrinsic discriminant features of face patterns from a high-dimensional, small-size training sample. Since the separability criteria used in traditional LDA algorithms are not directly related to their classification ability in the output space, an iterative rotation strategy of linear space was introduced to further optimize the low-dimensional face feature representation obtained from the JD-LDA approach.

- LDA assumes that each pattern class is subjected to a Gaussian distribution with identical covariance structure. To relax the assumption, the concept of JD-LDA was discussed and extended under the optimal Bayes classification framework. This leads to a more general quadratic discriminant learning method, called "RD-QDA", which can deal with classes subject to any Gaussian distribution. As a result, not only linear but also quadratic decision boundaries can be constructed by RD-QDA during the classification process. Meanwhile, the SSS problem that becomes worse due to increased algorithm complexity in quadratic learning is addressed by an extended regularization scheme.

- Both JD-LDA and RD-QDA were developed using the Gaussian assumption. In addition to this, a globally nonlinear discriminant learning algorithm, called "KDDA" was proposed by *kernelizing* or *nonlinearizing* JD-LDA with the kernel machine technique. As a result, nonlinear decision boundaries far more complicated than quadratic can be generalized for classification in high-dimensional, SSS settings.

- Instead of seeking a global but difficult and complex solution, we proposed two novel ensemble-based discriminant learning methods based on the principal of "divide and conquer". The first method, called "HCF-MSC" is based on a novel cluster analysis criterion developed from the viewpoint of classification. Using the criterion, the HCF-MSC method generalizes a mixture of locally linear JD-LDA models, based on which a hierarchical classification framework was then introduced to effectively address nonlinear, large-scale FR problems.

- Finally, we presented another ensemble-based learning method, called "B-JD-LDA" by incorporating JD-LDA into the boosting framework. The machine learning technique, AdaBoost, has been shown to be particularly robust in preventing overfitting and reducing generalization error. However, it is generally believed that the technique is not suited to a stable learner, for instance LDA. To this end, some novel concepts and theories regarding weakness analysis of a learning algorithm, generalization loss, and selection of good training examples have been introduced in the design of B-JD-LDA. As a result, the performance of JD-LDA is significantly boosted by B-JD-LDA, which is also shown by experimentation to outperform all other FR methods proposed in this thesis.

1.3 Organization

The rest of this thesis is organized as follows. First, a brief background review of FR research is given in Chapter 2. In Chapter 3, the SSS problem, one of the biggest challenges faced in the FR research is discussed in the context of the appearance-based learning paradigm. Then, a simple but effective linear solution, the JD-LDA method is introduced. In Chapter 4, the relationships between the SSS problem, regularization and LDA are further analyzed together with the optimal Bayes classification theory. Following that, the quadratic discriminant learning method, RD-QDA is proposed under the Gaussian framework. In Chapter 5, the kernel machine technique is introduced and discussed. Based on the technique, a global nonlinear FR method, KDDA, is developed by *kernelizing* JD-LDA. In addition, from the viewpoint of a mixture of locally linear models, we propose two ensemble-based discriminant learning methods using cluster analysis and boosting techniques respectively in Chapter 6 and 7. JD-LDA is used as the base learner in both of the two methods. Finally, Chapter 8 summarizes conclusions and provides directions for future research.

Chapter 2

Background Review

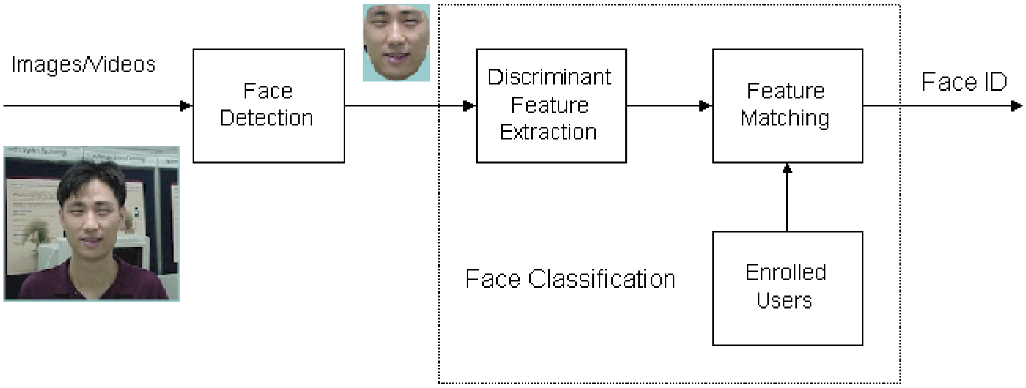

Figure 2.1: A general framework for automatic face recognition.

Taking as input images or videos of face objects, a general framework for automatic *face recognition* (FR) is as shown in Fig.2.1, with the *face detection* and the *face classification* forming its two central modules. *Face detection* provides information about the location and scale of each located face object. In the case of video, the found faces may be tracked. In many applications such as highly accurate face recognition and synthesis, an additional *face alignment* part is often included in *face detection*. In the part, some specific facial components, such as eyes, nose, mouth and facial outline are further located, and then the input face image is aligned and normalized in geometry and photometry. In *face classification*, features useful for distinguishing between different persons are extracted from the normalized face, and the extracted feature vector is then matched against those of known faces, outputting the identity of the input face when a match is found with a sufficient confidence or as an unknown face otherwise.

In the FR community, the research of the two modules, *face detection* and *face classification*, are often conducted separately. In this work, our focus is on *face classification*. Thus, through the thesis we actually implement *partially* automatic FR algorithms, which take as input the localized face images as shown in Fig.2.1. In general, the partially automatic FR algorithms consist of two processing parts, *discriminant feature extraction* and *feature matching* (see the *face classification* module depicted in Fig.2.1). In the past, most face research efforts have been made to address the issue of feature extraction, which has been shown to be the key to the particular task of face recognition due to the technique barriers summarized in Section 1.1. The main goal of feature extraction is to develop techniques that can generalize a low-dimensional feature representation intrinsic to face objects with enhanced discriminatory power from low-level image information, such as intensity, color and edges. Often the approaches used for the purpose are classified into two classes: (1) *geometric feature-based approaches*, and (2) *appearance-based approaches*.

2.1 Geometric Feature-based Approach

The geometric (also called *shape*) feature-based approach (see *e.g.* [14, 34, 50, 58, 102, 128]) is based on the traditional computer vision framework [77], whose central issue is to abstract away from raw image data (*i.e.* pixel values) to higher level, invariant representations such as 3D shape. Under the framework, facial characteristics such as eyes, nose, mouth and chin are required to be accurately located and marked at first. Properties and relations (*e.g.* areas, distances, angles) between the features are then used as descriptors of face patterns for recognition. Using this approach, Kanade built the first face recognition system in the world [50]. Although this class of methods are computationally attractive, efficient in achieving dimensionality reduction, and relatively insensitive to variations in illumination and viewpoint, they rely heavily on the accurate detection of facial features. Unfortunately, facial feature detection and measurement techniques developed to date have not been reliable enough to cater to this need [21]. Also, geometric information only is insufficient for face recognition.

2.2 Appearance-based Approach

In the past twenty or so years, great progress has been made in FR research. To a great extent, this can be attributed to advances in appearance-based approaches (see *e.g.* [5,7,15,38,61,64,73,74,86,92,119,120,132]). In contrast with the geometric feature-based approach, the appearance-based approach generally operates directly on raw image data and processes them as 2D holistic patterns to avoid difficulties associated with 3D modeling, and shape or landmark detection. Consequently, this class of methods tends to be easier to implement, more practical and reliable as compared to the geometric feature-based methods [14,119].

The source of the appearance-based approach can be backdated to the influential Eigenfaces method [120], presented by Turk and Pentland in 1991. In [120], a low-dimensional subspace of the original face image space, called “face space”, is constructed to best account for the variations of the face objects. The face space is spanned by a number of Eigenfaces [110] derived from a set of training face images by using *principal component analysis* (PCA) or *Karhunen-Loeve transform* [33]. A face image is linearly mapped to the face space, and then the obtained low-dimensional projection vector is used to represent the face object. Compared to the over-abstract facial features, a great deal of experiments have shown that such subspace features are more salient and informative for recognition [14, 119]. The Eigenfaces method looks simple from the viewpoints of both theory and implementation. Nevertheless, it started the era of the appearance-based approach to visual object recognition [119]. Thereafter, algorithms developed with the approach have almost dominated FR research (see *e.g.* [16,35,119,122,133] for a detailed survey). Due to its huge influences, [120] was awarded to be the "Most influential paper of the decade" at the 2000 IAPR Workshop on Machine Vision Applications.

Based on the above reasons, all the research presented in this thesis were conducted in the context of the appearance-based learning paradigm.

2.3 Face Databases and FR Evaluation Design

In addition to the advancement of the *feature extraction* and *matching* algorithms, the development of FR research depends on the availability of other two factors: (i) a large and representative database of face images, and (ii) a method for evaluating the performance of FR algorithms. In this section, we first briefly review the history of FR evaluations, and then discuss how these two issues were addressed in the subsequent work presented in this thesis.

2.3.1 Facial Recognition Technology (FERET) Program

To date, the so-called *Facial Recognition Technology* (abbreviated as FERET) program incepted in 1993 has made a significant contribution to the evaluation of FR algorithms by building the FERET database and the evaluation protocol, which have become de facto standards in the FR world [93,94]. Based on the FERET program, the first overall competition of FR algorithms was launched by P.J. Phillips *et al.* , Army Research Laboratory, USA in August 1994 [93]. Following that, two extensive competitions took place in March of 1995 and September of 1996 respectively [93,94]. In these competitions, an algorithm was given two sets of images: the *target* set and the *query* set. The *target set* is a set of known facial images, while the *query* set consists of unknown facial images to be identified. Furthermore, multiple *gallery* and *probe* sets can be constructed from the *target* and *query* sets respectively. For a given gallery and probe pair, the FR performance measure, *correct recognition rate* (CRR) is computed by examining the similarity between the two sets of images. Almost all the FR systems that attended the Sept96 evaluation adopted the appearance-based approach. PCA and LDA were the two most popular techniques used in these FR systems. Methods based on the two techniques [86, 120, 132] dominated among the top performers of the evaluations. The trio competitions were highly successive. This further leads to the regular Face Recognition Vendor Tests (FRVT) [37], which were developed to provide independent government evaluations of commercially available and mature prototype FR systems. The information obtained from these evaluations are used to assist U.S. Government and law enforcement agencies in determining where and how facial recognition technology can best be deployed. The latest FRVT 2002 reports can be found in its web site: http://www.frvt.org.

2.3.2 Face Databases

The FERET database can be considered the largest, most comprehensive and representative face database, provided to FR researchers to advance the state of the art in face recognition [93, 94]. Since the FERET program incepted in 1993, a total of 14051 face images of 1209 persons have been incorporated into the database. These images cover a wide range of variations in viewpoint, illumination, facial expression/details, acquisition time, races and others.

As mentioned earlier, through the thesis we only implement partially automatic FR algorithms, which require that the centers of the eyes are available during a preprocessing stage for the purpose of alignment and normalization [94]. Currently, only 3817 face images of 1200 persons in the FERET database are provided along with the coordinate information of eyes, nose tip and mouth center. Thus, we first extracted all the 3817 images to form a data set denoted as $\mathcal{G}_0$, which was the biggest evaluation database to be used in the experiments reported here. Also, it is required to study the sensitivity of the CRR measure to the number of training examples per subject in many simulations such as those depicted in Sections 4.5, 7.5. To this end, two more evaluation databases were generalized from $\mathcal{G}_0$ in the sequence. The first evaluation database denoted as $\mathcal{G}_1$ was formed by choosing in the set $\mathcal{G}_0$ all (606) images of 49 subjects with each subject having at least ten images. Similarly, the second evaluation database denoted as $\mathcal{G}_2$ (including $\mathcal{G}_1$) was constructed by choosing in $\mathcal{G}_0$ all (1147) images of 120 subjects with at least six images per subject. The details of the images included in $\mathcal{G}_0$, $\mathcal{G}_1$ and $\mathcal{G}_2$ are depicted in Table 2.1, where the naming convention for the imagery categories can be found in Table 2.2.

Table 2.1: No. of images divided into the standard FERET imagery categories in evaluation databases, and the pose angle (degree) of each category.

Category	fa	fb	ba	bj	bk	ql	qr	rb	rc	sum
$\mathcal{G}_0$	1604	1360	200	200	200	81	78	32	62	3817
$\mathcal{G}_1$	275	166	4	4	4	43	42	26	42	606
$\mathcal{G}_2$	567	338	5	5	5	68	65	32	62	1147
PoseAngle	0	0	0	0	0	-22.5	+22.5	10	-10	-

The original images in the FERET database are raw face images that include not only the face, but also some data irrelevant for the FR task, such as hair, neck, shoulder and background as shown in Fig.2.2:Left. In a great deal of previously reported FR works, only simple preprocessing operations such as cropping and resizing are applied to the raw images. As a consequence, irrelevant facial portions were retained, and presented to the FR algorithms together with the face portion. Recently it has been quantitatively shown that inclusion of these irrelevant facial portions may mislead the systems, resulting in in-

Table 2.2: Naming convention for the FERET imagery categories [93, 94].

Category	Pose Angle	Description
fa	0	Regular facial expression
fb	0	Alternative facial expression
ba	0	Frontal “b” series
bj	0	Alternative expression to ba
bk	0	Different illumination to ba
ql	-22.5	Quarter left
qr	+22.5	Quarter right
rb	+10	Random images
rc	-10	Random images

correct evaluations [18]. To this end, we follow the preprocessing sequence recommended in the FERET protocol [94], which includes four steps: (1) images are translated, rotated and scaled (to size 150×130) so that the centers of the eyes are placed on specific pixels; (2) a standard mask as shown in Fig.2.2:Middle is applied to remove the nonface portions; (3) histogram equalization is performed on the non masked facial pixels; (4) face data are further normalized to have zero mean and unit standard deviation. Fig.2.2:Right and Fig.2.3 depict some examples after the preprocessing sequence was applied. For computational purposes, each image is finally represented as a column vector of length $J = 17154$ prior to the recognition stage.

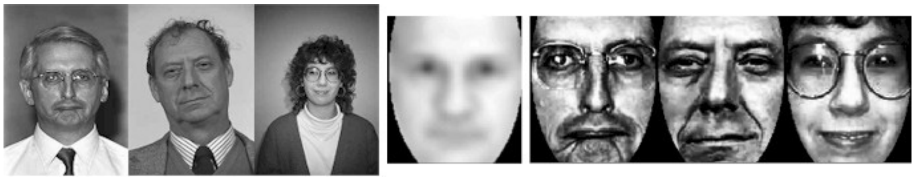

Figure 2.2: **Left**: Original samples in the FERET database. **Middle**: The standard mask. **Right**: The samples after the preprocessing sequence.

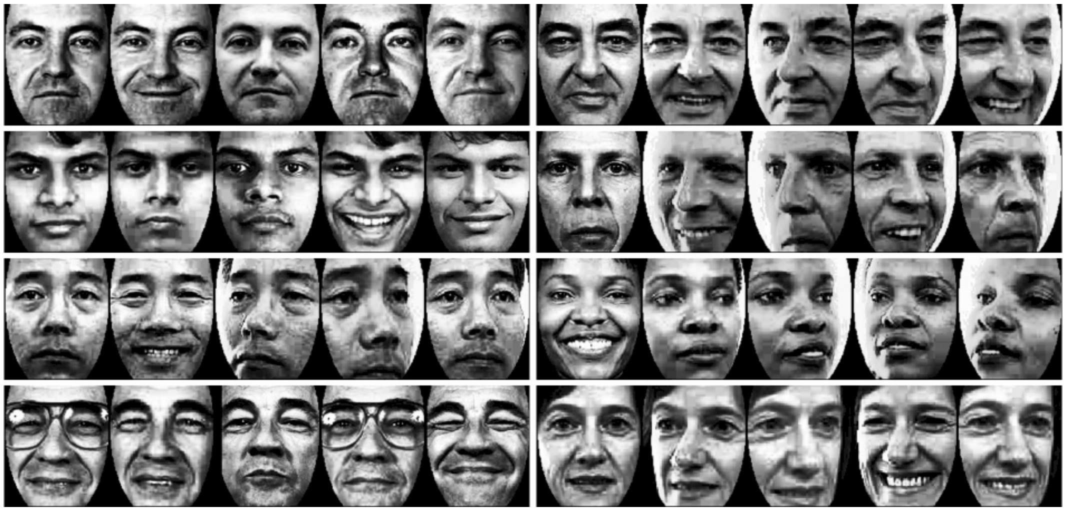

Figure 2.3: Some samples of eight people from the normalized FERET evaluation databases.

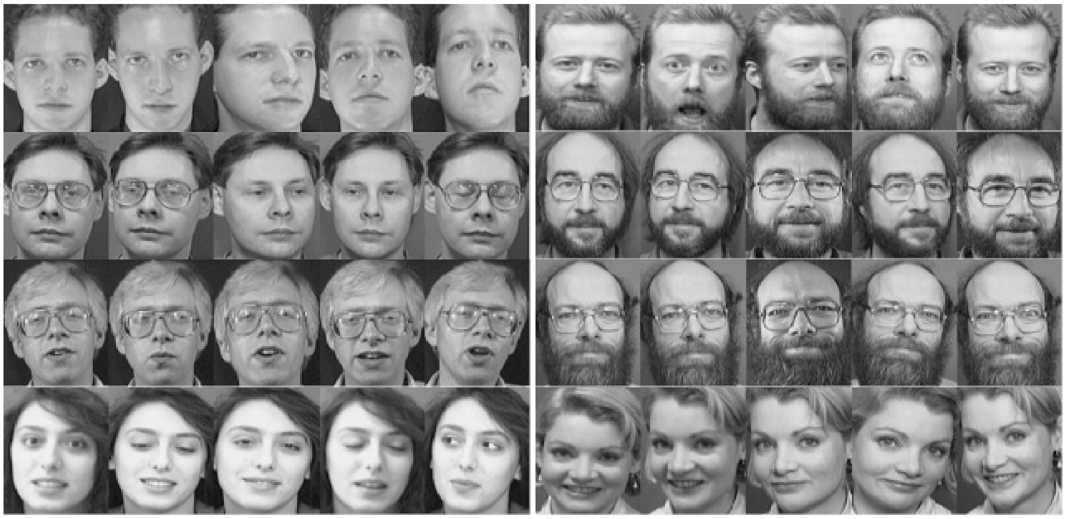

Figure 2.4: Some sample images of 8 people randomly chosen from the ORL database.

In addition to the FERET database, ORL [95, 103] and UMIST [36], are two very popular and specific face databases widely used in the literature. The ORL database that comes from the Olivetti Research Laboratory in University of Cambridge, UK, contains 40 distinct people with 10 images per person. The images were taken at different time instances, with varying lighting conditions, facial expressions (open/closed eyes,

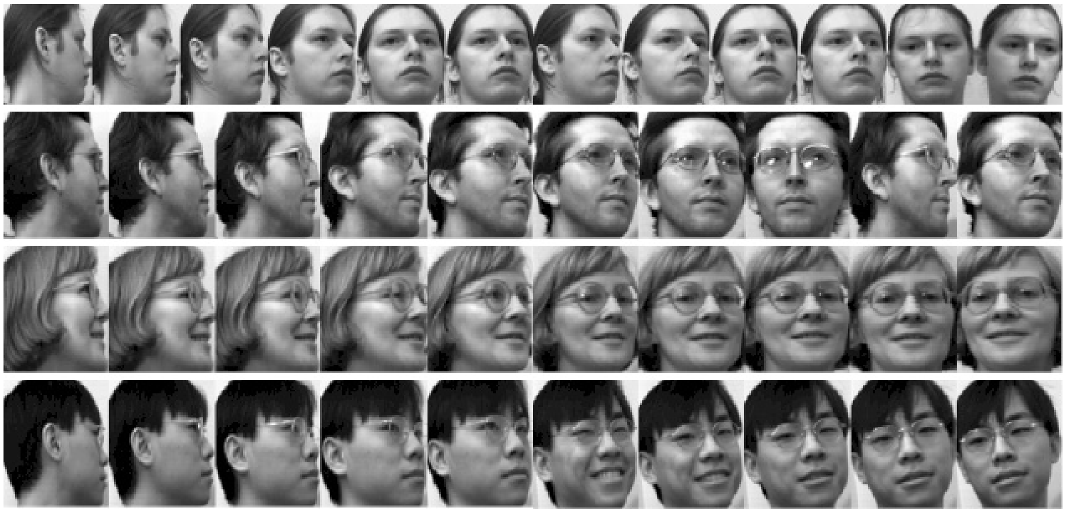

Figure 2.5: Some sample images of 4 people randomly chosen from the UMIST database.

smiling/non-smiling) and facial details (glasses/no-glasses). All persons were in the upright, frontal position, with tolerance for some side movement. The UMIST repository comes from the Image Engineering and Neural Computing Group in University of Manchester Institute of Science and Technology, UK. It is a multi-view database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views as well as a range of race/sex/appearance. Figs.2.4-2.5 depict some samples contained in the two databases, where each image is scaled into (112 × 92), resulting in an input dimensionality of $J = 10304$.

2.3.3 FR Evaluation Designs

The test protocol is designed based on the FERET protocol and the standard FR practices in the literature [95]. Any evaluation database ($\mathcal{G}$) used here is randomly partitioned into two subsets: the training set $\mathcal{Z}$ and the test set $\mathcal{Q}$. They correspond to the target set and the query set in the FERET language respectively. The training set consists of $N = \sum_{i=1}^{C} C_i$ images: C_i images per subject are randomly chosen, where C is the number of subjects. The remaining images are used to form the test set $\mathcal{Q} = \mathcal{G} - \mathcal{Z}$. Any FR method evaluated here is first trained with $\mathcal{Z}$, and the resulting face recognizer is then applied to $\mathcal{Q}$ for testing. For each image $\mathbf{q}_j$ in the query set $\mathcal{Q}$, the evaluated algorithm reports a similarity $s_j(k)$ between $\mathbf{q}_j$ and each image $\mathbf{z}_k$ in the target set. Then, the target images $\mathbf{z}_k$ are sorted by the similarity scores $s_j(\cdot)$. The top rank implies the closest match. In the FERET protocol, the test algorithm is required to answer not only "is the top rank correct?", but also "is the correct answer in the top n ranks?". Assuming that γ_n represents the number of the query images whose real identification correctly matches that of one of the top n ranked targets, the *correct recognition rate* (CRR) for rank n, denoted as R_n, is given by $R_n = \gamma_n/|\mathcal{Q}|$, where $|\mathcal{Q}|$ denotes the size of $\mathcal{Q}$. It is not difficult to see that the above criterion is equivalent to the classic nearest neighbor rule when $n = 1$.

To enhance the accuracy of performance assessment, the CRRs reported in this work are averaged over $t \geq 1$ runs. Each run is executed on a random partition of the evaluation database $\mathcal{G}$ into the training set $\mathcal{Z}$ and the test set $\mathcal{Q}$. Following the framework introduced in [26, 60, 61], the average CRR, denoted as $\bar{R}_n$, is given as follows,

$$\bar{R}_n = \frac{\sum_{i=1}^{t} \gamma_n^{\{i\}}}{|\mathcal{Q}| \cdot t} \tag{2.1}$$

For simplicity, we also use another popular performance measure, *classification error rate* (CER) instead of CRR in some cases. It should be noted at this point that there is no difference between the two measures but CER=(1-CRR).

Chapter 3

Linear Discriminant Learning for Face Recognition

3.1 Introduction

Low-dimensional feature representation with enhanced discriminatory power is of paramount importance to face recognition (FR) systems. We have learned from Chapter 2 that the most successful solution to the issue developed to date is the appearance-based approach. In the approach, *principal component analysis* (PCA) and *linear discriminant analysis* (LDA) are two powerful tools widely used for data reduction and feature extraction. Many state-of-the-art FR methods, including the Eigenfaces method [120] and the Fisherfaces method [7], built on the two techniques, have been shown to be very successful in both practical applications and the FERET competitions [94].

It is generally believed that, when it comes to solving problems of pattern classification, LDA based algorithms outperform PCA based ones, since the former deals directly with discrimination between classes, whereas the latter deals with optimal data compression without paying any attention to the underlying class structure [7, 17, 46]. However, the classification performance of traditional LDA is often degraded by the fact that their separability criteria are not directly related to their classification accuracy in the output space [70]. A solution to the problem is to introduce weighting functions into LDA. Object classes that are closer together in the output space, and thus can potentially result in mis-classification, should be more heavily weighted in the input space. This idea has been further extended in [70] with the introduction of the *fractional-step* LDA (F-LDA) algorithm, where the dimensionality reduction is implemented in an iterative mechanism allowing for the relevant distances to be more accurately weighted. Although the method has been successfully tested on low-dimensional patterns whose dimensionality is $J \leq 5$, it cannot be directly applied to high-dimensional patterns, such as those face images used in the experiments reported here due to two factors.

1. The computational difficulty of the eigen-decomposition of matrices in the high-dimensional face image space. It should be noted at this point that a typical face image pattern of size (112×92) (see *e.g.* Figs.2.4-2.5) results in a vector of dimension $J = 10304$. It is difficult to store a 10304×10304 matrix, which requires a memory space of 810M bytes.

2. The considerably degenerate sample scatter matrices caused by the so-called *small sample size* (SSS) problem, which widely exists in FR tasks where the number of training samples is much smaller than the dimensionality of the samples [7, 17, 46]. For example, in contrast with $J = 10304$, only $L \leq 10$ training samples per subject are available in most FR tasks.

The traditional solution to these two problems requires the incorporation of a PCA step into the LDA framework. In this approach, PCA is used as a pre-processing step for dimensionality reduction and removal of the null spaces of the sample scatter matrices. Then LDA is performed in the lower dimensional PCA subspace, as it was done for example in Fisherfaces [7]. However, it has been shown that the discarded null spaces may contain significant discriminatory information [17, 46]. To prevent this from happening, solutions without a separate PCA step, called *direct* LDA (D-LDA) methods have been presented recently [17, 46]. In the D-LDA framework, data are processed directly in the original high-dimensional input space avoiding the possible loss of significant discriminatory information due to the PCA pre-processing step.

In this chapter, we introduce a new LDA-based feature representation method for FR tasks. The method combines the strengths of the D-LDA and F-LDA approaches while at the same time overcomes their shortcomings and limitations. In the proposed framework, hereafter DF-LDA, we firstly lower the dimensionality of the original input space by introducing a new variant of D-LDA that results in a low-dimensional SSS-relieved subspace where the most discriminatory features are preserved. The variant of D-LDA developed here utilizes a regularized Fisher's discriminant criterion to avoid a problem resulting from the wage of the zero eigenvalues of the within-class scatter matrix as possible divisors in [46]. Also, a weighting function is introduced into the proposed variant of D-LDA, so that a subsequent F-LDA process can be applied to carefully re-orient the SSS-relieved subspace resulting in a set of optimal linear discriminant features for face representation.

3.2 Appearance-based Feature Extraction

To date, the appearance-based learning framework has been most influential in the face recognition (FR) research. Under this framework, the problem of learning low-dimensional feature representation from examples can be stated as follows: Given a training set, $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, containing C classes with each class $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ consisting of a number of localized face images $\mathbf{z}_{ij}$, a total of $N = \sum_{i=1}^{C} C_i$ face images are available in the set. For computational convenience, each image is represented as a column vector of length $J = I_w \times I_h$ by lexicographic ordering of the pixel elements, *i.e.* $\mathbf{z}_{ij} \in \mathbb{R}^J$, where $(I_w \times I_h)$ is the image size, and $\mathbb{R}^J$ denotes the J-dimensional real space. Taking as input such a set $\mathcal{Z}$, the objective of appearance-based learning is to find, based on optimization of certain separability criteria, a transformation φ which produces a low-dimensional feature representation $\mathbf{y}_{ij} = \varphi(\mathbf{z}_{ij})$, $\mathbf{y}_{ij} \in \mathbb{R}^M$ and $M \ll J$, intrinsic to face objects with enhanced discriminatory power for pattern classification.

Among various techniques available for the solution to the learning problem, Principal Component Analysis and Linear Discriminant Analysis are the two most widely used in the FR literature. Due to their huge influences and close relationship with the methods developed later, we begin with a brief review of the two techniques and their applications to FR research.

3.2.1 Principal Component Analysis (PCA)

In the statistical pattern recognition literature, Principal Component Analysis (PCA) [47] is one of the most popular tools for data reduction and feature extraction. The well-known FR method Eigenfaces [120], built on the PCA technique, has been proved to be very successful. In the Eigenfaces method [120], PCA is applied to the training set $\mathcal{Z}$ to find the N eigenvectors (with non zero eigenvalues) of the set's covariance matrix,

$$\mathbf{S}_{cov} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} (\mathbf{z}_{ij} - \bar{\mathbf{z}})(\mathbf{z}_{ij} - \bar{\mathbf{z}})^T \tag{3.1}$$

where $\bar{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} \mathbf{z}_{ij}$ is the average of the ensemble. The Eigenfaces are the first $M(\leq N)$ eigenvectors (denoted as Ψ_{ef}) corresponding to the largest eigenvalues, and they form a low-dimensional subspace, called "face space" where most energies of the original face manifold are supposed to lie. Fig.3.1 (1st row) shows the first six most significant Eigenfaces, which appear, as some researchers have said, as ghostly faces. Transforming to the M-dimensional face space is a simple linear mapping: $\mathbf{y}_{ij} = \Psi_{ef}^T(\mathbf{z}_{ij} - \bar{\mathbf{z}})$, where the basis vectors Ψ_{ef} are orthonormal. The subsequent classification of face patterns can be performed in the face space using any classifier.

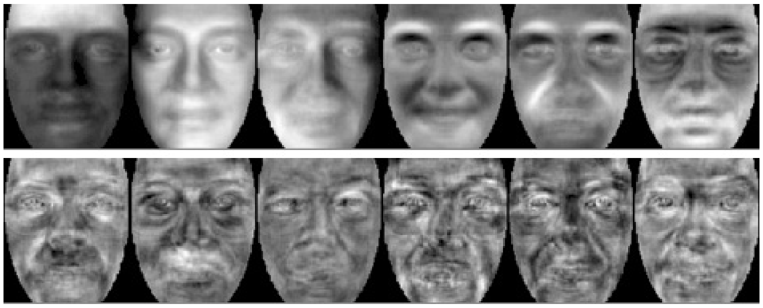

Figure 3.1: Visualization of two types of basis vectors obtained from a normalized subset of the FERET database. Row 1: the first 6 most significant PCA bases. Row 2: the first 6 most significant LDA bases.

3.2.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) [27] is also a representative technique for data reduction and feature extraction. In contrast with PCA, LDA is a class specific one that utilizes supervised learning to find a set of $M \ll J$ feature basis vectors, denoted as $\{\psi_m\}_{m=1}^{M}$, in such a way that the ratio of the between- and within-class scatters of the training sample set is maximized. The maximization is equivalent to solving the following eigenvalue problem,

$$\Psi = \arg\max_{\Psi} \frac{\left|\Psi^T \mathbf{S}_b \Psi\right|}{\left|\Psi^T \mathbf{S}_w \Psi\right|}, \quad \Psi = [\psi_1, \cdots, \psi_M], \quad \psi_m \in \mathbb{R}^J \tag{3.2}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between- and within-class scatter matrices, having the following expressions,

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C} C_i(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})^T = \sum_{i=1}^{C} \Phi_{b,i}\Phi_{b,i}^T = \Phi_b\Phi_b^T \tag{3.3}$$

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C_i}(\mathbf{z}_{ij} - \bar{\mathbf{z}}_i)(\mathbf{z}_{ij} - \bar{\mathbf{z}}_i)^T \tag{3.4}$$

where $\bar{\mathbf{z}}_i = \frac{1}{C_i}\sum_{j=1}^{C_i}\mathbf{z}_{ij}$ is the mean of class $\mathcal{Z}_i$, $\Phi_{b,i} = (C_i/N)^{1/2}(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})$ and $\Phi_b = [\Phi_{b,1}, \cdots, \Phi_{b,c}]$. When $\mathbf{S}_w$ is non-singular, the basis vectors Ψ sought in Eq.3.2 correspond to the first M most significant eigenvectors of $(\mathbf{S}_w^{-1}\mathbf{S}_b)$, where the “significant” means that the eigenvalues corresponding to these eigenvectors are the first M largest ones. For an input image $\mathbf{z}$, its LDA-based feature representation can be obtained simply by a linear projection, $\mathbf{y} = \Psi^T \mathbf{z}$.

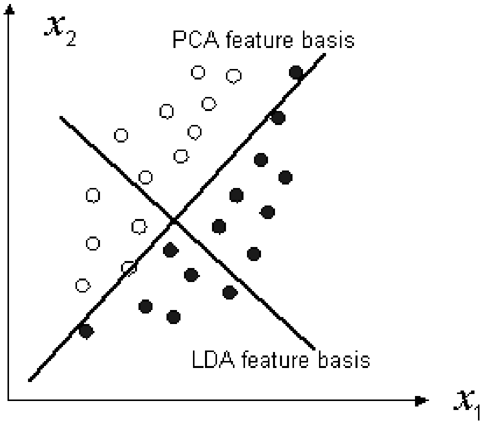

Figure 3.2: PCA feature basis *vs* LDA feature basis obtained from a set of 2D training samples consisting of two classes. Each sample $\mathbf{x} = [x_1, x_2]$ is represented by its projections in the two feature bases respectively. In this case, the PCA-based representation accounts most variations of the samples, but it is entirely unsuitable for the purpose of pattern classification.

Fig.3.1(2nd row) visualizes the first six most significant basis vectors $\{\psi_i\}_{i=1}^{6}$ obtained by using the LDA version of [74]. Comparing Fig.3.1(1st row) to Fig.3.1(2nd row), it can be seen that the Eigenfaces look more like a real human face than those LDA basis vectors. This can be explained by the different learning criteria used in the two techniques. LDA optimizes the low-dimensional representation of the objects with focus on the most discriminant feature extraction while PCA achieves simply object reconstruction in a least-square sense. The difference may lead to significantly different orientations of feature bases as shown in Fig.3.2, where it is not difficult to see that the representation obtained by PCA is entirely unsuitable for the task of separating the two classes. As a consequence, it is generally believed that when it comes to solving problems of pattern classification such as face recognition, the LDA based feature representation is usually superior to the PCA based one [7,17,46].

3.3 LDA in the Small-Sample-Size (SSS) Scenarios

3.3.1 The Small-Sample-Size (SSS) problem

Statistical learning theory tells us essentially that the difficulty of an estimation problem (*e.g.* for $\mathbf{S}_b$ and $\mathbf{S}_w$) increases drastically with the dimensionality J of the sample space, since in principle, as a function of J, one needs exponentially many patterns to sample the space properly [88,123,124]. Unfortunately, in the particular tasks of face recognition, the truth is that data are highly dimensional, while the number of available training samples per subject is usually much smaller than the dimensionality of the sample space. For example, a canonical example used for recognition is a 112 × 92 face image, which exists in a 10304-dimensional real space. Nevertheless, the number (C_i) of examples per class available for learning is not more than ten in most cases. This results in the so-called *small sample size* (SSS) problem, which is known to have significant influences on the design and performance of a *statistical pattern recognition* (SPR) system. During the last three decades, considerable research efforts have been made to deal with the SSS problem related to various SPR issues such as feature extraction, feature selection and classifier design (see *e.g.* [7,17,43,44,46,51,74,81,84,98,108,116,125,129]).

In the application of LDA into FR tasks, the SSS problem often gives rise to high variance in the sample-based estimation for the two scatter matrices, $\mathbf{S}_b$ and $\mathbf{S}_w$, which are either ***ill-*** or ***poorly-posed***. Roughly speaking, a problem is poorly-posed if the number of parameters to be estimated is comparable to the number of observations and ill-posed if that number exceeds the sample size. Compared to the PCA solution, the LDA solution is much more susceptible to the SSS problem given a same training set, since the latter requires many more training samples than the former due to the increased number of parameters needed to be estimated [125]. As a result, the general belief that

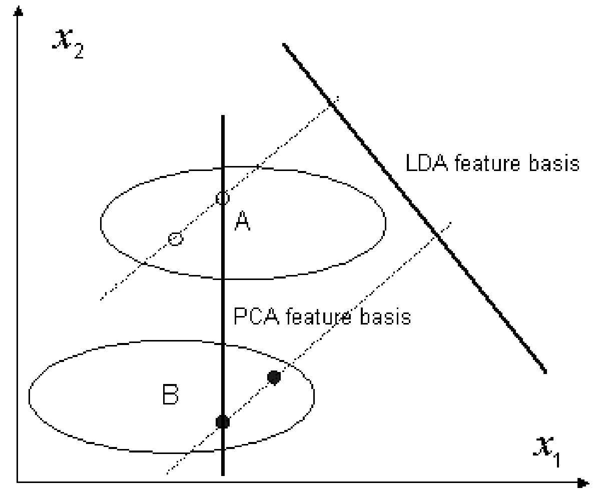

Figure 3.3: There are two different classes (A and B) subjected to two different “Gaussian-like” distributions respectively. However, only two examples per class are available for the learning procedure. Each example $\mathbf{x} = [x_1, x_2]$ is a 2D vector. In this case, the basis vector produced by PCA is more desirable than the one produced by LDA for the purpose of pattern classification.

LDA is superior to PCA in the context of pattern classification may not be correct in the SSS scenarios, where it has been shown recently in [78] that there is no guarantee that LDA will outperform PCA. The phenomenon of LDA over-fitting the training data in the SSS settings can be further illustrated by a simple example shown in Fig.3.3, where PCA yields a superior feature basis for the purpose of pattern classification.

3.3.2 Where are the optimal discriminant features?

From the above analysis, it can be seen that the biggest challenge that all LDA-based FR methods have to face is the SSS problem. Briefly, there are two ways to tackle the problem with LDA. One is to apply linear algebra techniques to solve the numerical problem of inverting the singular within-class scatter matrix $\mathbf{S}_w$. For example, the pseudo inverse is utilized to complete the task in [116]. Also, small perturbation may be added to $\mathbf{S}_w$ so that it becomes nonsingular [44, 132]. The other way is a subspace approach, such as the one followed in the development of the Fisherfaces method [7], where PCA is firstly used as a pre-processing step to remove the null space of $\mathbf{S}_w$, and then LDA is performed in the lower dimensional PCA subspace. However, it should be noted at this point that the maximum of the ratio in Eq.3.2 can be reached only when $\psi_i^T \mathbf{S}_w \psi_i = 0$ and $\psi_i^T \mathbf{S}_b \psi_i \neq 0$. In other words, the discarded null space that ψ_i belongs to may contain significant discriminatory information. To prevent this from happening, solutions without a separate PCA step, called *direct* LDA (D-LDA) methods have been presented recently in [17, 46, 67–69]. The basic premise behind the D-LDA approach is that the null space of $\mathbf{S}_w$ contains significant discriminant information if the projection of $\mathbf{S}_b$ is not zero in that direction, while no significant information, in terms of the maximization in Eq.3.2, will be lost if the null space of $\mathbf{S}_b$ is discarded. It is not difficult to see that when ψ_i belongs to the null space of $\mathbf{S}_b$, the ratio $\frac{|\psi_i^T \mathbf{S}_b \psi_i|}{|\psi_i^T \mathbf{S}_w \psi_i|}$ drops down to its minimal value, 0, due to $\psi_i^T \mathbf{S}_b \psi_i = 0$. In other words, assuming that $\mathcal{A}$ and $\mathcal{B}$ represent the null space of $\mathbf{S}_b$ and $\mathbf{S}_w$ respectively, while $\mathcal{A}' = \mathbb{R}^N - \mathcal{A}$ and $\mathcal{B}' = \mathbb{R}^N - \mathcal{B}$ are the complement spaces of $\mathcal{A}$ and $\mathcal{B}$, the optimal discriminant feature bases sought by D-LDA exist in the intersection space $(\mathcal{A}' \cap \mathcal{B})$.

In early D-LDA methods [67–69], only the two conditions ($\psi_i^T \mathbf{S}_w \psi_i = 0$ and $\psi_i^T \mathbf{S}_b \psi_i \neq 0$) are used in the search for the optimal discriminant feature bases. However, it is shown by recent research [17] that the vectors that satisfy the two conditions may not maximize the between-class separability. To address the shortcoming, one more condition, *i.e.* $\arg\max\limits_{\psi_i} (\psi_i^T \mathbf{S}_b \psi_i)$, is introduced in the D-LDA methods proposed recently in [17, 46]. The main difference between the two methods in [17, 46] is that, the version of [46] first diagonalizes $\mathbf{S}_b$ to find $\mathcal{A}'$ when seeking the solution of (3.2), while the version of [17] as early D-LDA methods [67–69] first diagonalizes $\mathbf{S}_w$ to find $\mathcal{B}$. Although it seems that there is no significant difference between the two methods, it may be intractable to calculate $\mathcal{B}$ when the size of $\mathbf{S}_w$ is large, which is the case in most FR tasks. For example, a typical face image of (112×92) results in the size of the two scatter matrices, $\mathbf{S}_w$ and $\mathbf{S}_b$, going up to (10304×10304). Fortunately, the rank of $\mathbf{S}_b$ is determined by $rank(\mathbf{S}_b) = min(J, C-1)$, with C the number of image classes, which is a much smaller value than J the dimensionality of the images in most cases, *e.g.* $C = 40$ in the ORL database used in the experiments reported here, resulting in $rank(\mathbf{S}_b) = 39$. $\mathcal{A}'$ can be easily found by solving an eigenvalue problem of a (39×39) matrix rather than the original (10304×10304) matrix through an algebraic transformation [46,120]. Then the intersection space $(\mathcal{A}' \cap \mathcal{B})$ can be found by solving the null space of $\mathbf{S}_w$'s projection into $\mathcal{A}'$, where the projection is a small matrix with size (39×39). Based on these reasons, we follow the D-LDA version of [46] (hereafter YD-LDA), *i.e.* diagonalizing $\mathbf{S}_b$ first instead of $\mathbf{S}_w$ in the algorithm to be developed in the following sections.

3.3.3 A modified Fisher's criterion

The performance of the YD-LDA method [46] may deteriorate rapidly due to two problems that may be encountered when the SSS problem becomes severe. One is that the zero eigenvalues of the within-class scatter matrix are used as possible divisors, so that the YD-LDA process can not be carried out. The other is that the worse of the SSS situations may significantly increase the variance in the estimation for small eigenvalues of the within-class scatter matrix, while the importance of the eigenvectors corresponding to these small eigenvalues is dramatically exaggerated. To avoid these two problems, a variant of D-LDA is developed here by introducing a modified Fisher's criterion.

The modified Fisher's criterion, which is utilized in this work instead of the conventional one (Eq.3.2), can be expressed as follows:

$$\Psi = \arg\max_{\Psi} \frac{\left|\Psi^T \mathbf{S}_b \Psi\right|}{\left|\eta(\Psi^T \mathbf{S}_b \Psi) + (\Psi^T \mathbf{S}_w \Psi)\right|} \tag{3.5}$$

where $0 \leq \eta \leq 1$ is a regularization parameter. Although Eq.3.5 looks different from Eq.3.2, it can be shown that the modified criterion is exactly equivalent to the conventional one by the following theorem.

Theorem 1 *Let $\mathbb{R}^J$ denote the J-dimensional real space, and suppose that $\forall \psi \in \mathbb{R}^J$, $u(\psi) \geq 0$, $v(\psi) \geq 0$, $u(\psi) + v(\psi) > 0$ and $0 \leq \eta \leq 1$. Let $q_1(\psi) = \frac{u(\psi)}{v(\psi)}$ and $q_2(\psi) = \frac{u(\psi)}{\eta \cdot u(\psi) + v(\psi)}$. Then, $q_1(\psi)$ has the maximum (including positive infinity) at point $\psi^* \in \mathbb{R}^J$ iff $q_2(\psi)$ has the maximum at point ψ^*.*

Proof: Since $u(\psi) \geq 0$, $v(\psi) \geq 0$ and $0 \leq \eta \leq 1$, we have $0 \leq q_1(\psi) \leq +\infty$ and $0 \leq q_2(\psi) \leq \frac{1}{\eta}$.

1. If $\eta = 0$, then $q_1(\psi) = q_2(\psi)$.

2. If $0 < \eta \leq 1$ and $v(\psi) = 0$, then $q_1(\psi) = +\infty$ and $q_2(\psi) = \frac{1}{\eta}$.

3. If $0 < \eta \leq 1$ and $v(\psi) > 0$, then

$$q_2(\psi) = \frac{\frac{u(\psi)}{v(\psi)}}{1 + \eta \frac{u(\psi)}{v(\psi)}} = \frac{q_1(\psi)}{1 + \eta q_1(\psi)} = \frac{1}{\eta}\left(1 - \frac{1}{1 + \eta q_1(\psi)}\right) \tag{3.6}$$

   It can be seen from Eq.3.6 that $q_2(\psi)$ increases iff $q_1(\psi)$ increases.

Combining the above three cases, the theorem is proven.

The modified Fisher's criterion is a function of the parameter η, which controls the strength of regularization. Within the variation range of η, two extremes should be noted. In one extreme where $\eta = 0$, the modified Fisher's criterion is reduced to the conventional one with no regularization. In contrast with this, strong regularization is introduced in another extreme where $\eta = 1$. In this case, Eq.3.5 becomes $\Psi = \arg\max\limits_{\Psi} \frac{\left|\Psi^T \mathbf{S}_b \Psi\right|}{|(\Psi^T \mathbf{S}_b \Psi) + (\Psi^T \mathbf{S}_w \Psi)|}$, which as a variant of the original Fisher's criterion has been also widely used for example in [17, 67–69]. The advantages of introducing the regularization will be seen during the development of the D-LDA variant proposed in the next section.

3.3.4 A variant of direct LDA: JD-LDA

For the reasons explained in Section 3.3.2, we start our D-LDA algorithm by solving the eigenvalue problem of $\mathbf{S}_b$. It is intractable to directly compute the eigenvectors of $\mathbf{S}_b$ which is a large size $(J \times J)$ matrix. Fortunately, the first m $(\leq C-1)$ most significant eigenvectors of $\mathbf{S}_b$, which correspond to nonzero eigenvalues, can be indirectly derived through an algebraic transform [120] from the eigenvectors of the matrix $(\Phi_b^T \Phi_b)$ with size $(C \times C)$, where $\Phi_b = [\Phi_{b,1}, \cdots, \Phi_{b,c}]$ defined in Eq.3.3. Let $\lambda_{b,i}$ and $\mathbf{e}_i$ be the i-th eigenvalue and its corresponding eigenvector of $(\Phi_b^T \Phi_b)$, $i = 1, \cdots, C$, sorted in **decreasing** eigenvalue order. Since $(\Phi_b \Phi_b^T)(\Phi_b \mathbf{e}_i) = \lambda_{b,i}(\Phi_b \mathbf{e}_i)$, $(\Phi_b \mathbf{e}_i)$ is an eigenvector of $\mathbf{S}_b$.

To remove the null space of $\mathbf{S}_b$, we use only its first m $(\leq C-1)$ most significant eigenvectors: $\mathbf{U}_m = \Phi_b \mathbf{E}_m$ with $\mathbf{E}_m = [\mathbf{e}_1, \cdots, \mathbf{e}_m]$, whose corresponding eigenvalues are nonzero, and discard the remaining $(J-m)$ eigenvectors. It is not difficult to see that $\mathbf{U}_m^T \mathbf{S}_b \mathbf{U}_m = \Lambda_b$, with $\Lambda_b = \mathbf{diag}[\lambda_{b,1}^2, \cdots, \lambda_{b,m}^2]$, a $(m \times m)$ diagonal matrix. Let $\mathbf{H} = \mathbf{U}_m \Lambda_b^{-1/2}$. Projecting $\mathbf{S}_b$ and $\mathbf{S}_w$ into the subspace spanned by $\mathbf{H}$, we have $\mathbf{H}^T \mathbf{S}_b \mathbf{H} = \mathbf{I}$ and $\mathbf{H}^T \mathbf{S}_w \mathbf{H}$. Then, we diagonalize $\mathbf{H}^T(\eta \mathbf{S}_b + \mathbf{S}_w)\mathbf{H}$ which is a tractable matrix with size $(m \times m)$. Let $\mathbf{p}_i$ be the i-th eigenvector of $\mathbf{H}^T(\eta \mathbf{S}_b + \mathbf{S}_w)\mathbf{H}$, where $i = 1, \cdots, m$, sorted in **increasing** order according to corresponding eigenvalues $\lambda_{w,i}$. In the set of ordered eigenvectors, those corresponding to the smallest eigenvalues maximize the ratio in Eq.3.5, and they should be considered as the most discriminatory features. We can discard the eigenvectors with the largest eigenvalues, and denote the $M \leq m$ selected eigenvectors as $\mathbf{P}_M = [\mathbf{p}_1, \cdots, \mathbf{p}_M]$. Defining a matrix $\mathbf{Q} = \mathbf{H}\mathbf{P}_M$, we have $\mathbf{Q}^T(\eta \mathbf{S}_b + \mathbf{S}_w)\mathbf{Q} = \Lambda_w$, where $\Lambda_w = \mathbf{diag}[\lambda_{w,1}, \cdots, \lambda_{w,M}]$, a $(M \times M)$ diagonal matrix.

Based on the derivation presented above, we can obtain a set of optimal discriminant feature basis vectors, $\Psi = \mathbf{H}\mathbf{P}_M \Lambda_w^{-1/2}$. To facilitate comparison, it should be mentioned at this point that the YD-LDA method uses the conventional Fisher's criterion of (3.2) with $(\eta \mathbf{S}_b + \mathbf{S}_w)$ replaced by $\mathbf{S}_w$. However, since the subspace spanned by Ψ may contain the intersection space $(\mathcal{A}' \cap \mathcal{B})$, it is possible that there exist zero or very small eigenvalues in $\Lambda_w (= \mathbf{Q}^T \mathbf{S}_w \mathbf{Q}$ for YD-LDA), so that the normalization $(\mathbf{Q}\Lambda_w^{-1/2})$, which has been shown to have a significant influence on the classification performance [132], can not be carried out successfully. In contrast with this, due to the utilization of the modified Fisher's criterion, the non-singularity of $\mathbf{Q}^T(\eta \mathbf{S}_b + \mathbf{S}_w)\mathbf{Q}$ when $\eta > 0$ can be guaranteed by the following lemma.

Lemma 1 *Suppose* $\mathbf{B}$ *is a real matrix of size* $(J \times J)$*. Furthermore, let us assume that it can be represented as* $\mathbf{B} = \Phi\Phi^T$ *where* Φ *is a real matrix of size* $(J \times M)$*. Then, the matrix* $(\eta \mathbf{I} + \mathbf{B})$ *is positive definite,* i.e. $\eta \mathbf{I} + \mathbf{B} > 0$*, where* $\eta > 0$ *and* $\mathbf{I}$ *is the* $(J \times J)$ *identity matrix.*

Proof: Since $B^T = \Phi\Phi^T = B$, $\eta \mathbf{I} + B$ is a real symmetric matrix. Let x be any $J \times 1$ non-zero real vector, we have $x^T(\eta \mathbf{I} + B)x = \eta(x^T x) + x^T B x = \eta(x^T x) + (\Phi^T x)^T(\Phi^T x) > 0$. According to [45], the matrix $(\eta \mathbf{I} + B)$ that satisfies the above condition is strictly positive definite, *i.e.* $\eta \mathbf{I} + B > 0$.

Similar to $\mathbf{S}_b$, $\mathbf{S}_w$ can be expressed as $\mathbf{S}_w = \Phi_w \Phi_w^T$, and then $\mathbf{Q}^T \mathbf{S}_w \mathbf{Q} = (\mathbf{Q}^T \Phi_w)(\mathbf{Q}^T \Phi_w)^T$, which is real symmetric. Based on the Lemma 1, it is not difficult to see that $\Lambda_w = \mathbf{Q}^T(\eta \mathbf{S}_b + \mathbf{S}_w)\mathbf{Q} = (\eta \mathbf{I} + \mathbf{Q}^T \mathbf{S}_w \mathbf{Q})$ is strictly positive definite. In addition to avoid possible zero eigenvalues appearing in Λ_w, the regularization introduced here helps to reduce the high variance related to the sample-based estimates, *e.g.* for the smallest nonzero eigenvalues in $(\mathbf{Q}^T \mathbf{S}_w \mathbf{Q})$. Without the regularization, *i.e.* $\Lambda_w = \mathbf{Q}^T \mathbf{S}_w \mathbf{Q}$, often an effect arising from the SSS problem is that the importance of the eigenvectors corresponding to these small eigenvalues is dramatically exaggerated due to the normalization $(\mathbf{Q}\Lambda_w^{-1/2})$.

For the sake of simplicity hereafter, we call "JD-LDA" the D-LDA variant developed above. The detailed process to implement the JD-LDA method is depicted in Fig.3.4, where it is not difficult to see that JD-LDA reduces to YD-LDA when $\eta = 0$. Also, as a powerful weapon against the SSS problem, the regularization issue will be further discussed under a more general discriminant analysis framework in Chapter 4, where it will be also shown that the JD-LDA method is a special case of the general framework.

```
Input: A training set $\mathcal{Z}$ with $C$ classes: $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, each class containing
    $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ face images, where $\mathbf{z}_{ij} \in \mathbb{R}^J$, and the regularization parameter $\eta$.
Output: An $M$-dimensional LDA subspace spanned by $\Psi$, an $J \times M$ matrix
    with $M \ll J$.
Algorithm:
    Step 1. Express $\mathbf{S}_b = \Phi_b \Phi_b^T$ (Eq.3.3).
    Step 2. Find the eigenvectors of $\Phi_b^T \Phi_b$ with non-zero eigenvalues, and denote
        them as $\mathbf{E}_m = [\mathbf{e}_1, \cdots, \mathbf{e}_m]$, $m \leq C - 1$.
    Step 3. Calculate the first $m$ most significant eigenvectors ($\mathbf{U}_m$) of $\mathbf{S}_b$ and
        their corresponding eigenvalues ($\Lambda_b$) by $\mathbf{U}_m = \Phi_b \mathbf{E}_m$ and $\Lambda_b = \mathbf{U}_m^T \mathbf{S}_b \mathbf{U}_m$.
    Step 4. Let $\mathbf{H} = \mathbf{U}_m \Lambda_b^{-1/2}$. Find eigenvectors of $(\eta \mathbf{I} + \mathbf{H}^T \mathbf{S}_w \mathbf{H})$, $\mathbf{P}$.
    Step 5. Choose the $M (\leq m)$ eigenvectors in $\mathbf{P}$ with the smallest eigenvalues.
        Let $\mathbf{P}_M$ and $\Lambda_w$ be the chosen eigenvectors and their corresponding
        eigenvalues respectively.
    Step 6. Return $\Psi = \mathbf{H} \mathbf{P}_M \Lambda_w^{-1/2}$.
```

Figure 3.4: Pseudo code implementation of the JD-LDA method

3.4 The Direct Fractional-Step LDA (DF-LDA)

3.4.1 Weighted between-class scatter matrix

The optimization process in most traditional LDA approaches including those D-LDA methods introduced above is not directly linked to the separability of samples in the output space, which determines the classification performance of the resulting LDA-based FR systems. To this end, an iterative weighting mechanism has been proposed recently in the so-called *fractional-step* LDA (F-LDA) algorithm [70], where a weighting function is integrated into the between-class scatter matrix in the input space, to penalize those classes that are close and can potentially lead to mis-classifications in the output space generalized previously. The weighted between-class scatter matrix can be expressed as:

$$\hat{\mathbf{S}}_b = \sum_{i=1}^{C} \hat{\Phi}_{b,i} \hat{\Phi}_{b,i}^T = \hat{\Phi}_b \hat{\Phi}_b^T, \quad \hat{\Phi}_b = \left[\hat{\Phi}_{b,1}, \cdots, \hat{\Phi}_{b,c} \right] \tag{3.7}$$

where

$$\hat{\Phi}_{b,i} = (C_i/N)^{1/2} \sum_{j=1}^{C} (w(d_{ij}))^{1/2} (\bar{\mathbf{z}}_i - \bar{\mathbf{z}}_j) \tag{3.8}$$

and $d_{ij} = \| \bar{\mathbf{z}}_i - \bar{\mathbf{z}}_j \|$ is the Euclidean distance between the means of class i and class j. The weighting function $w(d_{ij})$ is a monotonically decreasing function of the distance d_{ij}. The only constraint is that the weight should drop faster than the distance d_{ij} with the authors in [70] recommending weighting functions of the form $w(d_{ij}) = (d_{ij})^{-2p}$ with $p = 2, 3, ...$

3.4.2 Rotation, re-orientation and dimensionality reduction of the JD-LDA subspace

In the input face image space, the weighting mechanism introduced in F-LDA cannot be implemented due to high dimensionality and singularity of the between- and within-class scatter matrices. However, in the low-dimensional JD-LDA output space spanned by

Ψ, it can be seen that the between-class scatter matrix becomes non-singular and the denominator in Eq.3.5 has been whitened, $\Psi^T(\eta\mathbf{S}_b+\mathbf{S}_w)\Psi = \mathbf{I}$. Thus, following JD-LDA, we can apply a F-LDA like method to further optimize the feature bases Ψ and reduce the dimensionality of the final output space, for example from M to $M'(< M)$ now.

To this end, we firstly project the original face images into the M-dimensional JD-LDA subspace, obtaining a representation $\mathbf{x}_{ij} = \Psi^T\mathbf{z}_{ij}$, whose ensemble is denoted as $\mathcal{X} = \{\mathbf{x}_{ij}, i = 1, \cdots, C; j = 1, \cdots, C_i\}$. Let $\hat{\mathbf{S}}_b^{\{M\}}$ (defined by Eq.3.7) be the between-class scatter matrix of $\mathcal{X}$, and $\hat{\mathbf{u}}_M$ be the M-th eigenvector of $\hat{\mathbf{S}}_b^{\{M\}}$ which corresponds to the smallest eigenvalue of $\hat{\mathbf{S}}_b^{\{M\}}$. This eigenvector will be discarded when dimensionality is reduced from M to $(M-1)$. A problem may be encountered during such a dimensionality reduction procedure. If classes $\mathcal{Z}_i$ and $\mathcal{Z}_j$ are well-separated in the M-dimensional input space, where it means a large Euclidean distance d_{ij} between the two classes, this will produce a very small weight $w(d_{ij})$. As a result, the two classes may be over-penalized by the weight $w(d_{ij})$ so as to heavily overlap in the $(M-1)$-dimensional output space, which is orthogonal to $\hat{\mathbf{u}}_M$. To avoid the problem, a kind of "automatic gain control" is introduced to smooth the weighting procedure in F-LDA [70], where dimensionality is reduced from M to $(M-1)$ at $r \geq 1$ fractional steps instead of one step directly. In each step, $\hat{\mathbf{S}}_b$ and its eigenvectors are recomputed based on the changes of $w(d_{ij})$ in current input space, which the output space of last step. In this way, the $(M-1)$-dimensional subspace is rotated and re-oriented iteratively, and severe overlap between classes in the output space is avoided. The eigenvector $\hat{\mathbf{u}}_M$ will not be discarded until r iterations are done. The pseudo-code implementation of the procedure can be found in Figure 3.5, where the dimensionality of the LDA output space is finally reduced from M to M', with r iterations being applied for each dimensionality reduction by 1.

It should be noted at this point that the F-LDA approach [70] has only been applied in small (not more than five) dimensionality pattern spaces. To the best of the author's knowledge the work reported here constitutes the first attempt to introduce fractional-

```
Input: A training set: $\mathcal{X} = \left\{ \mathbf{x}_{ij} \in \mathbb{R}^M, i = 1, \cdots, C; j = 1, \cdots, C_i \right\}$, the scaling
    factor $\alpha (< 1)$, the number of fractional steps $r$, and the objective
    dimensionality of the output space, $M' (< M)$.
Output: An $M'$-dimensional LDA subspace spanned by $\hat{\Psi}$, an $M \times M'$ matrix.
Algorithm:
    Initialize $\hat{\Psi} = \mathbf{I}$, a $M \times M$ identity matrix.
    For $m = M$ to $M'$ with step $-1$
        For $l = 0$ to $(r - 1)$ with step 1
            Project the input data to the space spanned by $\hat{\Psi}$: $\mathbf{y} = \hat{\Psi}^T \mathbf{x}$.
            Compress the last component of $\mathbf{y}$ by a factor $\alpha^l$: $\mathbf{y}_m = \alpha^l \mathbf{y}_m$.
            Compute the between-class scatter matrix of $\mathbf{y}$: $\hat{\mathbf{S}}_b^{\{m\}}$ with Eq.3.7.
            Compute the ordered (in decreasing) eigenvalues $[\hat{\lambda}_{b,1}, \cdots, \hat{\lambda}_{b,m}]$ and
                corresponding eigenvectors $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_m]$ of $\hat{\mathbf{S}}_b^{\{m\}}$.
            Update $\hat{\Psi} = \hat{\Psi} \hat{\mathbf{U}}$.
        End for loop $l$
        Discard the last $(m^{th})$ column of $\hat{\Psi}$: $\hat{\Psi} = \hat{\Psi}(:, 1 : m - 1)$.
    End for loop $m$
```

Figure 3.5: Pseudo-code for the computation of the F-LDA algorithm

step rotations and re-orientations in a realistic application involving large dimensionality spaces. This becomes possible due to the integrated structure of the (JD-LDA+F-LDA) algorithm (denoted as DF-LDA hereafter).

The effect of the space rotation strategy used in the DF-LDA method can be illustrated by an example shown in Fig.3.6, where the visualized are the first two most significant features of each image (of 170 multi-view face images of 5 subjects from the

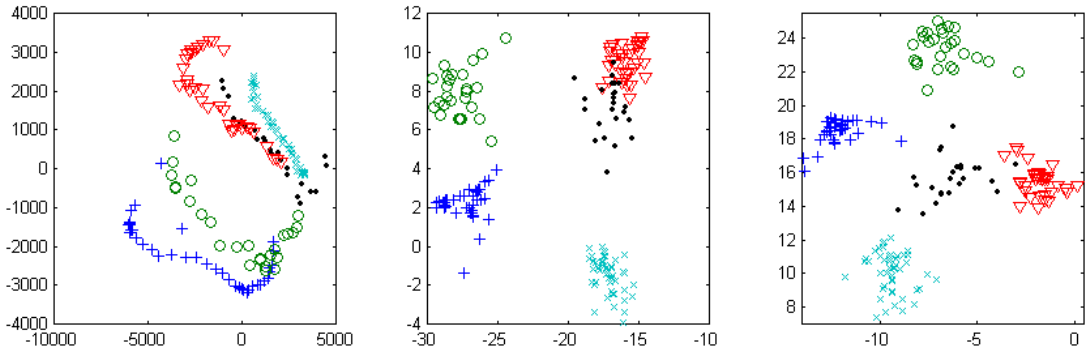

Figure 3.6: Distribution of 170 face images of 5 subjects (classes), projected into **left**: PCA-based subspace, **middle**: JD-LDA-based subspace and **right**: DF-LDA-based subspace. Each subspace is spanned by its two most significant feature bases. The five subjects are randomly selected from the UMIST database.

UMIST database [36]) extracted by PCA, JD-LDA (with $\eta = 1$ to ensure sufficient regularization) and DF-LDA respectively. The PCA-based representation as depicted in Fig.3.6-left is optimal in terms of image reconstruction, thereby provides some insight on the original structure of image distribution, which is highly complex and non-separable as expected for a multi-view face image set. Although the separability of subjects is greatly improved in the JD-LDA-based subspace, some classes still overlap as shown in Fig.3.6-middle. However, after a few fractional-step re-orientations of the JD-LDA subspace, it can be seen from Fig.3.6-right that the separability is further enhanced, and different classes tend to be equally spaced.

3.5 Experimental Results

Two popular face databases in the literature, the ORL [103] and the UMIST [36], are used to demonstrate the effectiveness of the proposed DF-LDA framework. The details of the two databases have been described in Section 2.3.2. It should be noted that each image was scaled into (112×92), resulting in an input dimensionality of $J = 10304$.

To start the FR experiments, each one of the two databases is randomly partitioned into a training set and a test set with no overlap between the two. The partition of the ORL database is done following the recommendation of [60, 61] which call for 5 images per person randomly chosen for training, and the other 5 for testing. Thus, a training set of 200 images and a test set of 200 images are created. For the UMIST database, 8 images per person are randomly chosen to produce a training set of 160 images. The remaining 415 images are used to form the test set. In the following experiments, the figures of merit are the *classification error rates* (CERs) averaged over 5 runs (4 runs in [61] and 3 runs in [60]), each run being performed on such random partitions in the two databases. It is worthy to mention here that both experimental setups introduce SSS conditions since the number of training samples are in both cases much smaller than the dimensionality of the input space. Also, we have observed some partition cases, where close-to-zero eigenvalues occurred in $(\mathbf{Q}^T\mathbf{S}_w\mathbf{Q})$ as discussed in Section 3.3.4. In these cases, in contrast with the failure of YD-LDA [46], JD-LDA and DF-LDA were still able to perform well.

In addition to YD-LDA [46], DF-LDA is compared against two popular appearance-based FR methods, namely: Eigenfaces [120] and Fisherfaces [7]. For each of the four methods, the FR procedure consists of: (i) a feature extraction step where four kinds of feature representation of each training or test sample are extracted by projecting the sample onto the four feature spaces generalized by Eigenface, Fisherface, YD-LDA and DF-LDA respectively, (ii) a classification step in which each feature representation obtained in the first step is fed into a simple nearest neighbor classifier. It should be noted at this point that, since the focus in this work is on feature extraction, a very simple classifier, namely nearest neighbor, is used in step (ii). We anticipate that the classification accuracy of all the four methods compared here will improve if a more sophisticated classifier is used instead of the nearest neighbor. Readers are recommended to see *e.g.* [72] for the performance of DF-LDA + *Support Vector Machines* (SVM), [118] for JD-LDA + Global Feedforward Neural Network, and [26] for JD-LDA + *Radial Basis Function* (RBF) Neural Network.

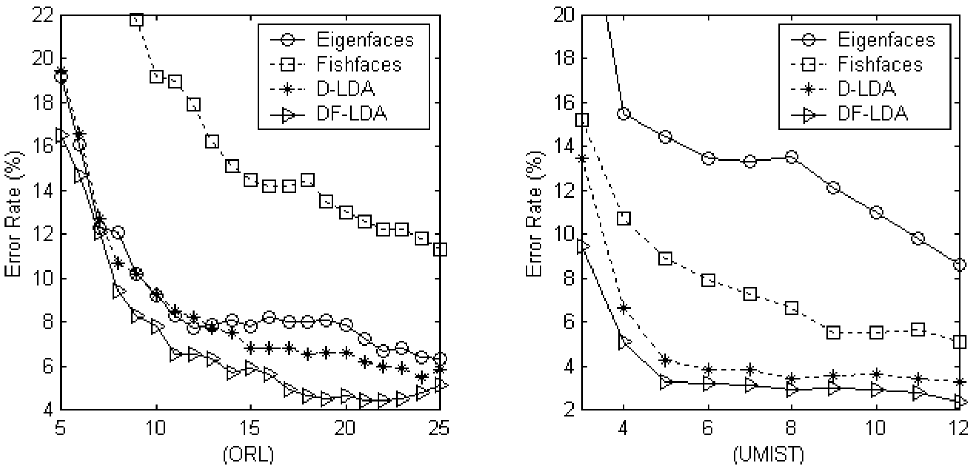

Figure 3.7: Comparison of CERs obtained by the four FR methods as functions of the number of feature vectors, where $w(d) = d^{-12}$ is used in DF-LDA for the ORL, $w(d) = d^{-8}$ for the UMIST, and $r = 20$ for both.

The CER curves obtained by the four methods are shown in Fig.3.7 as functions of the number of feature vectors used (*i.e.* output space dimensionality), M. In the DF-LDA simulation, for simplicity, the regularization parameter $\eta = 1$ is set to simply ensure sufficient regularization, the number of fractional steps used is $r = 20$ and the weight function utilized is $w(d) = d^{-8}$. In this work, we did not attempt to optimize the three parameters in terms of minimizing the resulting CERs. However, it can be seen from Fig.3.7 that the performance of DF-LDA even with such a sub-optimal parameter configuration is still overall superior to that of any other method compared here on both of the two databases. Let α_i and β_i be the CERs of DF-LDA and any one of the other three methods respectively, where i is the number of feature vectors used. We can obtain an average percentage of the CER of DF-LDA over that of the other method by $\mathcal{E} = \sum_{i=5}^{M} (\alpha_i/\beta_i)$ with $M = 25$ for the ORL database and $M = 12$ for the UMIST database. The results summarized in Table 3.1 indicate that the average CER of DF-LDA is approximately 50.5%, 43% and 80% of that of Eigenface, Fisherface and YD-LDA respectively. It is of interest to observe the performance of Eigenfaces vs that of Fisherfaces. Not surprisingly, Eigenfaces outperform Fisherfaces in the ORL database, because Fisherfaces may lose significant discriminant information due to the intermediate PCA step. The similar observation has also been found in [66, 78].

Table 3.1: The average percentage ($\mathcal{E}$) of the CER of DF-LDA over that of the other method.

Methods	Eigenfaces	Fisherfaces	YD-LDA
$\mathcal{E}_{orl}$	74.18%	38.51%	80.03%
$\mathcal{E}_{umist}$	26.75%	47.68%	79.6%
$(\mathcal{E}_{orl} + \mathcal{E}_{umist})/2$	50.47%	43.1%	79.82%

In addition to the regularization parameter η (to be further discussed under a more general framework in Chapter 4), the weighting function $w(d_{ij})$ influences the performance of the DF-LDA method. For different feature extraction tasks, appropriate values for the weighting function exponent should be determined through experimental methods such as leave-one-out using the available training set. However, it appears that there is a set of values for which good results can be obtained for a wide range of applications. Following the recommendation in [70] we examine the performance of the DF-LDA method for $w(d_{ij}) \in \{d^{-4}, d^{-8}, d^{-12}, d^{-16}\}$. Results obtained through the utilization of these weighting functions are depicted in Fig.3.8 where CERs are plotted against the number of feature vectors selected. The lowest CER on the ORL database is approximately 4.0% and it is obtained using a weighting function of $w(d) = d^{-16}$ and a set of only $M = 22$ feature basis vectors. This result is comparable to the best ones reported previously in the literature [60, 61].

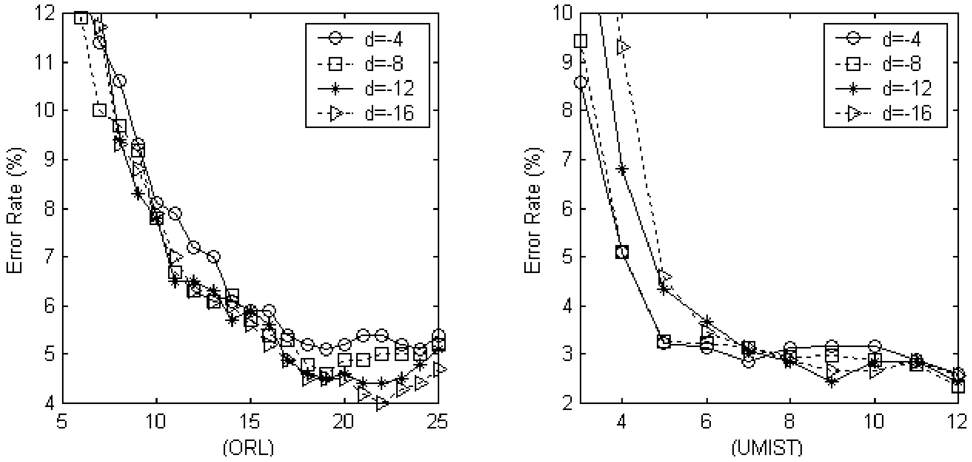

Figure 3.8: Classification error rates (CERs) of DF-LDA as a function of the number of feature vectors used with different weighting functions. In this example, $r = 20$ is used for each weighting function.

3.6 Summary

In this chapter, a new feature extraction method for face recognition tasks has been developed. The method introduced here utilizes the well-known framework of linear discriminant analysis and it can be considered as a generalization of a number of techniques which are commonly in use. The new method utilizes a new variant of D-LDA with regularization to safely remove the null space of the between-class scatter matrix and then applies a fractional-step rotation and re-orientation scheme to enhance the discriminatory power of the obtained SSS-relieved feature space. The effectiveness of the proposed method has been demonstrated through experimentation using two popular face databases.

The DF-LDA method presented here is a linear pattern recognition method. Compared with nonlinear models, a linear model is rather robust against noise and most likely will not overfit. Although it has been shown that distribution of face patterns is highly non convex and complex in most cases, linear methods are still able to provide cost effective solutions to the FR tasks through integration with other strategies, such as the principle of “divide and conquer” in which a large and nonlinear problem is divided into a few smaller and local linear sub-problems. The JD-LDA and DF-LDA methods launch the initiatives for the discriminant learning studies in the context of face recognition. Based on them, more sophisticate FR methods will be developed in the following chapters.

Chapter 4

Quadratic Discriminant Learning with Regularization

4.1 Introduction

Although successful in many circumstances, linear methods including the LDA-based ones often fail to deliver good performance when face patterns are subject to large variations in viewpoints, illumination or facial expression, which result in a highly non convex and complex distribution of face images. The limited success of these methods should be attributed to their linear nature [9]. LDA can be viewed as a special case of the optimal Bayes classifier when each class is subjected to a Gaussian distribution with identical covariance structure. Obviously, the assumption behind LDA is highly restrictive so that LDA often underfits the data in complex FR tasks. As a result, intuitively it is reasonable to assume that a better solution to this inherent nonlinear problem could be achieved using quadratic methods, such as the *Quadratic Discriminant Analysis* (QDA), which relaxes the identical covariance assumption and allows for quadratic discriminant boundaries to be formed. However, compared to LDA solutions, QDA solutions are more susceptible to the Small-Sample-Size (SSS) problem discussed in Section 3.3.1, since the latter requires many more training samples than the former due to the considerably increased number of parameters needed to be estimated [125]. To deal with such a situation, Friedman [32] proposed a *regularized discriminant analysis* (called RDA) technique within the Gaussian framework. The purpose of the regularization is to reduce the variance related to the sample-based estimates for the Gaussian models at the expense of potentially increased bias. Although RDA relieves to a great extent the SSS problem and performs well even when the number of training samples per subject (C_i) is comparable to the dimensionality of the sample space (J), it still fails when $C_i << J$, which is the case in most practical FR applications. For example, RDA cannot be successfully implemented in the experiments reported here where only $C_i \in [2, 7]$ training samples per subject are available while the dimensionality of the space is up to $J = 17154$. RDA's failure in this case should be attributed to two reasons: (1) RDA requires the sample covariance matrices of any single class and their ensemble to be estimated. The estimation results are highly inaccurate when the number of the available samples is far less than their dimensionality. (2) The sizes of these sample covariance matrices are up to 17154×17154. It leads to a challenging computational problem to invert these huge size matrices. It should be added at this point that, it is even difficult to store a 17154×17154 matrix, which requires a memory space of 2245M bytes.

In this chapter, we propose a new regularized quadratic discriminant analysis method called "RD-QDA", which is developed by incorporating the D-LDA technique introduced in last chapter into the RDA framework. The RD-QDA provides a more comprehensive solution and a more general framework to the SSS problem hampering both LDA and QDA than individual D-LDA or RDA. It will be seen that, by adjusting the parameters of the RD-QDA, we can obtain a number of new and traditional discriminant analysis methods such as YD-LDA [46], JD-LDA introduced in Section 3.3.4, *direct* QDA (hereafter D-QDA), *nearest center* (hereafter NC) and *weighted nearest center* (hereafter WNC) classifiers. Also, as will be shown in the experiments reported here, there exists an optimal RD-QDA solution, which greatly outperforms many successful FR approaches including Eigenfaces [120] and YD-LDA [46] as well as the JD-LDA method.

The rest of the chapter is organized as follows. Since RD-QDA is rooted in the optimal Bayes classifier, in Section 4.2, we start the analysis by briefly reviewing Bayes theory. Then a pre-processing step for dimensionality reduction and feature extraction is introduced in Section 4.3. Following that, RD-QDA is developed in Section 4.4, where the relationship of RD-QDA to YD-LDA, JD-LDA and other discriminant analysis methods is also discussed. In Section 4.5, a set of experiments conducted on the FERET database are presented to demonstrate the effectiveness of the proposed methods. Conclusions are drawn in Section 4.6.

4.2 Bayes Classification Theory

LDA has its roots in the optimal Bayes classifier. Given an input pattern $\mathbf{z} \in \mathbb{R}^J$, its class label is assumed to be $y \in \mathbb{Y}$, where $\mathbb{Y} = \{1, \cdots, C\}$ denotes a label set with C classes. Let $P(y = i)$ and $p(\mathbf{z}|y = i)$ be the prior probability of class i and the class-conditional probability density of $\mathbf{z}$ given the class label is i, respectively. Based on the Bayes formula, we have the following *a posteriori* probability $P(y = i|\mathbf{z})$, *i.e.* the probability of the class label being i given that the pattern $\mathbf{z}$ has been measured,

$$P(y = i|\mathbf{z}) = \frac{p(\mathbf{z}|y = i)P(y = i)}{\sum_{i=1}^{C} p(\mathbf{z}|y = i)P(y = i)} \tag{4.1}$$

The Bayes decision rule to classify the unlabeled input pattern $\mathbf{z}$ is then given as,

$$\text{Decide } y = j \text{ if } j = \arg\max_{i \in \mathbb{Y}} P(y = i|\mathbf{z}) \tag{4.2}$$

Eq.4.2 is also known as the *maximum a posteriori* (MAP) rule, and it achieves minimal misclassification risk among all possible decision rules.

The class-conditional densities $p(\mathbf{z}|y = i)$ are seldom known. However, often it is reasonable to assume that $p(\mathbf{z}|y = i)$ is subjected to a Gaussian distribution. Let μ_i and

Σ_i be the mean and covariance matrix of class i. We have

$$p(\mathbf{z}|y=i) = (2\pi)^{-J/2}|\Sigma_i|^{-1/2}\exp\left[-d_i(\mathbf{z})/2\right] \tag{4.3}$$

where

$$d_i(\mathbf{z}) = (\mathbf{z}-\mu_i)^T\Sigma_i^{-1}(\mathbf{z}-\mu_i) \tag{4.4}$$

is the squared Mahalanobis (quadratic) distance from the pattern $\mathbf{z}$ to the center of the class i, μ_i. With the Gaussian assumption, the classification rule of Eq.4.2 turns to

$$\text{Decide } y=j \text{ if } j = \arg\min_{i\in\mathbb{Y}}\left(d_i(\mathbf{z}) + \ln|\Sigma_i| - 2\ln P(y=i)\right) \tag{4.5}$$

The decision rule of Eq.4.5 produces quadratic boundaries to separate different classes in the J-dimensional real space. Consequently, this is also referred to as *quadratic discriminant analysis* (QDA). Often the two statistics (μ_i, Σ_i) are estimated by their sample analogs. For example, for the FR problem stated in Section 3.2, we have

$$\mu_i = \bar{\mathbf{z}}_i = \frac{1}{C_i}\sum_{j=1}^{C_i}\mathbf{z}_{ij} \tag{4.6}$$

$$\Sigma_i = \frac{1}{C_i}\sum_{j=1}^{C_i}(\mathbf{z}_{ij}-\bar{\mathbf{z}}_i)(\mathbf{z}_{ij}-\bar{\mathbf{z}}_i)^T \tag{4.7}$$

LDA can be viewed as a special case of QDA when the covariance structure of all classes are identical, *i.e.* $\Sigma_i = \Sigma$. However, the estimation for either Σ_i or Σ is ill-posed in the small sample size (SSS) settings, giving rise to high variance. This problem becomes extremely severe due to $C_i \ll J$ in FR tasks, where Σ_i is singular with $rank(\Sigma_i) \leq (C_i - 1)$. To deal with such a situation, it is necessary to conduct a pre-processing step for dimensionality reduction, so that a subsequent QDA process can be applied in a low-dimensional subspace, where most significant discriminant information are hoped to be kept, and whose dimensionality is comparable to C_i.

4.3 Determination of a low-dimensional discriminant subspace

Let us consider the appearance-based FR problem stated in Section 3.2. Based on the D-LDA subspace theory discussed in Section 3.3.2, it can be known that the optimal discriminant features exist in the complement space of the null space of the between-class scatter matrix, $\mathbf{S}_b$ (Eq.3.3), which has $M = (C-1)$ nonzero eigenvalues denoted as Λ_b. Let $\mathbf{U}_M = [\mathbf{u}_1, \cdots, \mathbf{u}_M]$ be the M eigenvectors of $\mathbf{S}_b$ corresponding to the M eigenvalues Λ_b. Then the complement space is spanned by $\mathbf{U}_M$, which is furthermore scaled by $\mathbf{H} = \mathbf{U}_M \Lambda_b^{-1/2}$ so as to have $\mathbf{H}^T \mathbf{S}_b \mathbf{H} = \mathbf{I}$, where $\mathbf{I}$ is an $(M \times M)$ identity matrix.

For the purpose of dimensionality reduction, we can project the original face images into the low-dimensional subspace spanned by $\mathbf{H}$ (called $\mathbf{H}$ space hereafter), where most significant discriminant information are retained. The projection is a simple linear mapping: $\mathbf{y}_{ij} = \mathbf{H}^T \mathbf{z}_{ij}$, where $\mathbf{y}_{ij} \in \mathbb{R}^M$ is the obtained feature representation of $\mathbf{z}_{ij}$ with $i = 1, \cdots, C$, $j = 1, \cdots, C_i$. Also, in most FR tasks, the number of face classes C is usually a small value comparable to the number of training samples N, *e.g.* $C = 49$ and $N \in [98, 343]$ in the experiments reported here. Thus, it is reasonable to perform a RDA [32] in the M-dimensional $\mathbf{H}$ space.

4.4 Regularized Direct QDA (RD-QDA)

In the $\mathbf{H}$ space, following the definition given in [32], the regularized sample covariance matrix estimate of class i, hereafter denoted by $\hat{\Sigma}_i(\lambda, \gamma)$, can be expressed as,

$$\hat{\Sigma}_i(\lambda, \gamma) = (1-\gamma)\hat{\Sigma}_i(\lambda) + \frac{\gamma}{M} tr[\hat{\Sigma}_i(\lambda)]\mathbf{I} \tag{4.8}$$

where

$$\hat{\Sigma}_i(\lambda) = \frac{1}{C_i(\lambda)}\left[(1-\lambda)\mathbf{S}_i + \lambda \mathbf{S}\right] \tag{4.9}$$

$$C_i(\lambda) = (1-\lambda)C_i + \lambda N \tag{4.10}$$

$$\mathbf{S}_i = \sum_{j=1}^{C_i} (\mathbf{y}_{ij} - \bar{\mathbf{y}}_i)(\mathbf{y}_{ij} - \bar{\mathbf{y}}_i)^T \tag{4.11}$$

$$\mathbf{S} = \sum_{i=1}^{C} \mathbf{S}_i = N \cdot \mathbf{H}^T \mathbf{S}_w \mathbf{H} \tag{4.12}$$

(λ, γ) is a pair of regularization parameters, and $\bar{\mathbf{y}}_i = \mathbf{H}^T \bar{\mathbf{z}}_i$ is the projection of the mean of class i in the $\mathbf{H}$ space.

In the FR procedure, any input query image $\mathbf{z}$ is firstly projected into the $\mathbf{H}$ space: $\mathbf{y} = \mathbf{H}^T\mathbf{z}$. Its class label i^* then can be inferred based on the QDA classification rule (Eq.4.5) through

$$i^* = \arg\min_{i \in \mathbb{Y}} \left(d_i(\mathbf{y}) + \ln \left| \hat{\Sigma}_i(\lambda, \gamma) \right| - 2 \ln \pi_i \right) \tag{4.13}$$

where $\pi_i = C_i/N$ is the estimate of the prior probability of class i, and

$$d_i(\mathbf{y}) = (\mathbf{y} - \bar{\mathbf{y}}_i)^T \hat{\Sigma}_i^{-1}(\lambda, \gamma)(\mathbf{y} - \bar{\mathbf{y}}_i) \tag{4.14}$$

is the squared Mahalanobis (quadratic) distance between the query $\mathbf{y}$ and the ith class center $\bar{\mathbf{y}}_i$.

The regularization parameter λ $(0 \leq \lambda \leq 1)$ controls the amount that the $\mathbf{S}_i$ are shrunk toward $\mathbf{S}$, the within-class scatter matrix of $\mathbf{y}_{ij}$. The other parameter γ $(0 \leq \gamma \leq 1)$ controls shrinkage of the class covariance matrix estimates toward a multiple of the identity matrix. Under the regularization scheme, a QDA can be performed without experiencing high variance of the plug-in estimates even when the dimensionality of the $\mathbf{H}$ space is comparable to the number of available training samples. We refer to the approach as *regularized direct* QDA, abbreviated as RD-QDA.

Since the RD-QDA is derived from the D-LDA and the RDA, it has close relationship with a series of traditional discriminant analysis methods. Firstly, the four corners defining the extremes of the (λ, γ) plane represent four well-known classification algorithms, as summarized in Table 4.1, where the prefix 'D-' means that all these methods are developed in the $\mathbf{H}$ space derived from the D-LDA technique. Based on the Fisher's criterion

Table 4.1: A series of discriminant analysis algorithms derived from RD-QDA.

Algs.	D-NC	D-WNC	D-QDA	YD-LDA	JD-LDA
λ	1	0	0	1	1
γ	1	1	0	0	γ_J
$\hat{\Sigma}_i(\lambda,\gamma)$	$\frac{1}{M}tr[\frac{\mathbf{S}}{N}]\mathbf{I}$	$\frac{1}{M}tr[\frac{\mathbf{S}_i}{C_i}]\mathbf{I}$	$\frac{\mathbf{S}_i}{C_i}$	$\frac{\mathbf{S}}{N}$	$\alpha\left(\frac{\mathbf{S}}{N}+\eta\mathbf{I}\right)$

of Eq.3.2 used in YD-LDA, it is obvious that the YD-LDA feature extractor followed by a nearest center classfier is actually a standard LDA classification rule implemented in the $\mathbf{H}$ space. Also, in order to introduce certain regularization, JD-LDA as mentioned earlier utilizes a modified optimization criterion seeking to maximize the ratio: $\frac{|\Psi^T\mathbf{S}_b\Psi|}{|\Psi^T(\eta\mathbf{S}_b+\mathbf{S}_w)\Psi|}$, whose kernel $\frac{\mathbf{S}_b}{\eta\mathbf{S}_b+\mathbf{S}_w}$ has the form: $\frac{\mathbf{I}}{\eta\mathbf{I}+\mathbf{H}^T\mathbf{S}_w\mathbf{H}}$ after projection in the $\mathbf{H}$ space. Meanwhile, we have $\hat{\Sigma}_i(\lambda,\gamma)=\alpha\left(\eta\mathbf{I}+\frac{\mathbf{S}}{N}\right)=\alpha\left(\eta\mathbf{I}+\mathbf{H}^T\mathbf{S}_w\mathbf{H}\right)$ when $(\lambda=1,\gamma=\gamma_J)$, where $\alpha=\left(\frac{tr[\mathbf{S}/N]}{tr[\mathbf{S}/N]+\eta M}\right)$ and $\gamma_J=\frac{\eta M}{tr[\mathbf{S}/N]+\eta M}$ both are constant for a given η value and a given training sample set. In this situation, it is not difficult to see that RD-QDA is equivalent to JD-LDA followed by a nearest center classifier. In addition, a set of intermediate discriminant classifiers between the five traditional ones can be obtained when we smoothly vary the two regularization parameters in their domains.

The objective of RD-QDA is to find the optimal (λ^*,γ^*) that give the best correct recognition rate for a particular FR task. The optimization of (λ,γ) is associated with the issue of model selection. A popular solution for this issue is the cross-validation approach, for example, the leave-one-out method. The basic idea behind the method is that generalize a model, RD-QDA$_v(\lambda,\gamma)$, based on the given (λ,γ) values and $(N-1)$ training samples exclusive of $\mathbf{z}_v\in\mathcal{Z}$, and then apply the model to classify the excluded sample $\mathbf{z}_v$. Each of the training sample $\mathbf{z}_v$ $(v=1,\cdots,N)$ is in turn taken out and then classified in this manner. The resulting misclassification loss averaged over the training sample is then used as an estimate of future or generalization classification error. Let

$\mathcal{E}_v(\lambda,\gamma)=0$ if the RD-QDA$_v(\lambda,\gamma)$ model correctly classifies its corresponding sample $\mathbf{z}_v$, and $\mathcal{E}_v(\lambda,\gamma)=1$ otherwise. The optimal (λ^*,γ^*) found by the leave-one-out method is the one that has

$$(\lambda^*,\gamma^*)=\arg\min_{(\lambda,\gamma)}\left(\frac{1}{N}\sum_{v=1}^{N}\mathcal{E}_v(\lambda,\gamma)\right) \tag{4.15}$$

4.5 Experimental Results

4.5.1 The Face Recognition Evaluation Design

A set of experiments is included in this chapter to assess the performance of the proposed RD-QDA method. To illustrate the high complexity of the face pattern distribution, the middle-size FERET evaluation database, $\mathcal{G}_1$ depicted in Section 2.3.2 is used in the experiments. The database $\mathcal{G}_1$ consists of 606 gray-scale images of 49 subjects, each one having more than 10 samples. For computational purposes, each image is finally represented as a column vector of length $J=17154$ prior to the recognition stage.

The number of available training samples per subject, C_i, has a significant influence on the plug-in covariance matrix estimates used in all discriminant analysis methods. For simplicity, we assume that each subject has the same number of training samples, $C_i=L$. To study the sensitivity of the *correct recognition rate* (CRR) measure to L, six tests were performed with various values of L ranging from $L=2$ to $L=7$. For a particular L, the FERET subset $\mathcal{G}_1$ is randomly partitioned into two datasets: a training set and a test set. The training set is composed of $(L\times 49)$ samples: L images per subject were randomly chosen. The remaining $(606-L\times 49)$ images are used to form the test set. There is no overlapping between the two. To enhance the accuracy of the assessment, five runs of such a partition were executed, and all of the CRRs reported later have been averaged over the five runs. As mentioned in Section 2.3.2, there is no difference between the two performance measures, CRR and CER (*classification error rate* used in last Chapter), but CRR=(1-CER).

4.5.2 The FR Performance Comparison

In addition to RD-QDA and its special cases listed in Table 4.1, the Eigenfaces method [120] was also implemented to provide a performance baseline as it did in the FERET competitions [93,94]. For JD-LDA, two special cases should be noted. One ($\eta = 0$) is JD-LDA with no regularization. In this case, JD-LDA is equivalent to YD-LDA or RD-LDA with ($\lambda = 1, \gamma = 0$). Another ($\eta = 1$) is the strong regularization case, called "JD-LDA.1" hereafter, which is equivalent to RD-QDA with $\left(\lambda = 1, \gamma = \frac{M}{tr[\mathbf{S}/N]+M}\right)$. The remaining cases ($\eta \in (0,1)$) of JD-LDA correspond to RD-LDA with $\lambda = 1$ and $\gamma \in \left(0, \frac{M}{tr[\mathbf{S}/N]+M}\right)$.

The testing grid of (λ, γ) values was defined by the outer product of $\lambda = [10^{-4} : 0.0099 : 1]$ and $\gamma = [10^{-4} : 0.0099 : 1]$, where $[10^{-4} : 0.0099 : 1]$ denotes a spaced vector consisting of 102 elements from 10^{-4} to 1 with step 0.0099: $[10^{-4}, 10^{-4} + 0.0099, 10^{-4} + 2 \times 0.0099, \cdots, 10^{-4} + 101 \times 0.0099]$, and both of λ and γ started from 10^{-4} instead of 0 in case $\mathbf{S}_i$ or even $\mathbf{S}(= N \cdot \mathbf{H}^T \mathbf{S}_w \mathbf{H})$ is singular so that some methods such as D-QDA and YD-LDA cannot be carried out. The CRRs obtained by RD-QDA in the grid are depicted in Fig.4.1. Since most peaks or valleys occur around the four corners, four 2D side faces of Fig.4.1 (only four representative cases $L = 2, 3, 4, 6$ are selected) are shown in Figs.4.2-4.3 for a clearer view. Also, a quantitative comparison of the best found CRRs and their corresponding standard deviations (STDs) (arising from the average of the five runs) obtained by Eigenfaces, those depicted in Table 4.1, and RD-QDA with corresponding parameters, is summarized in Table 4.2. It should be noted at this point that the "best" CRRs are only applicable for RD-QDA and PCA (or Eigenfaces). Since all other methods compared in Table 4.2 can be considered special cases of the RD-QDA method, no optimal regularization parameters can be determined for them. Their CRRs reported are the results of RD-QDA obtained using the regularization parameters shown in Table 4.1. Since the CRR of PCA is a function of the number of the Eigenfaces, the "best CRR" reported for PCA is the one with the best found Eigenfaces number, denoted as m^*.

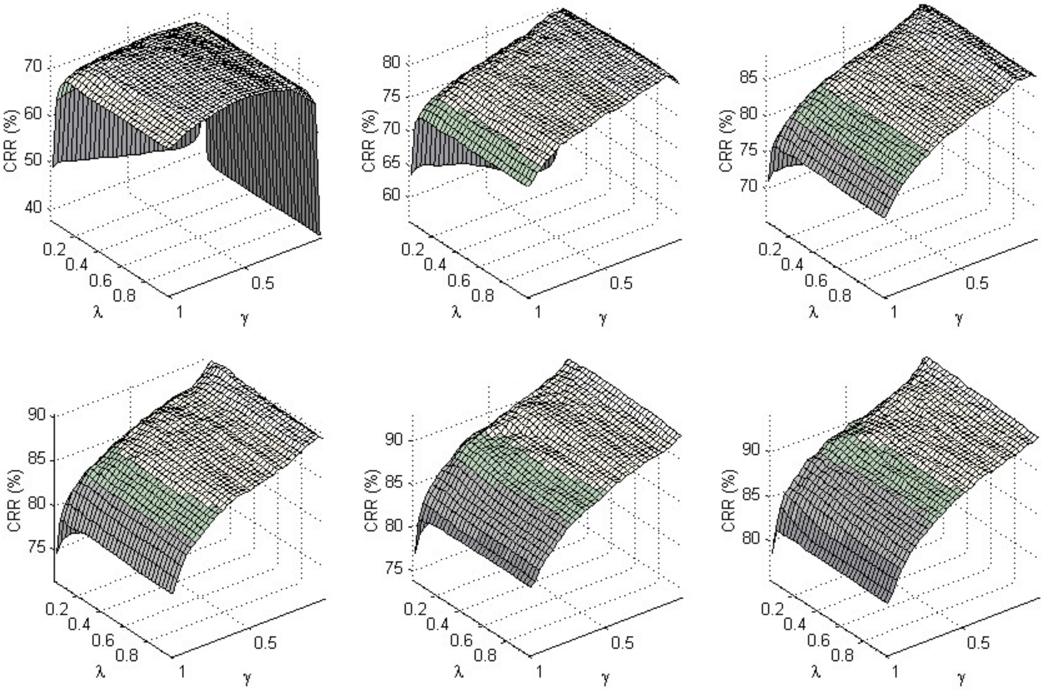

Figure 4.1: CRRs obtained by RD-QDA as a function of (λ, γ). **Top**: $L = 2, 3, 4$; **Bottom**: $L = 5, 6, 7$.

The parameter λ controls the degree of shrinkage of the individual class covariance matrix estimate $\mathbf{S}_i$ toward the within-class scatter matrix of the whole training set $(\mathbf{H}^T\mathbf{S}_w\mathbf{H})$. Varying the values of λ within $[0, 1]$ leads to a set of intermediate classifiers between D-QDA and YD-LDA. In theory, D-QDA should be the best performer among the methods evaluated here if sufficient training samples are available. It can be observed at this point from Fig.4.1 and Table 4.2 that the CRR peaks gradually moved from the central area toward the corner $(0, 0)$ that is the case of D-QDA as L increases. Small values of λ have been good enough for the regularization requirement in many cases $(L \geq 3)$ as shown in Fig.4.3:Left.

However, it is also can be seen from Fig.4.2:Right and Table 4.2 that both of D-QDA and YD-LDA performed poorly when $L = 2$. This should be attributed to the high variance in estimates of $\mathbf{S}_i$ and $\mathbf{S}$ due to insufficient training samples. In these cases, $\mathbf{S}_i$

Table 4.2: A comparison of best found CRRs/STDs (%).

L	2	3	4	5	6	7
PCA	59.8/0.93	67.8/2.50	73.0/2.62	75.8/3.27	81.3/1.35	83.7/1.77
(m^*)	67	103	131	151	183	227
D-NC	67.8/0.62	72.3/1.76	75.3/1.65	77.3/2.57	80.2/1.35	80.5/1.12
D-WNC	46.9/3.38	61.7/4.42	68.7/2.54	72.1/2.59	73.9/2.47	75.6/0.62
D-QDA	57.0/2.91	79.3/2.41	87.2/2.70	89.2/1.73	92.4/1.35	93.8/1.51
YD-LDA	37.8/7.03	79.5/3.24	87.8/2.70	89.5/1.71	92.4/1.23	93.5/1.24
JD-LDA.1	70.7/0.51	77.4/1.57	82.8/1.85	85.7/2.09	88.1/1.71	89.4/1.04
$(\gamma_J(\eta = 1))$	0.84	0.75	0.69	0.65	0.61	0.59
RD-QDA	73.2/1.05	81.6/1.85	88.5/2.22	90.4/1.38	93.2/1.35	94.4/0.87
(λ^*)	0.93	0.93	0.35	0.11	0.26	0.07
(γ^*)	0.47	0.10	0.07	0.01	1e-4	1e-4

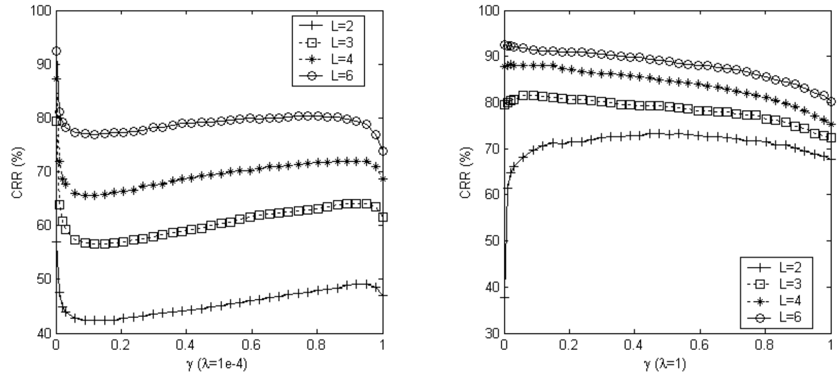

Figure 4.2: CRRs as a function of γ with fixed λ values.

and even $\mathbf{S}$ are singular or close to singular, and the resulting effect is to dramatically exaggerate the importance associated with the eigenvectors corresponding to the smallest eigenvalues. Against the effect, the introduction of another parameter γ helps to decrease

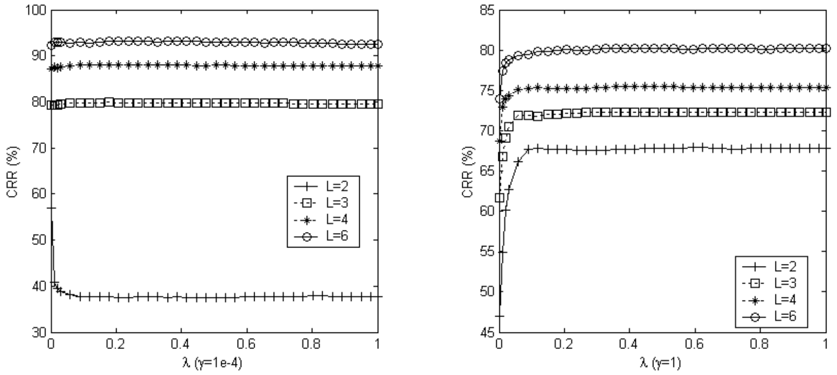

Figure 4.3: CRRs as a function of λ with fixed γ values.

the larger eigenvalues and increase the smaller ones, thereby counteracting for some extent the bias. This is also why JD-LDA.1 outperforms YD-LDA when L is small. Although JD-LDA.1 seems to be a little over-regularized compared to the optimal (λ^*, γ^*), the method almost guarantees a stable sub-optimal solution. A CRR difference of 4.5% on average over the range $L \in [2, 7]$ has been observed between the top performer RD-QDA(λ^*, γ^*) and JD-LDA.1. Also, it should be noted that in addition to L, the strength of regularization is dependent on the number of subjects, C, more accurate to say, how much the number of training examples per subject L is less than the number of subjects C. Stronger regularization is required when the extent of $L < C$ is larger. However, often the correct values of the regularization parameters are difficult to be estimated in advance. Therefore, it can be concluded that JD-LDA.1 should be preferred when insufficient prior information about the training samples is available and a cost effective processing method is sought. In addition to the CRRs, it can been seen from the STDs depicted in Table 4.2 that due to the introduction of regularization both JD-LDA.1 and RD-QDA are quite stable compared to other methods across various SSS setting scenarios.

For each of the methods evaluated here, the FR simulation process consists of 1) a training stage that includes all operations performed in the training set, *e.g.* the computations of $\mathbf{H}$, $\hat{\Sigma}_i(\lambda, \gamma)$ and $\bar{\mathbf{y}}_i$; 2) a test stage for the CRRs determination. The computational times consumed by these methods in the two stages are reported in Table 4.3. T_{trn} and T_{tst} are the amounts of time spent on training and test respectively, and the times of RD-LDA are the average ones consumed in a single point of the (λ, γ) grid. The simulation studies reported in this work were implemented on a personal computer system equipped with a 2.0GHz Intel Pentium 4 processor and 1.0 GB RAM. All programs are written in Matlab v6.5 and executed in MS Windows 2000. Since the discriminant analysis methods listed in Table 4.1 are some special cases of the RD-QDA method, it can been seen from Table 4.3 that all these methods have similar computational requirements. D-NC and D-WNC are slightly faster than the others since each class covariance matrix $\hat{\Sigma}_i(\lambda, \gamma)$ in the two methods is shrunk towards an identity matrix multiplied by a scalar. As a result, the computations of $\hat{\Sigma}_i(\lambda, \gamma)$ and $d_i(\mathbf{y})$ are simplified. It can be also observed from Table 4.3 that the PCA method is much slower compared to those discriminant analysis methods, especially as L increases. This is due to the fact that the computational complexity of all these methods is determined to a great extent by the dimensionality of the feature space, where most computations are conducted. The feature space for all the discriminant analysis methods is the $\mathbf{H}$ space, which has a dimensionality of $M = 48$, much smaller than the dimensionality of the best found Eigenfaces space, m^*, which is between 67 and 227 as shown in Table 4.2. Although RD-LDA is quite effective when it is performed in a single point of the (λ, γ) grid, the determination of its optimal parameter values is computationally demanding as it is based on exhaustive searches in the entire grid. For example, the 102×102 grid used here requires 10404 RD-QDAs to be computed for each run. However, compared to the exhaustive search, it is even more computationally expensive to use the leave-one-out method, which requires a computational cost of $(L \times 49 \times 10404)$ RD-QDAs for each run. Therefore, a fast and cost effective RD-QDA parameter optimization method will be the focus of future research.

Table 4.3: A comparison of computational times, $T_{trn} + T_{tst}$ (Seconds).

L	2	3	4	5	6	7
PCA	1.1+2.5	2.1+3.1	3.5+3.6	5.0+3.9	7.2+4.2	9.9+6.1
D-NC	0.7+1.3	0.9+1.2	1.0+1.1	1.2+0.9	1.4+0.8	1.5+0.7
D-WNC	0.7+1.4	0.9+1.2	1.1+1.1	1.2+1.0	1.4+0.9	1.5+0.7
D-QDA	0.8+1.6	0.9+1.4	1.1+1.3	1.3+1.1	1.4+1.0	1.6+0.8
YD-LDA	0.8+1.6	0.9+1.4	1.1+1.3	1.3+1.1	1.4+1.0	1.6+0.8
JD-LDA.1	0.8+1.6	0.9+1.4	1.1+1.3	1.3+1.1	1.4+1.0	1.6+0.8
RD-QDA	0.8+1.6	0.9+1.4	1.1+1.3	1.3+1.1	1.4+1.0	1.6+0.8

4.6 Summary

A new method for face recognition has been introduced in this chapter. The proposed method takes advantages of the D-LDA and regularized QDA techniques to effectively address the SSS and nonlinear(quadratic) problem commonly encountered in FR tasks. The D-LDA technique is utilized to map the original face patterns to a low-dimensional discriminant feature space, where a regularized QDA is then readily applied.

The regularization strategy used here provides a balance between the variance and the bias in sample-based estimates, and this significantly relieves the SSS problem. Also, it was shown that a series of traditional discriminant analysis methods including the recently introduced YD-LDA and JD-LDA can be derived from the proposed RD-QDA framework simply by adjusting the regularization parameters. Experimental results indicate that the best found RD-QDA solution outperforms the Eigenfaces method as well as other traditional or derived here discriminant analysis methods across various SSS settings. On the other hand, the process of how to effectively find the optimal balance between the variance and the bias currently remains as an open research problem. Rather than the leave-one-out method, a promising direction to approach this problem is to take advantage of the variable called “Learning Difficulty Degree”, which is introduced in Section 6.6.4 as a measure for the difficult extent of a SSS learning task.

In summary, RD-QDA can be seen as a general pattern recognition method capable of addressing both the nonlinear(quadratic) and SSS problems. We expect that in addition to FR, RD-QDA will provide excellent performance in applications, such as image/video indexing, retrieval, and classification.

Chapter 5

Nonlinear Discriminant Learning with Kernel Machines

5.1 Introduction

Both linear and quadratic discriminant analysis are based on the Gaussian framework, where each class is assumed to be subjected to a Gaussian distribution. Unfortunately, as highlighted in the introduction chapter, the distribution of face patterns in the real-world is far more complicated than Gaussian. For example, it is generally believed that a highly non-convex distribution is obtained when face patterns are subjected to large variations in viewpoints [9]. In general, such a complex distribution can be handled either by *globally nonlinear models* or by *a mixture of locally linear models* (AMLLM). We first investigate the globally nonlinear methods as the focus of this chapter, and then discuss the AMLLM-based methods in the next two chapters. Compared to the AMLIM-based approach, the globally nonlinear approach has some theoretical advantages, for example, its ability in generalizing a more compact feature representation.

Recently, the so-called "kernel machine" technique has become one of the most popular tools to design globally nonlinear algorithms in the communities of machine learning and pattern recognition [88, 100, 107, 123]. The idea behind the kernel-based learning methods is to construct a nonlinear mapping from the input space ($\mathbb{R}^J$) to an implicit high-dimensional feature space ($\mathbb{F}$) using a kernel function $\phi : \mathbf{z} \in \mathbb{R}^J \rightarrow \phi(\mathbf{z}) \in \mathbb{F}$. In the feature space, it is hoped that the distribution of the mapped data is linearized and simplified, so that traditional linear methods can perform well. A problem with the concept is that the dimensionality of the feature space may be arbitrarily large, possibly infinite, resulting in difficulty in computation. Fortunately, with some mathematical tricks, the exact $\phi(\mathbf{z})$ is not needed, and the nonlinear mapping can be performed implicitly in $\mathbb{R}^J$ by replacing dot products in the feature space $\mathbb{F}$ with a kernel function defined in the input space $\mathbb{R}^J$, $k(\mathbf{z}_i, \mathbf{z}_j) = \phi(\mathbf{z}_i) \cdot \phi(\mathbf{z}_j)$. The kernel functions that can be used for the task have to satisfy Mercer's condition [82], and typical examples include polynomial function, radial basis function (RBF) and multi-layer perceptrons. The field of the kernel machines is now extremely active. Examples based on such a design include *support vector machines* (SVM) [19, 106, 123], *kernel Mahalanobis* (KM) Distance [100], *kernel* PCA (KPCA) [3], *kernel* ICA (KICA) [4], and *Generalized Discriminant Analysis* (GDA, also known as kernel LDA) [6].

In this chapter, motivated by the success that these kernel-based methods have obtained in various regression and classification tasks, we propose a new kernel discriminant analysis algorithm to solve nonlinear pattern recognition problems specifically in SSS situations, such as face recognition. The algorithm is developed by attempting to *kernellize* the JD-LDA method introduced in Section 3.3.4, in other words, implement a JD-LDA process in the feature space $\mathbb{F}$. It is therefore supposed that the proposed algorithm is able to generalize the strengths of both the JD-LDA method and the kernel machines. Compared to the existing kernel discriminant analysis methods developed from the traditional LDA, such as GDA [6], the algorithm proposed here is more robust against the SSS problem, which should be noted to become worse in the feature space $\mathbb{F}$ due to the significantly increased dimensionality than in the original sample space $\mathbb{R}^J$. Also, it will be shown that the proposed algorithm reduces to JD-LDA when the feature space is linearly related to the sample space. Thus, following the notation conventions of kernel methods, we call the algorithm *kernel* JD-LDA method, abbreviated as "KDDA" hereafter.

The rest of the chapter is organized as follows. Firstly, the principle behind the kernel machines is introduced in Section 5.2. Then, two commonly used kernel-based feature extraction methods, KPCA and GDA, are briefly reviewed in Sections 5.3-5.4. Following that, KDDA is introduced and analyzed in Section 5.5. The relationship of KDDA to JD-LDA and GDA is also discussed in Section 5.6. In Section 5.7, two sets of experiments are presented to demonstrate the effectiveness of the KDDA algorithm in the case of highly non convex, highly complicated face pattern distributions. Conclusions are drawn in Section 5.8.

5.2 Sample Space *vs* Kernel Feature Space

The kernel machines provide an elegant way of dealing with nonlinear algorithms by reducing them to linear ones in some high-dimensional feature space $\mathbb{F}$ nonlinearly related to the sample space $\mathbb{R}^J$:

$$\phi : \mathbf{z} \in \mathbb{R}^J \to \phi(\mathbf{z}) \in \mathbb{F} \tag{5.1}$$

The idea can be illustrated by the toy example depicted in Fig.5.1, where two-dimensional input samples, say $\mathbf{z} = [z_1, z_2]$, are mapped to a three-dimensional feature space through a nonlinear transform: $\phi : \mathbf{z} = [z_1, z_2] \to \phi(\mathbf{z}) = [x_1, x_2, x_3] := \left[z_1^2, \sqrt{2}z_1 z_2, z_2^2\right]$. It can be seen from Fig.5.1 that in the sample space, a nonlinear ellipsoidal decision boundary is needed to separate classes A and B, in contrast with this, the two classes become linearly separable in the higher-dimensional feature space.

The feature space $\mathbb{F}$ could be considered as a "linearization space" [2], however, to reach the purpose, its dimensionality could be arbitrarily large, possibly infinite. Fortunately, the exact $\phi(\mathbf{z})$ is not needed and the feature space can become implicit by using

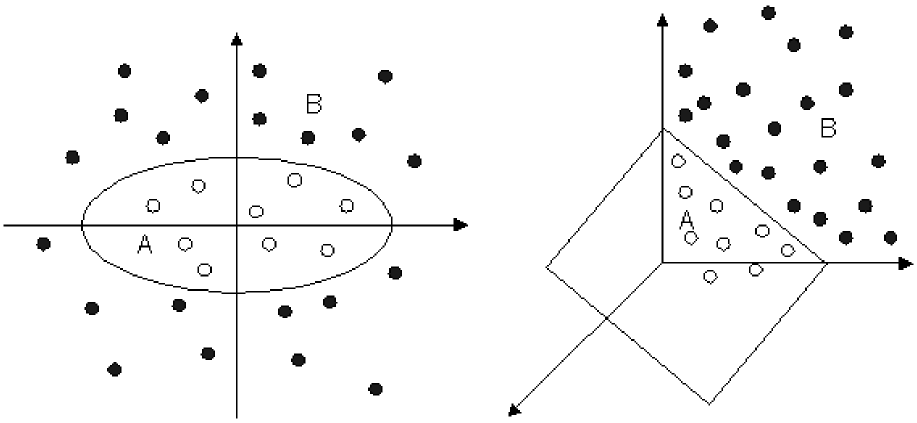

Figure 5.1: A toy example of two-class pattern classification problem. **Left:** samples lie in the 2-D input space, where it needs a nonlinear ellipsoidal decision boundary to separate classes A and B. **Right:** Samples are mapped to a 3-D feature space, where a linear hyperplane can separate the two classes.

kernel machine methods. The trick behind the kernel methods is to replace dot products in $\mathbb{F}$ with a kernel function in the input space $\mathbb{R}^J$ so that the nonlinear mapping is performed implicitly in $\mathbb{R}^J$. Let us come back to the toy example of Fig.5.1, where the feature space is spanned by the second-order monomials of the input sample. Let $\mathbf{z}_i \in \mathbb{R}^2$ and $\mathbf{z}_j \in \mathbb{R}^2$ be two examples in the input space, and the dot product of their feature vectors $\phi(\mathbf{z}_i) \in \mathbb{F}$ and $\phi(\mathbf{z}_j) \in \mathbb{F}$ can be computed by the following kernel function, $k(\mathbf{z}_i, \mathbf{z}_j)$, defined in $\mathbb{R}^2$,

$$\begin{aligned} \phi(\mathbf{z}_i) \cdot \phi(\mathbf{z}_j) &= \left[z_{i1}^2, \sqrt{2} z_{i1} z_{i2}, z_{i2}^2\right] \left[z_{j1}^2, \sqrt{2} z_{j1} z_{j2}, z_{j2}^2\right]^T \\ &= \left(\left[z_{i1}, z_{i2}\right] \left[z_{j1}, z_{j2}\right]^T\right)^2 \\ &= (\mathbf{z}_i \cdot \mathbf{z}_j)^2 =: k(\mathbf{z}_i, \mathbf{z}_j) \end{aligned} \tag{5.2}$$

In most cases, the dimensionality of the feature space is extremely high, possibly infinite, for example those mapped by RBF kernels. As a result, *the central issue to generalize a linear learning algorithm to its kernel version is to reformulate all the computations of the algorithm in the feature space in the form of dot products.* Based on the properties of the kernel functions used, the kernel generation gives rise to neural-network structures, splines, Gaussian, Polynomial or Fourier expansions, *etc.* . Any function satisfying Mercer's condition [82] can be used as a kernel. Table 5.1 lists some of the most widely used kernel functions, and more sophisticated kernels can be found in [48, 106, 112, 124, 135].

Table 5.1: Some of the most widely used kernel functions, where $\mathbf{z}_1 \in \mathbb{R}^J$ and $\mathbf{z}_2 \in \mathbb{R}^J$.

Gaussian RBF	$k(\mathbf{z}_1, \mathbf{z}_2) = \exp\left(\frac{-\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{\sigma^2}\right)$, $\sigma \in \mathbb{R}$
Polynomial	$k(\mathbf{z}_1, \mathbf{z}_2) = (a(\mathbf{z}_1 \cdot \mathbf{z}_2) + b)^d$, $a \in \mathbb{R}$, $b \in \mathbb{R}$, $d \in \mathbb{N}$
Sigmoidal	$k(\mathbf{z}_1, \mathbf{z}_2) = \tanh(a(\mathbf{z}_1 \cdot \mathbf{z}_2) + b)$, $a \in \mathbb{R}$, $b \in \mathbb{R}$
Inverse multiquadric	$\frac{1}{\sqrt{\|\mathbf{z}_1 - \mathbf{z}_2\|^2 + \sigma^2}}$, $\sigma \in \mathbb{R}$

5.3 Kernel Principal Component Analysis (KPCA)

To find principal components of a non convex distribution, the classic PCA has been generalized to the *kernel* PCA (KPCA) [3]. Given the appearance-based FR problem stated in Section 3.2, let $\phi : \mathbf{z} \in \mathbb{R}^J \to \phi(\mathbf{z}) \in \mathbb{F}$ be a nonlinear mapping from the input face image space to a high-dimensional feature space $\mathbb{F}$. The covariance matrix or total scatter matrix of the training sample in the feature space $\mathbb{F}$ can be expressed as

$$\tilde{\mathbf{S}}_{cov} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} (\phi(\mathbf{z}_{ij}) - \bar{\phi})(\phi(\mathbf{z}_{ij}) - \bar{\phi})^T \tag{5.3}$$

where $\bar{\phi} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} \phi(\mathbf{z}_{ij})$ is the average of the ensemble in $\mathbb{F}$. The KPCA is actually a classic PCA performed in the feature space $\mathbb{F}$. Let $\tilde{\mathbf{g}}_m \in \mathbb{F}$ $(m = 1, 2, \cdots, M)$ be the first M most significant eigenvectors of $\tilde{\mathbf{S}}_{cov}$, and they form a low-dimensional subspace, called "KPCA subspace" in $\mathbb{F}$. All these $\{\tilde{\mathbf{g}}_m\}_{m=1}^{M}$ lie in the span of $\{\phi(\mathbf{z}_{ij})\}_{\mathbf{z}_{ij} \in \mathcal{Z}}$, and have $\tilde{\mathbf{g}}_m = \sum_{i=1}^{C} \sum_{j=1}^{C_i} a_{ij} \phi(\mathbf{z}_{ij})$, where a_{ij} are the linear combination coefficients. For any face pattern $\mathbf{z}$, its nonlinear principal components can be obtained by the dot product,

$(\tilde{\mathbf{g}}_m \cdot (\phi(\mathbf{z}) - \bar{\phi}))$, computed indirectly through the kernel function $k()$. When $\phi(\mathbf{z}) = \mathbf{z}$, KPCA reduces to PCA, and the KPCA subspace is equivalent to the so-called “Eigenface space” introduced in [120].

5.4 Generalized Discriminant Analysis (GDA)

As such, *Generalized Discriminant Analysis* (GDA, also known as kernel LDA) [6] is a process to extract a nonlinear discriminant feature representation by performing a classic LDA in the high-dimensional feature space $\mathbb{F}$. Let $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ be the between- and within-class scatter matrices in the feature space $\mathbb{F}$ respectively, and they have following expressions:

$$\tilde{\mathbf{S}}_b = \frac{1}{N} \sum_{i=1}^{C} C_i (\bar{\phi}_i - \bar{\phi})(\bar{\phi}_i - \bar{\phi})^T \tag{5.4}$$

$$\tilde{\mathbf{S}}_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} (\phi(\mathbf{z}_{ij}) - \bar{\phi}_i)(\phi(\mathbf{z}_{ij}) - \bar{\phi}_i)^T \tag{5.5}$$

where $\bar{\phi}_i = \frac{1}{C_i} \sum_{j=1}^{C_i} \phi(\mathbf{z}_{ij})$ is the mean of class $\mathcal{Z}_i$. In the same way as LDA, GDA determines a set of optimal nonlinear discriminant basis vectors by maximizing the standard Fisher's criterion:

$$\tilde{\Psi} = \arg\max_{\tilde{\Psi}} \frac{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}\right|}{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_w \tilde{\Psi}\right|}, \quad \tilde{\Psi} = [\tilde{\psi}_1, \cdots, \tilde{\psi}_M], \quad \tilde{\psi}_m \in \mathbb{F} \tag{5.6}$$

Due to the extremely high dimensionality of $\mathbb{F}$, *the SSS problem is introduced essentially during the optimization process of Eq.5.6.* However, GDA, following traditional subspace approach, attempts to solve the SSS problem by removing the null space of $\tilde{\mathbf{S}}_w$, as was done in the Fisherfaces method [7]. As a result, it can be known from the analysis given in Section 3.3.2 that some significant discriminant information that may exist in the null space is lost inevitably due to such a process.

5.5 Kernel JD-LDA (KDDA)

To overcome the problems with the GDA method in the SSS situations, a kernel version of JD-LDA, named KDDA, is developed here.

5.5.1 Eigen-analysis of $\tilde{\mathbf{S}}_b$ in the Feature Space

Following the JD-LDA framework, we start by solving the eigenvalue problem of $\tilde{\mathbf{S}}_b$, which can be rewritten here as follows,

$$\tilde{\mathbf{S}}_b = \sum_{i=1}^{C} \left(\sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right) \right) \left(\sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right) \right)^T = \sum_{i=1}^{C} \tilde{\bar{\phi}}_i \tilde{\bar{\phi}}_i^T = \tilde{\Phi}_b \tilde{\Phi}_b^T \tag{5.7}$$

where $\tilde{\bar{\phi}}_i = \sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right)$, and $\tilde{\Phi}_b = \left[\tilde{\bar{\phi}}_1, \cdots, \tilde{\bar{\phi}}_c \right]$. Since the dimensionality of the feature space $\mathbb{F}$, denoted as J', could be arbitrarily large or possibly infinite, it is intractable to directly compute the eigenvectors of the $(J' \times J')$ matrix $\tilde{\mathbf{S}}_b$. Fortunately, the first m $(\leq C-1)$ most significant eigenvectors of $\tilde{\mathbf{S}}_b$, corresponding to non-zero eigenvalues, can be indirectly derived from the eigenvectors of the matrix $\tilde{\Phi}_b^T \tilde{\Phi}_b$ (with size $C \times C$) (see Section 3.3.4 for details). Computing $\tilde{\Phi}_b^T \tilde{\Phi}_b$, requires dot product evaluation in $\mathbb{F}$. This can be done in a manner similar to the one used in SVM, KPCA and GDA by utilizing kernel methods. For any $\phi(\mathbf{z}_i), \phi(\mathbf{z}_j) \in \mathbb{F}$, we assume that there exists a kernel function $k(\cdot)$ such that $k(\mathbf{z}_i, \mathbf{z}_j) = \phi(\mathbf{z}_i) \cdot \phi(\mathbf{z}_j)$. The introduction of the kernel function allows us to avoid the explicit evaluation of the mapping ϕ.

Using the kernel function, for two arbitrary classes $\mathcal{Z}_l$ and $\mathcal{Z}_h$, a $C_l \times C_h$ dot product matrix K_{lh} can be defined as:

$$K_{lh} = (k_{ij})_{\substack{i=1,\cdots,C_l \\ j=1,\cdots,C_h}} \tag{5.8}$$

where $k_{ij} = k(\mathbf{z}_{li}, \mathbf{z}_{hj}) = \phi_{li} \cdot \phi_{hj}$, $\phi_{li} = \phi(\mathbf{z}_{li})$ and $\phi_{hj} = \phi(\mathbf{z}_{hj})$.

For all of C classes $\{\mathcal{Z}_i\}_{i=1}^{C}$, we then define a $N \times N$ kernel matrix $\mathbf{K}$,

$$\mathbf{K} = (K_{lh})_{\substack{l=1,\cdots,C \\ h=1,\cdots,C}} \tag{5.9}$$

which allows us to express $\tilde{\Phi}_b^T\tilde{\Phi}_b$ as follows:

$$\begin{aligned}\tilde{\Phi}_b^T\tilde{\Phi}_b = \; & \tfrac{1}{N}\mathbf{B}\cdot(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{A}_{NC}-\tfrac{1}{N}(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{1}_{NC})- \\ & \tfrac{1}{N}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{A}_{NC})+\tfrac{1}{N^2}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{1}_{NC}))\cdot\mathbf{B}\end{aligned} \tag{5.10}$$

where $\mathbf{B}=\mathbf{diag}\left[\sqrt{C_1},\cdots,\sqrt{C_c}\right]$, $\mathbf{1}_{NC}$ is a $N\times C$ matrix with terms all equal to one, $\mathbf{A}_{NC}=\mathbf{diag}\left[\mathbf{a}_{c_1},\cdots,\mathbf{a}_{c_c}\right]$ is a $N\times C$ block diagonal matrix, and $\mathbf{a}_{c_i}$ is a $C_i\times 1$ vector with all terms equal to: $\frac{1}{C_i}$ (see Appendix A.1 for a detailed derivation of Eq.5.10.).

Let $\tilde{\lambda}_i$ and $\tilde{\mathbf{e}}_i$ $(i=1,\cdots,C)$ be the i-th eigenvalue and its corresponding eigenvector of $\tilde{\Phi}_b^T\tilde{\Phi}_b$, sorted in **decreasing** order of the eigenvalues. Since $(\tilde{\Phi}_b\tilde{\Phi}_b^T)(\tilde{\Phi}_b\tilde{\mathbf{e}}_i)=\tilde{\lambda}_i(\tilde{\Phi}_b\tilde{\mathbf{e}}_i)$, $\tilde{\mathbf{v}}_i=\tilde{\Phi}_b\tilde{\mathbf{e}}_i$ is an eigenvector of $\tilde{\mathbf{S}}_b$. In order to remove the null space of $\tilde{\mathbf{S}}_b$, we only use its first m $(\leq C-1)$ eigenvectors: $\tilde{\mathbf{V}}=[\tilde{\mathbf{v}}_1,\cdots,\tilde{\mathbf{v}}_m]=\tilde{\Phi}_b\tilde{\mathbf{E}}_m$ with $\tilde{\mathbf{E}}_m=[\tilde{\mathbf{e}}_1,\cdots,\tilde{\mathbf{e}}_m]$, whose corresponding eigenvalues are greater than 0. It is not difficult to see that $\tilde{\mathbf{V}}^T\tilde{\mathbf{S}}_b\tilde{\mathbf{V}}=\tilde{\Lambda}_b$, with $\tilde{\Lambda}_b=\mathbf{diag}[\tilde{\lambda}_1^2,\cdots,\tilde{\lambda}_m^2]$, an $(m\times m)$ diagonal matrix.

5.5.2 Eigen-analysis of $\tilde{\mathbf{S}}_w$ in the Feature Space

Let $\tilde{\mathbf{U}}=\tilde{\mathbf{V}}\tilde{\Lambda}_b^{-1/2}$, each column vector of which lies in the feature space $\mathbb{F}$. Projecting both $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ into the subspace spanned by $\tilde{\mathbf{U}}$, it can easily be seen that $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_b\tilde{\mathbf{U}}=\mathbf{I}$, an $(m\times m)$ identity matrix, while $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$ can be expanded as:

$$\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}=(\tilde{\mathbf{E}}_m\tilde{\Lambda}_b^{-1/2})^T(\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b)(\tilde{\mathbf{E}}_m\tilde{\Lambda}_b^{-1/2}) \tag{5.11}$$

Using the kernel matrix $\mathbf{K}$, a closed form expression of $\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b$ can be obtained as follows,

$$\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b=\frac{1}{N}(\mathbf{J}1-\mathbf{J}2) \tag{5.12}$$

where both $\mathbf{J}1$ and $\mathbf{J}2$ are defined in Appendix A.2 along with the detailed derivation of the expression in Eq.5.12.

We proceed by diagonalizing $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$, a tractable matrix with size $m\times m$. Let $\tilde{\mathbf{p}}_i$ be the i-th eigenvector of $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$, where $i=1,\cdots,m$, sorted in **increasing** order of its corresponding eigenvalue $\tilde{\lambda}_i'$. In the set of ordered eigenvectors, those that correspond to the smallest eigenvalues maximize the ratio in Eq.5.6, and should be considered the most discriminative features. Discarding the eigenvectors with the largest eigenvalues, the $M(\leq m)$ selected eigenvectors are denoted as $\tilde{\mathbf{P}}_M = [\tilde{\mathbf{p}}_1, \cdots, \tilde{\mathbf{p}}_M]$. Defining a matrix $\tilde{\mathbf{Q}} = \tilde{\mathbf{U}}\tilde{\mathbf{P}}_M$, we can obtain $\tilde{\mathbf{Q}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{Q}} = \tilde{\Lambda}_w$, with $\tilde{\Lambda}_w = \mathbf{diag}[\tilde{\lambda}'_1, \cdots, \tilde{\lambda}'_M]$, a $M \times M$ diagonal eigenvalue matrix.

Based on the calculations presented above, a set of optimal nonlinear discriminant feature vectors can be derived through $\tilde{\Gamma} = \tilde{\mathbf{Q}}(\eta\mathbf{I} + \tilde{\Lambda}_w)^{-1/2}$, where η is the regularization parameter introduced in Section 3.3. The feature bases $\tilde{\Gamma}$ form a low-dimensional subspace in $\mathbb{F}$, where the following regularized Fisher's criterion (originally introduced in Section 3.3.3) is optimized instead of the conventional one in Eq.5.6 used in GDA to relieve the SSS problem,

$$\tilde{\Psi} = \arg\max_{\tilde{\Psi}} \frac{\left|(\tilde{\Psi}^T\tilde{\mathbf{S}}_b\tilde{\Psi})\right|}{\left|\eta(\tilde{\Psi}^T\tilde{\mathbf{S}}_b\tilde{\Psi}) + (\tilde{\Psi}^T\tilde{\mathbf{S}}_w\tilde{\Psi})\right|} \tag{5.13}$$

5.5.3 Dimensionality Reduction and Feature Extraction

For any input pattern $\mathbf{z}$, its projection into the subspace spanned by the set of feature bases, $\tilde{\Gamma}$, derived in Section 5.5.2, can be computed by

$$\mathbf{y} = \tilde{\Gamma}^T\phi(\mathbf{z}) = \left(\tilde{\mathbf{E}}_m \cdot \tilde{\Lambda}_b^{-1/2} \cdot \tilde{\mathbf{P}}_M \cdot (\eta\mathbf{I} + \tilde{\Lambda}_w)^{-1/2}\right)^T \left(\tilde{\Phi}_b^T\phi(\mathbf{z})\right) \tag{5.14}$$

where $\tilde{\Phi}_b^T\phi(\mathbf{z}) = [\,\tilde{\bar{\phi}}_1 \quad \cdots \quad \tilde{\bar{\phi}}_c\,]^T\,\phi(\mathbf{z})$. Since

$$\begin{aligned} \tilde{\bar{\phi}}_i^{\,T}\,\phi(\mathbf{z}) &= \left(\sqrt{\tfrac{C_i}{N}}\,(\bar{\phi}_i - \bar{\phi})\right)^T \phi(\mathbf{z}) \\ &= \sqrt{\tfrac{C_i}{N}} \left(\tfrac{1}{C_i}\sum_{m=1}^{C_i}\phi_{im}^T\phi(\mathbf{z}) - \tfrac{1}{N}\sum_{p=1}^{C}\sum_{q=1}^{C_p}\phi_{pq}^T\phi(\mathbf{z})\right) \end{aligned} \tag{5.15}$$

we have

$$\tilde{\Phi}_b^T\phi(\mathbf{z}) = \frac{1}{\sqrt{N}}\mathbf{B} \cdot \left(\mathbf{A}_{NC}^T \cdot \nu(\phi(\mathbf{z})) - \frac{1}{N}\mathbf{1}_{NC}^T \cdot \nu(\phi(\mathbf{z}))\right) \tag{5.16}$$

where

$$\nu(\phi(\mathbf{z})) = [\,\phi_{11}^T\phi(\mathbf{z}) \quad \phi_{12}^T\phi(\mathbf{z}) \quad \cdots \quad \phi_{c(c_c-1)}^T\phi(\mathbf{z}) \quad \phi_{cc_c}^T\phi(\mathbf{z})\,]^T \tag{5.17}$$

is a $(N \times 1)$ kernel vector obtained by dot products of $\phi(\mathbf{z})$ and each mapped training sample $\phi(\mathbf{z}_{ij})$ in $\mathbb{F}$.

Combining Eq.5.14 and Eq.5.16, we obtain

$$\mathbf{y} = \Theta \cdot \nu(\phi(\mathbf{z})) \tag{5.18}$$

where

$$\Theta = \frac{1}{\sqrt{N}} \left(\tilde{\mathbf{E}}_m \cdot \tilde{\Lambda}_b^{-1/2} \cdot \tilde{\mathbf{P}}_M \cdot (\eta \mathbf{I} + \tilde{\Lambda}_w)^{-1/2} \right)^T \left(\mathbf{B} \cdot \left(\mathbf{A}_{NC}^T - \frac{1}{N} \mathbf{1}_{NC}^T \right) \right) \tag{5.19}$$

is a $(M \times N)$ matrix which can be computed offline. Thus, through Eq.5.18, a low-dimensional nonlinear representation $(\mathbf{y})$ of $\mathbf{z}$ with enhanced discriminant power has been introduced. The detailed steps for implementing the KDDA method are summarized in Fig.5.2.

5.6 Comments

KDDA implements the JD-LDA method in a high-dimensional feature space using the kernel machines. Its main advantages can be summarized as follows:

1. KDDA introduces a nonlinear mapping from the input space to an implicit high-dimensional feature space, where the non convex and complex distribution of patterns in the input space is “linearized” and “simplified” so that conventional LDA can perform well. It is not difficult to see that KDDA reduces to JD-LDA for $\phi(\mathbf{z}) = \mathbf{z}$. Thus, JD-LDA can be viewed as a special case of the proposed KDDA framework.

2. KDDA effectively solves the SSS problem in the high-dimensional feature space by employing the regularized Fisher's criterion and the D-LDA subspace technique. With the introduction of the two issues, KDDA can exactly extract the optimal discriminant features from both inside and outside of $\tilde{\mathbf{S}}_w$'s null space, while avoiding

```
Input: A training set $\mathcal{Z}$ with $C$ classes: $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, each class containing
    $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ face images, where $\mathbf{z}_{ij} \in \mathbb{R}^J$, and the regularization parameter $\eta$.
Output: The matrix $\Theta$; For an input example $\mathbf{z}$, its KDDA-based feature
    representation $\mathbf{y}$.
Algorithm:
    Step 1. Compute the kernel matrix $\mathbf{K}$ using Eq.5.9.
    Step 2. Compute $\tilde{\Phi}_b^T \tilde{\Phi}_b$ using Eq.5.10, and find $\tilde{\mathbf{E}}_m$ and $\tilde{\Lambda}_b$ from $\tilde{\Phi}_b^T \tilde{\Phi}_b$
        in the way shown in Section 5.5.1.
    Step 3. Compute $\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_w \tilde{\mathbf{U}}$ using Eq.5.11 and Eq.5.12, and find $\tilde{\mathbf{P}}_M$ and $\tilde{\Lambda}_w$
        from $\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_w \tilde{\mathbf{U}}$ in the way depicted in Section 5.5.2;
    Step 4. Compute $\Theta$ using Eq.5.19.
    Step 5. Compute the kernel vector of the input $\mathbf{z}$, $\nu(\phi(\mathbf{z}))$, using Eq.5.16.
    Step 6. The optimal nonlinear discriminant feature representation of $\mathbf{z}$ can
        be obtained by $\mathbf{y} = \Theta \cdot \nu(\phi(\mathbf{z}))$.
```

Figure 5.2: KDDA pseudo-code implementation

the risk of experiencing high variance in estimating the scatter matrices at the same time. This point makes KDDA significantly different from the existing nonlinear discriminant analysis methods such as GDA in the SSS situations.

3. In GDA, to remove the null space of $\tilde{\mathbf{S}}_w$, it is required to compute the pseudo inverse of the kernel matrix $\mathbf{K}$, which could be extremely ill-conditioned when certain kernels or kernel parameters are used. Pseudo inversion is based on inversion of the nonzero eigenvalues. Due to round-off errors, it is not easy to identify the true null eigenvalues. As a result, numerical stability problems often occur [100]. However,

it can been seen from the derivation of KDDA that such problems are avoided in KDDA. The improvement can be observed also in experimental results reported in Fig.5.5:**A** and Fig.5.6:**A**.

5.7 Experimental Results

Two sets of experiments are included here to illustrate the effectiveness of the KDDA algorithm. The distribution of face patterns is highly non-convex and complex, especially when the patterns are subject to large variations in viewpoints. Thus, the multi-view UMIST face database [36] is chosen to be utilized in the experiments reported here. The details of the database have been described in Section 2.3.2, where some sample images are shown in Fig.2.5. Each image (with size 112 × 92) is converted to a vector of dimensionality $J = 10304$ prior to being fed to an evaluation method.

5.7.1 Distribution of Multi-view Face Patterns

The first experiment aims to provide insights on how the KDDA algorithm linearizes and simplifies the complicated face pattern distribution.

For the sake of simplicity in visualization, we only use a subset of the database, which contains 170 images of 5 randomly selected subjects (classes). Four types of feature bases are generalized from the subset by utilizing the PCA, KPCA, JD-LDA and KDDA algorithms respectively. In the four subspaces produced, two are linear, produced by PCA and JD-LDA, and two are nonlinear, produced by KPCA and KDDA. In the sequence, all of the images are projected onto the four subspaces. For each image, its projections in the first two most significant feature bases of each subspace are visualized in Figs.5.3-5.4.

In Fig.5.3, the visualized projections are the first two most significant principal components extracted by PCA and KPCA, and they provide a low-dimensional representation for the samples, which can be used to capture the structure of data. Thus, we

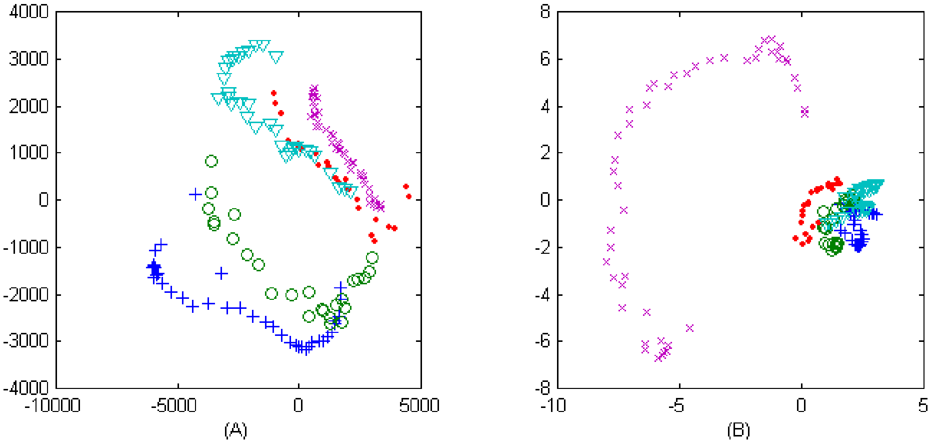

Figure 5.3: Distribution of 170 samples of 5 subjects projected into **A**: PCA-based subspace ($\subset \mathbb{R}^J$), and **B**: KPCA-based subspace ($\subset \mathbb{F}$) using a RBF kernel function with $\sigma^2 = 5\mathbf{e}6$. Each subspace is spanned by its most significant feature bases.

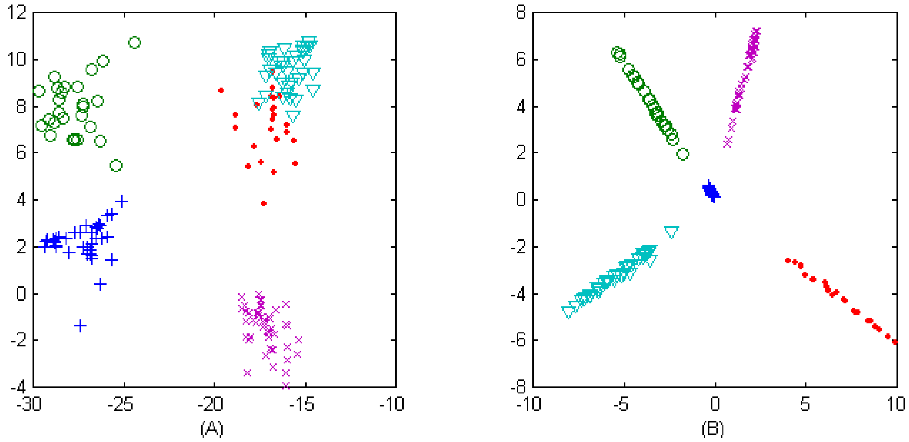

Figure 5.4: Distribution of 170 samples of 5 subjects projected into **A**: JD-LDA-based subspace ($\subset \mathbb{R}^J$), and **B**: KDDA-based subspace ($\subset \mathbb{F}$) using a RBF kernel function with $\sigma^2 = 5\mathbf{e}6$. Each subspace is spanned by its most significant feature bases.

can roughly learn the original distribution of face samples from Fig.5.3:**A**, which is non convex and complex as we expected based on the analysis presented in the previous sections. In Fig.5.3:**B**, KPCA generalizes PCA to its nonlinear counterpart using a RBF kernel function : $k(\mathbf{z}_1, \mathbf{z}_2) = \exp\left(\frac{-||\mathbf{z}_1-\mathbf{z}_2||^2}{\sigma^2}\right)$ with $\sigma^2 = 5\mathbf{e}6$. However, it is hard to find any useful improvement for the purpose of pattern classification from Fig.5.3:**B**. It can be therefore concluded here again that the low-dimensional representation obtained by PCA like techniques, achieve simply object reconstruction, and they are not necessarily useful for discrimination and classification tasks [7, 114].

Unlike the PCA approaches, LDA optimizes the low-dimensional representation of the objects based on separability criteria. Fig.5.4 depicts the first two most discriminant features extracted by utilizing JD-LDA and KDDA respectively. Simple inspection of Figs.5.3-5.4 indicates that these features outperform, in terms of discriminant power, those obtained using PCA like methods. However, subject to limitation of linearity, some classes are still non-separable in the JD-LDA-based subspace as shown in Fig.5.4:**A**. In contrast to this, we can see the linearization property of the KDDA-based subspace, as depicted in Fig.5.4:**B**, where all of classes are well linearly separable when a RBF kernel with $\sigma^2 = 5\mathbf{e}6$ is used.

5.7.2 Comparison with KPCA and GDA

The second experiment compares the performance of the KDDA algorithm, in terms of the *classification error rate* (CER), to two other commonly used kernel-based FR algorithms, KPCA and GDA. Following the standard evaluation process of appearance-based FR methods, the FR procedure is completed in two stages:

1. Feature extraction. The overall database is randomly partitioned into two subsets: the training set and test set. The training set is composed of 120 images: 6 images per person are randomly chosen. The remaining 455 images are used to form the test set. There is no overlapping between the two. After training is over, both sets are projected into the feature spaces derived from the KPCA, GDA and KDDA methods.

2. Classification. This is implemented by feeding feature vectors obtained in step-1 into a nearest neighbor classifier. It should be noted again at this point that, since the focus in this work is on feature extraction, a simple classifier is always preferred so that the FR performance is not mainly limited by the classifier but the feature selection algorithms. To enhance the accuracy of performance evaluation, the CERs reported in this work are averaged over 8 runs. Each run is based on a random partition of the database into the training and test sets. Following the evaluation protocol depicted in Section 2.3.3, the average CER is denoted as $\bar{E}_{ave} = 1 - \bar{R}_1$, where $\bar{R}_1$ is defined in Eq.2.1 with $n = 1$.

To evaluate the overall performance of the three methods, two typical kernel functions: namely the RBF and the polynomial function, and a wide range of parameter values are tested. Sensitivity analysis is performed with respect to the kernel parameters and the number of used feature vectors, M. Figs.5.5-5.6 depict the average CERs ($\bar{E}_{ave}$) of the three methods compared when the RBF and polynomial kernels are used.

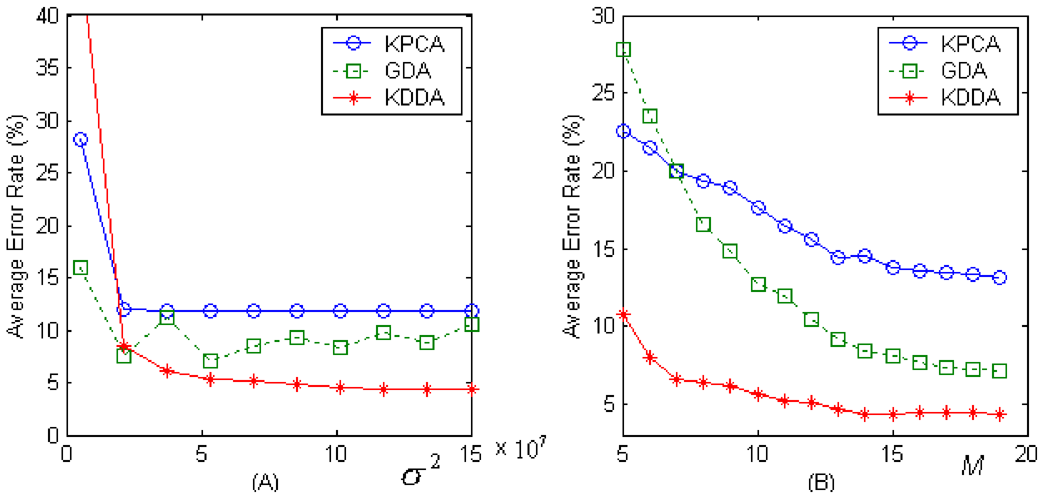

Figure 5.5: A comparison of average CERs ($\bar{E}_{ave}$) based on the RBF kernel function. **A**: CERs as a function of σ^2. **B**: CERs as a function of M.

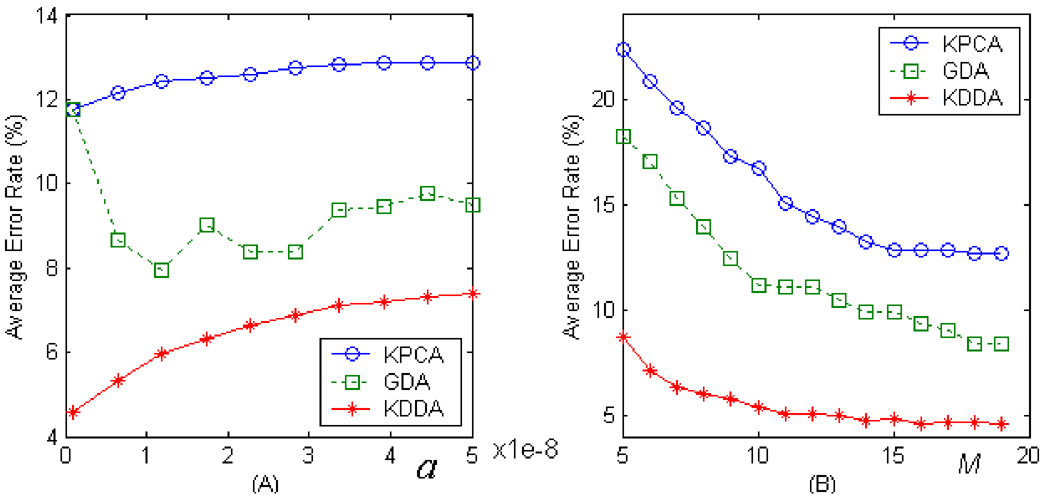

Figure 5.6: A comparison of average CERs ($\bar{E}_{ave}$) based on the Polynomial kernel function. **A**: CERs as a function of a. **B**: CERs as a function of M.

The only kernel parameter for RBF is the scale value σ^2. Fig.5.5:**A** shows the CERs as functions of σ^2 within the range from 0.5**e**7 to 1.5**e**8, when the optimal number of feature vectors, $M = M_{opt}$, is used. The optimal feature number is a result of the existence of the peaking effect in the feature selection procedure. It is well-known that the classification error initially declines with the addition of new features, attains a minimum, and then starts to increase [98]. The optimal number can be found by searching the number of used feature vectors that results in the minimal summation of the CERs over the variation range of σ^2. In Fig.5.5:**A**, $M_{opt} = 99$ is the value used for KPCA, while $M_{opt} = 19$ is used for GDA and KDDA. Fig.5.5:**B** depicts the CERs as functions of M within the range from 5 to 19, when optimal $\sigma^2 = \sigma^2_{opt}$ is used. Similar to M_{opt}, σ^2_{opt} is defined as the scale parameter that results in the minimal summation of the CERs over the variation range of M for the experiment discussed here. In Fig.5.5:**B**, a value $\sigma^2_{opt} = 1.5$**e**8 is found for KPCA, $\sigma^2_{opt} = 5.3333$**e**7 for GDA and $\sigma^2_{opt} = 1.3389$**e**7 for KDDA.

As such, the average CERs of the three methods with polynomial kernel ($k(\mathbf{z}_1, \mathbf{z}_2) = (a \cdot (\mathbf{z}_1 \cdot \mathbf{z}_2) + b)^d$) are shown in Fig.5.6. For the sake of simplicity, we only test the influence of a, while let $b = 1$ and $d = 3$ be fixed. Fig.5.6:**A** depicts the CERs as functions of a within the range from $1\mathbf{e}-9$ to $5\mathbf{e}-8$, where $M_{opt} = 100$ for KPCA, $M_{opt} = 19$ for GDA and KDDA. Fig.5.6:**B** shows the CERs as functions of M within the range from 5 to 19 with $a_{opt} = 1\mathbf{e}-9$ for KPCA, $a_{opt} = 2.822\mathbf{e}-8$ for GDA and $a_{opt} = 1\mathbf{e}-9$ for KDDA, determined similarly to σ^2_{opt} and M_{opt}.

Table 5.2: The average percentages of the CER of KDDA over that of KPCA or GDA.

Kernel	RBF	Polynomial	(RBF+Polynomial)/2
KDDA/KPCA	33.669%	35.081%	34.375%
KDDA/GDA	47.866%	47.664%	47.765%

Let α_M and β_M be the average CERs of KDDA and any one of other two methods respectively, where $M = [5, \cdots, 19]$. From Fig.5.5:**B** and Fig.5.6:**B**, we can obtain an interesting quantity comparison: the average percentages of the CER of KDDA over that of any other method by $\sum_{M=5}^{19}(\alpha_M/\beta_M)$. The results are tabulated in Table 5.2, where it can be seen that the average CER of KDDA is only about 34.375% and 47.765% of the average CERs of KPCA and GDA respectively. It should be also noted that Fig.5.5:**A** and Fig.5.6:**A** reveal the numerical stability problems existing in practical implementations of GDA. Comparing the GDA performance to that of KDDA we can easily see that the later is more stable and predictable, resulting in a cost effective determination of parameter values during the training phase.

5.8 Summary

A new nonlinear face recognition method has been introduced in this chapter. The proposed method combines kernel-based methodologies with discriminant analysis techniques. The kernel function is utilized to map the original face patterns to a high-dimensional feature space, where the highly non-convex and complex distribution of face patterns is linearized and simplified, so that linear discriminant techniques can be used for feature extraction. The small-sample-size (SSS) problem caused by high dimensionality of mapped patterns, is addressed by a regularized D-LDA technique which exactly finds the optimal discriminant subspace of the feature space without suffering from problems arising from the SSS settings, such as possible loss of significant discriminant information and high variance of parameter estimation. Experimental results indicate that the CER performance of the KDDA algorithm is overall superior to those obtained by the KPCA or GDA approaches. In conclusion, the KDDA algorithm provides a general pattern recognition framework for nonlinear feature extraction from high-dimensional input patterns in the SSS situations. We expect that in addition to face recognition, KDDA will provide excellent performance in applications where classification tasks are routinely performed, such as content-based image indexing and retrieval, video and audio classification.

Chapter 6

A Mixture of LDA Subspaces with Cluster Analysis

6.1 Introduction

Many state-of-the-art appearance-based methods have reported outstanding recognition performance (usually $> 90\%$ correct recognition rate). However, the FERET evaluation reports and other independent studies indicate that for most of these methods, such recognition rates can be achieved only for limited-size face databases (usually < 50 subjects), and that their performance deteriorates rapidly when they are applied to *large-scale face databases* (LSFDs) [39, 94]. The main reason is that the appearance-based methods use *statistical pattern recognition* (SPR) methodologies, which normally require face images to follow a convex and linear distribution. This condition can be approximately met only in small-scale databases with limited variations. Experimentation has revealed that although the images of appearances of the face patterns may vary significantly due to differences in imaging parameters such as lighting, scale, orientation, *etc.* , these differences have an approximately linear effect when they are small [119]. Nevertheless, as the size of the database increases so that more pattern variations are introduced, the distribution of the face images dramatically becomes highly non convex and complex [9]. As a result, the feature representations obtained by these linear models are not capable of generalizing all of these variations.

There are two approaches to handle the increased complexity of the face distribution in the case of LSFDs.

1. Model the complex distribution using *globally nonlinear* techniques. In the last Chapter, we have studied kernel machines [3,6,72,73,88,100,123], the most popular technique recently used to design nonlinear methods. From both theoretical and experimental analysis of KPCA, GDA and KDDA, it can be seen that the main problem with the kernel-based methods is the difficulty in the selection of the kernel functions and the optimization of involved parameters. All of these are issues that could significantly influence the performance of the resulting FR systems. In addition, given a small size training sample, these methods tend to overfit easily due to increased algorithm complexity, and they are also more computationally expensive compared to their linear counterparts due to the introduction of the high-dimensional feature space $\mathbb{F}$. The last point is particularly important for tasks like face recognition performed in a high-dimensional input space $\mathbb{R}^J$.

2. Piecewise learn the complex distribution by *a mixture of locally linear models.* This strategy is based on the principle of "divide and conquer", by which a complex FR problem is decomposed into a set of simpler ones, in each of which a locally linear pattern distribution could be generalized and dealt with by a relatively easy linear solution (see *e.g.* [52,53,55,90,99,109,115,126]). Compared to the globally nonlinear approaches, piecewise approximation is simpler, more cost effective and easier to implement. In addition, linear models are known to be rather robust against noise and most likely will not overfit [83]. A mixture of view-based PCA subspaces [92], a mixture of Gaussians (for face detection) [113] and a mixture of Factor Analyzers (MFAs) [30], are examples of piecewise learning methods that have been applied to

databases of $O(10^3)$ face images. It should be also noted at this point that attempts to combine the strengths of both solutions for face detection and pose estimation have been introduced recently [62, 63, 79].

Before we continue our presentation, it is important to distinguish two frequently used terms in this chapter: ***class*** and ***cluster***, where ***class*** means a set of face images from the same person or subject, while ***cluster*** means a group of ***classes***. Thus, all of members in one ***class*** will have a same ***cluster*** label.

From the designer's point of view, the central issue to the decomposition based approach is to find an appropriate criterion to partition the large-size training set. Existing partition techniques, whether nonparametric clustering such as K-means [42] or model-based clustering such as EM [80], unanimously adopt "similarity criterion", based on which similar samples are within the same cluster and dissimilar samples are in different clusters [131]. For example, in the view-based representation [92], every face pattern is manually assigned to one of several clusters according to its view angle with each cluster corresponding to a particular view. In the method considered in [113] and [30], the database partitions are automatically implemented using the K-means and EM clustering algorithms respectively. However, although such criterion may be optimal in the sense of approximating real face distribution for tasks such as face reconstruction, face pose estimation and face detection, they may not be good for the recognition task considered in this thesis. It is not hard to see that from a classification point of view, the database partition criterion should be aimed to maximize the difference or separability between classes within each "divided" subset or cluster, which as a sub-problem then can be more easily "conquered" by a traditional FR method.

Motivated by such concerns, we first propose in this chapter a novel clustering method specifically optimized for the purpose of pattern classification. Contrary to the conventional "similarity criterion", we introduce the concept of "separability criterion" for clustering. As we know from the analysis of previous chapters, a powerful tool to optimize the separability criterion is the *Linear Discriminant Analysis* (LDA) [27], which attempts to find optimal discriminatory feature representation by maximizing the between-class scatter of patterns. In the proposed method, the training LSFD is partitioned into a set of K *maximally separable clusters* (MSCs) by a LDA-like technique, and the separability between classes is maximized in each MSC. We then propose a two-level classification framework, which consists of a group of FR sub-systems to take advantage of the obtained MSCs. The first level is composed of K same but separate sub-systems, each trained with one MSC. For a given query, classification is firstly done independently in each sub-system, and thus K outputs can be produced. The second level is a classifier, which receives the K obtained outputs and chooses the optimum one as the final FR result. Through this process, the complex FR problem on a large database is gradually decomposed and addressed by a mixture of manageable FR sub-systems. Each sub-system is only required to deal with a part of the whole problem. This is not a difficult task for most of traditional FR methods to work as such a sub-system in a single MSC with limited-size subjects and high between-class separability.

The rest of the chapter is organized as follows. We first talk about the FR sub-system to be integrated into our methodologies in Section 6.2. In Section 6.3 and 6.4, the proposed clustering method and the hierarchical classification framework are introduced and discussed in details. Following that, Section 6.5 gives some important comments on the proposed methodologies. In Section 6.6, a set of experiments conducted on the FERET database are presented to demonstrate the effectiveness of the proposed methods. Finally, Section 6.7 summarizes conclusions and provides directions for future research.

6.2 FR Sub-Systems

In theory, the FR sub-system integrated into the proposed hierarchical classification framework could be any one of traditional appearance-based FR methods, whose performance is aimed to be improved in the case of LSFDs. In this chapter, we choose the JD-LDA methods introduced in Section 3.3.4. The selection was based on several considerations:

1. LDA-based methods have shown excellent performance, and outperform many other appearance-based approaches such as those based on PCA in limited-size face databases [7, 46, 73–75]. However, it has been also observed that the LDA-based methods tend to be easier to overfit than PCA-based ones when they are applied to a LSFD, and good results in this case have not been reported yet [78].

2. LDA-based methods optimize the face feature representation based on a separability criterion, and from this point of view are closely related to the clustering method to be proposed in Section 6.3.

3. Many LDA-based algorithms suffer from the *small sample size* (SSS) problem (see Section 3.3.1). The JD-LDA method provide a simple but cost effective solution to the SSS problem [74, 75].

For all these reasons, the JD-LDA method was selected for the feature extraction task in this work. The subsequent classification in the FR subsystem is performed by using a classic *nearest neighbor classifier* (NNC) as in Sections 3.5 and 5.7.2.

6.3 Clustering with Separability Criterion (CSC)

Motivated by LDA and its successful application to FR tasks, we introduce the concept of separability criterion in cluster analysis. Contrary to the traditional criteria, the one proposed here is developed from the standpoint of classification, which requires classes with more different properties to be grouped in the same cluster, so that discriminant learning within the cluster becomes easier. Similar to LDA, the criterion is optimized by maximizing a widely used separability measure, the *between-class scatter* (BCS).

Let us assume that a large-size training set as described in Section 3.2 is available. Furthermore, let Ω_k denote the k-th cluster, where $k = [1, \cdots, K]$ with K being the number of clusters. Representing each class $\mathcal{Z}_i$ by its mean: $\bar{\mathbf{z}}_i = \frac{1}{C_i}\sum_{j=1}^{C_i} \mathbf{z}_{ij}$, the total within-cluster BCS of the training set $\mathcal{Z}$ can be defined as,

$$S_t = \sum_{k=1}^{K} \sum_{\bar{\mathbf{z}}_i \in \Omega_k} C_i \cdot (\bar{\mathbf{z}}_i - \mathbf{w}_k)^T (\bar{\mathbf{z}}_i - \mathbf{w}_k) \tag{6.1}$$

where $\mathbf{w}_k = (\sum_{\bar{\mathbf{z}}_i \in \Omega_k} C_i \cdot \bar{\mathbf{z}}_i)/(\sum_{\bar{\mathbf{z}}_i \in \Omega_k} C_i)$ is the center of the cluster Ω_k. Eq.6.1 implies that a better class-separability intra clusters is achieved if S_t has a larger value. The clustering algorithm works as follows:

Firstly, an initial partition is formed by randomly assigning $\bar{\mathbf{z}}_i$ where $i = [1, \cdots, C]$ to one of clusters $\{\Omega_k\}_{k=1}^{K}$. Secondly, we find the class mean $\hat{\mathbf{z}}_k \in \Omega_k$ which has the minimal Euclidean distance to $\mathbf{w}_k$ by

$$\hat{\mathbf{z}}_k = \arg\min_{\bar{\mathbf{z}}_i \in \Omega_k} \left\{ (\bar{\mathbf{z}}_i - \mathbf{w}_k)^T (\bar{\mathbf{z}}_i - \mathbf{w}_k) \right\} \tag{6.2}$$

Next, we compute distances of $\hat{\mathbf{z}}_k$ to other cluster centers:

$$\mathbf{d}_{kq} = (\hat{\mathbf{z}}_k - \mathbf{w}_q)^T (\hat{\mathbf{z}}_k - \mathbf{w}_q), \tag{6.3}$$

and find the cluster $\hat{q}$ so that

$$\hat{q} = \arg\max_{q} \{\mathbf{d}_{kq}\} \tag{6.4}$$

where $q = [1, \cdots, K]$. We reassign the class represented by $\hat{\mathbf{z}}_k$ into cluster $\hat{q}$, *i.e.* set $\hat{\mathbf{z}}_k \in \Omega_{\hat{q}}$ if $\hat{q} \neq k$. The cluster centers $\mathbf{w}_k$ and the total scatter S_t are then updated and the above procedure is repeated until S_t stops increasing.

In summary, the clustering method (called CSC hereafter) presented above maximizes the total within-cluster BCS S_t by iteratively reassigning those classes whose means have the minimal distances to their own cluster centers, so that the overall separability between classes is enhanced gradually within each cluster. The detailed pseudo code implementation of the CSC method is depicted in Fig.6.1.

```
Input: A training set $\mathcal{Z}$ with $C$ classes: $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, each class contains
    $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ face images.
Output: $K$ maximally separable clusters $\{\Omega_k\}_{k=1}^{K}$, each class of images
    $\mathcal{Z}_i$ are assigned into one of $K$ clusters.
Algorithm:
    Step 1. Calculate $\bar{\mathbf{z}}_i = \frac{1}{C_i}\sum_{j=1}^{C_i}\mathbf{z}_{ij}$ for class $\mathcal{Z}_i$ where $i = [1, \cdots, C]$.
    Step 2. Randomly partition $\{\bar{\mathbf{z}}_i\}_{i=1}^{C}$ into $K$ initial clusters $\{\Omega_k\}_{k=1}^{K}$,
        calculate their cluster center $\{\mathbf{w}_k\}_{k=1}^{K}$, and initial $\hat{S}_t$ by Eq.6.1.
    Step 3. Find $\hat{\mathbf{z}}_k$ by Eq.6.2, where $k = [1, \cdots, K]$.
    Step 4. Compute the distances of $\hat{\mathbf{z}}_k$ to other cluster centers: $\mathbf{d}_{kq}$ by Eq.6.3,
        where $q = [1, \cdots, K]$.
    Step 5. Find the new cluster label $\hat{q}$ of $\hat{\mathbf{z}}_k$ by Eq.6.4, and re-set $\hat{\mathbf{z}}_k \in \Omega_{\hat{q}}$.
    Step 6. Update the cluster centers $\mathbf{w}_k$, and recompute the total scatter: $S_t$.
    Step 7. if $\hat{S}_t < S_t$ then $\hat{S}_t = S_t$; return to Step 3;
            else proceed to Step 8; /* Maximal $S_t$ has been found. */
    Step 8. Return current $K$ clusters $\{\Omega_k\}_{k=1}^{K}$ and their centers $\{\mathbf{w}_k\}_{k=1}^{K}$.
```

Figure 6.1: The pseudo code implementation of the CSC method.

6.4 Hierarchical Classification Framework (HCF)

The original training database $\mathcal{Z}$ is partitioned into a set of smaller and simpler subsets, called *maximally separable clusters* (hereafter MSCs) by the CSC method. Based on these MSCs, we then propose a *hierarchical classification framework* (hereafter HCF), which is able to take advantage of these obtained MSCs.

The two-level architecture of the HCF is as shown in Fig.6.2, where $(\cdot)_k^{(l)}$ denotes the

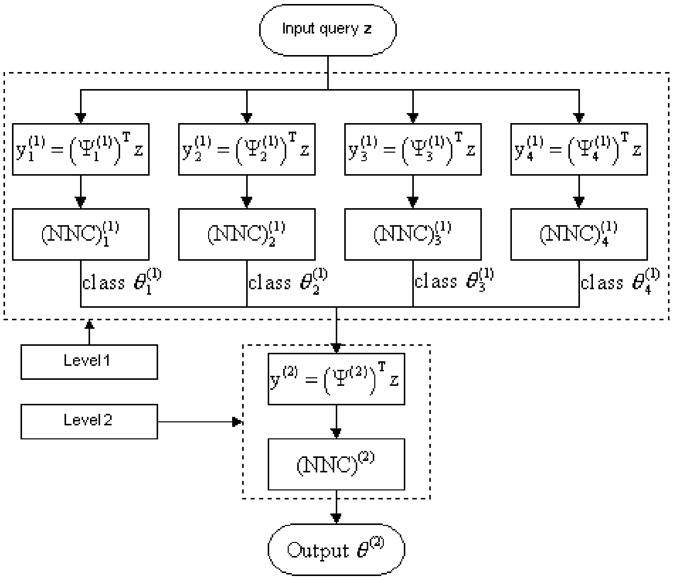

Figure 6.2: The architecture of the hierarchical classification framework (HCF), where the cluster number $K = 4$.

component corresponding to the l-th level and the k-th MSC (*i.e.* Ω_k), $k = [1, \cdots, K]$. The first level is composed of K JD-LDA FR sub-systems, each containing a JD-LDA feature extractor followed by a *nearest neighbor classifier* (NNC). The K sub-systems correspond to the K MSCs. During the learning stage, the k-th sub-system is trained with the k-th MSC in order to find a $M_k^{(1)}$-dimensional feature space spanned by $\Psi_k^{(1)}$ (for Ω_k). Upon completion, the training images $\mathbf{z}_{ij}$ are mapped to their corresponding feature spaces by $\mathbf{y}_{ij} = (\Psi_k^{(1)})^T \mathbf{z}_{ij}$ where $\mathbf{z}_{ij} \in \Omega_k$ respectively. These feature spaces are low-dimensional with enhanced discriminatory power.

In the FR procedure, any input query $\mathbf{z}$ is firstly fed to the first level of K FR sub-systems. Classification is independently performed in each sub-system by measuring Euclidean distances between $\mathbf{z}$'s projection $\mathbf{y}_k^{(1)} = (\Psi_k^{(1)})^T \mathbf{z}$ and the known examples $\mathbf{y}_{ij} \in \Omega_k$ based on the nearest neighbor rule. Thus, K classification results $\{\theta_k\}_{k=1}^K$ are produced, and they generalize a new subset, $\{\mathcal{Z}_{\theta_k}\}_{k=1}^K$, which is then passed to the next level. The second level contains only one NNC, $(\mathbf{NNC})^{(2)}$, which operates in the derived subset $\{\mathcal{Z}_{\theta_k}\}_{k=1}^{K}$. Here, we use a $\left(\sum_{k=1}^{K} M_k^{(1)}\right)$-dimensional joint feature space spanned by $\Psi^{(2)} = [\Psi_1^{(1)}, \cdots, \Psi_K^{(1)}]$ to take advantage of the K feature representations obtained from the first level. The input query $\mathbf{z}$ is projected onto the joint feature space by $\mathbf{y}^{(2)} = (\Psi^{(2)})^T\mathbf{z}$ as well as those training samples belonging to classes $\{\theta_k\}_{k=1}^{K}$ by the same projecting operations. Finally, the classification result $\theta^{(2)}$ is given as one of classes $\{\theta_k\}_{k=1}^{K}$ by performing the $(\mathbf{NNC})^{(2)}$ in the joint feature space. The feature fusion method is based on serial strategy, which is simple and easy to implement but may not be optimal. Future work will focus on more sophisticated fusion rules, such as parallel strategy and majority voting [52, 53, 130].

It should be noted that the proposed HCF is a flexible framework that can be used in conjunction with other clustering methodologies. For example, clusters generalized by utilizing the K-means approach can be used instead of those MSCs formed through the CSC approach. For the sake of simplicity in the sequence, we call the MSCs based HCF as HCF-MSC, while the K-means-clusters based HCF as HCF-Kmean. We intend to compare the two schemes in the experimental section to verify the claim that the proposed CSC method offers certain advantages over the popular similarity-based method K-means in large-scale FR tasks.

6.5 Issues to Build A Robust HCF-Based FR System

In the derivation of the HCF, a number of designing parameters have been introduced. They include: (1) the number of clusters selected, (2) the iterative method for the maximization of the total within-cluster BCS, S_t, (3) the type of partition, (4) the type of classifiers selected. All these elements affect the performance since they determine the features extracted and the classification accuracy. In the sequence, we provide some insights on the way we handle them in the context of this work.

1. **Determining the optimal number of clusters,** K. The problem of the cluster number determination is a major challenge in cluster analysis, underlined by the fact that there is no clear definition of a cluster [117]. In the experiments reported here, the optimal K is the one that gives the best CRR in the query sets, and is found through an exhaustive search within a relatively small range, $K = [4, \cdots, 25]$. On the other hand, the experiments show that the performance of the two methods, HCF-Kmean and HCF-MSC, tends to be closer as the cluster number increases. This reveals that the two clustering criteria, similarity and separability, may not be completely independent. An extreme example is that the same result is obtained by the CSC and K-mean methods when the cluster number K is equal to the class number C.

2. **Searching for the global maximum of the total within-cluster BCS,** S_t. The iterative procedure used for S_t maximization is quite often trapped in local maxima, so that multiple restarts for the CSC method are required to find a good solution in our experiments. This is a well-known problem that CSC shares with traditional clustering schemes such as K-means and EM. Future work in this area will focus on the introduction of parameterized re-weighting using methods such as deterministic annealing in order to avoid trapping of the iterative procedure in local minima [121]. It is worth pointing out in advance that, even if the clustering method (whether CSC or K-means) converges to a local minimum, significant performance improvement is delivered by the HCF strategy as opposed to a single JD-LDA FR system as shown in our experiments.

3. **The "soft split" of the training LSFD into a set of overlapping MSCs**. Like K-means, the proposed CSC method performs a "hard" database partition, and it does not impose any constraints between clusters. Extensions of the CSC method to "soft" partition using an EM-like algorithm is straightforward. The

soft-split approach will allow overlapping MSCs, and apply preferred weights to different MSCs.

4. **Potential improvement in the design of the second level of classifier**. For simplicity, the $(\mathbf{NNC})^{(2)}$ is used as the second level of classifier in this work. Instead it is possible to infer the cluster label of the query $\mathbf{z}$ in a probabilistic way by utilizing the optimal Bayes classifier [25]. We have noted that *mixtures of linear subspaces* (MLSes) such as PCAs and Factor Analyzers have been directly applied to class-level for face detection [113], handwritten character recognition [31] and FR [30] as well. However, it is worthy to mention here that, it is appropriate to model each class of samples using the MLSes only when the number of available training samples per class is large, *e.g.* 100 training face images for every person used in [30], a condition impossible to meet in practical FR tasks. For example, in the FERET evaluation protocol, only $1-3$ face images per person are available for learning. Due to the small sample number per class, it will be extremely ill-posed to apply a single or mixture probabilistic model to any single class. On the other hand, under the HCF, a mixture model can be applied to each MSC, which consists of many classes of face images. Thus, the density of the k-th mixture model trained with the k-th MSC can be described as $p(\mathbf{z}|\Omega_k) = \sum_t \int p(\mathbf{z}, \mathbf{v}, t|\Omega_k) d\mathbf{v}$, where $\mathbf{v}$ denotes the latent variables and t the component number in the mixture model. Then, the cluster label of $\mathbf{z}$ can be inferred using a posteriori probabilistic assignment, $P(\Omega_k|\mathbf{z}) = \frac{p(\mathbf{z}|\Omega_k)P(\Omega_k)}{p(\mathbf{z})}$.

6.6 Experimental Results

To assess the performance of the proposed CSC and HCF methodologies, we utilize the largest FERET subset, $\mathcal{G}_0$, which consists of 3817 gray-scale images of 1200 subjects as depicted in Section 2.3.2. To the best of the authors' knowledge such a database is among the largest utilized for the evaluation of FR algorithms. For computational convenience, each image is finally represented as a column vector of length $J = 17154$ prior to the recognition stage.

6.6.1 The FR Evaluation Design

To study the performance changes of the FR algorithms as the size of the test database increases, we further generalize eight test subsets with varying sizes (see Table 6.1 for details) from the evaluation database $\mathcal{G}_0$, and denote them as $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_8$. The relationship between the eight subsets can be described as $\mathbf{S}_1 \subset \mathbf{S}_2 \cdots \subset \mathbf{S}_7 \subset \mathbf{S}_8 = \mathcal{G}_0$. Each test dataset $\mathbf{S}_i$ is partitioned into two subsets: the training set $\mathcal{T}_i$ and the query set $\mathcal{Q}_i$. The LDA based algorithms require at least two training samples for each class. To meet the requirement, two frontal face images per person were chosen: one was from the **fa** set (regular facial expression, see [93, 94] for details) of the person, and another was from its corresponding **fb** set (alternative facial expression to **fa**), thus a training set with $|\mathcal{T}_i| = 2 \times |\mathcal{P}_i|$ images was obtained, where $|\mathcal{T}_i|$ denotes the size of $\mathcal{T}_i$, and $|\mathcal{P}_i|$ denotes the number of persons in $\mathbf{S}_i$. The remaining $|\mathbf{S}_i - \mathcal{T}_i|$ images in $\mathbf{S}_i$ were used to form the query set $\mathcal{Q}_i$. There was no overlapping between the two sets. Table 6.1 summarizes the details of the eight test databases and their partitions, and Fig.2.3 depicts some training and query samples.

Table 6.1: Sizes of the eight test datasets and their partitions.

Subsets	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$	$\mathbf{S}_6$	$\mathbf{S}_7$	$\mathbf{S}_8$
$\lvert\mathcal{T}_i\rvert$	362	788	1028	1308	1504	1784	2188	2400
$\lvert\mathcal{Q}_i\rvert$	137	212	473	692	998	1219	1314	1417
$\lvert\mathbf{S}_i\rvert$	499	1000	1501	2000	2502	3003	3502	3817
$\lvert\mathcal{P}_i\rvert$	181	394	514	654	752	892	1094	1200

The test protocol is same to the standard one depicted in Section 2.3.3. In addition to the CRR for top match, we tested the CRR for top n ranked matches, denoted as R_n with $n \geq 1$ in the experiments reported here.

6.6.2 CSC vs K-means

As required by the HCF-Kmean and HCF-MSC schemes, the training set $\mathcal{T}_i$ of the test database $\mathbf{S}_i$ was firstly partitioned into K clusters by using the standard K-means and the CSC method proposed here respectively. As the analysis given in Section 6.5, it may be easy for S_t to get trapped in local maximums during the clustering procedures. To relieve this, five runs of each method were executed on the training set $\mathcal{T}_i$. Since the cluster centers were randomly initialized, each run might converge to a different partition. All of the CRRs of HCF-Kmean and HCF-MSC reported later have been averaged over the five runs. It should be pointed out that the later experiments revealed a relative insensitivity of the CRR measure to the clustering initialization process.

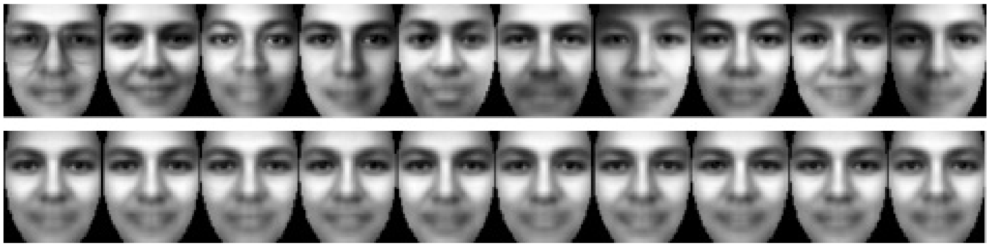

Figure 6.3: Means of $K = 10$ clusters obtained by performing the standard K-means (**Top**) and the CSC method (**Bottom**) on the training set $\mathcal{T}_8$ of $\mathbf{S}_8$.

Fig.6.3 and Table 6.2 (where S_t is calculated by Eq.6.1) depict quite different results obtained by the CSC method and the standard K-means. The total scatter S_t indicates the difference between classes within each cluster, while the cluster centers let us roughly know how different these clusters are. Not surprisingly, due to different clustering criteria, S_t obtained by CSC is approximately eight times of that by K-means as shown in

Table 6.2: Comparison of the total within-cluster BCS S_t in $\mathbf{S}_8$.

Cluster #	5	8	11	14	17	20	23	25
Kmean ($\times 10^7$)	1.53	1.45	1.39	1.35	1.33	1.30	1.28	1.27
CSC ($\times 10^7$)	10.50	10.49	10.47	10.45	10.45	10.43	10.42	10.41

Table 6.2. On the other hand, K-means obtained more compact clusters, each having its own certain common properties such that the difference between the clusters is more obvious compared to those clusters obtained by CSC as shown in Fig.6.3. The influence on the FR performance due to these difference will be embodied in the later simulation results obtained by HCF-Kmean and HCF-MSC.

6.6.3 The FR Performance Comparison

Since only two training examples are available for each subject, it can be seen from the experimental results presented in Section 4.5 that a strong regularization is required for the JD-LDA method in such a situation. Considering the high computational complexity to optimize the regularization parameter, η, we use in this work, JD-LDA.1, a special case of JD-LDA with η being set to 1, recommended in Section 4.5.2, where it has been shown that JD-LDA.1 provides a stable and cost-effective sub-optimal solution of JD-LDA across various SSS settings.

Both HCF-Kmean and HCF-MSC while employing JD-LDA.1 FR sub-systems can be considered as "a mixture of JD-LDA.1s", and they should be compared to the single JD-LDA.1 system to measure boosting performance in the case of LSFDs. Again, the Eigenfaces method [120] was implemented to provide a performance baseline. For all of the four methods evaluated here, the CRR (R_n) is a function of the number of feature vectors or dimensionality of feature spaces, M. In addition, the CRR of both HCF-Kmean and HCF-MSC is a function of the number of the clusters K as well. The best values of M

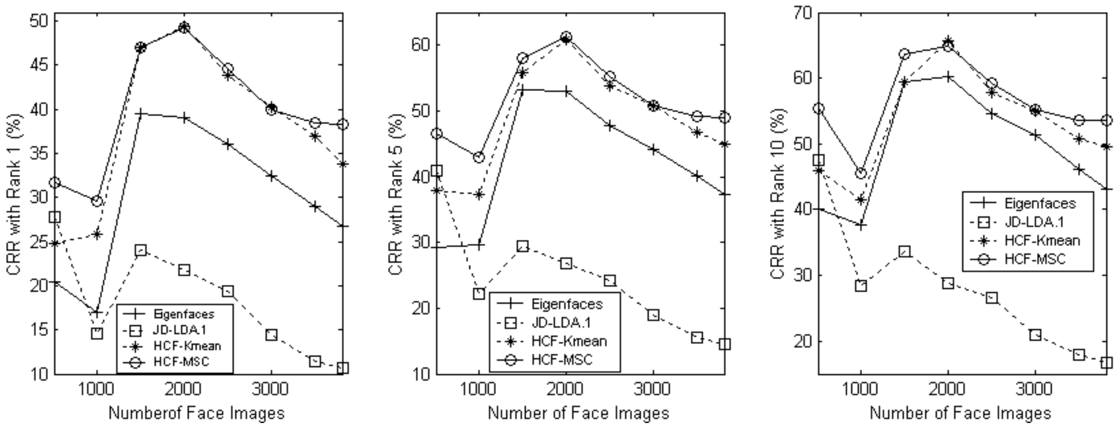

Figure 6.4: Comparison of CRRs (R_n) obtained by the four FR methods as functions of the number of face images ($\mathbf{S}_1 - \mathbf{S}_8$). **Left**: Rank 1 ($R_1$). **Middle**: Rank 5 ($R_5$). **Right**: Rank 10 ($R_{10}$).

and K can be found through exhaustive search with reasonable computational complexity. All of the four methods were tested on the eight test datasets, and CRRs obtained using the best found M and K are depicted in Fig.6.4, where HCF-MSC is clearly the top performer. It is worthy to mention that the results of Eigenfaces depicted in Fig.6.4 are comparable to those previously reported in [94] under similar test conditions. From the baseline results, we can roughly learn how difficult these test datasets are.

Algorithm performance depends critically on the test datasets selected. Since most query samples added to $\mathcal{Q}_3$ and $\mathcal{Q}_4$ were from the **FB** set, the simplest one in the FERET database (see [94] for definitions of various imagery categories), the CRRs of all methods were significantly improved in the corresponding sets: $\mathbf{S}_3$, $\mathbf{S}_4$. Especially, the CRR difference between HCF-MSC and HCF-Kmean was greatly reduced in the points: $\mathbf{S}_3$, $\mathbf{S}_4$ and $\mathbf{S}_5$ due to the introduction of these simple test samples.

To facilitate the performance comparison, let $R_n^{(JD-LDA.1)}$ and $R_n^{(\cdot)}$ denote the CRRs with rank n obtained by JD-LDA.1 and any one of other three methods respectively. Let us further define a quantitative statistic with respect to performance improvement against JD-LDA.1, denoted as $\xi_n^{(\cdot)} = R_n^{(\cdot)} - R_n^{(JD-LDA.1)}$. The results, summarized in

Table 6.3: CRR Improvement against JD-LDA.1 ξ_1 (%) with Rank 1.

ξ_1 (%)	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$	$\mathbf{S}_6$	$\mathbf{S}_7$	$\mathbf{S}_8$	Average
PCA	-7.30	2.36	15.43	17.20	16.63	17.97	17.58	15.95	11.98
HCF-Kmean	-2.92	11.16	22.90	27.60	24.38	25.81	25.44	23.10	19.67
HCF-MSC	3.89	14.94	22.90	27.41	25.15	25.49	26.99	27.64	21.80

Table 6.4: CRR Improvement against JD-LDA.1 ξ_5 (%) with Rank 5.

ξ_5 (%)	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$	$\mathbf{S}_6$	$\mathbf{S}_7$	$\mathbf{S}_8$	Average
PCA	-11.68	7.55	23.89	26.30	23.45	25.27	24.43	22.72	17.74
HCF-Kmean	-2.92	15.09	26.43	34.20	29.53	31.83	31.10	30.39	24.46
HCF-MSC	5.60	20.76	28.61	34.44	30.96	31.94	33.66	34.42	27.55

Table 6.5: CRR Improvement against JD-LDA.1 ξ_{10} (%) with Rank 10.

ξ_{10} (%)	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$	$\mathbf{S}_6$	$\mathbf{S}_7$	$\mathbf{S}_8$	Average
PCA	-7.30	9.43	25.79	31.50	27.96	30.52	28.31	26.54	21.59
HCF-Kmean	-1.46	13.21	25.72	36.90	31.33	34.02	32.95	32.93	25.70
HCF-MSC	8.03	17.30	30.02	36.03	32.67	34.24	35.72	36.98	28.87

Tables 6.3, 6.4, 6.5 clearly indicate that HCF-MSC, and even HCF-Kmean (except for in the case of $\mathbf{S}_1$) greatly outperform the single JD-LDA.1 system. The improvement is gradually enhanced with the size of the test dataset increasing. Also, it is of interest to observe the performance difference between JD-LDA.1 and Eigenfaces. LDA-based algorithms have shown many superior properties to PCA-based ones in the tasks of pattern classification [7, 17, 46, 74]. Not surprisingly, JD-LDA.1 outperforms Eigenfaces,

even HCF-Kmean in the smallest test dataset, $\mathbf{S}_1$, however, its performance advantage disappears rapidly, and JD-LDA.1 is outperformed by Eigenfaces as $|\mathbf{S}_i|$ increases. This is because LDA-based algorithms usually do not generalize as well as PCA-based ones and tend to be easier to overfit when they are applied to databases with a large number of classes and a small number of training samples per class. Results presented here are in line with those reported in [66, 78]. It will be further discussed at this point in Section 6.6.5. In addition, it is obvious in Fig.6.4 and Tables 6.3-6.5 that both of HCF-Kmean and HCF-MSC indeed can boost the performance of JD-LDA.1 in those larger test datasets. Thus, the advantage of the proposed HCF strategy is demonstrated.

Table 6.6: The improvement of the average CRRs by HCF-MSC against HCF-Kmean.

ζ_n (%)	$\mathbf{S}_2$	$\mathbf{S}_5$	$\mathbf{S}_8$	Average
R_1	1.8491	0.42011	2.6981	1.6558
R_5	5.6352	1.718	3.7817	3.7116
R_{10}	6.3711	2.0264	3.4961	3.9645

Since the number of clusters is a design parameter in the proposed HCF, an experimental evaluation regarding the influence of the number on the performance of HCF-Kmean and HCF-MSC is performed here. The results obtained in three representative in size databases, $\mathbf{S}_2$, $\mathbf{S}_5$ and $\mathbf{S}_8$, are as shown in Figs.6.5, 6.6, 6.7. Let $R_{n,i}$ be the CRR obtained utilizing i clusters. Similar to ξ_n, we define

$$\zeta_n = \sum_i \left(R_{n,i}^{(HCF-MSC)} - R_{n,i}^{(HCF-Kmean)} \right), \tag{6.5}$$

which quantitatively compares the CRRs of HCF-MSC and HCF-Kmean, as summarized in Table 6.6. It can be been seen from Figs.6.5-6.7 and Table 6.6 that HCF-MSC is far superior to HCF-Kmean in most cases, especially when the number of clusters used is small, and the averages of ζ_n obtained in the three test datasets are up to 1.66% (rank 1), 3.71% (rank 5) and 3.96% (rank 10). This indicates the advantage of performing cluster analysis using the proposed CSC method.

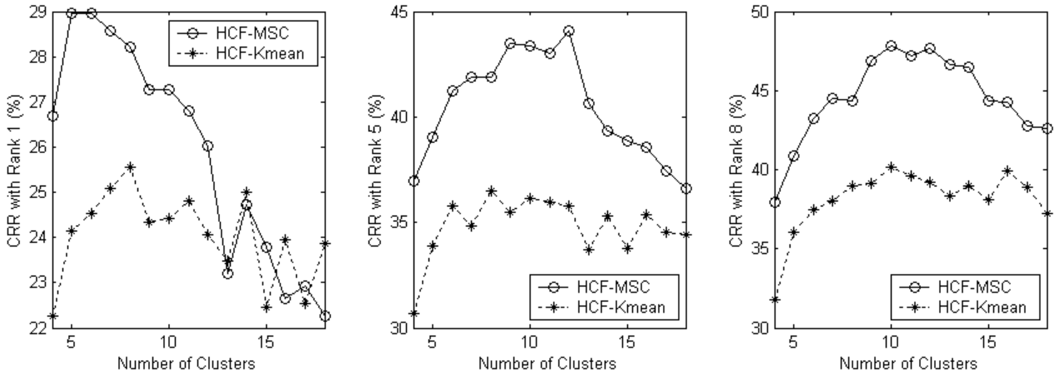

Figure 6.5: Comparison of CRRs obtained by HCF-MSC and HCF-Kmean in $\mathbf{S}_2$ as functions of the number of the clusters. **Left**: Rank 1 (R_1). **Middle**: Rank 5 (R_5). **Right**: Rank 8 (R_8).

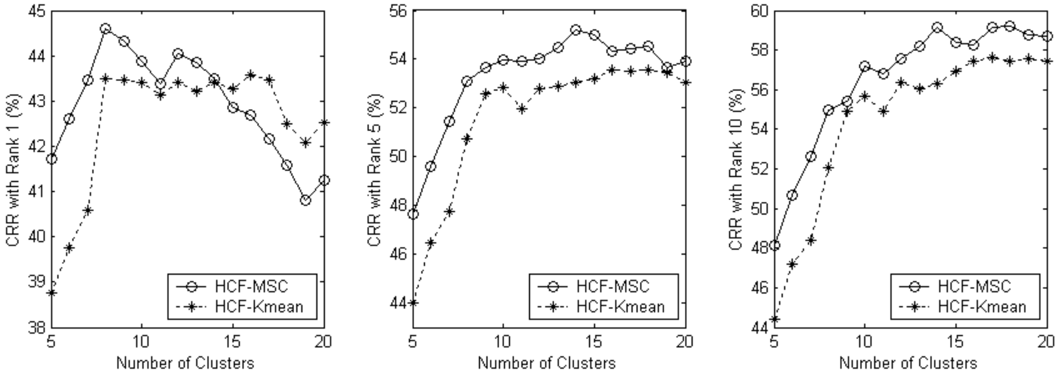

Figure 6.6: Comparison of CRRs obtained by HCF-MSC and HCF-Kmean in $\mathbf{S}_5$ as functions of the number of the clusters. **Left**: Rank 1 (R_1). **Middle**: Rank 5 (R_5). **Right**: Rank 10 (R_{10}).

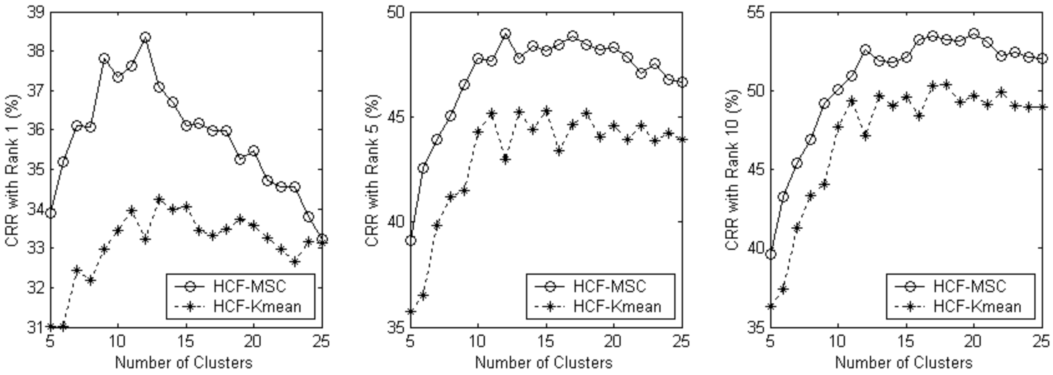

Figure 6.7: Comparison of CRRs obtained by HCF-MSC and HCF-Kmean in $\mathbf{S}_8$ as functions of the number of the clusters. **Left**: Rank 1 (R_1). **Middle**: Rank 5 (R_5). **Right**: Rank 10 (R_{10}).

6.6.4 The CRR Performance Analysis

Compared with previously published FR studies [7, 17, 46, 61, 74, 75, 120], the CRRs reported here are low, less than 70% even for the best performer HCF-MSC. In addition to the difficulty of the FERET database itself, which has been proved in the FERET FR competitions [94], there are two other reasons to which the weak performance of the algorithms evaluated here can be attributed, namely:

1. **The very small number of training samples per subject or class**. It is well-known that the learning capacity of the *discriminant feature extraction machines* (DFEMs) and the classifiers is directly proportional to the number of training samples per subject denoted as L, while reciprocally proportional to the number of the training subjects denoted as $|\mathcal{P}|$. Combining the two factors, we can define a variable called "*Learning Difficulty Degree*" (LDD): $\rho = \frac{L}{|\mathcal{P}|}$, to roughly estimate the difficulty of a discriminant learning task. Obviously, a smaller ρ value implies a more difficult task for both of the DFEMs and classifiers. In most previous FR studies that have reported outstanding CRRs, the LDD range is usually not less than $\frac{1}{25}$ (see *e.g.* [7,17,46,61,74,75,120]). However, it should be noted at this point

the LDD is between $\frac{1}{600}$ and $\frac{1}{90.5}$ ($2/1200$ in $\mathbf{S}_8 \rightarrow 2/181$ in $\mathbf{S}_1$) for the experiments presented here.

2. **A more general training process**. Of 1200 training/target subjects, only 482 were included in the query sets $\mathcal{Q}_i$. This would force each algorithm to some extent to have a general representation for faces, not a representation tuned to a specific query set. Compared with most previous FR research reports where the query sets usually contained all of training subjects, the simulation setting used in this set of experiments is more difficult. It is however a more realistic set up and matches the practical environment described by the FERET program [94].

3. **Insufficient representation of training samples for test samples**. In the experiments reported here, all the training samples were chosen from two frontal face categories, **fa** and **fb**, while most of the test samples come from the other, much more difficult to recognize, categories such as **ba**, **bj**, **bk**, **ql** *etc.* as shown in Table 2.1. As a result, it is significantly insufficient for the training samples to represent the test samples. In other word, the training data non-uniformly sample the underlying distribution. This is one of cases when LDA may fail and be outperformed by PCA as shown recently in [78].

To further substantiate our conclusions, it should be noted that another experiment conducted on a compound database using HCF-MSC with the D-LDA sub-systems inside has been reported in [71]. The compound database with 1654 face images of 157 subjects is composed of six widely used databases in the literature: the ORL database [103], the Berne database [1], the Yale database [7], the Harvard database [41], the UMIST database [36], and an Asian database constructed by our own [61]. In the experiment reported in [71], most subjects had $5 - 8$ samples chosen for learning, giving rise to $\rho = 1/35$ on average. The best found CRRs with rank 1 were 85.5% by D-LDA, 89.8% by HCF-Kmean, and 92.1% by HCF-MSC respectively.

6.6.5 Discussion

The following important conclusions can be drawn from the results presented above:

1. The size, type, and LDD of the evaluation database are the three factors that significantly impact the performance of all the FR algorithms evaluated here. The performance of the LDA-like algorithms deteriorates more rapidly than the PCA-like ones as the LDD decreases, although it is generally believed that algorithms based on LDA are superior in the FR tasks. This can be explained by the fact that PCA is an unsupervised learning method without paying any attention to the underlying class structure of the training set, so that compared to LDA, it is less sensitive to different values of the two factors, L and $|\mathcal{P}|$, defining the LDD. Therefore, we can draw here a conclusion similar to the one in [78] to some extent, that is, *PCA may outperform LDA when the LDD value of the data set considered for learning is very small or when the training data non-uniformly sample the underlying distribution* (see the example of Fig.3.3 in Section 3.3). However, through the utilization of the hierarchical scheme, it can be seen from the experiments that it is possible to consistently boost the performance of traditional LDA methods to acceptable levels across a number of environmental scenarios.

2. The hierarchical scheme introduced here seems to be relatively robust to settings of their design parameters, such as the number of clusters in which the LSFD is partitioned. Although an optimal number of clusters can be determined empirically or through the "leaving-one-out" strategy, performance is rather consistent for a class of values. On the other hand, the clustering criterion has an impact on the performance. The results summarized in Table 6.6 indicate that the separability-based criterion should be preferred to the similarity-based one for FR tasks.

3. Considering the computational complexities required for the implementation of the proposed HCF, it should be noted that it is comparable, if not less, to that of

traditional FR methods. Training at the different MSCs can be done in parallel reducing the overall processing time and memory space. The clustering procedure and the projection process at the second stage do not introduce significant additional computational cost. To the best of the authors' knowledge the partitioning mechanism introduced in this work is the only one capable of providing this form of parallel processing capability.

4. The proposed design is a scalable one. The designer controls the complexity of the procedure by determining the number and form of the individual feature learners. Depending on the problem specification and the computational constraints imposed by the design the appropriate number of clusters can be selected. Furthermore, it is possible to incorporate additional knowledge in the training process by augmenting the cluster set rather than re-training all existing learners. The exploitation of this important features, a direct result of the framework's parallel structure, is of paramount importance in practical implementations and it is currently under investigation.

6.7 Summary

This chapter introduced a general framework to improve performance of traditional FR methods when applied to LSFDs. The proposed framework utilizes a hierarchical classification scheme on top of traditional FR systems trained on database partitions obtained using a novel separability-based clustering method. Through the hierarchical design the problem of determining the appropriate distribution of the face patterns in a LSFD is transformed into the problem of combining a collection of admissible solutions obtained via traditional discriminant learning methods. This constitutes a problem of considerably reduced complexity since prior experimentation and analysis indicate that linear methods such as LDA or PCA provide cost effective solutions in smaller size databases.

Although the hierarchical implementation reported here was based on the JD-LDA approach, other appearance-based FR methods such as ICA [5], PCA, or their variants can be accommodated.

Experimentation with a set of FERET style large databases indicates that the proposed framework may be able to improve the performance of traditional FR methods in the case of LSFDs. It is anticipated that the performance enhancement obtained via the utilization of the proposed scheme will be more evident when larger sized databases are considered. Verification of the above claim, along with theoretical evaluation of the performance is currently under consideration. Future research in this area will also focus on the application of the proposed framework to content based indexing and retrieval tasks, and audio/video classification in large-scale databases.

Chapter 7

Ensemble-based Discriminant Learning with Boosting

7.1 Introduction

In this chapter, we continue the discussion of the approaches based on *a mixture of locally linear models* (AMLLM), but from the viewpoint of machine learning. It can be seen from previous presentations that most existing AMLLM-based FR methods, including the two just introduced in last chapter, HCF-MSC and HCF-Kmean, are developed based on traditional cluster analysis. As a consequence, a disadvantage to pattern classification tasks is that the sub-models' division/combination criteria used in these clustering techniques are not directly related to the *classification error rate* (CER) of the resulting classifiers, especially the true CER, which is often referred to as the *generalization error rate* in the machine-learning literature.

Recently, a machine-learning technique known as "boosting" has received considerable attention in the pattern recognition community, due to its usefulness in designing AMLLM-based classifiers, also called "ensemble-based classifiers" in the machine-learning literature [29,56,104]. The idea behind boosting is to sequentially employ a base classifier on a weighted version of the training sample set to generalize a set of classifiers of its kind. Often the base classifier is also called the "learner". These weights are updated at each iteration through a classification-error-driven mechanism. Although any individual classifier produced by the learner may only perform slightly better than random guessing, the formed ensemble can provide a very accurate (strong) classifier. It has been shown, both theoretically and experimentally, that boosting is particularly robust in preventing overfitting and reducing the generalization error by increasing the so-called ***margins*** of the training examples [13, 24, 104, 105]. The margin is defined as the minimal distance of an example to the decision surface of classification [123]. For a classifier, a larger expected margin of training data generally leads to a lower generalization error. However, the machine-learning community generally regards ensemble-based learning rules, including boosting and bagging [12], not suited to a stable learner, for instance LDA as shown in [13]. This reason is that the effectiveness of these rules depends to a great extent on the learner's "instability", which means that small changes in the training set could cause large changes in the resulting classifier. More recent simulation studies also demonstrated that boosting is not an effective method to be used in conjunction with the LDA-based learners [111].

In this chapter, we propose a new ensemble-based method to boost the performance of the traditional LDA-based algorithms in complex FR tasks. The main novelty is the introduction of the boosting technique, which is applied to address two issues central to all the ensemble-based approaches: 1) the generalization of a set of simple linear solutions, each of them targeting a particular sub-problem; 2) the formation of a globally strong solution through the aggregation of the multiple, relatively weak, local solutions. In this work, the JD-LDA method is again chosen as the learner due to its robustness against the SSS problem. However, the utilization of boosting with a strong learner such as JD-LDA contradicts the popular belief in the machine learning literature. To break this limitation, a novel weakness analysis theory is developed here. The theory attempts to boost a strong learner by increasing the diversity between the classifiers created by the learner, at the expense of decreasing their margins, so as to achieve a trad-off suggested by recent boosting studies [91] for a low generalization error. To this end, the so-called "*Learning Difficulty Degree*" (LDD), originally introduced in Section 6.6.4, is utilized in the theory to control and regulate the trade-off between the margins and the diversity of the classifiers produced during the boost process. Correspondingly, a novel loss function with respect to the LDD is proposed to quantitatively estimate the generalization power of the formed ensemble classifier. In addition, a new variable called "*pairwise class discriminant distribution*" (PCDD) is introduced to build an effective interaction mechanism between the booster and the learner. The PCDD is designed specifically for the LDA-based learner, so that the learner can be always manipulated to conquer current most *hard-to-separate pairs* (HTSP) of classes in each boosting iteration. In this way, the final result obtained by the boosting process is an ensemble of multiple relatively weak but very specific LDA solutions. The ensemble-based solution is able to take advantage of both boosting and LDA. It is shown by the FR experiments to greatly outperform any single solution created by the JD-LDA learner in various difficult learning scenarios, which include the cases with different SSS settings and the case with increased nonlinear variations.

The rest of this chapter is organized as follows. In Section 7.2, we design a learner based on the JD-LDA algorithm following the requirement of the boosting technique. In Section 7.3, we briefly review the classic AdaBoost method, and its multi-class extensions. Then, in Section 7.4, the theory and algorithm of how to boost a strong learner such as LDA are introduced and described in detail. Section 7.5 reports on a set of experiments conducted on the FERET database to demonstrate the effectiveness of the proposed methodologies. In addition, an experiment of comparing all the six discriminant learning algorithms developed in the thesis is introduced in Section 7.5.5. Finally, Section 7.6 summarizes conclusions and provides directions for future research.

7.2 A Strong Learner: JD-LDA

From the viewpoint of machine learning, the task of learning from examples can be formulated in the following way: Given a training set, $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, containing C classes with each class $\mathcal{Z}_i = \{(\mathbf{z}_{ij}, y_{ij})\}_{j=1}^{C_i}$ consisting of a number of face images $\mathbf{z}_{ij} \in \mathbb{R}^J$ and their class labels y_{ij}, a total of $N = \sum_{i=1}^{C} C_i$ face images are available in the set. The class label of the example $\mathbf{z}_{ij}$ is $y_{ij} = i$, which is in the label set $\mathbb{Y} = \{1, \cdots, C\}$. Taking as input such a set $\mathcal{Z}$, the objective of learning is to estimate a function or classifier $h(\mathbf{z}) : \mathbb{R}^J \to \mathbb{Y}$, such that h will correctly classify unseen examples $(\mathbf{z}, y)$.

Under the boosting framework, the learner works like a *classifier generator*, which iteratively creates classifiers of its kind, h_t, but each one with a different focus on accounting for the patterns under learning. In this work, the JD-LDA method [74] is chosen as the learner due to its cost-effective solution to the SSS problem. However, as we know from previous chapters, JD-LDA mainly functions as a feature extractor, which outputs an M-dimensional feature space spanned by Ψ, where any face image $\mathbf{z}$ is represented as $\mathbf{y} = \Psi^T \mathbf{z}$, $\mathbf{y} \in \mathbb{R}^M$ with enhanced discriminant power. To act as a learner, a subsequent classifier is needed. In theory, the classification in the feature space can be performed by using any classifier. However, from the viewpoint of reducing the overfitting chances in the context of boosting, a simple discriminant function that explains most of the data is preferable to a complex one. Consequently, a classic *nearest center classifier* (NCC) is adopted here for the classification task. Thus the so-called JD-LDA learner is a complete FR system consisting of a feature extractor followed by a classifier. For simplicity in the following presentations, we denote the feature extraction part of the JD-LDA learner in the form of a function $\mathcal{L}(\cdot)$, which has $(\Psi, \{\bar{\mathbf{z}}_i\}_{i=1}^{C}) = \mathcal{L}(\mathcal{Z})$, where $\bar{\mathbf{z}}_i = \frac{1}{C_i}\sum_{j=1}^{C_i} \mathbf{z}_{ij}$ is the known center of the class i. For completeness, the detailed pseudo code implementation of the JD-LDA feature extractor embedded in the boosting process is depicted in Fig.7.1.

The NCC adopted here is based on a normalized Euclidean distance, which is equivalent to the linear form of a generalized membership function defined in [96]. The nor-

```
Input: A training set $\mathcal{Z}_t$ with $C$ classes: $\mathcal{Z}_t = \{\mathcal{Z}_{i,t}\}_{i=1}^{C}$, each class containing
    $\mathcal{Z}_{i,t} = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ face images, $\mathbf{z}_{ij} \in \mathbb{R}^J$; the regularization parameter $\eta$.
Output: A $M$-dimensional LDA subspace spanned by $\Psi_t$, a $M \times J$ matrix with
    $M \ll J$, and the class centers $\{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C}$.
Algorithm:
    Step 1. Re-write $\hat{\mathbf{S}}_{b,t}$ of Eq.7.12: $\hat{\mathbf{S}}_{b,t} = \mathbf{W}_b \mathbf{W}_b^T$, where $\mathbf{W}_b = [\phi_1, \cdots, \phi_c]$.
    Step 2. Find the eigenvectors of $\mathbf{W}_b^T \mathbf{W}_b$ with non-zero eigenvalues, and
        denote them as $\mathbf{E}_m = [\mathbf{e}_1, \ldots, \mathbf{e}_m]$, $m \le C - 1$.
    Step 3. Calculate the first $m$ most significant eigenvectors $(\mathbf{V})$ of $\hat{\mathbf{S}}_{b,t}$ and their
        corresponding eigenvalues $(\Lambda_b)$ by $\mathbf{V} = \mathbf{W}_b \mathbf{E}_m$ and $\Lambda_b = \mathbf{V}^T \hat{\mathbf{S}}_{b,t} \mathbf{V}$.
    Step 4. Let $\mathbf{U} = \mathbf{V} \Lambda_b^{-1/2}$. Find eigenvectors of $\mathbf{U}^T(\hat{\mathbf{S}}_{b,t} + \eta \cdot \hat{\mathbf{S}}_{w,t})\mathbf{U}$, $\mathbf{P}$, where
        $\hat{\mathbf{S}}_{w,t}$ is defined in Eq.7.14.
    Step 5. Choose the $M(\le m)$ eigenvectors in $\mathbf{P}$ with the smallest eigenvalues.
        Let $\mathbf{P}_M$ and $\Lambda_w$ be the chosen eigenvectors and their corresponding
        eigenvalues respectively.
    Step 6. Return $\Psi_t = \mathbf{U} \mathbf{P}_M \Lambda_w^{-1/2}$ and $\{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C}$.
```

Figure 7.1: The pseudo code implementation of the JD-LDA feature extractor: $\mathcal{L}(\mathcal{Z}_t)$ in the t-th boosting iteration, where the input $\mathcal{Z}_t = \mathcal{R}_t$, and $\mathcal{R}_t \subset \mathcal{Z}$ is an adaptively updated subset defined in Section 7.4.2.

malized Euclidean distance can be expressed as

$$d(\mathbf{z}, i, \mathcal{L}) = (d_{max} - d_{\mathbf{z},i})/(d_{max} - d_{min}) \tag{7.1}$$

where $d_{\mathbf{z},i} = \left\| \Psi^T(\mathbf{z} - \bar{\mathbf{z}}_i) \right\|$, $d_{max} = \max(\{d_{\mathbf{z},i}\}_{i=1}^{C})$, and $d_{min} = \min(\{d_{\mathbf{z},i}\}_{i=1}^{C})$. Based on the NCC rule, the class label $y(\mathbf{z})$ of an input example $\mathbf{z}$ can be inferred through

$$y(\mathbf{z}) = \arg\max_{i} d(\mathbf{z}, i, \mathcal{L}). \tag{7.2}$$

The classification score $d(\mathbf{z}, i, \mathcal{L})$ has values in $[0, 1]$, and thus it can fulfill the functional requirement of the boosting algorithm (AdaBoost.M2 [29]), indicating a "degree of plausibility" for labeling $\mathbf{z}$ as the class i. Since a classifier h such as the NCC discussed here usually yields two outputs, the classification score $d(\mathbf{z}, i, \mathcal{L})$ and the class label $y(\mathbf{z})$, we denote

$$h(\mathbf{z}) = y(\mathbf{z}), \tag{7.3}$$

and

$$h(\mathbf{z}, i) = d(\mathbf{z}, i, \mathcal{L}) \tag{7.4}$$

for the distinguishing purposes.

7.3 AdaBoost and Its Multi-class Extensions

Since the boosting method proposed here is developed from AdaBoost [29]. We begin with a brief review of the algorithm and its multi-class extensions.

To find a strong (accurate) classifier $h(\mathbf{z}) : \mathbb{R}^J \rightarrow \mathbb{Y}$, AdaBoost works by applying a given weak learner to a weighted version of the training set repeatedly in a series of rounds $t = 1, \cdots, T$, and then linearly combines these classifiers h_t produced by the learner into a single composite classifier h_f. On each round t, the weights, also called "sample distribution" over $\mathcal{Z}$, denoted $D_t(\mathbf{z}_{ij})$, are updated through an error-driven mechanism. The weights of incorrectly classified examples are increased by

$$D_{t+1}(\mathbf{z}_{ij}) = D_t(\mathbf{z}_{ij}) \cdot \sqrt{(1 - \epsilon_t)/\epsilon_t} \tag{7.5}$$

where $\epsilon_t = \sum\limits_{i,j:h_t(\mathbf{z}_{ij}) \neq y_{ij}} D_t(\mathbf{z}_{ij})$ is the training error of h_t. In this way, during the next round the learner is forced to focus on the hard-to-classify examples. The pseudo code implementation of solving a general two-class problem using the AdaBoost algorithm is depicted in Fig.7.2.

```
Input: A set of $N$ training samples, $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^J$ and
    $y_i \in \mathbb{Y} = \{-1, +1\}$ for two classes.
Initialize sample distribution $D_1(\mathbf{x}_i) = \frac{1}{N}$.
Do for $t = 1, \cdots, T$:
    1. Train weak learner using the sample distribution $D_t$.
    2. Get weak classifier $h_t : \mathbb{R}^J \rightarrow \mathbb{Y}$ with training error: $\epsilon_t = \sum_{i:h_t(\mathbf{x}_i) \neq y_i} D_t(\mathbf{x}_i)$.
    3. Choose $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$.
    4. Update: $D_{t+1}(\mathbf{x}_i) = D_t(\mathbf{x}_i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(\mathbf{x}_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(\mathbf{x}_i) \neq y_i \end{cases}$
    5. Normalize $D_{t+1}$ so that it will be a distribution, $D_{t+1}(\mathbf{x}_i) \leftarrow \frac{D_{t+1}(\mathbf{x}_i)}{\sum_{i=1}^{N} D_{t+1}(\mathbf{x}_i)}$.
Output the final composite classifier,
    $$h_f(\mathbf{x}) = \text{sign}\left( \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right).$$
```

Figure 7.2: The AdaBoost algorithm.

One of the most interesting features of AdaBoost is its ability to reduce the potential of overfitting and the generalization error, even as T becomes large. Schapire *et al.* [105] show that the following upper bound on the generalization error $\mathbf{P}_{err}$

$$\mathbf{P}_{err} \leq P_{\mathcal{Z}}(\varrho_{h_f}(\mathbf{z}, y) \leq \theta) + O\left(\sqrt{\frac{\kappa \log^2(NC/\kappa)}{N\theta^2} - \frac{\log(\delta)}{N}} \right) \tag{7.6}$$

holds with probability at least $1 - \delta$ for all $\theta > 0$, where $0 < \delta < 1$, κ is the Vapnik-Chervonenkis (VC)-dimension of the function space or class that the weak learner belongs to, $\varrho_{h_f}(\mathbf{z}, y)$ denotes the margin of example $(\mathbf{z}, y)$ classified by the final classifier h_f, and $P_{\mathcal{Z}}(\varrho_{h_f}(\mathbf{z}, y) \leq \theta)$ is the so-called "*cumulative margin distribution*" (CMD). The VC dimension κ is used to capture the complexity of a function class [123,124]. Roughly speaking, the VC dimension measures how many (training) points can be shattered (*i.e.* separated) for all possible labellings using functions of the class. The CMD, $P_{\mathcal{Z}}(\varrho_{h_f}(\mathbf{z}, y) \leq \theta)$, represents the fraction of the training examples with margin $\varrho_{h_f}(\mathbf{z}, y) \leq \theta$. Following the definition given in [105], $\varrho_{h_f}(\mathbf{z}, y)$ is the quantity

$$\varrho_{h_f}(\mathbf{z}, y) = h_f(\mathbf{z}, y) - \max\{h_f(\mathbf{z}, k) : k \in \mathbb{Y}, k \neq y\} \tag{7.7}$$

where $h_f(\mathbf{z}, k)$ is the classification score produced when $\mathbf{z}$ is identified as the class k. The example $(\mathbf{z}, y)$ is misclassified by h_f if and only if $\varrho_{h_f}(\mathbf{z}, y) < 0$. The upper bound of Eq.7.6 with respect to the margin links Adaboost with Vapnik's maximal margin classifier theory [123], and helps to understand why often the generalization error of AdaBoost continues to decrease even after the training data has been fitted perfectly [105].

AdaBoost is originally developed to support binary classification tasks. Its multi-class extensions include two variants, AdaBoost.M1 and AdaBoost.M2 [29]. AdaBoost.M1 is the most straightforward generalization. However, the algorithm has to be halted if the training CER of the weak classifier h_t produced in any iterative step is $\geq 50\%$. For the multi-class problems, this means that these weak classifiers h_t need to be much stronger than random guessing, whose expected error is $(1 - 1/C)$ with C being the number of classes. The requirement is quite strong and may often be hard to meet. For example, our experimental analysis indicates that the limitation often stops the algorithm too early, resulting in insufficient classification capabilities [104, 105]. To avoid the problem, rather than the ordinary CER ϵ_t, AdaBoost.M2 targets minimization of a more sophisticated error measure called "pseudo-loss", $\hat{\epsilon}_t$, which has the following expression,

$$\hat{\epsilon}_t = \frac{1}{2} \sum_{(\mathbf{z}_{ij}, y) \in B} \Upsilon_t(\mathbf{z}_{ij}, y) \left(1 - h_t(\mathbf{z}_{ij}, y_{ij}) + h_t(\mathbf{z}_{ij}, y)\right) \tag{7.8}$$

where $\Upsilon_t(\mathbf{z}_{ij}, y)$ is the so-called "mislabel distribution" defined over the set of all mislabels:

$$B = \{(\mathbf{z}_{ij}, y) : \mathbf{z}_{ij} \in \mathcal{Z}, \mathbf{z}_{ij} \in \mathbb{R}^J, y \in \mathbb{Y}, y \neq y_{ij}\}. \tag{7.9}$$

Let $\beta_t = \hat{\epsilon}_t/(1 - \hat{\epsilon}_t)$, the mislabel distribution is updated through,

$$\Upsilon_{t+1}(\mathbf{z}_{ij}, y) = \Upsilon_t(\mathbf{z}_{ij}, y) \cdot \beta_t^{(1 + h_t(\mathbf{z}_{ij}, y_{ij}) - h_t(\mathbf{z}_{ij}, y))/2} \tag{7.10}$$

With the pseudo-loss $\hat{\epsilon}_t$, the boosting process can continue as long as the classifier produced has pseudo-loss slightly better than random guessing. Also, the introduction of the mislabel distribution enhances the communication between the learner and the booster, so that AdaBoost.M2 can focus the learner not only on hard-to-classify examples, but more specifically, on the incorrect labels [29]. Based on these reasons, we develop the ensemble-based discriminant algorithm proposed in the next section under the framework of AdaBoost.M2.

In this work, we choose the LDA variant, JD-LDA, as the learner. Compared to traditional learners used in the boosting algorithms, the LDA-based learner should be emphasized again at the following two points. (1) The utilization of JD-LDA obviously contradicts the common belief of weak learners in the boosting literature, given the fact that JD-LDA has been shown to be a rather strong and particularly stable learner in FR tasks [74, 75]. (2) The JD-LDA learner is composed of a LDA-based feature extractor and a nearest center classifier. However, it can be seen in Section 7.2 that the learning focus of JD-LDA is not on the classifier but on the feature extractor. It is rather different at this point from earlier boosting designs where the weak learners are used only as *pure* classifiers without concerning feature extraction. Therefore, accommodating a learner such as JD-LDA requires a more general boosting framework, which should be able to break the limitation of weak learners. To highlight these significant difference, we call "*gClassifier*" the more general classifier created by the JD-LDA learner in the rest of the chapter.

7.4 Boosting A Strong Learner: JD-LDA

7.4.1 Interaction between the LDA learner and the booster

To boost a learner, we first have to build a strong connection between the learner and the boosting framework. In AdaBoost, this is implemented by manipulating the so-called "sample distribution", which is a measure of how hard to classify an example. However, we need a more specific connecting variable in this work, given the fact that the nature of LDA is a feature extractor, whose objective is to find a linear mapping to enhance the separability of different subjects under learning. For this purpose, a new distribution called "pairwise class discriminant distribution" (PCDD), A_{pq}, is introduced here. The PCDD is developed from the mislabel distribution Υ_t of AdaBoost.M2. Defined on any one pair of classes $\{(p,q) : p, q \in \mathbb{Y}\}$, the PCDD can be computed at the t-th iteration as:

$$A_t(p,q) = \begin{cases} \frac{1}{2}\left(\sum\limits_{j=1}^{C_p} \Upsilon_t(\mathbf{z}_{pj}, q) + \sum\limits_{j=1}^{C_q} \Upsilon_t(\mathbf{z}_{qj}, p)\right), & \text{if } p \neq q \\ 0, \text{ otherwise} \end{cases} \tag{7.11}$$

where C_p and C_q are the element number in classes $\mathcal{Z}_p$ and $\mathcal{Z}_q$ respectively. As it is known from the AdaBoost.M2 development, the mislabel distribution $\Upsilon_t(\mathbf{z}_{ij}, y)$ indicates the extent of difficulty in distinguishing the example $\mathbf{z}_{ij}$ from the incorrect label y based on the feedback information from the preceding $(t-1)$ gClassifiers. Thus, $A_t(p,q)$ can be intuitively considered as a measure of how important it is to discriminate between the classes p and q when designing the current gClassifier h_t. Obviously, a larger value of $A_t(p,q)$ implies worse separability between the two classes. It is therefore reasonable to drive a LDA-based learner such as JD-LDA through $A_t(p,q)$, so that it is focused specifically on the *hard-to-separate pairs* (HTSP) of classes. To this end, rather than the ordinary definition of the between-class scatter matrix $\mathbf{S}_b(= (1/N)\sum_{i=1}^{C} C_i(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})(\bar{\mathbf{z}}_i - \bar{\mathbf{z}})^T$ where $\bar{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C_i} \mathbf{z}_{ij}$ is the average of the ensemble), we introduce a variant of $\mathbf{S}_b$, which can be expressed as:

$$\hat{\mathbf{S}}_{b,t} = \sum_{p=1}^{C} \phi_p \phi_p^T \tag{7.12}$$

where

$$\phi_p = (C_p/N)^{1/2} \sum_{q=1}^{C} A_t^{1/2}(p,q)(\bar{\mathbf{z}}_p - \bar{\mathbf{z}}_q). \tag{7.13}$$

It should be noted at this point that the variant $\hat{\mathbf{S}}_{b,t}$ weighted by A_t embodies the design principle behind the so-called "fractional-step" LDA presented in [70] (see also Section

3.4.2). According to this principle, object classes that are difficult to be separated in the low-dimensional output spaces $(\Psi_1, \cdots, \Psi_{t-1})$ generalized in previous rounds can potentially result in mis-classification. Thus, they should be paid more attention by being more heavily weighted in the high-dimensional input space of the current (t-th) round, so that their separability is enhanced in the resulting feature space Ψ_t. It can be easily seen that the variant $\hat{\mathbf{S}}_{b,t}$ reduces to $\mathbf{S}_b$ when $A_t(p,q)$ is equal to a constant.

Similarly, the weighted version of the within-class scatter matrix $\mathbf{S}_w$ can be given as follows,

$$\hat{\mathbf{S}}_{w,t} = N \cdot \sum_{i=1}^{C} \sum_{j=1}^{C_i} \hat{D}_t(\mathbf{z}_{ij})(\mathbf{z}_{ij} - \bar{\mathbf{z}}_i)(\mathbf{z}_{ij} - \bar{\mathbf{z}}_i)^T \tag{7.14}$$

where

$$\hat{D}_t(\mathbf{z}_{ij}) = \sum_{y \neq y_{ij}} \Upsilon_t(\mathbf{z}_{ij}, y) \tag{7.15}$$

is defined over $\mathcal{Z}$ as the sample distribution, similar to the $D_t(\mathbf{z}_{ij})$ given in AdaBoost. Since $\hat{D}_t(\mathbf{z}_{ij})$ is derived indirectly from the pseudo-loss $(\hat{\epsilon})$ through Υ_t, we call $\hat{D}_t(\mathbf{z}_{ij})$ a "pseudo sample distribution" for the purpose of distinguishing it from $D_t(\mathbf{z}_{ij})$. It is not difficult to see that a larger value of $\hat{D}_t(\mathbf{z}_{ij})$ implies a harder-to-classify example for those preceding gClassifiers.

Often the upper bound of Eq.7.6, which is only concerned with maximizing the margin, is found to be too loose in practical applications [91, 105]. Recently, Murua [91] proposed an improved bound with respect to the margin and the dependence between the classifiers involved in the linear combination. The theory behind Murua's bound reveals that, to achieve a low generalization error, the boosting procedure should not only create classifiers with large expected margins, but also keep their dependence low or weak. Obviously, classifiers trained with more overlapping examples will result in stronger dependence between them. A way to avoid building similar gClassifiers repeatedly is to artificially introduce some randomness in the construction of the training data.

To this end, we can introduce a modified PCDD, given by

$$\hat{A}_t(p,q) = \begin{cases} \frac{1}{2}\left(\sum\limits_{j:g_t(\mathbf{z}_{pj})=q} \hat{D}_t(\mathbf{z}_{pj}) + \sum\limits_{j:g_t(\mathbf{z}_{qj})=p} \hat{D}_t(\mathbf{z}_{qj})\right), \text{ if } p \neq q \\ 0, \text{ otherwise} \end{cases} \tag{7.16}$$

where $g_t(\mathbf{z}) = \arg\max\limits_{y\in\mathbf{Y}} h_t(\mathbf{z},y)$. As a result of using $\hat{A}_t(p,q)$ instead of Eq.7.11, it can be seen that only those subject sets $\mathcal{Z}_i$ that include the mislabeled examples by the last gClassifier h_{t-1} are contributing to the construction of the current gClassifier h_t (through $\hat{\mathbf{S}}_{b,t}$). Thus, by manipulating $\hat{A}_t(p,q)$, we can reduce the overlapping extent of training examples used to build different gClassifiers, and thereby reach the goal of weakening the dependence among these gClassifiers. Also, this has the effect of forcing every gClassifier to focus only on the HTSP classes indicated by its preceding gClassifier, resulting in a more diverse committee of gClassifiers to be generalized in the end. On the other hand, the classification ability of the individual gClassifier h_t is to some extent weakened due to less training examples being involved. This weakening may result in decrease in the examples' margins as will be shown in the experimental section 7.5.4. However, it should be noted at this point that there appears to be a trade-off between weak dependence and large expected margins to achieve a low generalization error as suggested by Murua's bound [91]. Experimentation to be reported later indicates that in many cases, the utilization of $\hat{A}_t(p,q)$ may yield a better balance than that obtained by $A_t(p,q)$, improving the classification performance.

Based on the introduction of $A_t(p,q)$, $\hat{A}_t(p,q)$, $\hat{D}_t(\mathbf{z}_{ij})$, $\hat{\mathbf{S}}_{b,t}$ and $\hat{\mathbf{S}}_{w,t}$, we can now give a new boosting algorithm, as depicted in Fig.7.3, from which it can be seen that the JD-LDA learner at every iteration is tuned to conquer a particular sub-problem generalized by the feedback Υ_t in a manner similar to "automatic gain control". As an effect, every produced solution (gClassifier) can offer complementary information about the patterns to be classified. The final solution can be considered as a mixture of T JD-LDA based recognizers obtained by linearly weighted combination. Either $A_t(p,q)$ or $\hat{A}_t(p,q)$ can be used during the boosting process. In the remainder of the chapter, we

```
Input: A set of training images $\mathcal{Z} = \{(\mathbf{z}_{ij}, y_{ij})_{j=1}^{C_i}\}_{i=1}^{C}$ with labels $y_{ij} = i \in \mathbb{Y}$, where
    $\mathbb{Y} = \{1, \cdots, C\}$; the chosen weak learner, JD-LDA; and the iteration number, $T$.
Let $B = \{(\mathbf{z}_{ij}, y) : \mathbf{z}_{ij} \in \mathcal{Z}, \mathbf{z}_{ij} \in \mathbb{R}^J, y \in \mathbb{Y}, y \neq y_{ij}\}$.
Initialize $\Upsilon_1(\mathbf{z}_{ij}, y) = \frac{1}{|B|} = \frac{1}{N(C-1)}$, the mislabel distribution over $B$.
( For simplicity, we denote the JD-LDA feature extractor as a function $\mathcal{L}(\cdot)$, which
  has $(\Psi_t, \{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C}) = \mathcal{L}(\mathcal{R}_t, \hat{D}_t, A_t)$. )
Do for $t = 1, \cdots, T$:
    1. Update the pseudo sample distribution: $\hat{D}_t(\Upsilon_t)$, and the PCDD: $A_t$ with Eq.7.11.
    2. If $t = 1$ then randomly choose $r$ samples per class to form a learning set $\mathcal{R}_1 \subset \mathcal{Z}$.
       else choose $r$ hardest samples per class based on $\hat{D}_t$ to form $\mathcal{R}_t \subset \mathcal{Z}$.
    3. Train a JD-LDA feature extractor with $\mathcal{L}(\mathcal{R}_t, \hat{D}_t, A_t)$ to obtain $(\Psi_t, \{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C})$.
    4. Build a gClassifier $h_t = d(\Psi_t, \{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C})$ with Eq.7.1, apply it into the entire
       training set $\mathcal{Z}$, and get back corresponding hypotheses, $h_t : \mathbb{R}^J \times \mathbb{Y} \rightarrow [0, 1]$.
    5. Calculate the pseudo-loss produced by $h_t$:
         $\hat{\epsilon}_t = \frac{1}{2} \sum_{(\mathbf{z}_{ij}, y) \in B} \Upsilon_t(\mathbf{z}_{ij}, y) \left(1 - h_t(\mathbf{z}_{ij}, y_{ij}) + h_t(\mathbf{z}_{ij}, y)\right)$.
    6. Set $\beta_t = \hat{\epsilon}_t / (1 - \hat{\epsilon}_t)$. If $\beta_t = 0$, then set $T = t - 1$ and abort loop.
    7. Update the mislabel distribution $\Upsilon_t$:
         $\Upsilon_{t+1}(\mathbf{z}_{ij}, y) = \Upsilon_t(\mathbf{z}_{ij}, y) \cdot \beta_t^{(1 + h_t(\mathbf{z}_{ij}, y_{ij}) - h_t(\mathbf{z}_{ij}, y))/2}$.
    8. Normalize $\Upsilon_{t+1}$ so that it is a distribution,
         $\Upsilon_{t+1}(\mathbf{z}_{ij}, y) \leftarrow \frac{\Upsilon_{t+1}(\mathbf{z}_{ij}, y)}{\sum_{(\mathbf{z}_{ij}, y) \in B} \Upsilon_{t+1}(\mathbf{z}_{ij}, y)}$.
Output the final composite gClassifier,
         $h_f(\mathbf{z}) = \arg\max_{y \in \mathbb{Y}} \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right) h_t(\mathbf{z}, y)$.
Note: the above pseudo code is for B-JD-LDA.$A$, simply replacing all $A_t$ with $\hat{A}_t$
      to obtain B-JD-LDA.$\hat{A}$.
```

Figure 7.3: The Algorithm of Boosting JD-LDA (B-JD-LDA).

call "B-JD-LDA.A" the algorithm utilizing $A_t(p,q)$, while "B-JD-LDA.$\hat{A}$" indicates the one employing $\hat{A}_t(p,q)$.

7.4.2 A Cross-validation Mechanism to Weaken the Learner

As we mentioned earlier, JD-LDA itself has been a rather strong and stable learner in terms of classification ability. As a consequence, two problems are often encountered: (1) the gClassifiers created exhibit a high similarity or mutual dependence, given the same training data; (2) the pseudo-loss $\hat{\epsilon}_t = 0$ is often obtained halting the boosting process too early. To solve the problems, we have to artificially weaken the gClassifiers and increase their diversity accordingly. Generally speaking, the learning capacity of any LDA-like algorithm is directly proportional to the number of training examples per subject, L, and reciprocally proportional to the number of the subjects, C. To take advantage of the two factors, we can utilize the so-called *Learning Difficulty Degree* (LDD): $\rho = \frac{L}{C}$, originally introduced in Section 6.6.4. With the LDD, it can be seen from previous experiments (Section 6.6) that we can roughly estimate the degree of difficulty for the discriminant learning task on hand. It should be noted that the average $L = \frac{1}{C}\sum_{i=1}^{C} C_i$ is considered as subjects are allowed to have different number of training examples, C_i. Obviously, a smaller ρ value implies a more difficult learning task. In other words, if a learner is trained with different sample sets, the classification strength of the obtained gClassifiers will be different: *a sample set with a smaller ρ value leads to a weaker gClassifier.* Thus, from the training data point of view, the LDD provides a qualitative measure of the weakness of the gClassifiers created by the same learner. For the purpose of distinguishing the two meanings, we denote the LDD as ρ_t when it is used to express the difficulty degree of a learning task (see *e.g.* Section 6.6.4), while ρ_l denotes the weakness of a gClassifier.

Based on the above analysis, we can introduce into the proposed B-JD-LDA framework the cross-validation mechanism depicted in Fig.7.4. With the mechanism in place, only a subset of the entire training set $\mathcal{Z}$, $\mathcal{R}_t \subset \mathcal{Z}$, is used to train the JD-LDA learner.

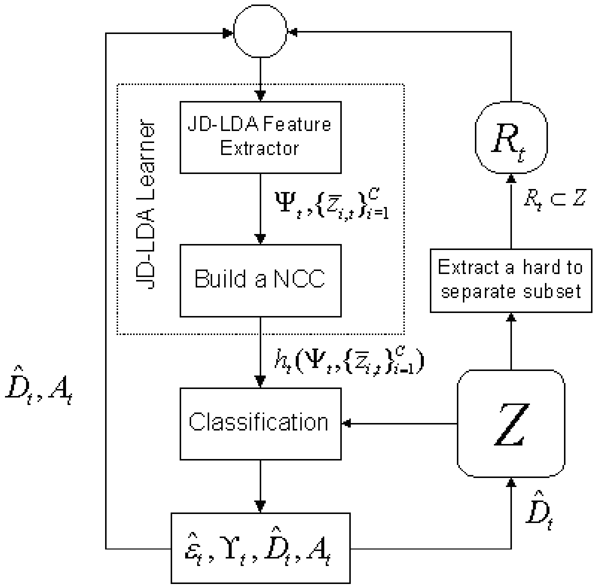
JD-LDA Learner
JD-LDA Feature Extractor
$\Psi_t, \{\bar{z}_{i,t}\}_{i=1}^{C}$
Build a NCC
$h_t(\Psi_t, \{\bar{z}_{i,t}\}_{i=1}^{C})$
Classification
$\hat{\varepsilon}_t, \Upsilon_t, \hat{D}_t, A_t$
$\hat{D}_t, A_t$
R_t
$R_t \subset Z$
Extract a hard to separate subset
Z
$\hat{D}_t$

Figure 7.4: The flow chart of the cross-validation mechanism embedded in B-JD-LDA to weaken the JD-LDA learner. The flow chart is based on one iteration, and the NCC denotes the nearest center classifier.

The subset $\mathcal{R}_t$ is formed in each iteration by choosing the $r \leq L$ hardest to classify examples per class based on current values of $\hat{D}_t(\mathbf{z}_{ij})$. Please note that $|\mathcal{R}_t| = C \cdot r$, where $|\mathcal{R}_t|$ denotes the size of $\mathcal{R}_t$. In the sequence, the obtained JD-LDA feature extractor $(\Psi_t, \{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C}) = \mathcal{L}(\mathcal{R}_t, \hat{D}_t, A_t \text{ or } \hat{A}_t)$ are used to build a gClassifier, $h_t = h(\Psi_t, \{\bar{\mathbf{z}}_{i,t}\}_{i=1}^{C})$, based on the nearest center rule. The gClassifier is applied to the entire training set $\mathcal{Z}$ including those unseen, to the learner, examples $(\mathcal{Z} - \mathcal{R}_t)$. All the variables defined on $\mathcal{Z}$ such as $\hat{\epsilon}_t$, Υ_{t+1}, $\hat{D}_t$, and $A_t/\hat{A}_t$ are then reported and used in the next iteration. The detailed implementation steps of the mechanism have been embedded in Fig.7.3.

It can be seen that under the proposed cross-validation strategy, the LDD (ρ_t) value of the sample set used to train the JD-LDA learner decreases to $\frac{r}{C}$ from $\frac{L}{C}$ (note: $r \leq L$) in each iteration. Following the weakness analysis described above, this equivalently weakens the gClassifiers produced by the learner. At the same time, since each iteration feeds the learner a different subset of the entire training set, this essentially increases the diversity among these gClassifiers. Also, it should be added at this point that one of side-effects of using only r examples per subject during the construction of each gClassifier is obtaining a better estimate of the pseudo-loss $\hat{\epsilon}_t$. This is achieved by using what Leo Breiman calls the "out-of-bag" samples (those samples not used during the training of the classifier) to estimate the error rate [11]. Hence finding the optimal r also provides a balance between good classifier performance and an improved estimate of the CER.

7.4.3 Estimation of Appropriate Weakness

The cross-validation mechanism introduced above greatly enhances the strength of the proposed boosting algorithm, but also raises the problem of model selection, that is, the determination of the optimal $\rho_l(r)(=r/C)$. As we know from the analysis in last section, a smaller/larger ρ_l value will equivalently lead to a weaker/stronger gClassifier, given the same learner. However, boosting may fail when either too weak (*e.g.* $r = 2$) or too strong (*e.g.* $r = L$) classifiers are constructed for combination [104]. Consequently, we can conjecture that a gClassifier produced with appropriate weakness should have a ρ_l value in between $\frac{2}{C}$ and $\frac{L}{C}$. Intuitively it is reasonable to further assume that a stronger gClassifier should lead to a lower empirical CER, while a learner, trained on a smaller fraction of the training set (*i.e.* a smaller size $\mathcal{R}_t$), should generalize a weaker but more diverse committee of gClassifiers with each one having a more honest estimate of misclassification. Thus, a sort of loss function with respect to r that balances the two factors can be used to drive the model selection process. To this end, the proposed here function is defined as follows,

$$\mathbf{R}(r) = \left(\frac{1}{T} \sum_{t=1}^{T} \sum_{i,j} Pr[h_{t,r}(\mathbf{z}_{ij}) \neq y_{ij}] \right) + \lambda \cdot \sqrt{\frac{\rho_l(r)}{\rho_l(L)}} \tag{7.17}$$

where $\sum\limits_{i,j} Pr[h_{t,r}(\mathbf{z}_{ij}) \neq y_{ij}]$ is the empirical CER obtained by applying the gClassifier h_t constructed by $\mathcal{L}(\mathcal{R}_{t,r})$ to the training set $\mathcal{Z}$, $\rho_l(r) = \frac{r}{C}$, $\rho_l(L) = \frac{L}{C}$, and λ is a regularization parameter that controls the trade-off between the classification strength and the diversity of the gClassifiers. It can be seen that the trade-off embodied in Eq.7.17 implements the design principles described earlier in the sense that in order to compensate for high empirical error, the gClassifiers should have low mutual dependence, and vice versa. The square root introduced in the last (penalty) term of Eq.7.17 embodies the traditional relationship between the empirical error and the number of the training samples as they appear in a loss function (see *e.g.* Eq.7.6) for the purpose of pattern classification [54, 76]. With the introduction of the loss, the task of finding gClassifiers with the appropriate weakness or the optimal $\rho_l(r)$ value can be translated to minimizing $\mathbf{R}(r)$ with respect to r. As will be seen in the experiments reported here, the estimation results through $\mathbf{R}(r)$ look rather accurate across various settings of the parameters (r, L).

In this work, the weakness analysis theory, including the cross-validation mechanism of weakening a strong learner and the subsequent estimation method of appropriate weakness, is developed for the JD-LDA learner. However, it can be seen from the above presentations that both the two methods are dependent only on the training set, where each subject is required to have at least two examples. As a result, a traditional boosting framework enhanced with the weakness analysis theory is applicable to work with any general (weak/strong) learners. This exhibits a considerably promising approach to break the traditional limitation of the weak learners in the boosting literature.

7.4.4 Determination of Convergence

Besides the weakness extent of the individual gClassifier, the iteration number T, *i.e.* the number of the gClassifiers considered in the mixture, significantly influences the performance of the final composite gClassifier h_f. As it was mentioned earlier, since boosting is particularly effective in increasing the fraction of training examples with large margin, the generalization error of h_f often continues to drop as T becomes large even long after the training error reaches zero [105]. However, the phenomenon also leads to the difficulty in determining when the boosting procedure should be stopped in order to avoid possible overfitting.

Considering the relationship between boosting and the margin theory, intuitively, it is reasonable to use the cumulative margin distribution of the training examples as an indicator to roughly estimate an appropriate value of T. In other words, we can observe the changes of the margins of the training examples at every boosting iteration, and consider it convergent when the margins of most training examples stop increasing or are increasing slowly. However, the upper bound of Eq.7.6 on which the margin theory of boosting was built has been shown to be quite coarse in many practical applications. It is therefore unrealistic to expect that the heuristic approach can accurately estimate the optimal value of T. It should be noted at this point that the determination of the optimal T value is still an open research problem for the machine learning community, the solution of which is beyond the scope of this chapter.

7.5 Experimental Results

7.5.1 The FR Evaluation Design

To show the high complexity of the face pattern distribution, in the experiments we use the two evaluation databases, $\mathcal{G}_1$ and $\mathcal{G}_2$ (see Section 2.3.2 for details), taken from the FERET database [93,94]. In the database $\mathcal{G}_1$, each subject has at least ten images so that we can generalize a set of learning tasks with wide LDD ($\rho_t(L)$) values, ranging from $\frac{3}{C}$ to $\frac{7}{C}$, to study the corresponding performance changes of the boosting algorithms. The other database, $\mathcal{G}_2 \supset \mathcal{G}_1$ consisting of more subjects but with less images per subject, is utilized to test the learning capacity of the algorithms as the size of the evaluation database becomes larger. The details of the images included in $\mathcal{G}_1$ and $\mathcal{G}_2$ are depicted in Table 2.1. For computational purposes, each image is again represented as a column vector of length $J = 17154$ prior to the recognition stage.

Following the FR evaluation design in Section 2.3.3, the database $\mathcal{G}(= \mathcal{G}_1$ or $\mathcal{G}_2)$ is randomly partitioned into two subsets: the training set $\mathcal{Z}$ and test set $\mathcal{Q}$. The training set is composed of $|\mathcal{Z}| = L \cdot C$ images: L images per subject are randomly chosen, where $|\mathcal{Z}|$ denotes the size of $\mathcal{Z}$. The remaining images are used to form the test set $\mathcal{Q} = \mathcal{G} - \mathcal{Z}$. Any FR method evaluated here is first trained with $\mathcal{Z}$, and the resulting face recognizer is then applied to $\mathcal{Q}$ to obtain a CER. To enhance the accuracy of the assessment, all the CERs reported later are averaged over five runs. Each run is executed on such a random partition of the database $\mathcal{G}$ into $\mathcal{Z}$ and $\mathcal{Q}$.

7.5.2 The Boosting Performance in Terms of CER

In general, both the LDD ρ and the size of the evaluation dataset $|\mathcal{G}|$ significantly influence the CER characteristics of any learning-based FR method. In order to study the overall boosting performance, two experiments corresponding to the two issues respectively are designed and reported here. Besides the two proposed boosting schemes, B-JD-LDA.A and B-JD-LDA.$\hat{A}$, the stand-alone JD-LDA FR method (without boosting, hereafter S-JD-LDA) was performed to measure the improvement brought by boosting. Meanwhile, two FR algorithms, the Eigenfaces method [120] and the Bayes matching method [86], were also implemented to provide performance baselines. The Bayes method is the top performer in the 1996/1997 FERET competitions [94]. Considering the high computational complexity to optimize the regularization parameter, η, we use again in this work, JD-LDA.1, the cost-effective special case of JD-LDA with η being set to 1, recommended in Section 4.5.2. To be fair, the nearest center rule is used for classification in all of these methods compared here.

The first experiment conducted on $\mathcal{G}_1$ is designed to test the sensitivity of the CER measure to $\rho_t(L)$ (*i.e.* various SSS learning tasks arising from different database partitions) and $\rho_l(r)$ (*i.e.* various weakness extents of gClassifiers in each task). For all the five methods compared here, the CER is a function of the number of extracted feature vectors, M, and the number of available training examples per subject, L. In addition,

Table 7.1: Comparisons of the lowest CERs (%) as a function of $(\rho_t(L), \rho_l(r))$ obtained on the database $\mathcal{G}_1$.

$\rho_t(L) =$	$\rho_l(r) =$	B-JD-LDA.A	B-JD-LDA.$\hat{A}$	S-JD-LDA	Eigenfaces	Bayes
L/C	r/C	$\bar{e}_{15}(\bar{T}^*)$	$\bar{e}_{15}(\bar{T}^*)$	$\bar{e}^*(M^*)$	$\bar{e}^*(M^*)$	$\bar{e}^*(M^*)$
3/49	2/49	15.73(35)	17.47(22)	20.09(39)	32.33(120)	21.61(88)
4/49	2/49	9.71(38)	9.80(51)	13.17(35)	29.22(124)	13.02(137)
	3/49	7.95(29)	10.59(15)			
5/49	2/49	8.37(51)	9.31(54)	11.69(35)	30.03(159)	9.97(179)
	3/49	6.54(30)	5.76(29)			
	4/49	7.20(36)	6.87(21)			
6/49	2/49	6.73(42)	7.24(47)	9.04(32)	25.13(140)	6.28(206)
	3/49	4.49(42)	3.97(44)			
	4/49	4.42(32)	3.97(16)			
	5/49	5.19(39)	5.38(10)			
7/49	2/49	6.54(53)	6.92(54)	7.38(28)	24.94(137)	5.02(235)
	3/49	4.26(35)	3.80(42)			
	4/49	3.73(34)	3.19(16)			
	5/49	3.88(41)	3.73(23)			
	6/49	4.87(40)	4.71(11)			

B-JD-LDA's performance is affected by r, the number of examples per subject that is used to control the weakness of the produced gClassifiers during the boosting process. Considering the huge computational cost, we simply fixed the feature number $M = 15$ (the value was chosen based on the trade-off between the computational cost and the CER) for B-JD-LDA rather than seek the optimal M^*, which yields the lowest CER. The maximal iteration number used in boosting was set as $T = 60$, beyond which it

Table 7.2: The best CER Performance improvement achieved by B-JD-LDA in the tasks $\rho_t = 3/49, \cdots, 7/49$.

Method	$\rho_t(L)$	3/49	4/49	5/49	6/49	7/49
B-JD-LDA.A	$\bar{\xi}^*/r^*$	-4.36/2	-5.22/3	-5.15/3	-4.62/4	-3.65/4
B-JD-LDA.$\hat{A}$	$\bar{\xi}^*/r^*$	-2.61/2	-3.37/2	-5.93/3	-5.06/3,4	-4.18/4

was empirically observed that boosting is very likely to overfit. The lowest CERs finally obtained by the five methods under various settings of $\rho_t(L)$ and $\rho_l(r)$ are depicted in Table 7.1, where $\bar{e}_{15}(\bar{T}^*)$ denotes the CER of B-JD-LDA with $M = 15$ and the best found iteration number $\bar{T}^*$, while $\bar{e}^*(\bar{M}^*)$ denotes the CER of the three non-boosting methods with the best found feature number $\bar{M}^*$. All these variables have been averaged over five runs as we mentioned earlier. To further facilitate the comparison of boosting performance, we define a quantitative statistic regarding the best found CER improvement of B-JD-LDA against S-JD-LDA, denoted as

$$\bar{\xi}^*(L) = \bar{e}_{15}^{\{b\}}(r^*, \bar{T}^*, L) - \bar{e}^{*\{s\}}(\bar{M}^*, L) \tag{7.18}$$

where $(\cdot)^{\{b\}}$ and $(\cdot)^{\{s\}}$ mean B-JD-LDA and S-JD-LDA respectively, and r^* is the value: $r^* = \arg\min_r \{\bar{e}_{15}^{\{b\}}(r, L)\}$. The results are summarized in Tables 7.2, from which it can be clearly seen that both of B-JD-LDA.A and B-JD-LDA.$\hat{A}$ with appropriate r values have greatly boosted the performance of S-JD-LDA across various SSS learning scenarios ranging from $\rho_t = 3/49$ to $\rho_t = 7/49$. The biggest improvement, $\bar{\xi}^* = -5.93\%$, is achieved by B-JD-LDA.$\hat{A}$ when $\rho_t = 5/49$ and $r^* = 3$.

The second experiment conducted on $\mathcal{G}_2$ is designed to test the performance changes as size of the evaluation dataset increases. Due to the computational demand, all the five methods are performed only in a representative partition case, *i.e.* $L = 5$, which leads to a SSS learning task with $\rho_t = 5/120$. Correspondingly, the lowest CERs obtained by the five methods are shown in Table 7.3. It can be seen from these results that the

Table 7.3: Comparisons of the lowest CERs (%) as a function of $\rho_l(r)$ obtained on the database $\mathcal{G}_2$.

$\rho_l(r) =$	B-JD-LDA.A	B-JD-LDA.$\hat{A}$	S-JD-LDA	Eigenfaces	Bayes
r/C	$\bar{e}_{15}(\bar{T}^*)$	$\bar{e}_{15}(\bar{T}^*)$	$\bar{e}^*(M^*)$	$\bar{e}^*(M^*)$	$\bar{e}^*(M^*)$
3/120	6.98(48)	9.32(51)	15.14(87)	30.71(258)	9.51(336)
4/120	7.13(52)	6.36(36)			

a bigger boosting performance is reached by the B-JD-LDA approach than in the first experiment. The quantitative statistic $\bar{\xi}^*$ goes up to 8.15% and 8.78% for B-JD-LDA.A and B-JD-LDA.$\hat{A}$ respectively. The reason is not difficult to be seen. As the size of the database increases so that more pattern variations are introduced, the non convex extent of the face distribution grows rapidly [9]. As a result, only a single linear feature representation generalized by S-JD-LDA appears too weak to account for the increased variations. In contrast with the deterioration of S-JD-LDA, the B-JD-LDA approach indicates a stable performance given a similar learning task in the two experiments. This should be attributed to the ensemble-based design of the approach.

In both of the two experiments, Eigenfaces[1] is the worst performer among the five methods. From the results delivered by the most popular benchmark method, we can roughly learn how difficult it is to conduct face recognition on the two evaluation datasets, $\mathcal{G}_1$ and $\mathcal{G}_2$. Also, it is of interest to compare the performance of B-JD-LDA with that of the Bayes method. Researches have shown that the latter generally outperforms, in terms of CER, most subspace-based FR approaches including those using traditional LDA, kernel PCA or ICA techniques by a margin of at least 10 percent [85]. However, it

[1] The performance of Eigenfaces in Table 7.1 differs from that in Table 4.2, due to two reasons: (1) The classification here is based on the nearest center rule instead of the nearest neighbor rule used in the experiment depicted in Table 4.2; (2) It should be noted that each of database partitions is done randomly.

can be seen from Table 7.1, 7.3 that both the two B-JD-LDA methods are overall superior to the Bayes method. Especially, B-JD-LDA.$\hat{A}$ leads the state-of-the-art method up to 5.88% in the task with the worst SSS setting ($\rho_t(L) = 3/49$). This further shows that it is possible to boost a traditional FR algorithm to the state-of-the-art level under the proposed framework. Finally, it should be mentioned again at this point that unlike the four non-boosting methods, we did not seek the CERs with the optimal M^* values for the B-JD-LDA approach. Obviously the $\bar{e}_{15}$, as a substitute for $\bar{e}_{M^*}$, is only sub-optimal. We expect that a higher boosting performance gain can be obtained when a better M value is found.

7.5.3 Weakness Analysis of the gClassifiers

As it was mentioned earlier, B-JD-LDA would fail, in theory, to perform well when too weak or too strong gClassifiers are utilized. Clearly, it can be experimentally observed at this point from the results shown in Table 7.1 and Figs.7.5,7.6, where the lowest CERs always appear along with the gClassifiers having $\rho_l(r)$ values in between $\frac{2}{49}$ and $\frac{L-1}{49}$. Based on the theory developed in Section 7.4.3, the gClassifiers with the best weakness or the optimal $\rho_l(r^*)$ can be found by minimizing a generalization loss function $\mathbf{R}(r)$ (Eq.7.17) with respect to r, *i.e.* $r^* = \arg\min_r \mathbf{R}(r)$. To test the estimation accuracy of the method, we applied the loss function to the various learning tasks designed in the first experiment. The obtained results including $\mathbf{R}(r)$, r^*, and the worst r value (r^-) are depicted in Table 7.4,7.5 for B-JD-LDA.A and B-JD-LDA.$\hat{A}$ respectively, where the values of λ were found empirically. It should be mentioned here that it is not a difficult task to find an appropriate λ value within $[0, 1]$. In fact, our experiments reveal that there exist a range of λ values which produce the same estimation for the preference rankings of the r values, for example, $\lambda \in [0.5, 0.61]$ for B-JD-LDA.A and $\lambda \in [0.01, 0.34]$ for B-JD-LDA.$\hat{A}$ found in the experiment. Comparing the results of the r rankings to those shown in Table 7.1, it is not difficult to see that the values of the losses correctly

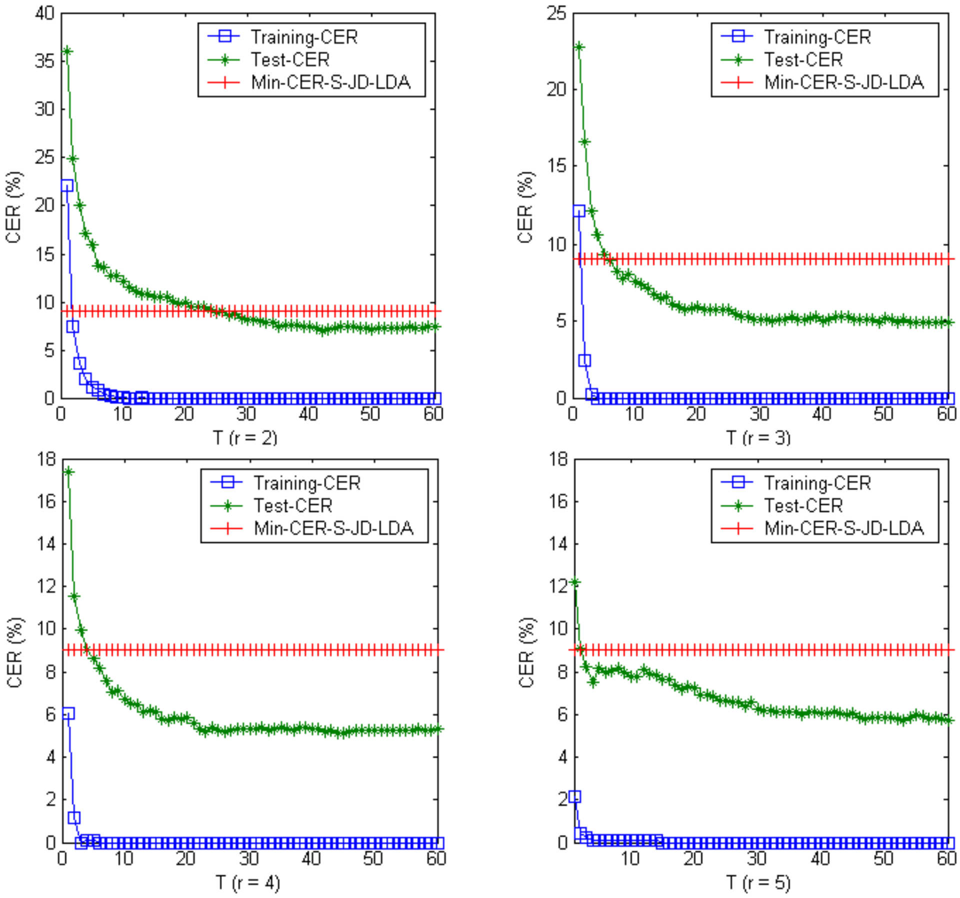

Figure 7.5: Training/test CER comparisons of B-JD-LDA.A with varying weakness extents of gClassifiers as a function of T in the task $\rho_t(L) = 6/49$. Min-CER-S-JD-LDA denotes the CER of S-JD-LDA with $M = M^*$.

indicate the optimal r^*, the worst r^-, and even the r rankings between them such as the 2nd, 3rd and 4th best r values in most cases. The unique mis-estimate occurs in the task $\rho_t(L) = 4/49$ for B-JD-LDA.$\hat{A}$. From Table 7.1, it can be observed that the test CER difference between the two candidate gClassifiers (with $r = 2$ and $r = 3$ respectively) in the mis-estimate case is only $10.59\% - 9.80\% = 0.79\%$, clearly indicating the difficulty of predicting which candidate could be better in advance.

Also, it is of interest to analyze the performance difference between B-JD-LDA.A and B-JD-LDA.$\hat{A}$ with Murua's theory regarding the generalization error [91], that is,

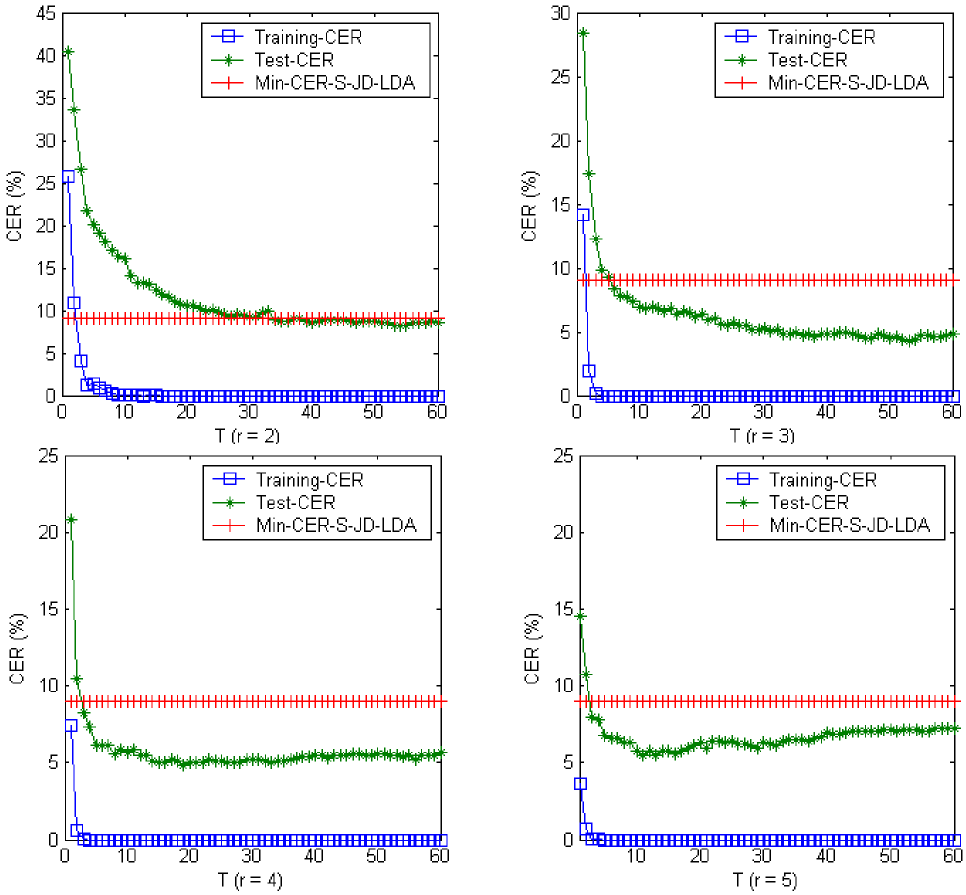

Figure 7.6: Training/test CER comparisons of B-JD-LDA.$\hat{A}$ with varying weakness extents of gClassifiers as a function of T in the task $\rho_t(L) = 6/49$. Min-CER-S-JD-LDA denotes the CER of S-JD-LDA with $M = M^*$.

to achieve a low generalization error, a trade-off between weak dependence and large expected margins has to be maintained for the gClassifiers involved in the combination. As we analyzed in Section 7.4, B-JD-LDA.$\hat{A}$ creates the gClassifiers with lower weakness and lower mutual dependence than B-JD-LDA.A does given the same $\rho_l(r)$ value. As a consequence, it seems reasonable to conjecture that when the individual gClassifier produced in both algorithms has been sufficiently strong, the one with lower mutual dependence may outperform the other. The demonstration at this point can be seen from Table 7.1, where B-JD-LDA.$\hat{A}$ is obviously superior to B-JD-LDA.A when they trained

Table 7.4: The generalization loss $\mathbf{R}(r, L)$ with $\lambda = 0.55$, the best r estimate (r^*) and the worst r estimate (r^-) obtained by B-JD-LDA.A on the database $\mathcal{G}_1$.

$\rho_t(L)$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	Estimation	
4/49	0.5540	0.5068	–	–	–	$r^* = 3$	$r^- = 2$
5/49	0.5563	0.4965	0.5031	–	–	$r^* = 3$	$r^- = 2$
6/49	0.5636	0.4897	0.4770	0.5050	–	$r^* = 4$	$r^- = 2$
7/49	0.5635	0.4860	0.4599	0.4727	0.5106	$r^* = 4$	$r^- = 2$

Table 7.5: The generalization loss $\mathbf{R}(r, L)$ with $\lambda = 0.25$, the best r estimate (r^*) and the worst r estimate (r^-) obtained by B-JD-LDA.$\hat{A}$ on the database $\mathcal{G}_1$.

$\rho_t(L)$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	Estimation	
4/49	0.4173	0.3111	–	–	–	$r^* = 3$	$r^- = 2$
5/49	0.4562	0.3128	0.4062	–	–	$r^* = 3$	$r^- = 2$
6/49	0.4882	0.3512	0.2819	0.4574	–	$r^* = 4$	$r^- = 2$
7/49	0.5047	0.3725	0.2779	0.3194	0.4398	$r^* = 4$	$r^- = 2$

the JD-LDA learner with $\rho_l(r) \geq \bar{\rho} = 3/49$ in most cases, where $\bar{\rho}$ denotes certain weakness threshold. On the other hand, the large margin factor in the balance may play a more important role than the mutual dependence when the individual gClassifier is very weak. It can be also observed at this point from Table 7.1 that B-JD-LDA.A performed much better than B-JD-LDA.$\hat{A}$ in those cases of $\rho_l(r) = 2/49$. Therefore, the preference for B-JD-LDA.A or B-JD-LDA.$\hat{A}$ should be dependent on the $\rho_l(r)$ value. Similar observations can be also found in the experiments on $\mathcal{G}_2$. The only difference is that the weakness threshold is changed to $\bar{\rho} = 4/120$ due to the increased class number $C = 120$.

7.5.4 Convergence and Cumulative Margin Distribution

It can be observed from Figs.7.5,7.6 that during the iteration, B-JD-LDA continued to improve the test CERs ($h_f(\mathcal{Q})$) when appropriately weak gClassifiers (*e.g.* $\rho_l(r) = 3/49$ for B-JD-LDA.$\hat{A}$ in Fig.7.6) were produced, even long after the training CERs ($h_f(\mathcal{Z})$) had dropped down to zero, clearly showing the beautiful property of boosting as a large margin classifier against overfitting. However, similar to $\rho_l(r)$, boosting may fail to perform well given both too few (underfit) or too many (overfit) iterations T. It can be also seen at this point from Figs.7.5,7.6 that there obviously exists an optimal T^* in between. As discussed in Section 7.4.4, a simple method to roughly estimate T^* is to observe the changes of the *cumulative margin distribution* (CMD) of training examples. Taken as a representative example, the CMDs of S-JD-LDA, B-JD-LDA.A and B-JD-LDA.$\hat{A}$ obtained in the task $\rho_t(L) = 5/49$ of the first experiment are shown in Fig.7.7, where it can be seen that there were almost no improvements in terms of the CMDs after the iteration $T = 40$ for B-JD-LDA.A and $T = 20$ for B-JD-LDA.$\hat{A}$ in the cases of $r = 3, 4$. Meanwhile, small improvement is still observable for both the two methods after $T = 40$ in the case of $r = 2$. These observations are roughly in agreement with those results of T^* as depicted in Table 7.1. However, it should be noted again at this point that, due to the incompleteness of the boosting margin theory, the estimation method is quite coarse, for example, S-JD-LDA with the best found M^* yielded much better CMDs than both the two boosting algorithms in all cases shown in Fig.7.7.

Another observation from Table 7.1 and Figs.7.5,7.6 is that B-JD-LDA.$\hat{A}$ generally needs fewer iterations to reach the best performance than B-JD-LDA.A except for the case when the individual gClassifier is too weak, *e.g.* $\rho_l(r) = 2/49$. This can be explained by B-JD-LDA.$\hat{A}$'s property of lower mutual dependence, which helps to produce a more effective combination of gClassifiers with each one having less overlapping performance, compared to B-JD-LDA.A. Also, it should be noted at this point that only around $T^* \leq 45$ iterations are required to find an excellent result using the B-JD-LDA algorithm for

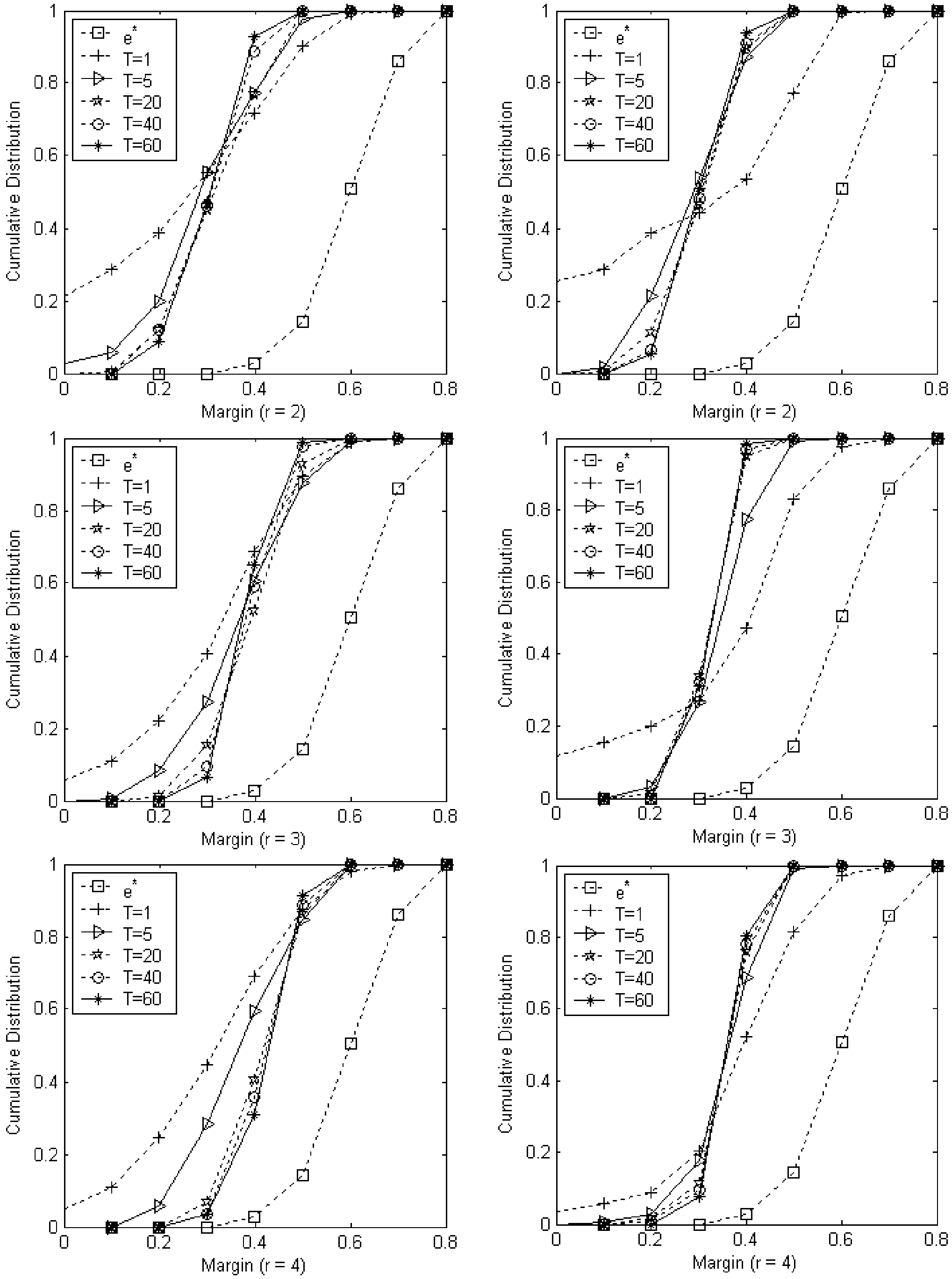

Figure 7.7: Cumulative margin distributions (CMDs) obtained in the task $\rho_t(L) = 5/49$ by **Left:** B-JD-LDA.A, and **Right:** B-JD-LDA.$\hat{A}$. (T1, T5, T20, T40, T60) are the CMDs after 1, 5, 20, 40, 60 iterations respectively. e^* is the result obtained by S-JD-LDA with $M = M^*$.

most cases shown in Table 7.1. Considering that each gClassifier works in a considerably lower-dimensional subspace ($M = 15$) compared to the four non-boosting methods, such a computational cost is affordable for most existing personal computers.

7.5.5 A Comparison of Six FR Methods Developed in the Thesis in terms of CER Performance

So far, we have developed six appearance-based discriminant learning algorithms for face recognition in this thesis, and they are JD-LDA, DF-LDA, RD-QDA, KDDA, HCF-MSC and B-JD-LDA. Each of them has its own theoretical properties, and advantages on some specific applications. Consequently, we believe that it is of interest and desire for readers to give a brief evaluation and benchmarking of these algorithms in FR tasks before we finish the presentations of the thesis.

Table 7.6: A comparison of eight FR methods in terms of CER (%) performance obtained with Rank=1 on the database $\mathcal{G}_2$.

Algs.	Min. CER	$\bar{M}$	Values of Parameters Involved
Eigenfaces	30.71	258	–
Bayes	9.51	336	–
JD-LDA	15.14	87	$\eta = 1$
DF-LDA	12.36	49	$\eta = 1$, $w(d) = d^{-8}$, $r = 10$
RD-QDA	12.03	119	$\lambda = 0.25$, $\gamma = 1\mathbf{e} - 4$
KDDA	12.58	103	Kernel=RBF, $\sigma^2 = 3\mathbf{e}5$
HCF-MSC	10.97	82	$K = 3$
B-JD-LDA	6.36	15	with $\hat{A}_t(p, q)$, $\rho_l(r) = 4/120$, $T = 36$

To this end, we built six FR systems corresponding to the six algorithms. Each

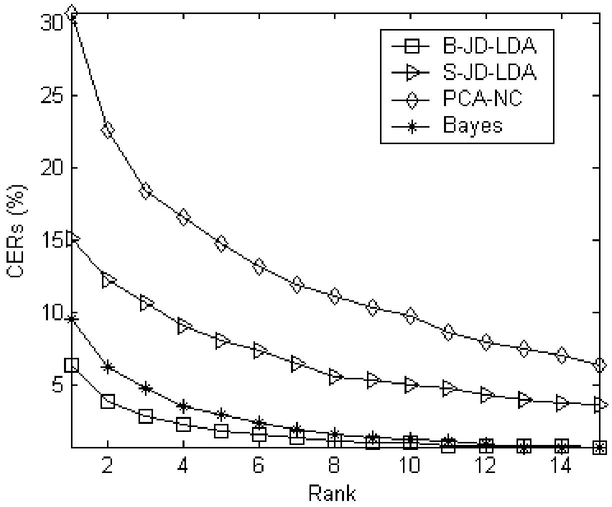

Figure 7.8: CER performance comparison of four FR methods, B-JD-LDA, S-JD-LDA, PCA-NC(Eigenfaces) and Bayes matching.

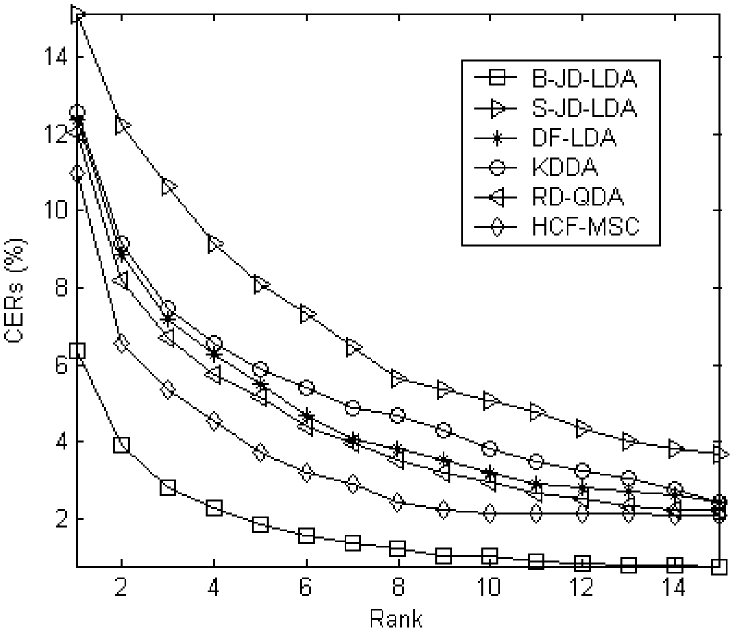

Figure 7.9: CER performance comparison of six FR methods developed in the thesis, B-JD-LDA, S-JD-LDA, DF-LDA, RD-QDA, KDDA and HCF-MSC.

system was constructed as did in previous simulations by one algorithm followed by a simple classifier, the NCC, so as to emphasize the contribution of the feature extractor in the FR performance. Also, two systems based on the Eigenfaces method and the Bayes matching method, were implemented again to provide performance baselines. All of the eight FR systems were applied to the evaluation database $\mathcal{G}_2$ with the same settings (partitions and runs) as the experiment reported in Table 7.3. The obtained results are summarized in Table 7.6, where the performance of Eigenfaces, Bayes matching, JD-LDA, and B-JD-LDA are the same as those reported in Table 7.3 due to the same experimental settings. JD-LDA is the base, from which other five discriminant learning methods are developed. To facilitate a comparison, we set $\eta = 1$ in all JD-LDA related methods, such as DF-LDA, HCF-MSC and B-JD-LDA *etc.* , although the optimization with respect to $\eta = 1$ could result in a better performance for each of them. A common parameter among all the methods evaluated here is the number of feature vectors used, M. The best found values of M, averaged over 5 runs, are shown as $\bar{M}$ in Table 7.6. In addition, Figs.7.8-7.9 depict the CER performance as a function of the rank number of top matching (see similar presentations in Section 6.6.3). It can be seen from these results that B-JD-LDA is the top performer, although it is the only method without being optimized with respect to M ($M = 15$ is set in each gClassifier as did in previous experiments). All the five methods developed from JD-LDA have shown certain performance improvement against JD-LDA, although these improvement may not be as big as those reported previously on other evaluation databases. The most impressive thing that can be observed here is that all the six discriminant methods have significantly outperformed the Eigenfaces method by a margin up to $\geq 15.57\%$. This demonstrates again the effectiveness of these methods proposed here.

In addition, it may be of interest to readers to observe those test examples wrongly classified in this experiment. For this purpose, Figs.7.8 depicts two test examples misclassified in terms of the top rank matching by four representative methods compared here, Eigenfaces, S-JD-LDA and B-JD-LDA. In the example **A**, the smiled query looks very similar to the top 1 rank returned by Eigenfaces, KDDA and B-JD-LDA, and actually

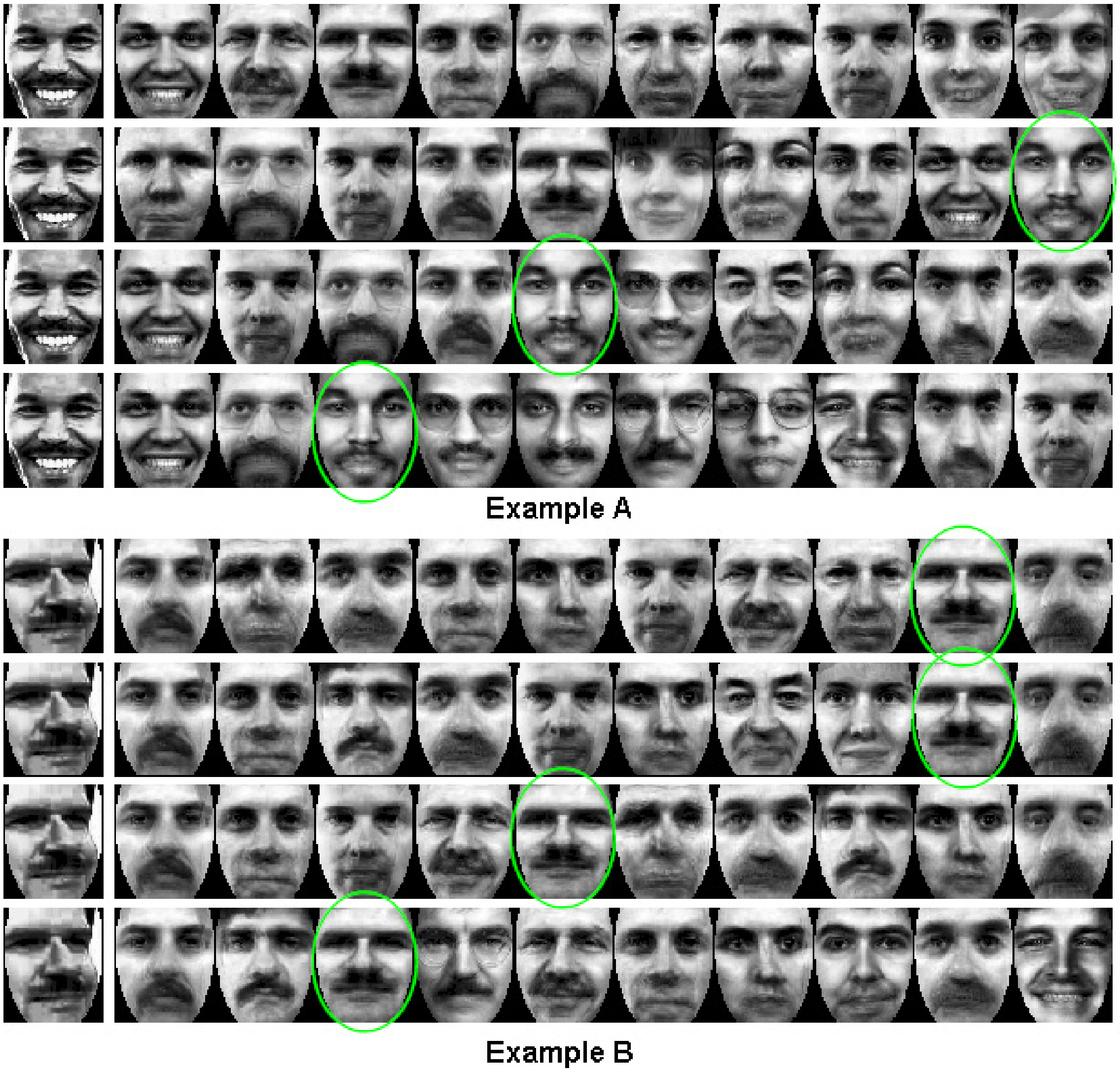

Figure 7.10: Two examples misclassified by Eigenfaces, JD-LDA, KDDA and B-JD-LDA. In each example, the most left image is the query, and the right images are its top ten matches ranked by matching scores obtained by Eigenfaces (1st row), JD-LDA(2nd row), KDDA(3rd row) and B-JD-LDA (4th row) with the parameter settings described in Table 7.6, respectively. Only circled images share the same label with the queries.

it is a difficult task even for a person to identify their difference. However, compared to the failure of Eigenfaces in this case, JD-LDA, KDDA and B-JD-LDA still returned the correct answering within the top 10, 5 and 3 ranks respectively. Similar observations can be found in the example **B**, where the difficulty is due to the big difference in pose angle between the query and its corresponding target. To improve the performance in this case, it may have to rely on the introduction of more sophisticated preprocessing technologies such as 3-D warping or 3-D face models.

Before we proceed to the conclusion, one point is worth mentioning here. Although the six discriminant learning algorithms proposed in the thesis gave different CERs on the evaluation database $\mathcal{G}_2$, it should be noted at this point that the performance of a learning-based pattern recognition system is very data/application-dependent, and there is no theory that is able to accurately predict them for unknown-distribution data/new applications. In other words, some methods that have reported almost perfect performance in certain scenarios may fail in other scenarios.

7.6 Summary

The contribution of the works presented in this chapter is twofold. (1) A novel weakness analysis theory has been developed to overcome the limitation of the weak learners in traditional boosting techniques. The theory proposed here is composed of a cross-validation mechanism of weakening a strong learner and a subsequent estimation method of appropriate weakness for the classifiers created by the learner. With the introduction of the weakness analysis theory, a traditional boosting algorithm can be used to work effectively with a general (strong or weak) learner. (2) The new boosting framework is applied to boost a strong and stable learner, JD-LDA. This leads to a novel ensemble-based discriminant learning approach, B-JD-LDA. In this approach, a novel variable accounting for the pairwise class discriminant information is also introduced to build an effective connection between the booster and the LDA-based learner. As a result, by manipulating the B-JD-LDA process, a set of specific LDA feature spaces can be constructed effectively in a manner of similar to "automatic gain control". Unlike most traditional mixture models of linear subspaces that are based on cluster analysis, these LDA subspaces are generalized in the context of classification error minimization.

The effectiveness of the proposed B-JD-LDA approach including boosting power, estimation accuracy of the loss function, and robustness against the overfitting and SSS problems has been demonstrated through the FR experimentation performed on the FERET database. It is further anticipated that in addition to JD-LDA, other existing traditional face recognizers such as those based on PCA or ICA techniques may be boosted to the state-of-the-art level through integration into the proposed boosting framework. On the other hand, the booster, AdaBoost, in the framework could be replaced by its superior variants such as the so-called Soft Margins AdaBoost [97], which has recently been shown to outperform the original AdaBoost in head pose classification [40].

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The focus of this research was on the development of discriminant learning methods capable of providing solutions to small-size-sample (SSS), high-dimensional face recognition problems. In this work, a simple but effective, linear discriminant analysis algorithm, called JD-LDA was first introduced. The algorithm is based on a regularized Fisher's criterion, and has been shown to be effective and stable in capturing the optimal linear discriminant features across various SSS scenarios. Based on the JD-LDA solution, a series of discriminant analysis algorithms were developed through the integration of advanced learning theories, such as Bayes discriminant analysis, kernel machines, mixture of multi-models, and AdaBoost. Each of the solutions proposed here has its own theoretical properties, and advantages on some specific applications. The DF-LDA method can be considered as a generalization of a number of linear techniques which are commonly in use. It can be chosen for use when the variations of patterns under learning are not too complex and a cost-effective solution is sought. RD-QDA takes advantages of the JD-LDA and regularized Bayes discriminant analysis techniques, and it is capable of dealing with patterns subject to general Gaussian distributions with an affordable computational cost. In addition, KDDA, HCF-MSC and B-JD-LDA are three methods developed to address recognition problems when the pattern distribution is far more complicated than Gaussian. KDDA is a globally nonlinear solution. In the method, the kernel function is utilized to map the original face patterns to a high-dimensional feature space, where the highly non-convex and complex distribution of face patterns is linearized and simplified, so that the JD-LDA can be applied for feature extraction. Due to extremely high dimensionality of the feature space, KDDA is more susceptible to the overfitting problem compared to other two solutions: HCF-MSC and B-JD-LDA, which are based on a mixture of linear models. Therefore, KDDA is recommended for use only if sufficient training samples are available. Otherwise both HCF-MSC and B-JD-LDA could be better choices especially when very small size, for example only $L \leq 5$ training samples per subject are available. HCF-MSC is a hierarchical classification scheme which operates on top of traditional FR systems trained on database partitions obtained using a novel separability-based clustering method. HCF-MSC is able to address large-scale FR problems with low computational complexity. During the course of this work, another ensemble-based algorithm, B-JD-LDA has been shown to be superior to HCF-MSC whether from a theoretical or experimental analysis point of view. A boosting technique is introduced in B-JD-LDA. As a result, the LDA sub-models can be generalized and mixed in the context of direct minimization of the generalization error instead of only the empirical error. A great deal of FR simulations have been presented to demonstrate the effectiveness of these proposed methods in various SSS situations.

It should be worthy to mention again that in this work, the proposed algorithms were designed, analyzed and tested specifically by using the face recognition paradigm. However, the recognition methods researched here may be proven useful in other applications where classification tasks are routinely performed, such as content-based image/video indexing and retrieval, face detection (generally considered a binary classification problem), and video/audio classification.

8.2 Directions of Future Research

In this section, a set of topics are presented to extend the discriminant learning algorithms developed in the thesis.

1. **Automatic parameter selection**. Automatic parameter optimization remains an open research problem in the pattern recognition and machine learning communities [134]. The difficulty arises from the following unknown facts regarding the patterns under learning:

   (i) What is the actual distribution of the patterns?

   (ii) How have the training data sampled the underlying distribution of the patterns?

   (iii) How many training samples are sufficient to capture all the variations of the patterns?

   Due to a lack of prior knowledge and the application-specific nature of the process, there is no systematic criteria capable of accurately predicting the generalization performance of chosen parameter values before a test set is provided and used for evaluation.

   Thus, in the experiments performed in this work, the design parameters in all the algorithms have been selected using heuristic and/or exhaustive searches. Although some routine techniques, such as leave-one-out, can be used to find good (yet suboptimal) parameter values in a systematic way, they often lead to significant increase in computational cost as it can be seen from the analysis in Chapter 4. Therefore, rather than develop some generic but complex parameter optimization methods, it may be of interest to determine a simple and application-dependent relationship between the parameters to be optimized, the CER, and prior information regarding the training data (such as the LDD) using a combination of analysis and heuristic knowledge. The introduction of the generalization loss (Eq.7.17), which is designed to find the best number (r) of training samples per subject for building a gClassifier in Chapter 7, can be seen as a step towards this direction. Similar developments are envisioned for the determination of the parameters involved in the JD-LDA, DF-LDA, KDDA and HCF-MSC methods introduced here.

2. **A mixture of multiple discriminant learning algorithms with boosting**. It can be seen from the previous analysis that each kind of facial feature representations discussed here, such as PCA, LDA, KPCA, KDDA and even ICA and KICA, has its own theoretical properties, and advantages on some specific problems. It is therefore reasonable to inquire if, "Is it possible to merge all these useful features into one FR system?". The boosting framework constructed for B-JD-LDA seems to provide a promising way to answer the above question. In the current framework, the learner always remains unchanged, although each gClassifier produced by the learner could be different. Alternatively, we can design a scheme to adaptively update the learner at each iteration either. For example, let

$$\mathcal{F} = \{\text{PCA, JD-LDA, KPCA, KDDA, ICA, KICA}\}$$

be a set of candidate learners. As we know from Chapter 7, boosting generalizes a different discriminant sub-problem at every iteration. Thus, we can choose a learner from $\mathcal{F}$ to best conquer the specific sub-problem produced in the current iteration. The best learner can be found, for example, by comparing the pseudo-loss $\hat{\epsilon}_t$ of the gClassifiers produced by these learners in $\mathcal{F}$. In this way, the final solution of boosting is a mixture of the gClassifiers generalized by various learners included in $\mathcal{F}$. On the other hand, the complexity of the entire boosting algorithm is inevitably increased due to the introduction of multiple learners. Therefore, it could be the focus of the research to solve the problems arising from the increased algorithmic complexity.

3. **A mixture of shape- and appearance- based models**. In the appearance-based methods, a face image is generally processed as a 2D holistic pattern represented by an array of its pixel values (*i.e.* image intensities), and the feature extraction is conducted directly on the pixel array. Obviously, in such an approach any structural information, such as the shapes of eyes, nose and mouth, and their geometric configuration, is ignored. As a result, these kind of methods are sensible to variations in illuminations and head poses. In contrast, shape-based methods (see Section 2.1) focus on the construction of a 3D facial shape model, completely removed from the pixel values by locating and analyzing the structure information. Thus, it should be advantageous to develop systems that are able to effectively merge the best aspects of both approaches. Initial efforts such as the works of Craw *et al.* [22, 23], Von der Malsburg *et al.* [57, 127, 128], Cootes *et al.* [59] and Blanz *et al.* [10] suggest that improvement in performance should be expected. Particularly, the method of [127] was one of the top performers among the partially automatic FR algorithms included in the 1996/1997 FERET competitions [94]. Therefore, it is only reasonable to assume that the incorporation of more powerful appearance-based methods such as those introduced in this work, will boost further the performance of such hybrid solutions.

Appendix A

Some Derivations in Chapter 5

A.1 Computation of $\tilde{\Phi}_b^T \tilde{\Phi}_b$

Expanding $\tilde{\Phi}_b^T \tilde{\Phi}_b$, we have

$$\tilde{\Phi}_b^T \tilde{\Phi}_b = \left[\tilde{\bar{\phi}}_1 \ \cdots \ \tilde{\bar{\phi}}_C\right]^T \left[\tilde{\bar{\phi}}_1 \ \cdots \ \tilde{\bar{\phi}}_C\right] = \left(\tilde{\bar{\phi}}_i^{\ T} \tilde{\bar{\phi}}_j\right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} \tag{A.1}$$

where

$$\begin{aligned} \tilde{\bar{\phi}}_i^{\ T} \tilde{\bar{\phi}}_j &= \left(\sqrt{\tfrac{C_i}{N}}\left(\bar{\phi}_i - \bar{\phi}\right)\right)^T \left(\sqrt{\tfrac{C_j}{N}}\left(\bar{\phi}_j - \bar{\phi}\right)\right) \\ &= \tfrac{\sqrt{C_i C_j}}{N}\left(\bar{\phi}_i^T \bar{\phi}_j - \bar{\phi}_i^T \bar{\phi} - \bar{\phi}^T \bar{\phi}_j + \bar{\phi}^T \bar{\phi}\right) \end{aligned} \tag{A.2}$$

We develop each term of Eq.A.2 according to the kernel matrix $\mathbf{K}$ as follows,

- $$\begin{aligned} \bar{\phi}^T \bar{\phi} &= \left(\tfrac{1}{N}\sum_{l=1}^{C}\sum_{k=1}^{C_l} \phi_{lk}\right)^T \left(\tfrac{1}{N}\sum_{h=1}^{C}\sum_{m=1}^{C_h} \phi_{hm}\right) = \tfrac{1}{N^2}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{h=1}^{C}\sum_{m=1}^{C_h} \phi_{lk}^T \phi_{hm} \\ &= \tfrac{1}{N^2}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{h=1}^{C}\sum_{m=1}^{C_h} \left(k_{km}\right)_{lh} \end{aligned}$$

  $$\Rightarrow \left(\bar{\phi}^T \bar{\phi}\right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \tfrac{1}{N^2}\left(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{1}_{NC}\right);$$

- $$\begin{aligned} \bar{\phi}^T \bar{\phi}_j &= \left(\tfrac{1}{N}\sum_{l=1}^{C}\sum_{k=1}^{C_l} \phi_{lk}\right)^T \left(\tfrac{1}{C_j}\sum_{m=1}^{C_j} \phi_{jm}\right) = \tfrac{1}{NC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{m=1}^{C_j} \phi_{lk}^T \phi_{jm} \\ &= \tfrac{1}{NC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{m=1}^{C_j} \left(k_{km}\right)_{lj} \end{aligned}$$

  $$\Rightarrow \left(\bar{\phi}^T \bar{\phi}_j\right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \tfrac{1}{N}\left(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{A}_{NC}\right);$$

- $$\begin{aligned}\bar{\phi}_i^T\bar{\phi} &= \left(\tfrac{1}{C_i}\sum_{m=1}^{C_i}\phi_{im}\right)^T\left(\tfrac{1}{N}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\phi_{lk}\right) = \tfrac{1}{NC_i}\sum_{m=1}^{C_i}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\phi_{im}^T\phi_{lk}\\ &= \tfrac{1}{NC_i}\sum_{m=1}^{C_i}\sum_{l=1}^{C}\sum_{k=1}^{C_l}(k_{mk})_{il}\end{aligned}$$

$$\Rightarrow \left(\bar{\phi}_i^T\bar{\phi}\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \tfrac{1}{N}\left(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{1}_{NC}\right);$$

- $$\begin{aligned}\bar{\phi}_i^T\bar{\phi}_j &= \left(\tfrac{1}{C_i}\sum_{m=1}^{C_i}\phi_{im}\right)^T\left(\tfrac{1}{C_j}\sum_{n=1}^{C_j}\phi_{jn}\right) = \tfrac{1}{C_iC_j}\sum_{m=1}^{C_i}\sum_{n=1}^{C_j}\phi_{im}^T\phi_{jn}\\ &= \tfrac{1}{C_iC_j}\sum_{m=1}^{C_i}\sum_{n=1}^{C_j}(k_{mn})_{ij}\end{aligned}$$

$$\Rightarrow \left(\bar{\phi}_i^T\bar{\phi}_j\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \left(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{A}_{NC}\right).$$

Applying the above derivations into Eq.A.2, we obtain the Eq.5.10.

A.2 Computation of $\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b$

Expanding $\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b$, we have

$$\tilde{\Phi}_b^T\tilde{\mathbf{S}}_w\tilde{\Phi}_b = \left[\tilde{\bar{\phi}}_1\ \cdots\ \tilde{\bar{\phi}}_C\right]^T\tilde{\mathbf{S}}_w\left[\tilde{\bar{\phi}}_1\ \cdots\ \tilde{\bar{\phi}}_C\right] = \left(\tilde{\bar{\phi}}_i^T\tilde{\mathbf{S}}_w\tilde{\bar{\phi}}_j\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} \tag{A.3}$$

where

$$\begin{aligned}\tilde{\bar{\phi}}_i^T\tilde{\mathbf{S}}_w\tilde{\bar{\phi}}_j &= \tfrac{1}{N}\tilde{\bar{\phi}}_i^T\left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}(\phi_{lk}-\bar{\phi}_l)(\phi_{lk}-\bar{\phi}_l)^T\right)\tilde{\bar{\phi}}_j\\ &= \tfrac{1}{N}\tilde{\bar{\phi}}_i^T\left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}(\phi_{lk}\phi_{lk}^T-\bar{\phi}_l\phi_{lk}^T-\phi_{lk}\bar{\phi}_l^T+\bar{\phi}_l\bar{\phi}_l^T)\right)\tilde{\bar{\phi}}_j\\ &= \tfrac{1}{N}\tilde{\bar{\phi}}_i^T\left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\phi_{lk}\phi_{lk}^T-\sum_{l=1}^{C}\bar{\phi}_l\left(\sum_{k=1}^{C_l}\phi_{lk}^T\right)-\sum_{l=1}^{C}\left(\sum_{k=1}^{C_l}\phi_{lk}\right)\bar{\phi}_l^T+\sum_{l=1}^{C}C_l\bar{\phi}_l\bar{\phi}_l^T\right)\tilde{\bar{\phi}}_j\\ &= \tfrac{1}{N}\tilde{\bar{\phi}}_i^T\left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\phi_{lk}\phi_{lk}^T-\sum_{l=1}^{C}C_l\bar{\phi}_l\bar{\phi}_l^T-\sum_{l=1}^{C}C_l\bar{\phi}_l\bar{\phi}_l^T+\sum_{l=1}^{C}C_l\bar{\phi}_l\bar{\phi}_l^T\right)\tilde{\bar{\phi}}_j\\ &= \tfrac{1}{N}\left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\tilde{\bar{\phi}}_i^T\phi_{lk}\phi_{lk}^T\tilde{\bar{\phi}}_j-\sum_{l=1}^{C}C_l\tilde{\bar{\phi}}_i^T\bar{\phi}_l\bar{\phi}_l^T\tilde{\bar{\phi}}_j\right)\end{aligned} \tag{A.4}$$

Firstly, expand the term $\sum_{l=1}^{C}\sum_{k=1}^{C_l}\tilde{\bar{\phi}}_i^T\phi_{lk}\phi_{lk}^T\tilde{\bar{\phi}}_j$ in Eq.A.4, and have

$$\sum_{l=1}^{C}\sum_{k=1}^{C_l}\tilde{\bar{\phi}}_i^T\phi_{lk}\phi_{lk}^T\tilde{\bar{\phi}}_j = \frac{\sqrt{C_iC_j}}{N}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\left(\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j-\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi}-\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j+\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi}\right) \tag{A.5}$$

We then develop each term of Eq.A.5 according to the kernel matrix $\mathbf{K}$ as follows,

- $$\begin{aligned}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j &= \tfrac{1}{C_iC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\left(\sum_{m=1}^{C_i}\phi_{im}^T\phi_{lk}\right)\left(\sum_{n=1}^{C_j}\phi_{lk}^T\phi_{jn}\right)\\ &= \tfrac{1}{C_iC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{m=1}^{C_i}\sum_{n=1}^{C_j}(k_{mk})_{il}(k_{kn})_{lj}\end{aligned}$$

$$\Rightarrow \left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \left(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{A}_{NC}\right);$$

- $$\begin{aligned}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi} &= \tfrac{1}{NC_i}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\left(\sum_{n=1}^{C_i}\phi_{in}^T\phi_{lk}\right)\left(\sum_{h=1}^{C}\sum_{m=1}^{C_h}\phi_{lk}^T\phi_{hm}\right)\\ &= \tfrac{1}{NC_i}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{n=1}^{C_i}\sum_{h=1}^{C}\sum_{m=1}^{C_h}(k_{nk})_{il}(k_{km})_{lh}\end{aligned}$$

$$\Rightarrow \left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}_i^T\phi_{lk}\phi_{lk}^T\bar{\phi}\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \tfrac{1}{N}\left(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{1}_{NC}\right);$$

- $$\begin{aligned}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j &= \tfrac{1}{NC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\left(\sum_{h=1}^{C}\sum_{m=1}^{C_h}\phi_{hm}^T\phi_{lk}\right)\left(\sum_{n=1}^{C_j}\phi_{lk}^T\phi_{jn}\right)\\ &= \tfrac{1}{NC_j}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{h=1}^{C}\sum_{m=1}^{C_h}\sum_{n=1}^{C_j}(k_{mk})_{hl}(k_{kn})_{lj}\end{aligned}$$

$$\Rightarrow \left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi}_j\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \tfrac{1}{N}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{A}_{NC});$$

- $$\begin{aligned}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi} &= \tfrac{1}{N^2}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\left(\sum_{h=1}^{C}\sum_{m=1}^{C_h}\phi_{hm}^T\phi_{lk}\right)\left(\sum_{p=1}^{C}\sum_{q=1}^{C_p}\phi_{lk}^T\phi_{pq}\right)\\ &= \tfrac{1}{N^2}\sum_{l=1}^{C}\sum_{k=1}^{C_l}\sum_{h=1}^{C}\sum_{m=1}^{C_h}\sum_{p=1}^{C}\sum_{q=1}^{C_p}(k_{mk})_{hl}(k_{kq})_{lp}\end{aligned}$$

$$\Rightarrow \left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\bar{\phi}^T\phi_{lk}\phi_{lk}^T\bar{\phi}\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}} = \tfrac{1}{N^2}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{1}_{NC}).$$

Defining $\mathbf{J}1 = \left(\sum_{l=1}^{C}\sum_{k=1}^{C_l}\tilde{\bar{\phi}}_i^{\,T}\phi_{lk}\phi_{lk}^T\tilde{\bar{\phi}}_j\right)_{\substack{i=1,\cdots,C\\ j=1,\cdots,C}}$, we conclude:

$$\begin{aligned}\mathbf{J}1 = \quad & \tfrac{1}{N}\mathbf{B}\cdot(\mathbf{A}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{A}_{NC} - \tfrac{1}{N}(A_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{1}_{NC}) - \\ & \tfrac{1}{N}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{A}_{NC}) + \tfrac{1}{N^2}(\mathbf{1}_{NC}^T\cdot\mathbf{K}\cdot\mathbf{K}\cdot\mathbf{1}_{NC}))\cdot\mathbf{B}\end{aligned} \tag{A.6}$$

Expanding the term $\sum_{l=1}^{C} C_l \tilde{\bar{\phi}}_i^{\,T} \bar{\phi}_l \bar{\phi}_l^T \tilde{\bar{\phi}}_j$ in Eq.A.4, we obtain:

$$\sum_{l=1}^{C} \sum_{k=1}^{C_l} \tilde{\bar{\phi}}_i^{\,T} \bar{\phi}_l \bar{\phi}_l^T \tilde{\bar{\phi}}_j = \frac{\sqrt{C_i C_j}}{N} \sum_{l=1}^{C} C_l \left(\bar{\phi}_i^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi}_j - \bar{\phi}_i^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi} - \bar{\phi}^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi}_j + \bar{\phi}^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi} \right) \tag{A.7}$$

Using the kernel matrix $\mathbf{K}$, the terms in Eq.A.7 can be developed as follows,

- $\left(\sum\limits_{l=1}^{C} C_l \bar{\phi}_i^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi}_j \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{A}_{NC},$
- $\left(\sum\limits_{l=1}^{C} C_l \bar{\phi}_i^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi} \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \frac{1}{N} \left(\mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{1}_{NC} \right),$
- $\left(\sum\limits_{l=1}^{C} C_l \bar{\phi}^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi}_j \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \frac{1}{N} \left(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{A}_{NC} \right),$
- $\left(\sum\limits_{l=1}^{C} C_l \bar{\phi}^T \bar{\phi}_l \bar{\phi}_l^T \bar{\phi} \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \frac{1}{N^2} \left(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{1}_{NC} \right),$

where $\mathbf{W} = \mathbf{diag}\left[\mathbf{w}_1 \ \cdots \ \mathbf{w}_c\right]$ is a $N \times N$ block diagonal matrix, and $\mathbf{w}_i$ is a $C_i \times C_i$ matrix with terms all equal to: $\frac{1}{C_i}$.

Defining $\mathbf{J}2 = \left(\sum\limits_{l=1}^{C} \sum\limits_{k=1}^{C_l} \tilde{\bar{\phi}}_i^{\,T} \bar{\phi}_l \bar{\phi}_l^T \tilde{\bar{\phi}}_j \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}}$, and using the above derivations, we conclude that:

$$\begin{aligned} \mathbf{J}2 = \quad & \tfrac{1}{N}\mathbf{B} \cdot (\mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{A}_{NC} - \tfrac{1}{N}(\mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{1}_{NC}) - \\ & \tfrac{1}{N}(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{A}_{NC}) + \tfrac{1}{N^2}(\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{K} \cdot \mathbf{1}_{NC})) \cdot \mathbf{B} \end{aligned} \tag{A.8}$$

Thus,

$$\tilde{\Phi}_b^T \tilde{\mathbf{S}}_w \tilde{\Phi}_b = \left(\tilde{\bar{\phi}}_i^{\,T} \tilde{\mathbf{S}}_w \tilde{\bar{\phi}}_j \right)_{\substack{i=1,\cdots,C \\ j=1,\cdots,C}} = \frac{1}{N}(\mathbf{J}1 - \mathbf{J}2) \tag{A.9}$$

Bibliography

[1] B. Achermann and H. Bunke. “Combination of face classifiers for person identification”. In *Proceedings of International Conference Pattern Recognition*, pages 416–420, Vienna, Austria, August 1996.

[2] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoér. “Theoretical foundations of the potential function method in pattern recognition learning”. *Automation and Remote Control*, 25:821–837, 1964.

[3] Schölkopf B., Smola A., and Müller K. R. “Nonlinear component analysis as a kernel eigenvalue problem”. *Neural Computation*, 10:1299–1319, 1999.

[4] Francis R. Bach and Michael I. Jordan. “Kernel independent component analysis”. *Computer Science Division, University of California Berkeley, Available as Technical Report No. UCB/CSD-01-1166*, November 2001.

[5] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. “Face recognition by independent component analysis”. *IEEE Transactions on Neural Networks*, 13(6):1450–1464, Nov. 2002.

[6] G. Baudat and F. Anouar. “Generalized discriminant analysis using a kernel approach”. *Neural Computation*, 12:2385–2404, 2000.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[8] Anthony J. Bell and Terrence J. Sejnowski. "An information-maximization approach to blind separation and blind deconvolution". *Neural Computation*, 7(6):1129–1159, 1995.

[9] M. Bichsel and A. P. Pentland. "Human face recognition and the face image set's topology". *CVGIP: Image Understanding*, 59:254–261, 1994.

[10] V. Blanz and T. Vetter. "Face recognition based on fitting a 3d morphable model". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, September 2003.

[11] L. Breiman. "Out-of-bag estimation". *Unpublished. Available at ftp://ftp.stat.berkeley.edu/pub/users/breiman/OOBestimation.ps.Z*, pages 1–13, 1996.

[12] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[13] Leo Breiman. "Arcing classifiers". *Ann Statistics*, 26(3):801–849, 1998.

[14] R. Brunelli and T. Poggio. "Face recognition: Features versus templates". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1042–1052, 1993.

[15] Kyong Chang, K.W. Bowyer, S. Sarkar, and B. Victor. "Comparison and combination of ear and face images in appearance-based biometrics". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1160–1165, September 2003.

[16] R. Chellappa, C.L. Wilson, and S. Sirohey. "Human and machine recognition of faces: A survey". *The Proceedings of the IEEE*, 83:705–740, 1995.

[17] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. “A new LDA-based face recognition system which can solve the small sample size problem”. *Pattern Recognition*, 33:1713–1726, 2000.

[18] Li-Fen Chen, Hong-Yuan Mark Liao, Ja-Chen Lin, and Chin-Chuan Han. “Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof”. *Pattern Recognition*, 34(7):1393–1403, 2001.

[19] C. Cortes and V. N. Vapnik. “Support vector networks”. *Machine Learning*, 20:273–297, 1995.

[20] E. Cosatto, J. Ostermann, H.P. Graf, and J. Schroeter. “Lifelike talking faces for interactive services”. *The Proceedings of the IEEE*, 91(9):1406 –1429, September 2003.

[21] I. J. Cox, J. Ghosn, and P.N. Yianilos. “Feature-based face recognition using mixture-distance”. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 209–216, 1996.

[22] I. Craw, N. Costen, T. Kato, G. Robertson, and S. Akamatsu. “Automatic face recognition: Combining configuration and texture”. *Proc. Internation Workshop on Automatic Face- and Gesture-Recognition*, pages 53–58, 1995.

[23] Ian Craw, Nick P. Costen, Takashi Kato, and Shigeru Akamatsu. “How should we represent faces for automatic recognition?”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:725–736, 1999.

[24] Harris Drucker and Corinna Cortes. “Boosting decision trees”. In *Advances in Neural Information Processing Systems 8*, pages 479–485, 1996.

[25] Richard O. Duda, Peter E. Hart, and David G. Stork. *"Pattern Classification (2nd ed)"*. Wiley, New York, NY, 2000.

[26] Meng Joo Er, Shiqian Wu, Juwei Lu, and Hock Lye Toh. "Face recognition with radial basis function (RBF) neural networks". *IEEE Transactions on Neural Networks*, 13(3):697–710, May 2002.

[27] R.A. Fisher. "The use of multiple measures in taxonomic problems". *Ann. Eugenics*, 7:179–188, 1936.

[28] Y. Freund and R. Schapire. "A short introduction to boosting". *Journal of Japanese Society for Artificial Intelligence*, 14:771–780, 1999.

[29] Yoav Freund and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[30] B. J. Frey, A. Colmenarez, and T. S. Huang. "Mixtures of local linear subspaces for face recognition". In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 32–37, Santa Barbara, CA, June 1998.

[31] Brendan J. Frey and Nebojsa Jojic. "Transformation-invariant clustering using the EM algorithm". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):1–17, JANUARY 2003.

[32] Jerome H. Friedman. "Regularized discriminant analysis". *Journal of the American Statistical Association*, 84:165–175, 1989.

[33] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, Boston, 2 edition, 1990.

[34] A. J. Goldstein, L. D. Harmon, and A. B. Lesk. "Identification of human faces". *The Proceedings of the IEEE*, 59(5):748–760, May 1971.

[35] Shaogang Gong, Stephen J McKenna, and Alexandra Psarrou. *"Dynamic Vision From Images to Face Recognition"*. Imperial College Press, World Scientific Publishing, May 2000.

[36] Daniel B Graham and Nigel M Allinson. "Characterizing virtual eigensignatures for general purpose face recognition". In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, volume 163, pages 446–456. 1998.

[37] P.J. Grother, R.J. Micheals, and P. J. Phillips. "Face recognition vendor test 2002 performance metrics". In *Proceedings 4th International Conference on Audio Visual Based Person Authentication*, Guildford, UK, June 2003.

[38] B.K. Gunturk, A.U. Batur, Y. Altunbasak, M.H. Hayes III, and R.M. Mersereau. "Eigenface-domain super-resolution for face recognition". *IEEE Transactions on Image Processing*, 12(5):597 –606, May 2003.

[39] G.D. Guo, H.J. Zhang, and S.Z. Li. "Pairwise face recognition". In *Proceedings of The Eighth IEEE International Conference on Computer Vision*, volume 2, pages 282–287, Vancouver, Canada, July 2001.

[40] Ying Guo, Geoff Poulton, Jiaming Li, Mark Hedley, and Rong yu Qiao. "Soft margin AdaBoost for face pose classification". In *Proceedings of the 28th IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages III:221–224, Hong Kong, China, April 2003.

[41] P. Hallinan. "A low-dimensional representation of human faces for arbitrary lighting conditions". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 995–999, 1994.

[42] J. Hartigan. “Statistical theory in clustering”. *Journal of Classification*, 2:63–76, 1985.

[43] Trevor Hastie, Andreas Buja, and Robert Tibshirani. “Penalized discriminant analysis”. *The Annals of Statistics*, 23(1):73–102, 1995.

[44] Zi-Quan Hong and Jing-Yu Yang. “Optimal discriminant plane for a small number of samples and design method of classifier on the plane”. *Pattern Recognition*, 24(4):317–324, 1991.

[45] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1992.

[46] Yu Hua and Yang Jie. “A direct LDA algorithm for high-dimensional data - with application to face recognition”. *Pattern Recognition*, 34:2067–2070, October 2001.

[47] L.T. Jolliffe. *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[48] A. J.Smola. “Learning with kernels”. *Ph.D. dissertation: Technische Universität Berlin*, 1998.

[49] Comon P. Jutten and J. Herault. “Blind separation of sources”. *Signal Processing*, 24:11–20, 1991.

[50] T. Kanade. “Picture processing by computer complex and recognition of human faces”. *PhD thesis, Kyoto University*, 1973.

[51] L. Kanal and B. Chandrasekaran. “On dimensionality and sample size in statistical pattern classification”. *Pattern Recognition*, 3:238–255, 1971.

[52] J. Kittler and F.M. Alkoot. “Sum versus vote fusion in multiple classifier systems”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):110–115, JANUARY 2003.

[53] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. "On combining classifiers". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[54] Vladimir Koltchinskii and Dmitriy Panchenko. "Empirical margin distributions and bounding the generalization error of combined classifiers". *Annals of Statistics*, 30(1), February 2002.

[55] S. Kumar and J. Ghosh. "GAMLS: a generalized framework for associative modular learning systems". In *Proceedings of the Applications and Science of Computational Intelligence II*, pages 24–34, Orlando, Florida, 1999.

[56] Samuel Kutin. "Algorithmic stability and ensemble-based learning". *PhD Thesis, The Faculty of The Division of The Physical Sciences, The University of Chicago*, June 2002.

[57] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, Christoph von der Malsburg, R. P. Würtz, and W. Konen. "Distortion invariant object recognition in the dynamic link architecture". *IEEE Transactions on Computers*, 42:300–311, 1993.

[58] Kin-Man Lam and Hong Yan. "An analytic-to-holistic approach for face detection based on a single frontal view". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:673–686, 1998.

[59] Andreas Lanitis, Chris J. Taylor, and Timothy F. Cootes. "Automatic interpretation and coding of face images using flexible models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:743–756, 1997.

[60] Steve Lawrence, C. Lee Giles, A.C. Tsoi, and A.D. Back. "Face recognition: A convolutional neural network approach". *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[61] Stan Z. Li and Juwei Lu. “Face recognition using the nearest feature line method”. *IEEE Transactions on Neural Networks*, 10:439–443, March 1999.

[62] S.Z. Li, X.G. Lv, and H.J.Zhang. “View-based clustering of object appearances based on independent subspace analysis”. In *Proceedings of The Eighth IEEE International Conference on Computer Vision*, volume 2, pages 295–300, Vancouver, Canada, July 2001.

[63] S.Z. Li, J. Yan, X. W. Hou, Z. Y. Li, and H. J. Zhang. “Learning low dimensional invariant signature of 3-d object under varying view and illumination from 2-d appearances”. In *Proceedings of The Eighth IEEE International Conference on Computer Vision*, volume 1, pages 635–640, Vancouver, Canada, July 2001.

[64] Chengjun Liu and H. Wechsler. “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition”. *IEEE Transactions on Image Processing*, 11(4):467–476, April 2002.

[65] Chengjun Liu and H. Wechsler. “Independent component analysis of gabor features for face recognition”. *IEEE Transactions on Neural Networks*, 14(4):919–928, July 2003.

[66] Chengjun Liu and Harry Wechsler. “Evolutionary pursuit and its application to face recognition”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):570–582, June 2000.

[67] K. Liu, Y. Cheng, and J. Yang. “A generalized optimal set of discriminant vectors”. *Pattern Recognition*, 25:731–739, 1992.

[68] K. Liu, Y. Cheng, and J. Yang. “Algebraic feature extraction for image recognition based on an optimal discriminant criterion”. *Pattern Recognition*, 26:903–911, 1993.

[69] 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". *Int. J. Pattern Recog. Artif. Intell.*, 6:817–829, 1992.

[70] Rohit Lotlikar and Ravi Kothari. "Fractional-step dimensionality reduction". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):623–627, 2000.

[71] Juwei Lu and K.N. Plataniotis. "Boosting face recognition on a large-scale database". In *Proceedings of the IEEE International Conference on Image Processing*, pages II.109–II.112, Rochester, New York, USA, September 2002.

[72] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. "Face recognition using feature optimization and ν-support vector learning". In *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing*, pages 373–382, Falmouth, MA., USA, September 2001.

[73] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. "Face recognition using kernel direct discriminant analysis algorithms". *IEEE Transactions on Neural Networks*, 14(1):117–126, January 2003.

[74] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. "Face recognition using LDA based algorithms". *IEEE Transactions on Neural Networks*, 14(1):195–200, January 2003.

[75] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. "Regularized discriminant analysis for the small sample size problem in face recognition". *Pattern Recognition Letter*, 24(16):3079–3087, December 2003.

[76] Gábor Lugosi and Kenneth Zeger. "Concept learning using complexity regularization". *IEEE Transactions on Information Theory*, 42(1):48–54, January 1996.

[77] D. Marr. *"Vision"*. W. H. Freeman and Co, San Francisco, 1982.

[78] Aleix M. Martínez and Avinash C. Kak. “PCA versus LDA”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[79] S. Gongand S. McKenna and J. Collins. “An investigation into face pose distribution”. *Proc. IEEE International Conference on Face and Gesture Recognition*, pages 265–270, 1996.

[80] G. McLachlan and D. Peel. *“Finite Mixture Models”*. John Wiley & Sons, 2000.

[81] G.J. McLachlan. *“Discriminant Analysis and Statistical Pattern Recognition”*. Wiley, New York, 1992.

[82] J. Mercer. “Functions of positive and negative type and their connection with the theory of integral equations”. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

[83] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[84] Abdallah Mkhadri. “Shrinkage parameter for the modified linear discriminant analysis”. *Pattern Recognition Letters*, 16:267–275, March 1995.

[85] B. Moghaddam. “Principal manifolds and probabilistic subspaces for visual recognition”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, June 2002.

[86] B. Moghaddam, T. Jebara, and A. Pentland. “Bayesian face recognition”. *Pattern Recognition*, 33:1771–1782, 2000.

[87] Y. Moses, Y. Adini, and S. Ullman. “Face recognition: The problem of compensating for changes in illumination direction”. In *Proceedings of the European Conference on Computer Vision*, volume A, pages 286–296, 1994.

[88] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. "An introduction to kernel-based learning algorithms". *IEEE Transactions on Neural Networks*, 12(2):181–201, March 2001.

[89] H. Murase and S.K. Nayar. "Visual learning and recognition of 3-D objects from appearance". *International Journal of Computer Vision*, 14:5–24, 1995.

[90] R. Murray-Smith and T. A. Johansen. "Multiple model approaches to modelling and control". Taylor and Francis, London, UK, 1997.

[91] Alejandro Murua. "Upper bounds for error rates of linear combinations of classifiers". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):591–602, May 2002.

[92] Alex Pentland, Baback Moghaddam, and Thad Starner. "View-based and modular eigenspaces for face recognition". *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1–7, June 1994.

[93] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. "The FERET database and evaluation procedure for face recognition algorithms". *Image and Vision Computing J*, 16(5):295–306, 1998.

[94] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. "The FERET evaluation methodology for face-recognition algorithms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[95] P.J. Phillips and E.M. Newton. "Meta-Analysis of face recognition algorithms". In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 235–241, Washinton D.C., USA, May 20-21 2002.

[96] K.N. Plataniotis, D. Androutsos, and A.N. Venetsanopoulos. "Adaptive fuzzy systems for multichannel signal processing". *Proceedings of the IEEE*, 87(9):1601–1622, September 1999.

[97] G. Rätsch, T. Onoda, and K.-R. Müller. "Soft margins for AdaBoost". *Machine Learning*, 42(3):287–320, March 2001.

[98] Sarunas J. Raudys and Anil K. Jain. "Small sample size effects in statistical pattern recognition: Recommendations for practitioners". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[99] Fabio Roli and Josef Kittler (eds.). *"Multiple Classifier Systems: Third International Workshop"*. Lecture Notes in Computer Science, Volume 2364. Springer, Cagliari, Italy, June 24-26 2002.

[100] A. Ruiz and P.E. López de Teruel. "Nonlinear kernel-based statistical pattern analysis". *IEEE Transactions on Neural Networks*, 12(1):16–32, January 2001.

[101] H.S. Sahambi and K. Khorasani. "A neural-network appearance-based 3-d object recognition using independent component analysis". *IEEE Transactions on Neural Networks*, 14(1):138–149, JANUARY 2003.

[102] Ashok Samal and Prasana A.Iyengar. "Automatic recognition and analysis of human faces and facial expressions: A survey". *Pattern Recognition*, 25:65–77, 1992.

[103] Ferdinando Samaria and Andy Harter. "Parameterisation of a stochastic model for human face identification". In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pages 138 – 142, Sarasota FL, December 1994.

[104] Robert E. Schapire. "The boosting approach to machine learning: An overview". In *MSRI Workshop on Nonlinear Estimation and Classification*, pages 149–172, Berkeley, CA, 2002.

[105] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. “Boosting the margin: A new explanation for the effectiveness of voting methods”. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 322–330. Morgan Kaufmann, 1997.

[106] B. Schölkopf. *“Support Vector Learning”*. Oldenbourg-Verlag, Munich, Germany, 1997.

[107] Bernhard Schölkopf, Chris Burges, and Alex J. Smola. *“Advances in Kernel Methods - Support Vector Learning”*. MIT Press, Cambridge, MA, 1999.

[108] Behzad M. Shahshahani and David A. Landgrebe. “The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon”. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, September 1994.

[109] A. Sharkey. *“Combining Artificial Neural Nets”*. Springer-Verlag, 1999.

[110] L. Sirovich and M. Kirby. “Low-dimensional procedure for the characterization of human faces”. *Journal of the Optical Society of America A*, 4(3):519–524, March 1987.

[111] Marina Skurichina and Robert P. W. Duin. “Bagging, boosting and the random subspace method for linear classifiers”. *Pattern Analysis & Applications*, 5(2):121–135, June 2002.

[112] A. J. Smola, B. Schölkopf, and K.-R.Müller. “The connection between regularization operators and support vector kernels”. *Neural Networks*, 11:637–649, 1998.

[113] K.-K. Sung and T. Poggio. “Example- based learning for view-based human face detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.

[114] D. L. Swets and J. Weng. “Using discriminant eigenfeatures for image retrieval”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:831–836, 1996.

[115] Dan L. Swets and John J. Weng. “Hierarchical discriminant analysis for image retrieval”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):386 – 401, May 1999.

[116] Q. Tian, M. Barbero, Z.H. Gu, and S.H. Lee. “Image classification by the foley-sammon transform”. *Opt. Eng.*, 25(7):834–840, 1986.

[117] R. Tibshirani, G. Walther, and T. Hastie. “Estimating the number of clusters in a dataset via the gap statistic”. *Journal of the Royal Statistical Society B,* **63**, pages 411–423, 2001.

[118] Kar-Ann Toh, Juwei Lu, and Wei-Yun Yau. “Global feedforward neural network learning for classification and regression”. In Mário Figueiredo, Josiane Zerubia, and Anil K. Jain, editors, *Proceedings of the Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 407–422, Sophia Antipolis, France, September 3-5 2001.

[119] Matthew Turk. “A random walk through eigenspace”. *IEICE Trans. Inf.* & *Syst.*, E84-D(12):1586–1695, December 2001.

[120] Matthew A. Turk and Alex P. Pentland. “Eigenfaces for recognition”. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[121] N. Ueda and R. Nakano. “Deterministic annealing EM algorithm”. *Neural Networks*, 16:271–282, 1998.

[122] Dominique Valentin, Herve Abdi Alice, J. O. Toole, and Garrison W. Cottrell. "Connectionist models of face processing: A survey". *Pattern Recognition*, 27(9):1209–1230, 1994.

[123] V. N. Vapnik. *"The Nature of Statistical Learning Theory"*. Springer-Verlag, New York, 1995.

[124] V. N. Vapnik. *"Statistical Learning Theory"*. Wiley, New York, 1998.

[125] P.W. Wald and R.A. Kronmal. "Discriminant functions when covariance are unequal and sample sizes are moderate". *Biometrics*, 33:479–484, 1977.

[126] D. Windridge and J. Kittler. "A morphologically optimal strategy for classifier combination: multiple expert fusion as a tomographic process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):343 –353, March 2003.

[127] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg. "Face recognition by elastic bunch graph matching". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[128] Laurenz Wiskott and Christoph von der Malsburg. "Recognizing faces by dynamic link matching". *NeuroImage*, 4(3):S14–S18, December 1996.

[129] J. Yang, D. Zhang, and J.-Y. Yang. "A generalised K-L expansion method which can deal with small sample size and high-dimensional problems". *Pattern Anal. Applic.*, 6:47–54, 2003.

[130] Jian Yang, Jing yu Yang, David Zhang, and Jian feng Lu. "Feature fusion: parallel strategy vs. serial strategy". *Pattern Recognition*, 36(6):1369–1381, June 2003.

[131] D.S. Yeung and X.Z. Wang. "Improving performance of similarity-based clustering by feature weight learning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):556–561, April 2002.

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

[133] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. "Face recognition: A literature survey". *Technical Report, CFAR-TR00-948, University of Maryland*, 2000.

[134] Shaohua Zhou. "Probabilistic analysis of kernel principal components: mixture modeling, and classification". *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[135] A. Zien, G. Rätsch, S.Mika, B.Schölkopf, T.Lengauer, and K.-R.Müller. "Engineering support vector machine kernels that recognize translation initiation sites in dna". *Bioinformatics*, 16:799–807, 2000.