FEATURE EXTRACTION TECHNIQUES IN HIGH-DIMENSIONAL SPACES: LINEAR AND NONLINEAR APPROACHES

By

Hakan Cevikalp

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

In partial fulfillment of the requirements

For the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

August, 2005

Nashville, Tennessee

Approved: Date:

_____ _____

_____ _____

_____ _____

_____ _____

_____ _____

ACKNOWLEDGEMENTS

Most of all, I would like to express my sincere appreciation to my supervisors, Prof. Marian Neamtu and Dr. Mitch Wilkes, for their invaluable support and guidance. I am grateful to Prof. Neamtu for introducing me to the magical world of mathematics during our discussions in his office. He has provided proofs of all lemmas and theorems given in this work and without his help I would not have completed this study in such a short time.

Secondly, I would like to express my gratitude to Dr. David Noelle and Prof. Atalay Barkana for their technical feedback and encouragement. I also thank my committee members, Prof. Benoit Dawant, Prof. Richard Shiavi, and Dr. Alan Peters, for their assistance. In addition, I am very grateful to my former supervisor, Prof. John Mike Fitzpatrick, for providing his support, friendship, and encouragement when I needed them most.

I would like to thank Osmangazi University for supporting my studies in USA and providing me an inspiring atmosphere for scientific research. I also would like to thank anonymous referees, who reviewed our published papers, for their valuable comments, which helped to improve the quality of this work.

Finally, I would like to thank my mother and father, Nuriye and Adem Çevikalp, who sacrificed so much to provide me the opportunity to pursue and appreciate education. I dedicate this work to my parents. Last but not least, I would like to thank my sisters, Vildan and Şükran Çevikalp, for their love and prayers.

Thank you all for being so supportive in pursuit of my Ph. D.

TABLE OF CONTENTS

Page

Appendix

LIST OF TABLES

LIST OF FIGURES

FEATURE EXTRACTION TECHNIQUES IN HIGH-DIMENSIONAL SPACES: LINEAR AND NONLINEAR APPROACHES

HAKAN CEVIKALP

Dissertation under the direction of Professor Mitch Wilkes

In this thesis, feature extraction methods for pattern recognition tasks in high-dimensional spaces are investigated. High-dimensional spaces are quite different from the three-dimensional (3-D) space in terms of geometrical and statistical properties. Although high-dimensional sample spaces contain more information regarding capability to discriminate different class samples with more accuracy, pattern classification techniques that carry out computations at full dimensionality may not deliver the advantages of high-dimensional sample spaces if there are insufficient training sample patterns. In such cases, reliable density estimation is extremely difficult. Therefore, the dimensionality of the sample space must be reduced via feature extraction methods before the application of the classifier to data samples in high-dimensional spaces. However, in order to retain discriminatory information which the high-dimensional sample spaces provide, good dimension reduction methods are needed.

In this study, a linear feature extraction method which exploits the advantages of high-dimensional spaces was proposed. Then, this linear method was generalized to the nonlinear case by utilizing kernel functions. There is no loss of discriminatory information content in the sense that the proposed methods achieve 100% recognition rate with respect to training data under certain conditions. Experimental results using different databases also show that

the proposed methods are superior to other feature extraction methods in terms of generalization and real-time performances.

Approved_____ Date _____

CHAPTER I

INTRODUCTION

A pattern is the description of an object, and this pattern can be anything, perceivable with the five senses. An image of a face, a fingerprint, a spoken word, a hand written character, and a biological waveform are, depending on the application, some pattern examples. Patterns are represented by a set of features (attributes). Each feature numerically expresses a property of the pattern, and the total number of features determines the size of the original feature space. This space is also called the sample space. Pattern samples with similar properties form the pattern classes, and pattern recognition can be defined as the classification of patterns into a number of categories or classes via the extraction of significant features from a background of irrelevant detail [106].

Human beings can do most pattern recognition tasks well. We receive data through our senses and most of the time we can easily identify the source of the data. However, technology has introduced many new pattern recognition tasks which must be performed more cheaply and faster than human beings can. Therefore, much research is being done to design and build machines that recognize patterns. Indeed, machines that recognize patterns are used in many areas including fingerprint identification, speech recognition, face recognition, optical character recognition, DNA sequence identification, and many more. Although human beings can solve many pattern recognition problems with little effort, pattern recognition of machines is an extremely difficult task. Rapidly growing computing power has facilitated the use of complex and diverse methods for data analysis and

recognition. At the same time, because of the availability of large databases and strict performance requirements, (speed, accuracy, and cost), demands on automatic pattern recognition systems have been rising constantly [57].

A pattern recognition system consists of a series of stages, of which the feature extraction and classification are the most crucial for its overall performance. The feature extraction reduces the dimensionality of the sample space by keeping the most discriminatory information. The performance of the feature extraction stage significantly affects the design and performance of classification stage. If the best set of features is selected, the job of subsequent classifiers will be trivial. On the other hand, if the features with little discriminatory power are chosen, a more sophisticated classification model may be needed.

Feature extraction is more problem and domain dependent. For example, a good feature extractor for pattern recognition tasks with high-dimensional sample spaces might not work well for the pattern recognition tasks where the dimensionality of the sample space is small. In this thesis, the feature extraction techniques for high-dimensional spaces are investigated. High-dimensional spaces are quite different from the three-dimensional (3-D) space in terms of geometrical and statistical properties. Although high-dimensional sample spaces contain more information regarding capability to discriminate different class samples with more accuracy, pattern classification techniques that carry out computations at full dimensionality may not deliver the advantages of high-dimensional sample spaces if there are insufficient training sample patterns. In such cases, reliable density estimation is extremely difficult. Therefore, the dimensionality of the sample space must be reduced via feature extraction methods before the application of the classifier to data samples in high-dimensional spaces.

However, in order to retain discriminatory information which the high-dimensional sample spaces provide, good dimension reduction methods are needed.

In this study, a linear feature extraction method which exploits the advantages of high-dimensional spaces was proposed. Then, this linear method was generalized to the nonlinear case by utilizing kernel functions. In addition, we proposed a variation of a linear subspace classifier which is suitable for pattern recognition tasks in high-dimensional spaces. This method was also generalized to the nonlinear case by employing kernel functions. The usefulness of the proposed methods was demonstrated with experiments using various databases.

1.1 Outline

This thesis is divided into four major parts. An overview of the stages of a typical pattern recognition system is given in Chapter 2. Then, some basic concepts from pattern recognition area are explained.

Chapter 3 introduces the characteristic properties of high-dimensional sample spaces first. Then linear feature extraction methods are examined extensively, and a novel linear feature extraction method, called the Discriminative Common Vector (DCV) Method, is proposed. Finally, we compare the proposed method to other linear feature extraction methods using a wide range of different databases and formulate our conclusions at the end of the chapter.

Chapter 4 presents a general introduction to nonlinear feature extraction methods employing kernel functions. We generalize the linear DCV Method to the nonlinear case by utilizing kernel functions in this chapter. Then, a large scale comparison of feature extraction

methods is carried out and its results are examined. Finally, we draw our conclusions based on the experimental results at the end of the chapter.

Chapter 5 describes the linear and nonlinear subspace classifiers. We introduce a variation of a subspace classifier here and then generalize it to the nonlinear case by using kernel functions. Finally, we give experimental results and draw our conclusions at the end of the chapter.

CHAPTER II

PATTERN RECOGNITION SYSTEM

Pattern recognition can be defined as the classification of patterns into a number of categories or classes via the extraction of significant features from a background of irrelevant detail. A typical pattern recognition system has four stages as shown in Figure 2.1. There is an unknown pattern sample presented as a set of features at the input of a pattern recognition system, and there is a set of predefined classes at the output. The task of the system is to assign the unknown pattern sample to one of the classes.

In the following sections each stage of the pattern recognition system is described. Then, we explain some basic concepts of pattern recognition used throughout this thesis.

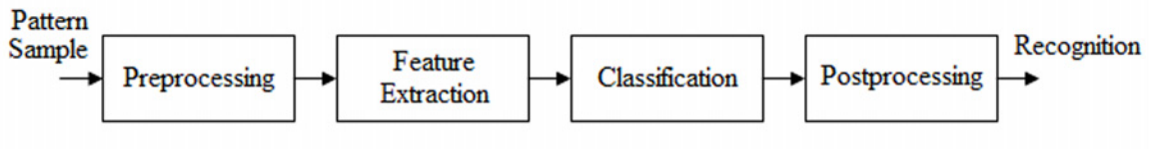


Figure 2.1: A typical pattern recognition system.

2.1 Pre-processing

This stage typically includes operations that improve the representation of the patterns. Therefore, it may include data registration, noise removal, segmentation, and data normalization, depending on the nature of pattern recognition task. In face recognition problems, the face images are registered so as to make sure that the eyes appear in the same coordinates of the images. Pattern samples usually contain some noise, which may need to be

reduced before classification. The term “noise” is usually defined in a wide sense in pattern recognition area. Any property of the pattern that hinders a pattern recognition system’s task and is not due to the true underlying model is regarded as noise. In speech recognition problems, filters are usually used to remove noise and enhance higher frequencies. Some recognition tasks may require segmentation of individual patterns. For instance, we may need to segment faces in an image to create meaningful patterns for the feature extraction step. Normalization is to scale the features of data so as to fall within a small specified range. Some neural network models require normalization of data samples to be in the range of -1 to 1 or 0 to 1. All these operations contribute to defining a compact representation of patterns [32].

2.2 Feature Extraction

The selection of the best set of features for dimension reduction is one of the most important issues of pattern recognition. The aim of feature extraction is to reduce the number of features of patterns and at the same time retain as much as possible of their discriminatory information. Therefore, a good feature extractor chooses features which are similar for patterns in the same class and very different for patterns in different classes. Since the dimensionality of the sample space is reduced after the feature extraction step, extraction will yield savings in memory and time consumption. The feature extraction step may also alleviate the worst effects of the so-called *curse of dimensionality,* which will be explained in detail in the following sections.

2.3 Classification

The task of the classification is to assign the feature vector provided by the feature extractor to a class. The output of the classifier is typically a discrete selection of one of the pre-defined classes. All the preceding components of a pattern recognition system are designed and tuned for improving the performance of the classifier. The degree of difficulty of the classification depends on the similarity relations between the patterns of different classes. Therefore, its success is significantly affected by the feature extraction stage.

2.4 Post-processing

The post-processing stage aims to improve overall classification accuracy. It tries to minimize the classification error rate based on the classification outputs. This stage usually utilizes a priori information about the problem to accomplish its task.

2.5 Some Concepts of Pattern Recognition

Learning and generalization are two important concepts of pattern recognition. In the following sections they are explained in detail.

2.5.1 Learning Types

A pattern recognition system produces a mathematical model which maps the patterns to the corresponding classes. Typically, it is not possible to determine a reliable mapping without the help of data samples. Finding this model is called learning or training, and the sample patterns used during this process called the training set samples. Any method that incorporates information from the training set samples in pattern classification employs

learning. There are three basic types of learning methods depending upon the nature of the pattern recognition task.

1. Supervised Learning: In supervised learning, class labels or costs of training set samples are known before the training phase begins. The training phase computes the model which minimizes the total cost for the training set patterns. This kind of learning involves human labor. It is typically the most common learning method, and it has many applications in pattern recognition area.

2. Unsupervised Learning: In unsupervised learning, the training set samples are not labeled, and the main objective is to unravel the underlying similarities and group similar patterns together. Unsupervised learning does not require human labor for labeling, and it has many applications in engineering, such as image segmentation and multi-spectral remote sensing.

3. Reinforcement Learning: In reinforcement learning, a feedback is provided in order to compute the model that maps the patterns to the classes. Typically, the teaching feedback is the information of the fact that the tentative class is right or wrong instead of the patterns' true label information.

2.5.2 Generalization

To compute the model that maps the pattern samples to their corresponding classifiers, we train the pattern recognition system by using available training set samples. However, a pattern recognition system, which is trained to maximize the performance in recognizing training set samples, may not recognize new test samples well. This is the issue of generalization. Therefore, generalization ability of a pattern recognition system refers to its

performance in recognizing new samples not used in the learning stage. There are basically two causes of poor generalization in pattern recognition systems:

1. Pattern recognition system is intensively optimized (overtrained) on the training set, and it ends up memorizing the training set samples. This is also referred to as an overfitting problem.

2. The number of features is too large compared to the number of training samples. This is also called the curse of dimensionality.

Overfitting

A successfully designed pattern recognition system produces a reliable input-output mapping model and recognizes well the new samples that are slightly different from the training set samples. However, an overtrained pattern recognition system will produce a mapping model which describes the training data well but does not generalize to unseen samples. Figure 2.2 adopted from [10] illustrates two different mapping models which are produced by successful and overtrained pattern systems. Although the overtrained system recognizes all the training set samples correctly, it is very unlikely perform well on new patterns.

Overtrained pattern recognition systems usually take into consideration the features which are present in the training data but not part of the correct underlying input-output mapping model. These features may stem from noise in the system. Thus, the mapping model becomes more complex and loses the ability to generalize between similar input-output patterns.

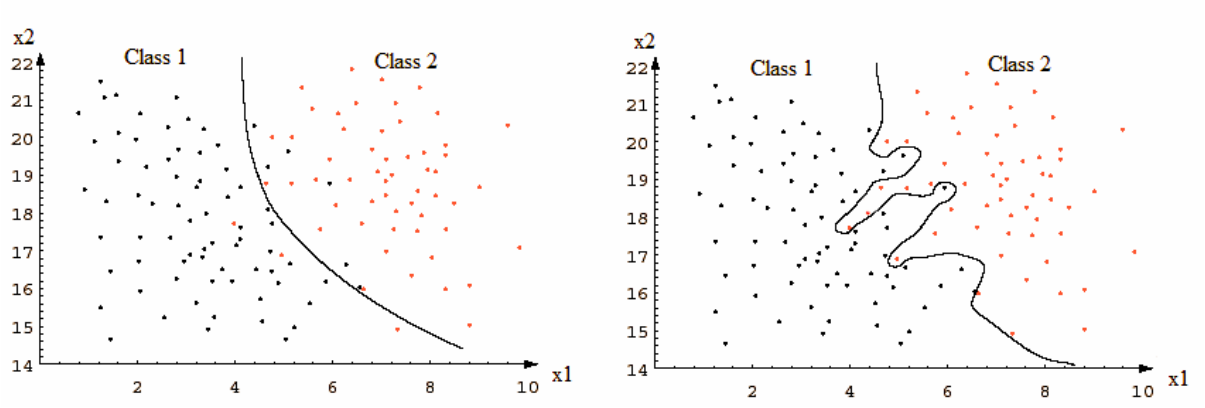


Figure 2.2: Succesfully designed and overtrained pattern recognition systems.

Curse of Dimensionality and the Relation between the Dimensionality of the Sample Space and the Training Set Size

The curse of dimensionality was first introduced by Richard Bellman in the adaptive control processes area [8]. It states that for a fixed training set size, increasing the number of features first enhances the performance of the recognition system, but beyond a certain point, adding new features degrades the performance of the system. This occurs because increasing the dimensionality of the sample space leads to sparseness which in turn leads to poor representation of the vector densities and input-output model.

In order to understand this phenomenon, consider the following example from [10]. Let x_i $(i = 1,2,...,N)$ represent a training set vector in a *d*-dimensional space, where there are a total of *N* samples in the training set. Suppose the function $f(x_i)$ is the nonlinear function model that assigns the patterns to the desired classes. Assume the function $f(x_i)$ is arbitrarily complex and completely unknown. We first divide each of the training samples into a large number of boxes or cells as shown in the Figure 2.3. The desired class $f(x_i)$ of a sample is specified by the cell in which it lies. Each of the training samples corresponds to a

point in the cells. We classify a new test pattern from an unknown class by using the cell in which it falls. If we increase the number of cells along each axis, this process increases the precision. However, assuming that each input sample is divided into M divisions, the total number of cells will be M^d, which grows exponentially with the dimensionality d of the sample space. Since each cell must contain at least one data sample, the number of training samples to specify the mapping model grows exponentially. If we have a limited number of training samples, then increasing the dimensionality of the sample space will lead to the point where the data samples are very sparse, which in turn provides a very poor representation of the model assigning the patterns to desired classes.

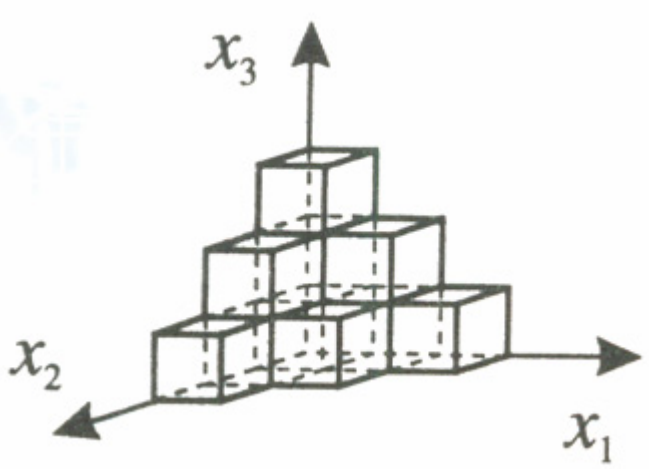


Figure 2.3: A mapping from a d-dimensional space to an output variable y can be accomplished by dividing the input space into a number of cells, and assigning each cell to a class. However, the number of cells grows exponentially with the dimensionality d.

Therefore, we should design pattern recognition systems by using a small number of features that have the most discriminatory information. It has been proved that the required number of training set samples is linearly related to dimensionality for a recognition system using a linear classifier; it is linearly related to the square of the dimensionality for a

quadratic classifier. In a nonlinear case, as given in the example, the training set size must increase exponentially for a good mapping model [59].

CHAPTER III

LINEAR FEATURE EXTRACTION METHODS IN HIGH-DIMENSIONAL SPACES

The objective of this study is to investigate pattern recognition methods for high-dimensional sample spaces. It has been demonstrated that high-dimensional space is significantly different from the three-dimensional (3-D) space, and that our experience in 3-D space tends to mislead our intuition of geometrical and statistical properties in high-dimensional sample spaces [59]. Therefore, we first review some characteristic properties of high-dimensional spaces which motivate the use of feature extraction techniques in pattern recognition tasks with high-dimensional sample spaces. Then we introduce our basic notation and examine linear feature extraction methods extensively. In addition, a novel feature extraction method that exploits the advantages of high-dimensional sample spaces is proposed in this chapter. We compare the proposed method to other discussed linear feature extraction methods in terms of recognition accuracy, numerical stability, and real-time performance using various databases. Finally, we formulate our conclusions based on the experimental results at the end of the chapter.

3.1 Characteristic Properties of High-Dimensional Spaces

For a fixed number of training samples, increasing the dimensionality of the sample space spreads the data over a greater volume. This process reduces overlap between the classes and enhances the potential for discrimination. Therefore, it is reasonable to expect that high-dimensional sample spaces contain more information of capability to detect more classes

with more accuracy. However, from the curse of dimensionality, we know that there is a penalty in classification accuracy as the number of features increases beyond some point. Therefore, techniques of carrying out computations at full dimensionality may not deliver the advantages of high-dimensional sample spaces if there are insufficient training samples.

Experiments have shown that high-dimensional sample spaces are mostly empty since data typically concentrate in an outside shell of the sample space far from the origin as the dimensionality increases [59]. This implies that the data samples are usually in a lower-dimensional structure. As a consequence, high-dimensional data can be projected to a lower-dimensional subspace without losing significant information in terms of separability among the classes by employing some feature extraction techniques. It has been also proved that as the dimensionality of the sample space goes to infinity, lower-dimensional linear projections approach a normality model with a probability approaching one. Here normality implies either a normal or a mixture of normal distributions.

It turns out that the normally distributed high-dimensional data concentrate in the tails and uniformly distributed high-dimensional data concentrate in the corners. This makes density estimation task for high-dimensional sample spaces a difficult task. In this case, local neighborhoods become empty, which in turn produces the effect of losing detailed density estimation.

Another interesting observation was related to the first and the second order statistics of data samples. It has been shown that for low-dimensional sample spaces, class means representing first order statistics play a more important role in discriminating between classes than the class covariances representing second order statistics. However, as dimensionality increases, class covariance differences become more important.

In summary, the dimensionality of the sample space must be reduced before the application of the classifier to data samples in high-dimensional sample spaces. However, in order to keep the discriminatory information, which the high-dimensional sample spaces provide, good dimension reduction techniques are needed. In this study, the dimension reduction techniques for high-dimensional sample spaces are investigated.

3.2 Dimensionality Reduction

Dimensionality reduction usually improves the accuracy of recognition of a pattern recognition system beside saving memory and time consumptions, as described in the previous chapter. This seems somewhat paradoxical since dimensionality reduction usually reduces the information content of the input data. However, a good dimensionality reduction technique keeps the features with the high discriminative information and discards the features with redundant information. Thus, the worst effects of the curse of dimensionality are reduced after the dimensionality reduction process, and often improved performance is achieved over the application of the selected classifier in the original sample space. But given a set of features, how can the best set of features for classification be selected?

Given a set of features, selection of the best set of features can be achieved in two different ways. The first approach is to identify the features that contribute most to class separability. Therefore, our task is the selection of previously decided $\tilde{d}$ features out of our initial d features. This is called feature selection. The second approach is to compute a transformation which will map the original input space to a lower-dimensional space by keeping the most of the discriminative information. This transformation can be linear or nonlinear combinations of the samples in the training set. This approach is usually called the

feature extraction. Both approaches require a criterion function, *J*, which is used to judge whether one subset of features is better than another.

3.2.1 Feature Selection

In this approach we select the best set of $\tilde{d}$ features for classification out of original *d* features. We must first define a criterion function, *J*, to accomplish this task. The selected criterion is evaluated for all possible combinations of $\tilde{d}$ features systematically selected from *d* features. Then, we select the set of features for which the criterion is maximum as our final features. However, this task is not very straightforward because there are $\frac{d!}{(d-\tilde{d})!\tilde{d}!}$ possible combinations for evaluation. As a consequence, this procedure may not be feasible even for moderate values of *d* and $\tilde{d}$. Therefore, we will not consider the feature selection methods in this study since we are only interested in the data sets with high-dimensional spaces. Some detailed information about the feature selection methods can be found in [10] and [120].

3.2.2 Feature Extraction

In this approach we seek a transformation which will map the original input space to a lower-dimensional space by keeping the features offering high classification power. The optimization is evaluated over all possible transformations of the data samples. Let $\tilde{W}$ denote the sought transformation for which $J(W) = \max_{\tilde{W} \in \varpi} J(\tilde{W}(x))$, where ϖ is the family of allowable transformations and *x* refers to the training set samples. The new samples in the

transformed space are computed by $y = W(x)$. The criterion function is typically a measure of distance or similarity between training set samples.

3.3 Linear Feature Extraction Methods

Feature extraction has been one of the most important issues of pattern recognition. Most of the feature extraction literature has centered on finding linear transformations, which map the original high-dimensional sample space into a lower-dimensional space that hopefully contains all discriminatory information. As explained previously, the principal motivation behind dimensionality reduction by feature extraction is that it may reduce the worst effects of the curse of dimensionality [10]. Also linear feature extractions techniques are often used as pre-processors before more complex nonlinear classifiers. In the following sections we discuss these linear methods.

3.4 Definitions

Let the training set be composed of *C* classes, where the *i*-th class denoted by $\omega^{(i)}$ contains N_i samples, and let x_m^i be a *d*-dimensional column vector, which denotes the *m*-th sample from the *i*-th class. There will be a total of $M = \sum_{i=1}^{C} N_i$ samples in the training set. The within-class scatter matrix S_W, the between-class scatter matrix S_B, and the total scatter matrix S_T are defined as

$$S_W = \sum_{i=1}^{C} \sum_{m=1}^{N_i} (x_m^i - \mu_i)(x_m^i - \mu_i)^T = A_W A_W{}^T, \tag{3.1}$$

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T = A_B A_B{}^T, \tag{3.2}$$

and

$$S_T = \sum_{i=1}^{C} \sum_{m=1}^{N_i} (x_m^i - \mu)(x_m^i - \mu)^T = A_T A_T{}^T = S_W + S_B, \tag{3.3}$$

where $\mu = \frac{1}{M} \sum_{i=1}^{C} \sum_{m=1}^{N_i} x_m^i$ is the mean of all samples, and $\mu_i = \frac{1}{N_i} \sum_{m=1}^{N_i} x_m^i$ is the mean of samples in $\omega^{(i)}$. The matrices $A_W \in \Re^{dxM}$, $A_B \in \Re^{dxC}$, and $A_T \in \Re^{dxM}$ are defined as

$$A_W = [x_1^1 - \mu_1 \quad ... \quad x_{N_1}^1 - \mu_1 \quad x_1^2 - \mu_2 \quad ... \quad x_{N_C}^C - \mu_C], \tag{3.4}$$

$$A_B = [\sqrt{N_1}(\mu_1 - \mu) \quad ... \quad \sqrt{N_C}(\mu_C - \mu)], \tag{3.5}$$

and

$$A_T = [x_1^1 - \mu \quad ... \quad x_{N_1}^1 - \mu \quad x_1^2 - \mu \quad ... \quad x_{N_C}^C - \mu]. \tag{3.6}$$

3.5 Principal Component Analysis (PCA)

One of the most popular feature extraction methods is the PCA Method. The main idea behind the PCA Method is to find a lower-dimensional space in which the data samples are optimally represented [10], [32], [113]. Therefore, the objective is to find the best set of projection directions in the sample space that will maximize the total scatter across all samples such that the criterion $J_{PCA}(W_{opt}) = \max |W^T S_T W|$ is maximized under the constraint that the columns of the projection matrix W be orthonormal (i.e., $w_i w_j = \delta_{ij}$, where δ_{ij} is the Kronecker's delta). Geometrically, PCA can be seen as the rotation of the axes of the original coordinate system to a new set of orthogonal axes which are ordered according to the amount of variation of the original data they account for. The criterion J_{PCA} is maximized when the most significant eigenvectors (the eigenvectors corresponding to the

largest eigenvalues of S_T) are chosen as the projection vectors for feature extraction. The eigenvectors of S_T are called as *principal components*. The total scatter matrix S_T is a symmetric positive semi-definite matrix. Therefore, all eigenvectors of S_T are orthogonal and, all eigenvalues of S_T are greater than or equal to zero. Let us assume the eigenvalues are ordered such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{rt}$, where *rt* ($rt \leq d$) is the rank of S_T. Then we select *n* ($n \leq rt$) eigenvectors corresponding to the largest eigenvalues and form the projection matrix $W = [w_1 \quad w_2 \quad ... \quad w_n]$. Any sample can be approximated by a linear combination of the significant eigenvectors. The sum $\sum_{j=n+1}^{rt} \lambda_j$ of the eigenvalues corresponding to the eigenvectors not used in reconstruction gives the mean square error. Thus, the number of eigenvectors *n* can be chosen such that the ratio of the sum of the eigenvalues corresponding to the retained eigenvectors to the sum of all eigenvalues exceeds a percentage η, i.e., $\sum_{j=1}^{n} \lambda_j / \sum_{j=1}^{rt} \lambda_j \geq \eta$. Typical values of this percentage lie between $0.9 \leq \eta < 1$.

Since Principal Component Analysis is a scale dependent method, a standardization procedure is usually carried out before applying PCA. Data are usually transformed to have zero mean and unit variance in each axis during the standardization procedure. This gives equal importance to each axis such that the PCA Method is not affected by the different units used to measure the axes.

The algorithm for the PCA Method can be summarized as follows:

Step 1: Find the mean μ of the training set samples and center the samples by subtracting mean from each sample such that

$$\tilde{x}_m^i = x_m^i - \mu, \qquad i = 1,...,C, \qquad m = 1,...,N_i . \tag{3.7}$$

Optionally, samples can also be standardized to have unit variance.

Step 2: Form the total scatter matrix $S_T = \sum_{i=1}^{C} \sum_{m=1}^{N_i} \tilde{x}_m^i \tilde{x}_m^{i\,T}$ and compute its eigenvectors and corresponding eigenvalues. Select the most significant *n* eigenvectors such that the sum of corresponding eigenvalues is 95% of the sum of all eigenvalues. Then form the matrix $W = [w_1 \quad w_2 \quad ... \quad w_n]$, where columns are the computed eigenvectors.

Step 3: Find the new feature vectors by

$$y_m^i = W^T \tilde{x}_m^i, \qquad i = 1,...,C, \qquad m = 1,...,N_i . \tag{3.8}$$

In the transformed space each new sample vector will have *n* entries. Thus, the original dimensionality *d* of the sample space is reduced to *n* by this process.

Sometimes the new feature vectors are normalized by the eigenvalues in the transformed space in order to minimize the within-class scatter, i.e.,

$$y_m^i = [\frac{y_{m1}^i}{\lambda_1} \quad \frac{y_{m2}^i}{\lambda_2} \quad ... \quad \frac{y_{mn}^i}{\lambda_n}], \qquad i = 1,...,C, \qquad m = 1,...,N_i . \tag{3.9}$$

3.5.1 Computational Considerations

If the dimensionality *d* of the sample space is too large, the total scatter matrix $S_T \in \Re^{dxd}$ will be a huge matrix, e.g., in face recognition tasks images of size 256 by 256 yield scatter matrices of size 65,536 by 65,536. Computing the eigenvalues and eigenvectors of $S_T \in \Re^{dxd}$ will be difficult and numerically unstable in these cases. However, the eigenvectors and corresponding eigenvalues can be obtained by calculating the eigenvectors of the smaller *M* by *M* matrix, $A_T^{\,T} A_T$, defined such that $S_T = A_T A_T^{\,T}$, where A_T is given in (3.6). Let λ_k and v_k be the *k*-th nonzero eigenvalue and the corresponding eigenvector of $A_T^{\,T} A_T \in \Re^{MxM}$, i.e.,

$$(A_T{}^T A_T)v_k = \lambda_k v_k, \qquad k = 1,...,rt. \tag{3.10}$$

Then,

$$(A_T A_T{}^T)A_T v_k = \lambda_k A_T v_k, \qquad k = 1,...,rt, \tag{3.11}$$

which means that $w_k = A_T v_k$ will be the eigenvector corresponding to the k-th nonzero eigenvalue of S_T.

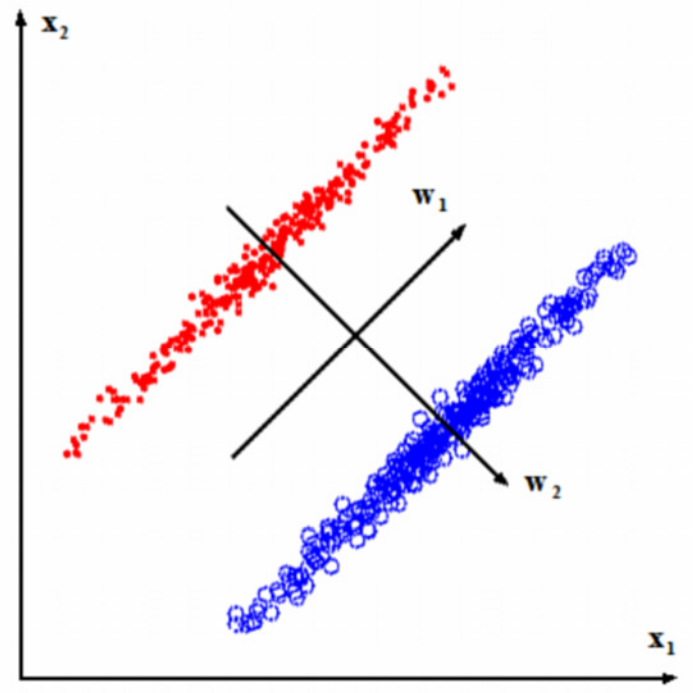


Figure 3.1: Eigenvectors found by the PCA Method. The PCA Method suggests choosing the most significant vector w_1 for feature extraction since it shows the direction of the maximum variation. This will cause misclassification in the transformed space. However, if the less significant eigenvector w_2 is chosen for feature extraction, all samples can be classified correctly. Therefore, PCA Method may not be suitable for pattern recognition tasks.

3.5.2 Drawbacks of PCA

The PCA Method is an unsupervised method since it does not consider the classes within the training set data. Although it is optimal for reconstruction, it is not necessarily optimal from a

discrimination point of view [7], [104]. Thus, the projection vectors chosen for optimal reconstruction may obscure the existence of separate classes. This fact is illustrated in Figure 3.1. In this figure two linearly separable classes with Gaussian distributions having similar covariance matrices are plotted. As can be seen in the figure, choosing the first significant eigenvector w_1 for feature extraction discards the discriminatory information and causes classification errors. Therefore, the PCA Method may cause loss of very important discriminatory information for classification even though this information is not very important for representation of data.

3.6 Linear Discriminant Analysis (LDA)

A typical way to attack a pattern recognition problem is first to estimate Gaussian (normal) density functions of classes assuming Gaussian distributions for all classes and then construct the quadratic discriminant functions that specify the decision boundaries by using the estimated density functions. However, it has been proved that the required number of training sample patterns is linearly related to the square of dimensionality of the feature space for a quadratic classifier. Therefore, it is almost impossible to obtain acceptable recognition rates by utilizing density estimation procedures when the dimensionality of the sample space is large compared to the number of training sample patterns. One way to simplify the problem is to assume that all classes have Gaussian distributions with identical covariance structures. In this case, the discriminant functions are linear, and the required number of training samples is linearly related to the dimensionality of the sample space. Linear Discriminant Analysis techniques are based on these assumptions, and they seek projection directions that maximize the between-class separability and minimize the within-class variability. Thus, by

applying these approaches, we find projection directions that on one hand maximize the distance between the samples of different classes, and on the other, minimize the distance between samples of the same class. Although LDA techniques are based on heavy assumptions that may not hold in many applications, it turned out that the linear discriminant functions can produce acceptable results even when the covariance structures are different. Thus, LDA approaches have been successfully applied in many classification problems such as image recognition, multimedia information retrieval, and medical applications. In the following sections we will examine these approaches in more detail.

3.7 Linear Discriminant Methods that Use Non-Orthogonal Projection Vectors for Feature Extraction

All methods in this category use projection vectors satisfying the orthogonality constraints $w_i^T S_W w_j = \delta_{ij}$ or $w_i^T S_T w_j = \delta_{ij}$. Therefore, the data samples in the transformed space will be uncorrelated after feature extraction step.

3.7.1 The Fisher's Linear Discriminant Analysis Method

This method was originally proposed by Fisher for taxanomic problems [36]. Although this method can be applied to the classes with different distributions, it becomes optimal Bayes classifier when all classes have Gaussian distribution with identical covariance structures. The method aims to maximize the Fisher's Linear Discriminant Analysis criterion,

$$J_{FLDA}(W_{opt}) = \max \frac{|\widetilde{S}_B|}{|\widetilde{S}_W|} = \max \frac{|W^T S_B W|}{|W^T S_W W|}, \tag{3.12}$$

subject to the normalization constraint $w_i^T S_W w_j = \delta_{ij}$. Here $\widetilde{S}_B$ and $\widetilde{S}_W$ represent new between-class and within-class scatter matrices in the transformed space, and $|.|$ represents

the determinant operation. The FLDA criterion simply measures the square of the hyperellipsoidal scattering volume. This criterion is maximized when the column vectors w_k of the projection matrix W are the eigenvectors corresponding to the nonzero eigenvalues of the matrix $S_W^{-1}S_B$. However, to find these eigenvectors, S_W must be nonsingular. Since the rank of the between-class scatter matrix S_B cannot be bigger than C-1, we cannot obtain more than C-1 projection directions for feature extraction. Therefore, new dimensionality of the transformed space can be at most C-1. If S_W is isotropic, the projection directions will span the range space of S_B. In this special case, the projection directions can be found by applying the Gram-Schmidt orthonormalization procedure to the C-1 vectors, $\mu_i - \mu, \quad i = 1,...,C-1.$

Computational Considerations

The matrix $S_W^{-1}S_B$ is typically not symmetric. Therefore, the eigen-decomposition of $S_W^{-1}S_B$ may be unstable. To avoid this problem, the simultaneous diagonalization algorithm is often employed to obtain a stable eigen-decomposition [39], [104]. Assuming S_W is nonsingular, this algorithm can be summarized as follows:

Step 1: Find the eigenvalues and corresponding eigenvectors of S_W. Let $U = [u_1 \quad u_2 \quad ... \quad u_d]$ be the orthogonal matrix whose columns are computed eigenvectors and Λ_W be a diagonal matrix with nonzero eigenvalues. We assume S_W is nonsingular hence, there are d nonzero eigenvalues. Then, $S_W = U\Lambda_W U^T$. Choose the transformation $Z = U\Lambda_W^{-1/2}$ that whitens S_W such that

$$(U\Lambda_W^{-1/2})^T S_W (U\Lambda_W^{-1/2}) = I \Leftrightarrow Z^T S_W Z = I \,. \tag{3.13}$$

Step 2: Find the nonzero eigenvalues and corresponding eigenvectors of $Y = Z^T S_B Z$. Let V be the matrix whose columns are computed eigenvectors and Λ_Y be the diagonal matrix of nonzero eigenvalues.

$$V = [v_1 \quad v_2 \quad ... \quad v_{rb}] \tag{3.14}$$

and

$$\Lambda_Y = diag(\lambda_1, \lambda_2, ..., \lambda_{rb}), \tag{3.15}$$

where *rb* is the rank of S_B and *rb* cannot be bigger than *C*-1.

Step 3: Let $W = U\Lambda_W^{-1/2}V$. Then W diagonalizes S_B and S_W at the same time. Since $S_W^{-1} = U\Lambda_W^{-1}U^T$, the matrix $S_W{}^{-1}S_B$ becomes

$$\begin{aligned} S_W^{-1}S_B &= U\Lambda_W^{-1}U^T U\Lambda_W^{1/2}V\Lambda_Y V^T \Lambda_W^{1/2}U^T \\ &= U\Lambda_W^{-1/2}V\Lambda_Y V^T \Lambda_W^{1/2}U^T \\ &= W\Lambda_Y W^{-1}. \end{aligned} \tag{3.16}$$

Therefore Λ_Y is the diagonal matrix of eigenvalues of $S_W{}^{-1}S_B$ and W is the matrix whose columns are the eigenvectors of $S_W{}^{-1}S_B$.

The projection matrix W diagonalizes both S_B and S_W, and all projection vectors are orthogonal with respect to the scatter matrices, i.e., $w_i^T S_W w_j = \delta_{ij}$, $w_i^T S_B w_j = \lambda_i \delta_{ij}$, $w_i^T S_T w_j = (\lambda_i + 1)\delta_{ij}$, where λ_i is the nonzero eigenvalue of the matrix *Y*. Recently Jin *et al.* proposed the Uncorrelated Optimal Discriminant Vector Method (UODV) [60], [61]. This method finds the projection vectors that maximize the FLDA criterion subject to the

constraint $w_i^T S_T w_j = \delta_{ij}$. Authors proposed the following iterative algorithm to find these projection vectors successively.

Step 1: *Finding the first projection direction* w_1*:*

The first projection direction is the vector maximizing the criterion $J_{FLDA}(w) = \max \frac{w^T S_B w}{w^T S_W w}$.

It is the eigenvector corresponding to the maximum eigenvalue of $S_W{}^{-1} S_B$.

Step 2: *Finding remaining projection directions* w_j *,* $j = 2,...,C-1$*:*

The *j*-th uncorrelated optimal discriminant vector, w_j, is the eigenvector corresponding to the maximum eigenvalue of the following eigen-equation:

$$U_j S_B w_j = \lambda_j S_W w_j, \tag{3.17}$$

where

$$U_1 = I, \tag{3.18}$$

$$U_j = I - S_T D_j^T (D_j S_T S_W^{-1} S_T D_j^T)^{-1} D_j S_T S_W^{-1}, \tag{3.19}$$

$$D_j = [w_1 \quad w_2 \quad ... \quad w_{j-1}]^T. \tag{3.20}$$

In these equations $I \in \Re^{dxd}$ represents the identity matrix. Although the UODV Method is called differently, it gives rise to the same projection vectors as the FLDA Method with the exception that the norms of the projection vectors are different. The iterative algorithm proposed by Jin *et al.* is computationally too expensive and unstable compared to the simultaneous diagonalization algorithm used for the FLDA Method. Thus, it makes more sense to compute the projection directions using the simultaneous diagonalization algorithm. Then projection vectors can be easily normalized such that they satisfy the constraint, $w_i^T S_T w_j = \delta_{ij}$.

3.7.2 The PCA+FLDA Method

The FLDA criterion $J_{FLDA}(W_{opt}) = \max \frac{|W^T S_B W|}{|W^T S_W W|}$ is maximized when the column vectors of the projection matrix W are the eigenvectors of $S_W{}^{-1} S_B$. In pattern recognition tasks with high dimensional spaces, the FLDA Method cannot be applied directly. This stems from the fact that the rank of S_W is at most *M-C*, and, in general, the number of the samples in the training set, *M*, is smaller than the dimensionality of the sample space *d.* As a consequence, S_W is singular in this case. This problem is also known as the "small sample size problem" [39].

In the last decade numerous methods have been proposed to solve this problem. These methods can be classified into two basic groups. The methods in the first group apply linear algebra techniques to solve the numerical problem of inverting the singular matrix S_W. For instance, Tian *et al*. [108] used the Pseudo-Inverse Method of replacing $S_W{}^{-1}$ with its pseudo-inverse. The Perturbation Method is used in [53] and [136], where a small perturbation matrix Δ is added to S_W in order to make it nonsingular. However, the above methods are typically computationally expensive since the scatter matrices are very large (e.g., images of size 256 by 256 yield scatter matrices of size 65,536 by 65,536). The methods in the second group reduce the dimensionality of the original sample space for solving the singularity problem. Swets and Weng [104] proposed a two stage PCA+FLDA method, also known as the Fisherface Method since it was first proposed for face recognition, in which PCA is first used for dimension reduction so as to make S_W nonsingular before the application of LDA. The algorithm for the PCA+FLDA Method is given below:

Step 1: Find the nonzero eigenvalues and corresponding eigenvectors of S_T by using a smaller matrix, $A_T^T A_T \in \Re^{MxM}$, where $S_T = A_T A_T^T$. Select the most significant *rw* eigenvectors u_k and form the matrix

$$U = [u_1 \quad u_2 \quad \ldots \quad u_{rw}], \tag{3.21}$$

where *rw* is the rank of S_W and *rw* cannot be bigger than *M-C*.

Step 2: Reduce the dimensionality of the sample space by applying the transformation

$$\widetilde{x}_m^i = U^T x_m^i, \quad i = 1,...,C, \quad m = 1,...,N_i. \tag{3.22}$$

In the transformed space, the new within-class scatter matrix $\widetilde{S}_W \in \Re^{rwxrw}$, and the new between-class scatter matrix $\widetilde{S}_B \in \Re^{rwxrw}$ will be

$$\widetilde{S}_W = U^T S_W U, \tag{3.23}$$

$$\widetilde{S}_B = U^T S_B U. \tag{3.24}$$

Step 3: Find the eigenvectors corresponding to the nonzero eigenvalues of $\widetilde{S}_W^{-1}\widetilde{S}_B$ by applying the simultaneous diagonalization algorithm. Let *V* be the matrix whose columns are the computed eigenvectors. The final projection vector matrix *W* that will then be used for feature extraction is given by

$$W = UV. \tag{3.25}$$

Although this method is computationally feasible, some directions corresponding to the small eigenvalues of S_T are discarded in the PCA step in order to make S_W nonsingular. Thus, applying PCA for dimensionality reduction has the potential to remove dimensions that contain discriminative information [55], [129], [133].

3.7.3 The Direct-LDA Method

The Direct LDA Method has been recently proposed for small sample size problems [133]. This method also reduces the dimensionality of the sample space for solving the singularity problem of S_W. It uses the simultaneous diagonalization method to find the optimal projection vectors in the range space of S_B. The algorithm can be summarized as follows:

Step 1: *Diagonalizing and Whitening of* S_B*:*

(i) Find the nonzero eigenvalues and corresponding eigenvectors of S_B. Let $U = [u_1 \quad u_2 \quad .... \quad u_{rb}]$ be the matrix whose columns are computed eigenvectors corresponding to the nonzero eigenvalues and Λ_B be a diagonal matrix with nonzero eigenvalues. Here *rb* represents the rank of S_B and *rb* cannot be larger than *C*-1.

$$\Lambda_B = U^T S_B U \tag{3.26}$$

(ii) Choose the transformation $Z = U\Lambda_B^{-1/2}$ that whitens S_B. Then,

$$(U\Lambda_B^{-1/2})^T S_B (U\Lambda_B^{-1/2}) = I \Leftrightarrow Z^T S_B Z = I \text{ and } Z^T S_W Z = K, \tag{3.27}$$

where *I* is the identity matrix.

Step 2: *Diagonalizing and Whitening of* S_W*:*

(i) Calculate the eigenvalues and corresponding eigenvectors of *K*. Let *V* be the matrix whose columns are the computed eigenvectors and Λ_K be the diagonal matrix of eigenvalues.

$$V = [v_1 \quad v_2 \quad ... \quad v_{rb}] \tag{3.28}$$

and

$$\Lambda_K = diag(\lambda_1, \lambda_2, ..., \lambda_{rb}), \tag{3.29}$$

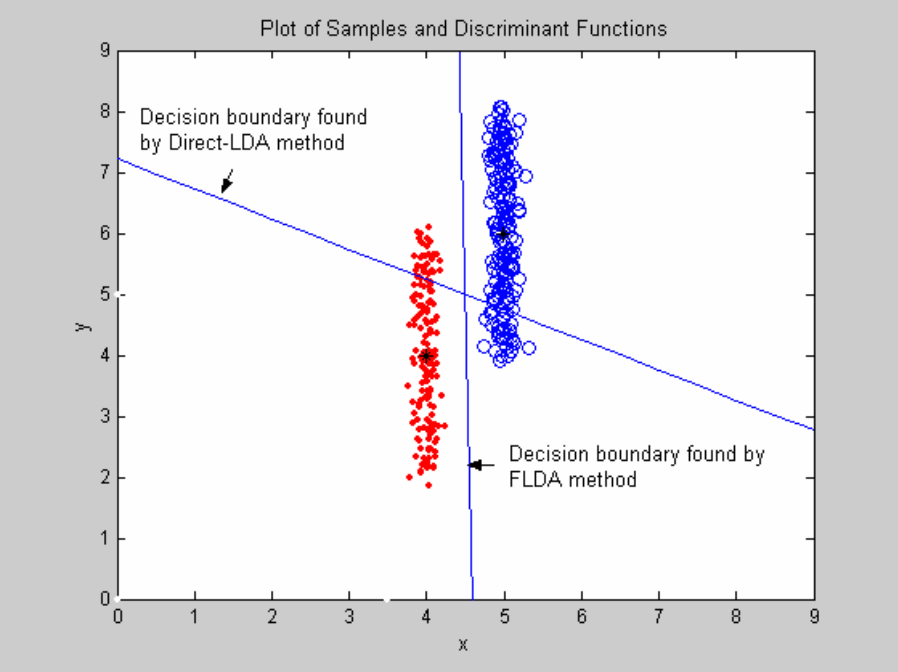
where $V^T Z^T S_W Z V = \Lambda_K$ or $V^T K V = \Lambda_K$.

(ii) Let $Y = ZV$. Then, $Y^T S_W Y = \Lambda_K$ and $Y^T S_B Y = I$

(iii) Final transformation, which spheres the data, will be $W = Y\Lambda_K^{-1/2}$. Then, $W^T S_W W = I$ and $W^T S_B W = \Lambda_K^{-1}$.

However, the range space of S_B is not always a good choice to obtain the optimal projection vectors [22], [55], [129]. This phenomenon can be clearly seen in the following example. In Figure 3.2, we plotted two linearly separable classes with Gaussian distribution having similar covariance matrices. Class means are shown as stars. Since the class distributions are Gaussian and similar, we expect the LDA functions to be optimal in this case. As can be seen in the figure, although the FLDA Method successfully discriminates all samples, the Direct-LDA Method fails for this example. Thus, the optimal projection vectors are not necessarily in the range space of S_B . These two methods produce the same results if the ranks of both S_B and S_W are equal to the dimensionality of the sample. The Direct-LDA Method extracts optimal discriminant features if the within-class scatter is isotropic or the range space $R(S_B)$ of the between-class scatter matrix includes the range space $R(S_W)$ of the within-class scatter matrix (i.e., $R(S_B) \supseteq R(S_W)$ or $R(S_B) = R(S_T)$). However, these conditions are not typically satisfied for the small sample size case. Therefore, the Direct-LDA Method fails to extract optimal projection vectors for feature extraction in most cases.

Furthermore, S_B is whitened as a part of this method. This whitening process can be shown to be redundant and therefore should be skipped. In [79], [80], and [81], the authors claim that the Direct-LDA Method finds the projection vectors in the intersection space of the null space of S_W and the range space of S_B . However, this statement is not correct, and this issue will be explained in the upcoming sections.



3.2: Two different linearly separable classes are plotted. Stars represent class means, and lines represent the decision boundaries found by the Direct-LDA and FLDA methods.

3.8 Linear Discriminant Methods that Use Orthonormal Projection Vectors for Feature Extraction

All methods in this category use orthonormal projection vectors for feature extraction. Since the projection vectors are not necessarily orthogonal with respect to the scatter matrices, the data samples in the transformed space may be correlated.

3.8.1 The Generalized Optimal Discriminant Vector Method

The FLDA Method defines a linear transformation in terms of eigenvectors corresponding to the nonzero eigenvalues of $S_W{}^{-1}S_B$ subject to the orthogonality constraint $w_i^T S_W w_j = \delta_{ij}$. Since the rank of S_B can be at most C-1, we cannot use more than C-1 projection vectors for feature extraction. In some pattern recognition tasks, these projection vectors may not be

sufficient for a good recognition performance. Therefore, an alternative method for the FLDA was proposed by Foley and Sammon [37] for the two class problem to maximize the FLDA criterion, $J_{FLDA}(w_{opt}) = \max \frac{w^T S_B w}{w^T S_W w}$, subject to the constraint $w_i^T w_j = \delta_{ij}$. This method was generalized to the multi-class case by Okada and Tomita and called as the Orthonormal Discriminant Vector (ODV) Method [92]. They proposed an iterative algorithm to obtain the optimal projection vectors successively. The first projection vector is the normalized eigenvector corresponding to the largest eigenvalue of $S_W{}^{-1} S_B$. Thus, the first projection vectors of the ODV Method and the FLDA Method are the same. The second projection vector maximizes the FLDA criterion subject to the orthogonality criterion $w_2^T w_1 = 0$, the third projection vector maximizes the FLDA criterion subject to orthogonality criteria $w_3^T w_1 = 0$ and $w_3^T w_2 = 0$, and so on. The proposed algorithm involves a search for a subspace, taking the inverse of a symmetric positive definite matrix, and eigen-decomposition of a non-symmetric matrix in each iteration. Thus, this algorithm is computationally too expensive. Another drawback of the algorithm is that it is not suitable for the small sample size case. Additionally, Duchene and Leclercq proposed an iterative algorithm for obtaining the orthonormal optimal projection vectors based on the Lagrange's method [31]. However, this algorithm is also computationally too expensive and is not suitable for the small sample size case similar to the ODV Method. Liu *et al*. introduced a new method called the Generalized Optimal Discriminant Vector (GODV) based on the modified FLDA criterion, $J_{MFLDA}(w_i) = \max \frac{w_i^T S_B w_i}{w_i^T S_T w_i}$ subject to the constraint $w_i w_j = \delta_{ij}$ [74]. They showed that the original FLDA criterion can be replaced by the modified FLDA

criterion in the course of solving the discriminant vectors of the optimal set. They modified the algorithm proposed by Okada and Tomita such that it uses the modified FLDA criterion and can be used for the small sample size case. However, this algorithm is also computationally too expensive since it is iterative and includes a search for a subspace, taking the inverse of a symmetric positive definite matrix, and the eigen-decomposition of a non-symmetric matrix in each iteration. The details of the algorithm for the GODV Method can be summarized as follows:

The Algorithm implementing the GODV Method:

Let $N(S_T) = \{x \mid S_T x = 0, x \in \Re^d\}$ (i.e., null space of S_T) and $R(S_T)$ be the complementary subspace (i.e., range of S_T) of $N(S_T)$.

Step 1: *Calculating the first optimal projection vector* w_1*:* Let

$$R(S_T) = span\{\varphi_1^{(1)}, \varphi_2^{(1)}, ..., \varphi_{rt}^{(1)}\} \tag{3.30}$$

where $\varphi_1^{(1)}, \varphi_2^{(1)}, ..., \varphi_{rt}^{(1)}$ are the orthonormal vectors.

Case 1 ($rt = d$): in this case, w_1 is the unit eigenvector corresponding to the largest eigenvalue of $S_T{}^{-1} S_B$.

Case 2 ($1 < rt < d$): Let $P^{(1)} = [\varphi_1^{(1)} \quad \varphi_2^{(1)} \quad ... \quad \varphi_{rt}^{(1)}]$ and $Z^{(1)}$ be the eigenvector corresponding to the largest eigenvalue of $(P^{(1)^T} S_T P^{(1)})^{-1} (P^{(1)^T} S_B P^{(1)})$. Then, w_1 is determined by the following formula

$$w_1 = P^{(1)} Z^{(1)} / \| P^{(1)} Z^{(1)} \|. \tag{3.31}$$

Step 2: *Calculating the j-th discriminant vector* w_j *:*

Let $N(S_T) = span\{\alpha_1, \alpha_2, ..., \alpha_{d-rt}\}$ where the orthonormal vector set $\{\alpha_1, \alpha_2, ..., \alpha_{d-rt}\}$ is orthogonal to the orthonormal vector set $\{\varphi_1^{(1)}, \varphi_2^{(1)}, ..., \varphi_{rt}^{(1)}\}$. Suppose $V^{(j)} = span\{w_1, w_2, ..., w_{j-1}, \alpha_1, ..., \alpha_{d-rt}\}$ is the subspace spanned by the optimal projection vectors $\{w_1, w_2, ..., w_{j-1}\}$ which have been found previously, and the vectors $\{\alpha_1, \alpha_2, ..., \alpha_{d-rt}\}$. Let $\overline{V}^{(j)} = span\{\varphi_1^{(j)}, \varphi_2^{(j)} ..., \varphi_{rt-j+1}^{(j)}\}$ is the complementary subspace of $V^{(j)}$, where $\{\varphi_1^{(j)}, \varphi_2^{(j)} ..., \varphi_{rt-j+1}^{(j)}\}$ are the orthonormal vectors. Let, $P^{(j)} = [\varphi_1^{(j)} \quad \varphi_2^{(j)} \quad ... \quad \varphi_{rt-j+1}^{(j)}]$ and $Z^{(j)}$ be the eigenvector corresponding to the largest eigenvalue of $(P^{(j)^T} S_T P^{(j)})^{-1} (P^{(j)^T} S_B P^{(j)})$. Then w_j is determined by the following formula

$$w_j = P^{(j)} Z^{(j)} / \| P^{(j)} Z^{(j)} \|. \tag{3.32}$$

3.8.2 The Null Space Based Methods

The modified FLDA criterion $J_{MFLDA}(W_{opt}) = \max \dfrac{\left| W^T S_B W \right|}{\left| W^T S_T W \right|}$ attains its maximum, 1 if the projection vectors are chosen from the null space $N(S_W)$ of S_W. However, this criterion is not appropriate since its maximum is not unique for the small sample size case. In particular, every projection vector matrix W such that $W^T S_W W = 0$ and $W^T S_B W \neq 0$ maximizes the modified FLDA criterion. Note that if S_W is singular, which is always the case for the small sample size problem, there are many such projection vector matrices *W*. However, it is not reasonable to use matrices *W* with a small number of projection vectors since they may not

be sufficient for an optimal feature extraction. On the other hand, the following criterion given in [7] and [19] has a unique maximum and it also maximizes the modified FLDA criterion

$$J(W_{opt}) = \max_{|W^T S_W W|=0} |W^T S_B W| = \max_{|W^T S_W W|=0} |W^T S_T W| . \quad (3.33)$$

Therefore, to find the optimal projection vectors w in the null space $N(S_W)$ of S_W, we first project the training set samples onto $N(S_W)$ and then obtain the projection vectors by performing PCA. After this operation, we obtain a set of orthonormal vectors that is a basis for a space which we call the *optimal discriminant subspace*. The optimal discriminant subspace is the intersection of $N(S_W)$ and the range space $R(S_T)$ of the total scatter matrix S_T. The modified FLDA criterion and the criterion given in (3.33) attain their maximum for orthonormal vectors that form a basis for the optimal discriminant subspace. This method was first proposed by Chen *et al*. for face recognition and called the Null Space Method. However, they did not propose an efficient algorithm for applying this method in the original sample space. Instead, the so-called pixel grouping method is applied to extract geometric features and reduce the dimension of the sample space. Then they applied the Null Space Method in this new reduced space. However, it has been observed that the performance of the Null Space Method depends on the dimension of the null space of S_W in the sense that larger dimension provides better performance. Thus, any kind of pre-processing that reduces the original sample space should be avoided [19].

The Optimal Discriminant Subspace Concept

If the dimensionality *d* of the sample space is larger than *M*-1, all scatter matrices will be rank deficient. Thus, if we apply eigen-decomposition to the scatter matrices, we will obtain some eigenvectors corresponding to the zero eigenvalues that span the null spaces of the scatter matrices. As explained previously, if the projection directions are chosen from $N(S_W)$, the modified FLDA criterion attains its maximum, 1. Therefore, we must first project the training set data onto $N(S_W)$. Then, optimal projection vectors can be obtained by applying PCA to the samples, which are projected onto $N(S_W)$. The fact that the optimal projection vectors span the optimal discriminant subspace follows from the following lemma.

Lemma 3.1: Suppose $\overline{U}$ is a matrix whose column vectors u_k ($k = rt+1,...,d$, where rt is the rank of S_T) are orthonormal vectors that span the null space $N(S_T)$ of S_T. If all samples in the training set are projected onto $N(S_T)$, they produce a unique common vector such that

$$x = \overline{U}\,\overline{U}^T x_m^i, \quad i = 1,...,C, \quad m = 1,...,N_i, \tag{3.34}$$

where *x* is independent of indices *i* and *m*.

Proof: By definition, a vector $u \in \Re^d$ is in $N(S_T)$ if $S_T u = 0$. Let μ be the mean vector of the samples in the training set, $1_M \in \Re^{MxM}$ be the matrix with all elements equal to M^{-1}, and $X \in \Re^{dxM}$ be the matrix whose columns are the training set samples. Thus, by multiplying both sides of identity $S_T u = 0$ by u^T, we obtain

$$0 = \sum_{i=1}^{C}\sum_{m=1}^{N_i} u^T (x_m^i - \mu)(x_m^i - \mu)^T u = u^T X(I - 1_M)(I - 1_M)^T X^T u = \| (I - 1_M)X^T u \|^2, \tag{3.35}$$

where $\|.\|$ denotes the Euclidean norm. Thus, (3.35) holds if $(I - 1_M)X^T u_k = 0$, or $X^T u_k = 1_M X^T u_k$. From this relation, it can be seen that

$$(x_m^i)^T u_k = \mu^T u_k, \qquad i = 1,...,C, \quad m = 1,...,N_i, \quad k = rt+1,...,d. \tag{3.36}$$

Thus the projection of any x_m^i onto $N(S_T)$,

$$x = \sum_{k=rt+1}^{d} \langle x_m^i, u_k \rangle u_k = \sum_{k=rt+1}^{d} \langle \mu, u_k \rangle u_k, \quad i = 1,...,C, \quad m = 1,...,N_i, \tag{3.37}$$

is independent of *m* and *i*, which proves the lemma. □

This lemma shows that, $N(S_T)$ does not contain any discriminative information, which can be used in the course of obtaining the optimal projection vectors. Therefore, the null space of S_T can be removed. Then, the remaining subspace for extracting the optimal features of discrimination will be the intersection of $N(S_W)$ and $R(S_T)$.

There are numerous algorithms to find the optimal discriminant subspace and optimal projection vectors that span it. The following observation proposed by Therrien [107] can be used to find optimal projection vectors and the optimal discriminant subspace.

Observation 3.1: Let $L^{(i)}$, $i = 1,...,n$, be a subspace of $\Re^d$. A vector *e* is contained in $\bigcap_{i=1}^{n} L^{(i)}$ if and only if it is an eigenvector of Ψ corresponding to an eigenvalue of 1, where

$$\Psi = \sum_{i=1}^{n} a_i P^{(i)} \tag{3.38}$$

with $P^{(i)}$ being the projection matrix (also called the orthogonal projection operator) of the *i*-th subspace and $0 < a_i < 1$, $\sum_{i=1}^{n} a_i = 1$.

In our case we can choose $L^{(1)}$ and $L^{(2)}$ as $R(S_T)$ and $N(S_W)$, respectively, to find orthonormal vectors that span the optimal discriminant space. However, this approach is not always practical for real applications since the size of projection matrices of subspaces may

be too large (e.g., images of size 256 by 256 yield projection matrices of size 65,536 by 65,536). We will use this observation for the numerical example that will be given later.

There are computationally better ways to find the optimal projection vectors by using smaller sets of basis vectors instead of projection matrices . This is a result of the fact that the projection matrices of $N(S_W)$ and $R(S_T)$ commute, as shown in Theorem 3.1 below, namely $P^{(1)}P^{(2)} = P^{(2)}P^{(1)}$, where $P^{(1)}$ and $P^{(2)}$ represent the projection matrices of $R(S_T)$ and $N(S_W)$, respectively. In this case, the projection matrix of the intersection $N(S_W) \cap R(S_T)$ is found by the equation

$$P_{opt} = P^{(1)}P^{(2)} = P^{(2)}P^{(1)}, \tag{3.39}$$

where P_{opt} is the projection matrix of the optimal discriminant subspace [121]. A consequence of Theorem 3.1 is that to obtain the optimal projection vectors, we can first project the training set samples onto $N(S_W)$ and then apply PCA or, alternatively, we can first project the training set samples onto $R(S_T)$ through PCA, and then find the null space in the transformed space. The DCV Method [19] uses the first approach, whereas the PCA+Null Space Method [55] uses the second approach. All projections are performed economically by using the basis vectors.

Before we prove Theorem 3.1 given below, we need the following auxiliary lemma.

Lemma 3.2: Let $L^{(1)}$, $L^{(2)}$ be subspaces of $\Re^d$, $L^{(1)^\perp}$, $L^{(2)^\perp}$ be their orthogonal complements, and $P^{(1)}$, $P^{(2)}$ be the orthogonal projection matrices onto $L^{(1)}$ and $L^{(2)}$, respectively. If $L^{(1)^\perp} \perp L^{(2)^\perp}$, then $P^{(1)}$ and $P^{(2)}$ commute, that is $P^{(1)}P^{(2)} = P^{(2)}P^{(1)}$.

Proof: If $L^{(1)^\perp} \perp L^{(2)^\perp}$ then clearly $(I - P^{(1)})(I - P^{(2)}) = 0$ and $(I - P^{(2)})(I - P^{(1)}) = 0$. Thus,

$$(I - P^{(1)})(I - P^{(2)}) = (I - P^{(2)})(I - P^{(1)}) = 0, \tag{3.40}$$

$$I - P^{(1)} - P^{(2)} - P^{(1)}P^{(2)} = I - P^{(1)} - P^{(2)} - P^{(2)}P^{(1)}, \tag{3.41}$$

which implies $P^{(1)}P^{(2)} = P^{(2)}P^{(1)}$. □

We are now ready to prove the following theorem:

Theorem 3.1: Let $P^{(1)}$ and $P^{(2)}$ be the projection matrices of the subspaces $R(S_T)$ and $N(S_W)$, respectively. Then $P^{(1)}$ and $P^{(2)}$ commute, i.e., $P^{(1)}P^{(2)} = P^{(2)}P^{(1)}$.

Proof: Let $L^{(1)} = R(S_T)$ and let $L^{(2)} = N(S_W)$. Clearly, $L^{(1)^\perp} = N(S_T)$ and $L^{(2)^\perp} = R(S_W)$. By using Lemma 1 from [13],

$$\begin{aligned} N(S_T) &= N(S_B + S_W) \\ &= N(S_B) \cap N(S_W), \end{aligned} \tag{3.42}$$

and, in particular, $N(S_T) \subset N(S_W)$, which, together with the fact that $N(S_W) \perp R(S_W)$, shows that

$$N(S_T) \perp R(S_W) \text{ or } L^{(1)^\perp} \perp L^{(2)^\perp}. \tag{3.43}$$

The assertion of the theorem now follows from Lemma 3.2. □

In [79], [80], and [81], the authors claim that the Direct-LDA Method finds the projection vectors in the intersection space of $N(S_W)$ and $R(S_B)$. Thus, the projection vectors found by this method should be optimal and equivalent to the ones found by the Null Space Method (equivalently the DCV Method and the PCA+Null Space Method). However this statement is incorrect. In fact, neither the Direct-LDA Method nor the Null Space Method finds the projection vectors in the intersection space of $R(S_B)$ and $N(S_W)$. The projection directions obtained by the Direct-LDA Method come from $R(S_B)$, and the intersection of $R(S_B)$ and $N(S_W)$ is in fact often trivial. Indeed, in all the database examples with the small sample size

case considered in this study, the intersection was trivial. Therefore, the intersection space of $R(S_B)$ and $N(S_W)$ cannot be used for recognition. This fact is also illustrated in Figure 3.3. In Figure 3.3, two classes with the same covariance matrices having two samples each in a 3-dimensional space are plotted. $R(S_W)$ and $R(S_B)$ are shown in the figure. In this example, $R(S_T)$ is the plane spanned by the vectors representing $R(S_W)$ and $R(S_B)$, and $N(S_T)$ is the line perpendicular to this plane. Note that it is also the intersection of $N(S_B)$ and $N(S_W)$. The optimal discriminant subspace, $R(S_T) \cap N(S_W)$, is the line in this plane that is perpendicular to $R(S_W)$. $N(S_W)$ is the plane spanned by the vectors representing $N(S_T)$ and $R(S_T) \cap N(S_W)$. As can be seen in the figure, the intersection of $N(S_W)$ and $R(S_B)$ is the trivial space.

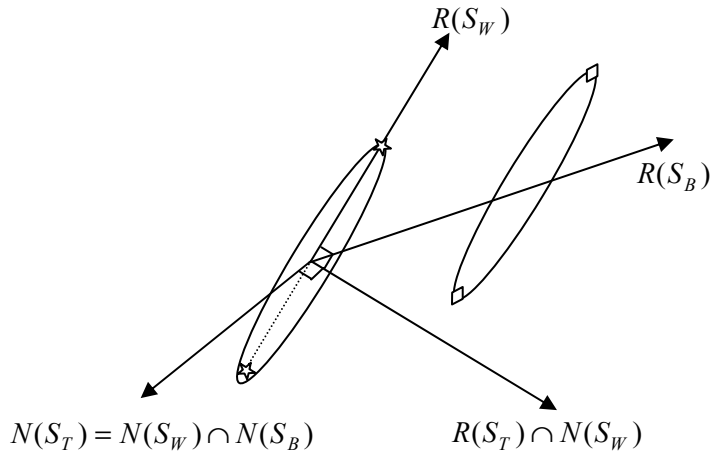


Figure 3.3: Illustration of the optimal discriminant subspace.

The projection vectors found by the Direct-LDA Method and the Null Space Method also differ in terms of orthogonality properties. The projection vectors found by the Direct-LDA Method satisfy the orthogonality property, $w_i^T S_W w_j = \delta_{ij}$, whereas the projection vectors found by the Null Space Method satisfy the property, $w_i w_j = \delta_{ij}$.

The Discriminative Common Vector Method

The Discriminative Common Vector Method suggests the projection of all training set samples onto $N(S_W)$ as the Null Space Method. Then, the final optimal projection vectors, which will be used for feature extraction, are found by applying PCA to the projected samples. However, the DCV Method omits the dimension reduction step of the Null Space Method and therefore it exploits the original high-dimensional space.

The idea of *common vectors* was originally introduced for isolated word recognition problems in the case where the number of samples in each class was less than or equal to the dimensionality of the sample space [44], [45]. These approaches extract the common properties of classes in the training set by eliminating the differences of the samples in each class. A common vector for each individual class is obtained by removing all features that are in the direction of the eigenvectors corresponding to the nonzero eigenvalues of the scatter matrix of its own class. The common vectors are then used for recognition. In our study, instead of using a given class's own scatter matrix, we use the within-class scatter matrix of all classes to obtain the common vectors. We also give an alternative algorithm to obtain the common vectors based on the subspace methods and the Gram-Schmidt orthogonalization procedure. Then, a new set of vectors, called the *discriminative common vectors*, which will

be used for classification, are obtained from the common vectors. We introduce algorithms for obtaining the common vectors and the discriminative common vectors below.

Obtaining the Discriminative Common Vectors by Using the Range Space of the Within-Class Scatter Matrix

To find the optimal projection vectors w in the null space of S_W, we project the training set samples onto the null space of S_W and then obtain the projection vectors by performing PCA. To do so, vectors that span the null space of S_W must first be computed. However, this task is computationally intractable since the dimension of this null space can be very large. A more efficient way to accomplish this task is by using the orthogonal complement of the null space of S_W, which typically is a significantly lower-dimensional space.

Let $\Re^d$ be the original sample space, $R(S_W)$ be the range space of S_W, and $N(S_W)$ be the null space of S_W. Equivalently,

$$R(S_W) = span\{\alpha_k \mid S_W\alpha_k \neq 0, \qquad k = 1,...,rw\} \tag{3.44}$$

and

$$N(S_W) = span\{\alpha_k \mid S_W\alpha_k = 0, \qquad k = rw+1,...,d\}, \tag{3.45}$$

where $rw < d$ is the rank of S_W, $\{\alpha_1,....,\alpha_d\}$ is an orthonormal set, and $\{\alpha_1,....,\alpha_{rw}\}$ is the set of orthonormal eigenvectors corresponding to the nonzero eigenvalues of S_W.

Consider the matrices $Q = [\alpha_1 \quad .... \quad \alpha_{rw}]$ and $\overline{Q} = [\alpha_{rw+1} \quad .... \quad \alpha_d]$. Since $\Re^d = R(S_W) \oplus N(S_W)$, every sample $x_m^i \in \Re^d$ has a unique decomposition of the form

$$x_m^i = y_m^i + z_m^i, \tag{3.46}$$

where $y_m^i = Px_m^i = QQ^T x_m^i \in R(S_W)$, $z_m^i = \overline{P}x_m^i = \overline{Q}\,\overline{Q}^T x_m^i \in N(S_W)$, and P and $\overline{P}$ are the orthogonal projection operators onto $R(S_W)$ and $N(S_W)$, respectively. Our goal is to compute

$$z_m^i = x_m^i - y_m^i = x_m^i - Px_m^i. \tag{3.47}$$

To do this, we need to find a basis for $R(S_W)$, which can be accomplished by an eigen-analysis of S_W. In particular, the normalized eigenvectors α_k corresponding to the nonzero eigenvalues of S_W will be an orthonormal basis for $R(S_W)$. The eigenvectors can be obtained by calculating the eigenvectors of the smaller *M* by *M* matrix, $A_W^T A_W$, defined such that $S_W = A_W A_W^T$, where A_W is a *d* by *M* matrix given in (3.4). Let λ_k and v_k be the *k*-th nonzero eigenvalue and the corresponding eigenvector of $A_W^T A_W$, where $k \le M - C$. Then $\alpha_k = A_W v_k$ will be the eigenvector that corresponds to the *k*-th nonzero eigenvalue of S_W. The sought-for projection onto $N(S_W)$ is achieved by using (3.47). After this process, we obtain the same vector for all samples of the same class,

$$x_{com}^i = x_m^i - QQ^T x_m^i = \overline{Q}\,\overline{Q}^T x_m^i, \quad m=1,\ldots,N, \quad i=1,...,C, \tag{3.48}$$

i.e., the vector on the right-hand side of (3.48) is independent of the sample index *m*. We refer to the vectors x_{com}^i as common vectors. The above fact is proved in the following theorem.

Theorem 3.2: Suppose $\overline{Q}$ is a matrix whose column vectors are the orthonormal vectors that span the null space $N(S_W)$ of S_W. Then, the projections of the samples x_m^i of the class *i* onto $N(S_W)$ produce a unique common vector x_{com}^i such that

$$x_{com}^{i} = \overline{Q}\,\overline{Q}^{T} x_{m}^{i}, \quad m=1,\ldots,N, \quad i=1,\ldots,C. \tag{3.49}$$

Proof: By definition, a vector $\alpha \in \Re^{d}$ is in $N(S_W)$ if $S_W \alpha = 0$. Let μ_i be the mean vector of the *i*-th class, G be the *N* by *N* matrix whose entries are all N^{-1}, and X^i be the *d* by *N* matrix whose *m*-th column is the sample x_m^i. Thus, multiplying both sides of identity $S_W \alpha = 0$ by α^T and writing

$$S_W = \sum_{i=1}^{C} S_i \,, \tag{3.50}$$

with

$$S_i = \sum_{m=1}^{N} (x_m^i - \mu_i)(x_m^i - \mu_i)^T = (X^i - X^i G)(X^i - X^i G)^T \,, \tag{3.51}$$

immediately leads to

$$0 = \sum_{i=1}^{C} \alpha^T X^i (I-G)(I-G)^T (X^i)^T \alpha = \sum_{i=1}^{C} \| (I-G)(X^i)^T \alpha \|^2 \,, \tag{3.52}$$

where ||.|| denotes the Euclidean norm. Thus, (3.52) holds if $(I-G)(X^i)^T \alpha_k = 0$, or $(X^i)^T \alpha_k = G(X^i)^T \alpha_k$. From this relation we can see that,

$$(x_m^i)^T \alpha_k = (\mu_i)^T \alpha_k \,, \quad m = 1,...,N, \quad i = 1,...,C, \quad k = rw+1,...,d \,. \tag{3.53}$$

Thus, the projection of x_m^i onto $N(S_W)$,

$$x_{com}^{i} = \sum_{k=rw+1}^{d} \langle x_m^i, \alpha_k \rangle \alpha_k = \sum_{k=rw+1}^{d} \langle \mu_i, \alpha_k \rangle \alpha_k \,, \tag{3.54}$$

is independent of *m*, which proves the theorem. □

The theorem states that it is enough to project a single sample from each class. This will greatly reduce the computational burden of the calculations. After obtaining the common

vectors x_{com}^{i}, optimal projection vectors will be those that maximize the total scatter of the common vectors,

$$J(W_{opt}) = \arg\max_{|W^T S_W W|=0} |W^T S_B W| = \arg\max_{|W^T S_W W|=0} |W^T S_T W| = \arg\max |W^T S_{com} W|, \quad (3.55)$$

where W is a matrix whose columns are the orthonormal optimal projection vectors w_k, and S_{com} is the scatter matrix of the common vectors,

$$S_{com} = \sum_{i=1}^{C} (x_{com}^{i} - \mu_{com})(x_{com}^{i} - \mu_{com})^T, \quad i = 1,...,C, \quad (3.56)$$

where μ_{com} is the mean of all common vectors, $\mu_{com} = \frac{1}{C}\sum_{i=1}^{C} x_{com}^{i}$.

In this case optimal projection vectors w_k can be found by an eigen-analysis of S_{com}. In particular, all eigenvectors corresponding to the nonzero eigenvalues of S_{com} will be the optimal projection vectors for feature extraction. S_{com} is typically a large d by d matrix and thus we can use the smaller matrix, $A_{com}^{T} A_{com}$, of size C by C, to find nonzero eigenvalues and the corresponding eigenvectors of $S_{com} = A_{com} A_{com}^{T}$, where A_{com} is the d by C matrix of the form

$$A_{com} = [x_{com}^{1} - \mu_{com} \quad .... \quad x_{com}^{C} - \mu_{com}]. \quad (3.57)$$

There will be C-1 optimal projection vectors since the rank of S_{com} is C-1 if all common vectors are linearly independent. If two common vectors are identical, then the two classes which are represented by this vector cannot be distinguished. Since the optimal projection vectors w_k belong to the null space of S_W, it follows that when the image samples x_m^i of the i-th class are projected onto the linear span of the projection vectors w_k, the feature vector

$\Omega_i = [< x_m^i, w_1 > \quad .... \quad < x_m^i, w_{C-1} >]^T$ of the projection coefficients $< x_m^i, w_k >$ will also be independent of the sample index *m*. Thus, we have

$$\Omega_i = W^T x_m^i, \quad m = 1,..., N_i, \quad i = 1,..., C. \tag{3.58}$$

We call the feature vectors Ω_i discriminative common vectors, and they will be used for classification of samples. Note that 100% recognition accuracy with respect to the training set data is achieved if the discriminative common vectors are distinct among themselves. To recognize a test image x_{test}, the feature vector of this test image is found by

$$\Omega_{test} = W^T x_{test}, \tag{3.59}$$

which is then compared with the discriminative common vector Ω_i of each class using the Euclidean distance. The discriminative common vector found to be the closest to Ω_{test} is used to identify the test image.

Since Ω_{test} is compared only to a single vector for each class, the recognition is very efficient for real-time recognition tasks.

The above method can be summarized as follows:

Step 1: Compute the nonzero eigenvalues and corresponding eigenvectors of S_W by using the matrix $A_W^T A_W$, where $S_W = A_W A_W^T$ and A_W is given by (3.4). Set $Q = [\alpha_1 \quad .... \quad \alpha_{rw}]$, where *rw* is the rank of S_W.

Step 2: Choose any sample from each class and project it onto the null space of S_W to obtain the common vectors

$$x_{com}^i = x_m^i - QQ^T x_m^i, \quad m = 1,..., N, \quad i = 1,..., C. \tag{3.60}$$

Step 3: Compute the eigenvectors w_k of S_{com}, corresponding to the nonzero eigenvalues, by using the matrix $A_{com}^T A_{com}$, where $S_{com} = A_{com} A_{com}^T$ and A_{com} is given in (3.57). There are at most *C*-1 eigenvectors that correspond to the nonzero eigenvalues. Use these eigenvectors to form the projection matrix $W = [w_1 \quad .... \quad w_{C-1}]$, which will be used to obtain feature vectors in (3.58) and (3.59).

Note that the training set samples will be uncorrelated in the feature space since the new total scatter matrix in the transformed space is

$$\widetilde{S}_T = \widetilde{S}_B = W^T S_T W = W^T S_B W = \Lambda, \tag{3.61}$$

where Λ is the diagonal matrix of nonzero eigenvalues of S_{com}.

Distinctness of Discriminative Common Vectors

If all samples in each class are projected onto the null space $N(S_W)$ of S_W, they give rise to a unique vector called common vector as in (3.49). A natural question arises whether the common vectors x_{com}^i, $i = 1,...,C$ are distinct or not i.e., whether each of these vectors can be uniquely associated with the *i*-th class. Or, put yet another way, whether there is one-to-one correspondence between the common vectors and the classes. For if this is not the case, i.e., if $x_{com}^i = x_{com}^j$, for some $i \neq j$, then the DCV method would not be able to discriminate between the two classes *i* and *j*, which would render this method less useful.

The next result shows that this situation is in practice very unlikely, even though it is possible in theory. First we state the following necessary condition for the common vectors to be distinct.

Observation 3.2: Let $i \neq j$. For the common vectors x_{com}^{i}, x_{com}^{j} to be distinct it is necessary to that the samples x_m^i, x_n^j in the corresponding two classes are mutually independent in the sense that one cannot find real numbers α_m, β_n satisfying $\sum_{m=1}^{N_i} \alpha_m = 1$, $\sum_{n=1}^{N_j} \beta_n = 1$ and such that

$$\sum_{m=1}^{N_i} \alpha_m x_m^i = \sum_{n=1}^{N_j} \beta_m x_n^j . \tag{3.62}$$

To explain this, let us first reformulate the above observation. To this end, recall that the *affine hull*, $\mathrm{aff}(A)$, of a finite set $A \in \Re^k$ is the set (called an affine space)

$$\mathrm{aff}(A) := \{ \sum_{a \in A} \lambda_a a, \lambda_a \in \Re \} . \tag{3.63}$$

Thus, the above observation can be rephrased by saying that a necessary condition for the common vectors of classes *i* and *j* to be distinct is that

$$A_i \cap A_j = \emptyset, \tag{3.64}$$

where A_i, A_j are the affine hulls of the vectors in the *i*-th and *j*-th classes, respectively. We already know that the common vectors x_{com}^{i} can be obtained by projecting any $x \in A_i$ onto $N(S_W)$ (for example, *x* can be chosen to be μ_i). Recall that $N(S_W) = \bigcap_{i=1}^{C} N(S_i)$, where S_i represents the scatter matrix of the *i*-th class. Let us denote the orthogonal projection operator of $N(S_W)$ by $\overline{P}$. With this notation, we have

$$x_{com}^{i} = \overline{P} x , \tag{3.65}$$

whenever $x \in A_i$, and

$$x_{com}^{j} = \overline{P} x , \tag{3.66}$$

for $x \in A_j$. Thus, if $A_i \cap A_j \neq \emptyset$, then clearly $x_{com}^i = x_{com}^j$ since above one can take $x \in A_i \cap A_j$, which would give $x_{com}^i = \overline{P}x = x_{com}^j$.

Unfortunately, the above observation does not constitute a sufficient condition for the common vectors to be distinct. This can be easily seen by taking classes of vectors satisfying $A_i \cap A_j = \emptyset$, for all $i \neq j$, but such that $N(S_W) = \{0\}$, in which case all common vectors will be trivial vectors. To arrive at a sufficient condition, it will therefore be necessary to impose a condition on linear separability of the considered classes.

For the purpose of the following result, we will say that the given classes $i = 1,...,C$ are *linearly separable* if for each pair $i \neq j$ there exists a hyperplane $H \in \Re^d$ strictly separating the affine spaces A_i and A_j such that $A_k \cap H = \emptyset$, for all $k \neq i, j$. As usual, A_i and A_j are said to be strictly separated by H if A_i and A_j are one the opposite sides of H and if $A_i \cap H = A_j \cap H = \emptyset$. Thus, this concept of separability is stronger than the usual "one-against-one" separability, but weaker than the "one-against-all" separability. As is well known, the above definition is equivalent to saying that there exists a linear functional φ on $\Re^d$ such that $\varphi A_i < \varphi H < \varphi A_j$ and $\varphi A_k \neq \varphi H$, $k \neq i, j$.

We are now ready to prove the following sufficient condition for existence of distinct common vectors.

Theorem 3.3: Suppose the classes $i = 1,...,C$ are linearly separable. Then, the corresponding common vectors are distinct.

Proof: We will show that for any pair $i \neq j$, we have $x_{com}^i \neq x_{com}^j$. To this end, let φ be the linear functional whose existence is guaranteed by the definition of separability. Let l be the

unique one-dimensional subspace of $\Re^d$ such that $l \perp H$ and let P_l the orthogonal projection operator onto this subspace. Clearly, also $l \perp A_i, A_j$. We have

$$\varphi P_l A_i = \varphi A_i < \varphi H < \varphi A_j = \varphi P_l A_j, \tag{3.67}$$

or in particular, $P_l A_i \neq P_l A_j$. Also not that l is a subspace of every S_i, $i = 1,...,C$, which is a direct consequence of the fact that $A_i \parallel H$, for all i. Consequently, $l \subset \bigcap_{i=1}^{C} N(S_i) = N(S_W)$. Combining this with the fact that $P_l A_i \neq P_l A_j$, we thus obtain

$$x_{com}^i = P_l A_i \neq P_l A_j = x_{com}^j \tag{3.68}$$

(since clearly, if the orthogonal projection onto a subspace l are distinct, then so are the projections onto a larger space $N(S_W)$). □

Note that if there are only two classes with corresponding affine hulls A_1 and A_2, then linear separability is equivalent to the condition $A_1 \cap A_2 = \emptyset$, which is a simple consequence of the Hahn-Banach theorem. Thus, for two classes, the above necessary condition is also sufficient in this case.

Corollary 3.1: If all samples x_m^i, $i = 1,...,C, m = 1,...,N_i$, are linearly independent then the common vectors x_{com}^i, $i = 1,...,C$, are distinct.

If the common vectors are distinct, then clearly so are the discriminative common vectors. The sufficient conditions of the discriminative common vectors being distinct are typically satisfied for the data sets in high-dimensional sample spaces. For instance, for a typical face recognition problem with 256-level gray scale face images of size 128x128, the volume of the sample space is $(16384)^{256}$. Since the dimension is so high, it is very likely that the training set samples will be linearly independent, and therefore the DCV method can be

applied safely for pattern recognition. It has been reported that the generalization performance of the DCV method is superior to competing methods for high-dimensional pattern classification tasks. In fact, the generalization performance is related to the dimensionality of $N(S_W)$ in the sense that the higher dimensions yield better results [13].

In some cases, the dimensionality of the sample space may not be large enough to make sure that the discriminative common vectors are distinct. There are three basic approaches to cope with this situation. First, we can discard all dependent samples. A second solution is to add new orthonormal projection vectors, maximizing the between-class scatter, to the projection vectors spanning the optimal discriminant subspace from outside the optimal discriminant subspace. In this case, since the new projection vectors will be from the range space $R(S_W)$ of the within-class scatter matrix of the training samples, the feature vectors will not yield the same discriminative common vectors anymore. As a result, 100% recognition accuracy is no longer guaranteed since some training samples may be misclassified in this case. A third solution would be to map the training samples into a higher-dimensional space where the new discriminative common vectors of classes are unique, as in the Kernel DCV method introduced in the next chapter.

Obtaining the Discriminative Common Vectors by Using Difference Subspaces and the Gram-Schmidt Orthogonalization Procedure

To find an orthonormal basis for the range of S_W, the algorithm, utilizing the eigenvectors of S_W, uses the eigenvectors corresponding to the nonzero eigenvalues of the M by M matrix $A_W^T A_W$, where $S_W = A_W A_W^T$. Assuming that $rank(S_W) = M - C$, then $l(\frac{4M^3}{3} + 2M^3 - M^2) + 2dM(M-C) + dC$ floating point operations (flops) are required to

obtain an orthonormal basis set spanning the range of S_W by using this approach. Here l represents the number of iterations required for convergence of the eigen-decomposition algorithm. However, the computations may become expensive and numerically unstable for large values of *M*. Since we do not need to find the eigenvalues (i.e., an explicit symmetric Schur decomposition) of S_W, the following algorithm can be used for finding the common vectors efficiently. It requires only $(2d(M-C)^2 + d(M-C))$ flops to find an orthonormal basis for the range of S_W and is based on the subspace methods and the Gram-Schmidt orthogonalization procedure.

Suppose that *d*>*M*-*C*. In this case, the subspace methods can be applied to obtain the common vectors x_{com}^i for each class $\omega^{(i)}$. To do this, we choose any one of the sample vectors from the *i*-th class as the subtrahend vector and then obtain the difference vectors b_j^i of the so-called difference subspace of the *i*-th class [45]. Thus, assuming that the first sample of each class is taken as the subtrahend vector, we have $b_j^i = x_{j+1}^i - x_1^i$, $j = 1,..., N_i - 1$.

The difference subspace B_i of the *i*-th class is defined as $B_i = span\{b_1^i,....,b_{N_i-1}^i\}$. These subspaces can be summed up to form the complete difference subspace

$$B = B_1 + ... + B_C = span\{b_1^1,....,b_{N_1-1}^1, b_1^2,....,b_{N_C-1}^C\}. \quad (3.69)$$

The number of independent difference vectors b_j^i will be equal to the rank of S_W. For simplicity, suppose there are *M*-*C* independent difference vectors. Since by Theorem 3.5 given below, *B* and the range space $R(S_W)$ of S_W, are the same spaces, the projection matrix

onto B is the same as the matrix P (projection matrix onto the range space of S_W) defined previously. This matrix can be computed as

$$P = D(D^T D)^{-1} D^T , \tag{3.70}$$

where $D = [b_1^1 \quad .... \quad b_{N_1-1}^1 \quad b_1^2 \quad .... \quad b_{N_C-1}^C]$ is a d by M-C matrix [90]. This involves finding the inverse of an M-C by M-C nonsingular, positive definite symmetric matrix $D^T D$. A computationally efficient method of applying the projection uses an orthonormal basis for B. In particular, the difference vectors b_j^i can be orthonormalized by using the Gram-Schmidt orthogonalization procedure to obtain orthonormal basis vectors $\beta_1,....,\beta_{M-C}$. The complement of B is the indifference subspace $B^\perp$ such that

$$U = [\beta_1 \quad .... \quad \beta_{M-C}], \quad P = UU^T , \tag{3.71}$$

$$\overline{U} = [\beta_{M-C+1} \quad .... \quad \beta_d], \quad \overline{P} = \overline{U}\,\overline{U}^T , \tag{3.72}$$

where P and $\overline{P}$ are the orthogonal projection operators onto B and $B^\perp$, respectively. Thus matrices P and $\overline{P}$ are symmetric and idempotent, and satisfy $P + \overline{P} = I$. Any sample from each class can now be projected onto the indifference subspace $B^\perp$ to obtain the corresponding common vectors of the classes,

$$\begin{aligned} x_{com}^i &= \overline{P} x_m^i = x_m^i - P x_m^i \\ &= \overline{U}\,\overline{U}^T x_m^i = x_m^i - UU^T x_m^i , \qquad m = 1,...,N, \quad i = 1,...,C. \end{aligned} \tag{3.73}$$

The common vectors do not depend on the choice of the subtrahend vectors, and they are identical to the common vectors obtained by using the null space of S_W. This follows from Theorem 3.5 below, which uses the results of Lemma 3.3 and Theorem 3.4.

Theorem 3.4: Let $N(S_i)$ be the null space of the scatter matrix S_i, and $B_i^{\perp}$ be the orthogonal complement of the difference subspace B_i. Then $N(S_i) = B_i^{\perp}$ and $R(S_i) = B_i$.

Proof: See [45].

Lemma 3.3: Suppose that $S_1,....,S_C$ are positive semi-definite scatter matrices. Then

$$N(S_1 + ... + S_C) = \bigcap_{i=1}^{C} N(S_i), \tag{3.74}$$

where $N(.)$ denotes the null space.

Proof: The null space on the left-hand side of the above identity contains elements α such that

$$(S_1 + ... + S_C)\alpha = 0 \tag{3.75}$$

or

$$\alpha^T (S_1 + ... + S_C)\alpha = \alpha^T S_1 \alpha + ... + \alpha^T S_C \alpha = 0, \tag{3.76}$$

by the positive semi-definiteness of $S_1 + ... + S_C$. Thus, again by the positive semi-definiteness, $\alpha \in N(S_1 + ... + S_C)$ if and only if

$$\alpha^T S_i \alpha = 0, \quad i=1,...,C, \tag{3.77}$$

or, equivalently, $\alpha \in \bigcap_{i=1}^{C} N(S_i)$. □

Theorem 3.5: Let $S_1,....,S_C$ be positive semi-definite scatter matrices. Then

$$B = R(S_W) = R(S_1 + ... + S_C) = R(S_1) + ... + R(S_C) = B_1 + ... + B_C, \tag{3.78}$$

where $R(.)$ denotes the range.

Proof: Since it is well known that the null space and the range of a matrix are complementary spaces, using the previous Lemma 3.3, we have

$$R(S_1 + ... + S_C) = (N(S_1 + ... + S_C))^{\perp} = (\bigcap_{i=1}^{C} N(S_i))^{\perp} = (N(S_1))^{\perp} + ... + (N(S_C))^{\perp}$$
$$= R(S_1) + ... + R(S_C) = B_1 + ... + B_C, \tag{3.79}$$

where the last equality is a consequence of Theorem 3.4. □

After calculating the common vectors, the optimal projection vectors can be found by performing PCA as described previously. The eigenvectors corresponding to the nonzero eigenvalues of S_{com} will be the optimal projection vectors. However, optimal projection vectors can also be obtained more efficiently by computing the basis of the difference subspace B_{com} of the common vectors, since we are only interested in finding an orthonormal basis for the range of S_{com}.

The algorithm based on the Gram-Schmidt orthogonalization can be summarized as follows.

Step 1: Find the linearly independent vectors b_j^i that span the difference subspace *B* and set $B = span\{b_1^1,....,b_{N_1-1}^1, b_1^2,....,b_{N_C-1}^C\}$. There are a total of *rw* linearly independent vectors, where *rw* is at most *M*-*C*.

Step 2: Apply the Gram-Schmidt orthogonalization procedure to obtain an orthonormal basis $\beta_1,....,\beta_{rw}$ for *B* and set $U = [\beta_1 \quad .... \quad \beta_{rw}]$.

Step 3: Choose any sample from each class and project it onto *B* to obtain common vectors by using (3.73).

Step 4: Find the difference vectors that span B_{com} as

$$b_{com}^j = x_{com}^{j+1} - x_{com}^1, \quad j = 1,...,C-1, \tag{3.80}$$

and apply the Gram-Schmidt orthogonalization to obtain an orthonormal basis $\widetilde{w}_1,...,\widetilde{w}_{C-1}$ for B_{com}. These vectors will be the optimal projection vectors to be used to form the projection matrix $\widetilde{W} = [\widetilde{w}_1 \quad .... \quad \widetilde{w}_{C-1}]$, which will in turn be used to obtain feature vectors in (3.58) and (3.59). Note that columns of $\widetilde{W}$ and columns of the projection matrix W (described in the previous subsection) span the same space, and hence the matrices obey the equation $WW^T = \widetilde{W}\widetilde{W}^T$. However, the training set samples are not necessarily uncorrelated in the transformed feature space since the new total scatter matrix $\widetilde{S}_T = \widetilde{S}_B = \widetilde{W}^T S_T \widetilde{W}$ may not be a diagonal matrix.

The PCA+Null Space Method

In this method, in order to obtain the optimal projection vectors, the training set samples are first projected onto the range space of S_T through PCA, and then the vectors that span the null space of the new within-class scatter matrix in the transformed space are computed. The algorithm is given below:

Step 1: Compute the nonzero eigenvalues and corresponding eigenvectors u_k of S_T by using the matrix $A_T{}^T A_T \in \Re^{MxM}$, where $S_T = A_T A_T{}^T$ and A_T is given by (3.6). Set $U = [u_1 \quad ... \quad u_{rt}]$, where rt is the rank of S_T. Then transform the training set samples by the equation, $U^T x_m^i$. Compute the new within-class scatter matrix in the transformed space by,

$$\widetilde{S}_W = U^T S_W U \,. \tag{3.81}$$

Step 2: Find the orthonormal vectors set that span the null space of $\widetilde{S}_W$. This can be done through an eigen-decomposition of $\widetilde{S}_W$. The eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_W$ span the null space of $\widetilde{S}_W$. Let V be the matrix whose columns are the computed eigenvectors such that $V^T \widetilde{S}_W V = 0$. In the transformed space the new scatter matrices will be

$$\hat{S}_T = (UV)^T S_T UV = \hat{S}_B = (UV)^T S_B UV \tag{3.82}$$

Step 3 (optional): Remove the null space of $\hat{S}_T$ if it exists and rotate the projection directions so that the new total and between-scatter matrices are diagonal (i.e., the scatter matrices of the feature vectors of the training set samples are uncorrelated). That is,

$$\hat{S}_T = L \Lambda L^T . \tag{3.83}$$

Then the final projection matrix $\hat{W}$ will be

$$\hat{W} = UVL . \tag{3.84}$$

The optimal projection vector matrix $\hat{W}$ obtained by the PCA+Null Space Method and the optimal projection vector matrix W obtained by the DCV Method are the same if Step 3 is carried out. If Step 3 is not used (i.e., $\widetilde{W} = UV$), then columns of $\widetilde{W}$ and columns of the projection matrix W span the same space and hence the matrices obey the equation $\hat{W}\hat{W}^T = \widetilde{W}\widetilde{W}^T = WW^T$.

Numerical Example

In this subsection we present a numerical example to show techniques to compute the optimal projection vectors from the optimal discriminant subspace. The samples of each class

given below are randomly chosen from the Gaussian distributions with different means and same identity covariance matrix. Let

$$x_1^1 = [0.7310 \quad 0.0403 \quad 0.5689 \quad -0.3775 \quad -1.4751 \quad 0.7812]^T,$$

$$x_2^1 = [0.5779 \quad 0.6771 \quad -0.2556 \quad -0.2959 \quad -0.2340 \quad -0.2656]^T;$$

$$x_1^2 = [2.1184 \quad 3.4435 \quad 2.6232 \quad 2.9409 \quad 2.2120 \quad 2.5690]^T,$$

$$x_2^2 = [2.3148 \quad 1.6490 \quad 2.7990 \quad 1.0079 \quad 2.2379 \quad 0.8122]^T;$$

$$x_1^3 = [-3.0078 \quad -0.9177 \quad -1.6101 \quad -2.6355 \quad -1.5563 \quad -2.8217]^T,$$

$$x_2^3 = [-2.7420 \quad -2.1315 \quad -1.9120 \quad -2.5596 \quad -2.9499 \quad -1.0137]^T.$$

Thus there are $C = 3$ classes, each of which contains 2 samples in a 6-dimensional sample space. The within-class scatter matrix is

$$S_W = S_1 + S_2 + S_3 = \begin{bmatrix} 0.0663 & -0.3863 & 0.0403 & -0.1860 & -0.2777 & 0.1479 \\ -0.3863 & 2.5495 & -0.2370 & 1.7143 & 1.2177 & 0.1457 \\ 0.0403 & -0.2370 & 0.4009 & -0.2150 & -0.2990 & 0.0042 \\ -0.1860 & 1.7143 & -0.2150 & 1.8745 & -0.0273 & 1.7239 \\ -0.2777 & 1.2177 & -0.2990 & -0.0273 & 1.7416 & -1.9322 \\ 0.1479 & 0.1457 & 0.0042 & 1.7239 & -1.9322 & 3.7255 \end{bmatrix}.$$

The eigenvalues and corresponding eigenvectors of S_W are

$$\lambda_1 = 5.5764,\ \alpha_1 = [0.0152 \quad 0.1408 \quad -0.0030 \quad 0.4428 \quad -0.3656 \quad 0.8064]^T;$$

$$\lambda_2 = 4.3672,\ \alpha_2 = [0.1215 \quad -0.7426 \quad 0.1043 \quad -0.4227 \quad -0.4721 \quad 0.1459]^T;$$

$$\lambda_3 = 0.4147,\ \alpha_3 = [0.0387 \quad -0.2703 \quad -0.9231 \quad 0.0397 \quad 0.2352 \quad 0.1279]^T;$$

$$\lambda_4 = 0,\ \alpha_4 = [0.9099 \quad 0.0229 \quad -0.0283 \quad 0.2937 \quad -0.1483 \quad -0.2498]^T;$$

$$\lambda_5 = 0,\ \alpha_5 = [0.0630 \quad -0.5236 \quad 0.3629 \quad 0.3660 \quad 0.6496 \quad 0.1851]^T;$$

$$\lambda_6 = 0,\ \alpha_6 = [0.3893 \quad 0.2843 \quad 0.0668 \quad -0.6351 \quad 0.3798 \quad 0.4642]^T.$$

If we project samples onto $N(S_W)$, we obtain the same unique vector for all samples of the same class. We call these vectors common vectors. The common vectors of the classes are

$$\begin{aligned} x_{com}^1 &= <x_1^1,\alpha_4>\alpha_4 + <x_1^1,\alpha_5>\alpha_5 + <x_1^1,\alpha_6>\alpha_6 = x_1^1 - <x_1^1,\alpha_1>\alpha_1 - <x_1^1,\alpha_2>\alpha_2 - <x_1^1,\alpha_3>\alpha_3 \\ &= <x_2^1,\alpha_4>\alpha_4 + <x_2^1,\alpha_5>\alpha_5 + <x_2^1,\alpha_6>\alpha_6 = x_2^1 - <x_2^1,\alpha_1>\alpha_1 - <x_2^1,\alpha_2>\alpha_2 - <x_2^1,\alpha_3>\alpha_3 \\ &= [0.6131 \quad 0.4971 \quad -0.2522 \quad -0.3373 \quad -0.4085 \quad -0.0993]^T, \end{aligned}$$

$$\begin{aligned} x_{com}^2 &= <x_1^2,\alpha_4>\alpha_4 + <x_1^2,\alpha_5>\alpha_5 + <x_1^2,\alpha_6>\alpha_6 = x_1^2 - <x_1^2,\alpha_1>\alpha_1 - <x_1^2,\alpha_2>\alpha_2 - <x_1^2,\alpha_3>\alpha_3 \\ &= <x_2^2,\alpha_4>\alpha_4 + <x_2^2,\alpha_5>\alpha_5 + <x_2^2,\alpha_6>\alpha_6 = x_2^2 - <x_2^2,\alpha_1>\alpha_1 - <x_2^2,\alpha_2>\alpha_2 - <x_2^2,\alpha_3>\alpha_3 \\ &= [2.6391 \quad -0.5379 \quad 0.9154 \quad 0.0059 \quad 2.0184 \quad 0.9593]^T, \end{aligned}$$

and

$$\begin{aligned} x_{com}^3 &= <x_1^3,\alpha_4>\alpha_4 + <x_1^3,\alpha_5>\alpha_5 + <x_1^3,\alpha_6>\alpha_6 = x_1^3 - <x_1^3,\alpha_1>\alpha_1 - <x_1^3,\alpha_2>\alpha_2 - <x_1^3,\alpha_3>\alpha_3 \\ &= <x_2^3,\alpha_4>\alpha_4 + <x_2^3,\alpha_5>\alpha_5 + <x_2^3,\alpha_6>\alpha_6 = x_2^3 - <x_2^3,\alpha_1>\alpha_1 - <x_2^3,\alpha_2>\alpha_2 - <x_2^3,\alpha_3>\alpha_3 \\ &= [-3.1844 \quad 0.9010 \quad -1.0584 \quad -0.6488 \quad -2.1055 \quad -0.6994]^T. \end{aligned}$$

The optimal projection vectors are those that maximize the scatter across the common vectors. In other words, the optimal projection vectors are the eigenvectors corresponding to the nonzero eigenvalues of S_{com}, where $S_{com} = \sum_{i=1}^{3}(x_{com}^i - \mu_{com})(x_{com}^i - \mu_{com})^T$ and $\mu_{com} = \sum_{i=1}^{3} x_{com}^i / 3$ The nonzero eigenvalues and the corresponding eigenvectors of S_{com} are

$\lambda_1 = 30.1010$, $w_1 = [0.7560 \quad -0.1830 \quad 0.2528 \quad 0.0842 \quad 0.5283 \quad 0.2119]^{\mathrm{T}}$;

$\lambda_2 = 0.6670$, $w_2 = [-0.6418 \quad -0.3751 \quad 0.2628 \quad 0.0432 \quad 0.5364 \quad 0.2981]^{\mathrm{T}}$.

The projection matrix of the subspace spanned by the optimal projection vectors is

$$P_{opt} = [w_1 \quad w_2][w_1 \quad w_2]^T = \begin{bmatrix} 0.9834 & 0.1024 & 0.0225 & 0.0359 & 0.0551 & -0.0311 \\ 0.1024 & 0.1741 & -0.1448 & -0.0316 & -0.2978 & -0.1506 \\ 0.0225 & -0.1448 & 0.1329 & 0.0326 & 0.2745 & 0.1319 \\ 0.0359 & -0.0316 & 0.0326 & 0.0089 & 0.0676 & 0.0307 \\ 0.0551 & -0.2978 & 0.2745 & 0.0676 & 0.5668 & 0.2718 \\ -0.0311 & -0.1506 & 0.1319 & 0.0307 & 0.2718 & 0.1337 \end{bmatrix}.$$

As explained before, optimal projection vectors form an orthonormal basis for the intersection subspace of $N(S_W)$ and $R(S_T)$. Thus, Observation 3.1 can also be used to find the projection matrix P_{opt} of this intersection subspace. Let $P^{(1)}$ and $P^{(2)}$ represent the projection matrices of $R(S_T)$ and $N(S_W)$ respectively. Then

$$\Psi = 0.5P^{(1)} + 0.5P^{(2)} = \begin{bmatrix} 0.9917 & 0.0512 & 0.0112 & 0.0180 & 0.0276 & -0.0156 \\ 0.0512 & 0.5871 & -0.0724 & -0.0158 & -0.1489 & -0.0753 \\ 0.0112 & -0.0724 & 0.5665 & 0.0163 & 0.1372 & 0.0659 \\ 0.0180 & -0.0158 & 0.0163 & 0.5045 & 0.0338 & 0.0153 \\ 0.0276 & -0.1489 & 0.1372 & 0.0338 & 0.7834 & 0.1359 \\ -0.0156 & -0.0753 & 0.0659 & 0.0153 & 0.1359 & 0.5669 \end{bmatrix},$$

where

$$P^{(1)} = \begin{bmatrix} 0.9999 & 0.0039 & -0.0006 & -0.0071 & 0.0013 & 0.0038 \\ 0.0039 & 0.8186 & 0.0269 & 0.3340 & -0.0623 & -0.1799 \\ -0.0006 & 0.0269 & 0.9960 & -0.0495 & 0.0092 & 0.0266 \\ -0.0071 & 0.3340 & -0.0495 & 0.3853 & 0.1146 & 0.3312 \\ 0.0013 & -0.0623 & 0.0092 & 0.1146 & 0.9786 & -0.0618 \\ 0.0038 & -0.1799 & 0.0266 & 0.3312 & -0.0618 & 0.8216 \end{bmatrix},$$

and

$$P^{(2)} = \begin{bmatrix} 0.9835 & 0.0985 & 0.0231 & 0.0431 & 0.0538 & -0.0349 \\ 0.0985 & 0.3556 & -0.1717 & -0.3655 & -0.2355 & 0.0294 \\ 0.0231 & -0.1717 & 0.1369 & 0.0821 & 0.2653 & 0.1052 \\ 0.0431 & -0.3655 & 0.0821 & 0.6236 & -0.0470 & -0.3005 \\ 0.0538 & -0.2355 & 0.2653 & -0.0470 & 0.5882 & 0.3336 \\ -0.0349 & 0.0294 & 0.1052 & -0.3005 & 0.3336 & 0.3122 \end{bmatrix}.$$

The eigenvectors corresponding to the eigenvalue 1 are

$e_1 = [0.9630 \quad 0.1968 \quad -0.0649 \quad 0.0143 \quad -0.1254 \quad -0.1175]^{\mathrm{T}}$, and

$e_2 = [-0.2369 \quad 0.3680 \quad -0.3588 \quad -0.0935 \quad -0.7423 \quad -0.3463]^{\mathrm{T}}$.

These vectors also span the same space spanned by the optimal projection vectors computed before, since the projection matrix found by using these vectors is the same as P_{opt} computed before, i.e., $P_{opt} = [e_1 \quad e_2][e_1 \quad e_2]^T$.

Now let $P^{(3)}$ be the projection matrix of the range space of S_B . We need to compute the following matrix to find the intersection of the null space of S_W and the range space of S_B ,

$$\widetilde{\Psi} = 0.5P^{(2)} + 0.5P^{(3)} = \begin{bmatrix} 0.8436 & 0.0966 & 0.0258 & 0.1110 & -0.0888 & 0.1507 \\ 0.0966 & 0.2561 & 0.0068 & -0.1011 & -0.0025 & 0.0623 \\ 0.0258 & 0.0068 & 0.1833 & 0.1324 & 0.2927 & 0.0909 \\ 0.1110 & -0.1011 & 0.1324 & 0.4020 & 0.0736 & -0.0833 \\ -0.0888 & -0.0025 & 0.2927 & 0.0736 & 0.5699 & 0.1569 \\ 0.1507 & 0.0623 & 0.0909 & -0.0833 & 0.1569 & 0.2451 \end{bmatrix}.$$

There is no eigenvalue of $\widetilde{\Psi}$ that corresponds to 1. Thus, the intersection of $N(S_W)$ and $R(S_B)$ is trivial, which clearly indicates that the optimal projection vectors are not in this intersection. Hence the intersection of $N(S_W)$ and $R(S_B)$ alone cannot be used for recognition tasks.

We can also compute the projection matrix of the optimal discriminant subspace directly with the following formula,

$$P_{opt} = P^{(1)}P^{(2)} = P^{(2)}P^{(1)}$$

since $P^{(1)}$ and $P^{(2)}$ commute. Thus, the optimal projection vectors that span the optimal discriminant subspace can also be obtained by the PCA+Null space Method. Note also that the projection matrix $P^{(2)}$ of $N(S_W)$ and $P^{(3)}$ of $R(S_B)$ do not commute, i.e., $P^{(2)}P^{(3)} \neq P^{(3)}P^{(2)}$. That is why the Direct-LDA Method does not extract features from the intersection of $N(S_W)$ and $R(S_B)$.

Now we can use the optimal projection vectors for dimension reduction. In this case, every sample in each class produces the same feature vector, called the discriminative common vector. In particular,

$$\Omega_1 = [< x_1^1, w_1 > \quad < x_1^1, w_2 >]^T = [< x_2^1, w_1 > \quad < x_2^1, w_2 >]^T = [\text{-}0.9094 \quad 0.0436]^T,$$

$$\Omega_2 = [< x_1^2, w_1 > \quad < x_1^2, w_2 >]^T = [< x_2^2, w_1 > \quad < x_2^2, w_2 >]^T = [0.1174 \quad 3.5951]^T,$$

$$\Omega_3 = [< x_1^3, w_1 > \quad < x_1^3, w_2 >]^T = [< x_2^3, w_1 > \quad < x_2^3, w_2 >]^T = [0.0618 \quad \text{-}4.1549]^T.$$

As a consequence a 100 % recognition rate is guaranteed for the vectors in the training set in the reduced 2-dimensional space.

3.9 Experimental Results

The Yale [7], AR [83], and ORL (Olivetti-Oracle Research Lab) face databases were used to test the recognition accuracy of the DCV Method. In addition to our proposed method, we also tested the PCA Method (also called the Eigenface Method for face recognition tasks), the PCA+FLDA Method (also called the Fisherface Method for face recognition tasks), and

the Direct-LDA Method. We did not test the PCA+Null Space Method since it has the same recognition accuracy as our method.

3.9.1 Experiments with the Yale Face Database

The Yale face database consists of images from $C = 15$ different people, using 11 images from each person, for a total of 165 images. The images contain variations with the following facial expressions or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised and wink. For subjects numbered 2, 3, 6, 7, 8, 9, 12 and 14, the normal facial expression and the without-glasses (or with glasses if subject normally wears glasses) images were copies of each other. Thus, we removed the image without glasses (or with glasses if subject normally wears glasses) from every subject in order to make all classes have an equal number of samples and so that all sample images were distinct. Thus, we had 10 samples per subject yielding a face database size of 150. We pre-processed these images by aligning and scaling them so that the distances between the eyes were the same for all images, and also ensuring that the eyes occurred in the same coordinates of the image. The resulting image was then cropped. The final image size was 126x152. The recognition rates were computed by the "leave-one-out" strategy [39] since the training set size is relatively small. The nearest-neighbor algorithm was employed using the Euclidean distance for classification. For the PCA Method, the most significant eigenvectors were chosen such that corresponding eigenvalues contained 95 % of the total energy. For the PCA+FLDA Method, all images were first projected onto a (*M-C=134)*-dimensional space, where S_W was nonsingular. The results for the Yale Database are given in Table 3.1. As can be seen in table, the DCV Method achieved the highest recognition rate.

TABLE 3.1
Recognition Rates for the Yale Face Database

Methods	Recognition Rates
PCA	76%
PCA+FLDA	96%
Direct-LDA	92%
Discriminative Common Vector	**97.33%**

3.9.2 Experiments with the AR Face Database

The AR face database includes 26 frontal images with different facial expressions, illumination conditions, and occlusions for 126 subjects. Images were recorded in two different sessions 14 days apart. Thirteen images were recorded under controlled circumstances in each session. The size of the images in the database is 768x576 pixels, and each pixel is represented by 24 bits of RGB color values.

We randomly selected $C = 50$ individuals (30 males and 20 females) for the experiment. Only nonoccluded images ((a)-(g) and (n)-(t) as in Figure 3.4) were chosen for every subject. Thus, our face database size was 700 with 14 images per subject. Next, these images were converted to grayscale, aligned, scaled, localized and cropped using the same procedure described previously for the Yale face database experiment. The final size of the images was 222x299. The training set consisted of $N = 7$ images randomly selected from each subject, and the rest of the images were used as the test set. Thus, a training set of $M = 350$ images and a test set of 350 images were created. A nearest-neighbor algorithm was employed using the Euclidean distance for classification. This process was repeated 4 times and the recognition rates were found by averaging the error rates of each run. The results are shown in Table 3.2. As can be seen in the table, the DCV Method achieved the lowest error rate on the AR face database.

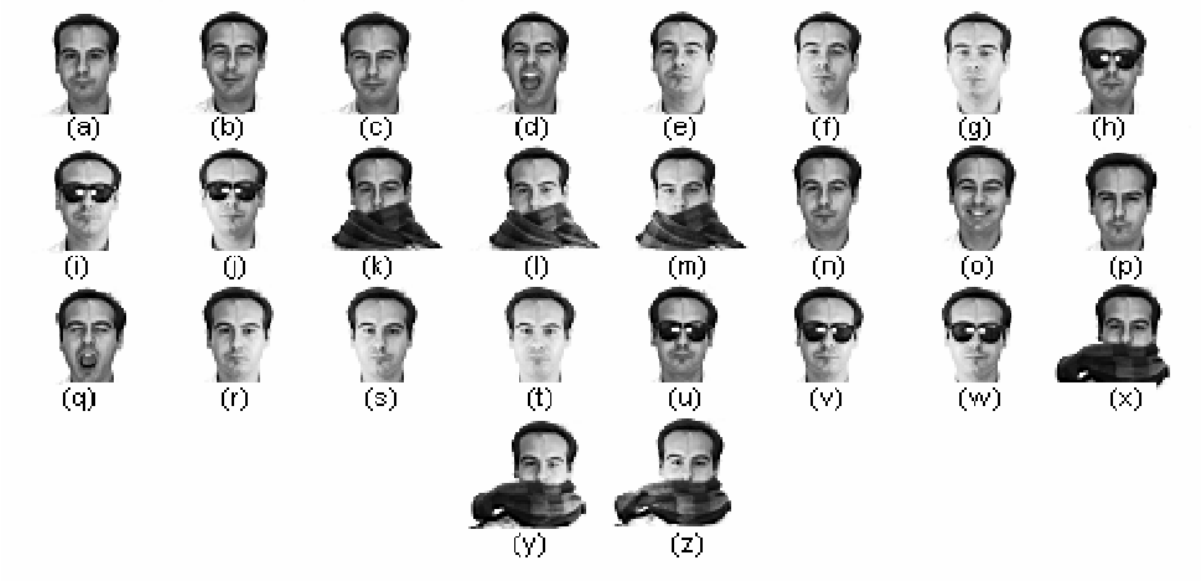


Figure 3.4: Images of one subject from the AR face database. First 13 images (a)-(m) were taken in one session and the others (n)-(z) in another session. Only nonoccluded images (a)-(g) and (n)-(t) were used in our experiments.

TABLE 3.2
Recognition Rates for the AR Face Database

Methods	**Recognition Rates**
PCA	79.14%
PCA+FLDA	98.85%
Direct-LDA	98.64%
Discriminative Common Vector	**99.35%**

The success of the proposed method depends on the size of the null space of the within-class scatter matrix, S_W. When the size of the null space is small, recognition rates are expected to be poor, since there will not be sufficient space for obtaining the optimal projection vectors. This is also mentioned in [22]. To verify this effect, we performed experiments using the pre-processed AR face database images. We randomly selected 7 images from each class for training and used the rest for testing. Thus, a training set of 350 images and a test set of 350 images were created. To observe the decrease in performance due to a small null space, we would have to have a huge number of classes for a training set

with sample space size 222x299. Unfortunately, we had a very limited number of classes in the training set. Thus, we had to take the approach of decreasing the dimensionality of the sample space by sub-sampling the images. Based on empirical observations, a new sample space size was chosen by down-sampling the images to 24x18. Then, we gradually decreased the number of classes from 50 to 5. This procedure was repeated 8 times using randomly chosen subsets of the 50 classes, and recognition rates were found by averaging the rates of each run. The results are shown in Figure 3.5. As can be seen, the performance decreases as the dimension of the null space decreases. This suggests that the initial sample space reduction step given in [22] is likely to reduce the achievable performance.

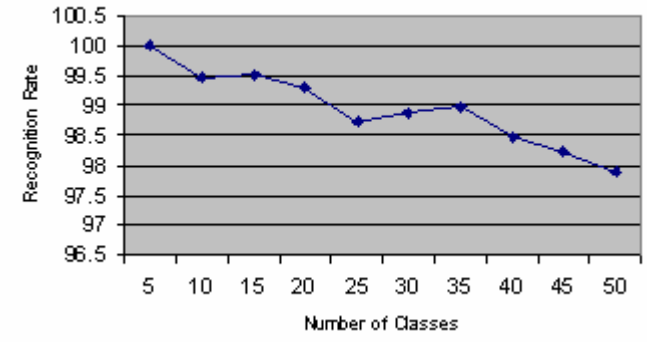


Figure 3.5: The recognition rates as functions of the number of classes for subsampled images.

3.9.3 Experiments with the ORL Face Database

The ORL face database contains $C = 40$ individuals with 10 images per person. The images are taken at different times with varying lighting conditions, facial expressions, and facial details. All individuals are in an upright, frontal position (with tolerance for some side movement). The size of the each image is 92x112 pixels. Some individuals from the ORL face database are shown in Figure 3.6.

Figure 3.6: Three sample sets from the ORL face database.

We randomly selected $N = 3,5,7$ samples from each class for training, and the remaining $(10 - N)$ samples of each class were used for testing. This process was repeated 20 times, and 20 different training and test sets were created. We did not apply any pre-processing to the images. The recognition rates for the experiment were found by averaging the recognition rates of each trial. The computed recognition rates on the ORL face database are given in Table 3.3. The best recognition was obtained by the DCV Method in all cases. The recognition performance of the DCV Method is especially superior to the other linear methods when $N = 3$ samples are used for training. As the number of training samples is increased, the difference between the recognition rates of the DCV Method and other linear methods decreases.

TABLE 3.3
Recognition Rates for the ORL Face Database

Number of training samples in each class	**Recognition Rates**			
	PCA	**PCA+FLDA**	**Direct-LDA**	**DCV**
$N = 3$	86.82%	86.35%	85.48%	**90.60%**
$N = 5$	93.75%	92.10%	95.70%	**95.95%**
$N = 7$	96.29%	94.33%	97.58%	**97.74%**

3.9.4 Gray Level Adjustment for Discriminative Common Vectors

Since we utilized the face image vectors in our experiments, it is possible to visualize the projection directions obtained by the linear methods (the eigenfaces, the fisherfaces, the projection vectors of the Direct-LDA, and the optimal projection vectors of the DCV). Furthermore, we can display the images of the reconstructed images by using these projection vectors. The depicted images of the projection directions and the reconstructed images may vary from person to person since a standard for the visualization of such images has not been established yet. Although the appearance of the projection directions found by the linear methods will not affect the performance of these methods at all, more meaningful images of projection directions may be helpful in understanding how these methods work. For instance, it has been observed that the first three significant eigenvectors model the illumination differences in face recognition problems with different illumination conditions. Thus, some researchers neglect these eigenvectors during classification since they do not carry any discriminative information. Here, we will propose a standard procedure for visualizing the common vectors obtained by the DCV Method. Since the final optimal projection vectors are obtained by applying PCA to the common vectors, each common vector represents the reconstructed images of the corresponding class by employing the optimal projection vectors. The depicted images of common vectors may help to understand which parts of face images carry the discriminatory information.

The pixel values of the original images in gray scale vary between 0 and 255. When the image is projected to a lower-dimensional subspace, the pixel values of the projected images may become negative. It is not well understood how researchers handle the negative valued pixels in depicting or reconstructing the projected images. When a projected image is

displayed in a Matlab medium, the negative gray levels of the pixels in the projected image are taken as zero. Therefore, these images seem to be darker than their normal appearance since zero gray level of pixels corresponds to the black color in Matlab. Dark and/or unclear images do not affect the recognition results, but one can wonder about the somewhat real appearance of the reconstructed images and try to see their usefulness during the recognition process. One may use the absolute values of elements of the projection vectors in order to display the images of those vectors. If the common vectors are displayed in the same manner, the resulting images are mostly very dark and obscure the interesting details in the darker areas. Thus, in order to visualize the common vectors, we took the absolute value followed by the logarithm before displaying them. However, there is a better approach for visualizing the common vectors of the DCV Method. In the following paragraphs we describe the procedure for visualization of the common vectors.

While reconstructing the images of the common vectors using Matlab, we have to work with positive gray levels. Usually these gray levels are between the integers 0 and 255. Since the algorithms given for the DCV Method may result in negative values for some of the gray levels of pixels in the common vectors, the reconstruction of the common vectors will not be meaningful due to the negative gray levels. Therefore, the best thing before obtaining the common vectors is to subtract an integer of 128 from the gray levels of each pixel in the images of all the persons in the database, that is,

$$\tilde{x}_m^i = x_m^i - 128 \text{ x } 1'_d, \; i = 1,...,C, \quad m = 1,...,N_i, \tag{3.81}$$

where the vector $1'_d$ includes all ones. After obtaining the common vectors from the new feature vectors, the integer 128 multiplied with $1'_d$ can be added to them so that they can be reconstructed using Matlab.

The absolute value of the pixel gray levels of the projected image in DCV may be larger than 128. This may cause an overflow problem for depicting projected images. The reason why some of the gray levels turn out to be larger than 128 is explained below.

Suppose that the elements of a new feature vector $\tilde{x}_m^i$ are $\tilde{x}_{m1}^i, \tilde{x}_{m2}^i, ..., \tilde{x}_{md}^i$ and they represent the gray levels in each pixel. If the absolute value of each $\tilde{x}_{mj}^i$ ($j = 1,...,d$) is bounded by a positive number Є (Є=128), then the absolute value of $\tilde{x}_{mj}^i$ is less than or equal to Є, that is, $\tilde{x}_{mj}^i \le$ Є, $j = 1,...,d$, for the whole picture elements. Then the norm of $\tilde{x}_m^i$ will be $\| \tilde{x}_m^i \| \le \sqrt{d}$ Є, where the square root of d times Є is an upper bound of the norm. Then an upper bound for the norm of the common vector can be calculated easily by

$$\begin{aligned} \| x_{com}^i \| &\le \| \sum_{k=rw+1}^{d} < \beta_k, \tilde{x}_m^i > \beta_k , \quad i = 1,...,C \\ &\le \sum_{k=rw+1}^{d} \| \tilde{x}_m^i \| \\ &\le (d - rw)\sqrt{d}\ \text{Є}. \end{aligned} \tag{3.82}$$

For example, for $d = 2$ and $rw = 1$ case, $\| x_{com}^i \| \le \sqrt{2}$ Є. This implies that some of the elements of the common vector may be larger than Є when absolute values are taken. Although this may or may not happen in practice at all, one must be careful in reconstructing the images projected onto the optimal projection vectors. We know that $\| x_{com}^i \|$ will become smaller as *rw* approaches *d*. In fact, when the indifference subspace disappears (becomes a null space when $rw = d$), then $\| x_{com}^i \| = 0$. Thus, our expectation for the elements of the common vectors is that they will approach zero values as the number of feature vectors (images) increases in the database. Therefore, only adding 128 to each element of the common vector may not be too meaningful for the reconstruction of the common vector

pictures. The reconstructed image of the common vector will have almost the gray level of 128 in each of the pixels. This image would have no meaning.

In order to improve the visualization of common vectors, the gray level of each pixel must be readjusted before the addition of 128. This is similar to the case of increasing or decreasing the contrast levels in the CRT tube. Each of elements in the common vector can be multiplied by ϵ/ζ, where ζ is the largest of the absolute values of the elements of a common vector. If ϵ is larger than ζ, the multiplication will increase the contrast level. If ϵ is smaller than ζ, it will decrease the contrast level. After this multiplication, 128 can be added to each element of the common vector. Therefore, the following three steps must be applied to visualize the common vector images:

1. Before starting to compute the common vectors, the gray level of 128 must be subtracted from each element of the image vectors.
2. After the common vectors are calculated, the contrast level of the image must be adjusted by multiplying the common vector by ϵ/ζ.
3. The gray level of 128 must be added to each element of the common vector before it is reconstructed.

The eigenfaces and common vectors obtained from the Yale, AR, and ORL face databases are shown in Figure 3.7 and Figure 3.8, respectively. Figure 3.7 displays the absolute values of the elements of the eigenfaces in an image form. In Figure 3.8 we first display the images of common vectors by taking the absolute value followed by the logarithm. We also display them by using the procedure described above. Eigenfaces characterize the variations resulting from differences in lighting conditions, facial expression, and so on, between face images. Thus, using the most significant eigenfaces (i.e., the ones

corresponding to the largest eigenvalues) may not be the best choice from a discrimination point of view. In contrast, common vectors represent the invariant regions of faces. Thus, the eyes, nose, part of the forehead above the eye brows, and cheeks are dominant in common vectors.

Figure 3.7: Most 10 significant eigenfaces obtained from the Yale, AR, and ORL face databases. The first row shows 10 significant eigenfaces obtained from one of the training sets of the AR face database, the second row shows 10 significant eigenfaces obtained from one of the training sets of the Yale face database, and the last row shows 10 significant eigenfaces obtained from one of the training sets of the ORL face database.

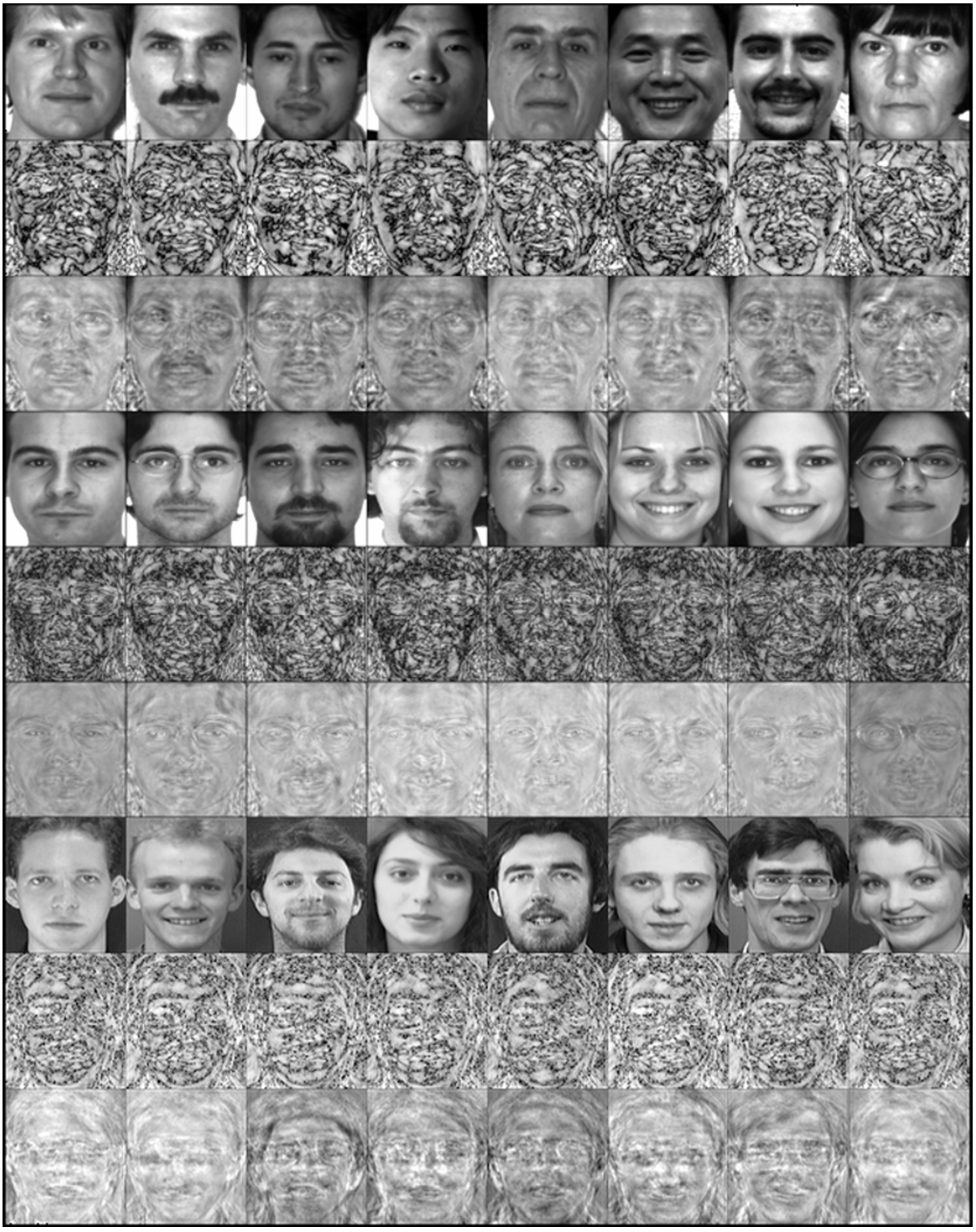


Figure 3.8: Some of the common vectors obtained from the Yale, AR, and ORL face databases. The first, second, and third rows show some individuals from the AR face database and their corresponding common vectors obtained by utilizing absolute values and the common vector visualization procedure, respectively. Similarly, the second three rows show some individuals from the Yale face database and their corresponding common vectors, and the last three rows show some individuals of the ORL face database and their corresponding common vectors.

3.10 Discussion

Accuracy, training cost, execution speed, and storage requirements are some factors that may be used to judge a pattern recognition method. Here we discuss the differences of these factors among the methods considered in this chapter.

Experimental results show that the proposed method (as well as the PCA+Null Space Method) yielded the highest performance in terms of accuracy. The PCA Method typically yielded lower recognition rates. In particular, its recognition rate for the Yale face database was notably poor. The misclassified images for the PCA Method were typically images that were not taken under the standard ambient light conditions used for most of the data (i.e., illumination was center-light, left-light, or right-light). Given that projection directions found by the PCA Method are chosen for optimal reconstruction, this method is expected to work well when the testing samples of a subject are similar to the samples of the subject used for training as in the ORL face database. Since the leave-one-out method was used for testing and there was only one sample for these non-ambient light illumination conditions per class, these unusual illumination images behaved as data outliers (i.e., these images were far from the samples used for training) for the Yale face database. We would expect better results if there were more than one example with these illumination conditions. The other tested methods produced better results since projection directions minimizing the total within-class scatter were used. A significant part of the total within-class scatter was produced by the non-ambient lighting cases in all of the classes. This variation due to lighting conditions appears to produce similar deviations from class mean across all classes. Thus, we believe the resulting projection reduces variation due to lighting in all classes, even classes in which such variation did not appear in the training set.

The proposed method and the PCA+Null Space Method require the same storage space, which is the smallest of all the methods studied. We need to store at most (C-1) d-dimensional projection vectors and C (C-1)-dimensional discriminative common vectors for comparison (In the PCA+Null Space Method it is not necessary to save all the training sample feature vectors, only the smaller set of discriminative common vectors, although this has not been reported in the literature.) Secondly, the Direct-LDA and the PCA+FLDA methods have the same storage requirements, which are higher than those of the proposed method and the PCA+Null Space Method. For these methods, we must save at most (C-1) d-dimensional projection vectors and M (C-1)-dimensional sample feature vectors of the training set for comparison. Hence, the only difference among storage requirements of the four methods is the number of feature vectors saved for comparison (The difference is the need to store additional (M-C) (C-1)-dimensional vectors for the Direct-LDA and the PCA+FLDA methods.) If M is small and d is large, this difference is negligible. However, if M is increased, this difference will also increase and become significant. Finally, for $n>C$-1 the PCA Method has the largest storage space requirements. Here n is the number of the chosen significant eigenvectors and has been chosen such that the corresponding eigenvectors contain 95% of the total energy in our experiments. It was found to be a minimum of 65 for the Yale face database, 108 for the AR face database, and 76 for the ORL face database.

Training cost is the number of computations required to find the optimal projection vectors and the sample feature vectors of the training set for comparison. We compare the training cost of the methods based on their computational complexities (number of flops). The Direct-LDA Method yields the highest efficiency in terms of computation complexity.

The next most efficient method is the proposed method, followed by the PCA Method, the PCA+FLDA Method and the PCA+Null Space Method. The computational comparison that is most interesting to us is between the PCA+Null Space Method and the proposed method, since these two methods yield the same accuracy, which is higher than the other methods. We estimated the computational complexities of these two algorithms and found PCA+Null Space to require approximately $(4dM^2 + 2l(\frac{4M^3}{3} + 2M^3 - M^2))$ flops and the proposed method required approximately $(2d(M-C)^2 + 4dMC)$ flops. Here l represents the number of iterations required for convergence of the eigen-decomposition algorithm. As d (the sample space size) and M (the number of training samples) become large, the proposed method requires less than half of the computations as the PCA+Null Space Method.

Execution speed or testing time is the time that is required to classify a new test sample. To do this, a test sample must be projected onto the linear span of the projection vectors and compared to the sample feature vectors of the training set. Testing time determines the real-time efficiency of a method. We also compare testing times based on computational complexities in this study. Our proposed method and the PCA+Null Space Method yield the highest efficiency in terms of computation. In these methods, a test image is projected onto (C-1) d-dimensional vectors and compared to the C (C-1)-dimensional vector set. The Direct-LDA and PCA+FLDA methods follow them in cost. In these methods, a test image is projected onto (C-1) d-dimensional vectors and compared to M (C-1)-dimensional vectors. As a result, the only difference between the testing times of these four methods is the time that is spent on comparison. In the Direct-LDA and the PCA+FLDA methods, a projected test image must be compared to all sample feature vectors of the training set instead of being compared to only one representative for each class. Thus, as with the storage requirements,

when the number of samples M is increased, the difference between testing times of these methods will also increase and become significant. Finally, the PCA Method yields the maximum test time in the case $n>C$-1.

In summary, the proposed method becomes progressively more efficient, compared to the other methods, as the size of the sample space M is increased. In Table 3.4 we present the overall results of our comparisons. The top row of the table lists the four criteria on which the methods were compared. The left column of the table is a qualitative ranking of how each method performed, and the cells in the table contain methods with comparable performance.

TABLE 3.4
Comparisons of Performance Across Methods for $n>C$-1

Performance Rank	**Accuracy**	**Training Time**	**Testing Time**	**Storage Requirements**
1	Discriminative Common Vector, PCA+Null Space	Direct-LDA	Discriminative Common Vector, PCA+Null Space	Discriminative Common Vector, PCA+Null Space
2	PCA+FLDA	Discriminative Common Vector	PCA+FLDA, Direct-LDA	PCA+FLDA, Direct-LDA
3	Direct-LDA	PCA	PCA	PCA
4	PCA	PCA+FLDA		
5		PCA+Null Space		

3.11 Conclusion

In this chapter we proposed a new method for addressing computational difficulties encountered in obtaining the optimal projection vectors in the optimal discriminant subspace. We showed that every sample in a given class produces the same unique common vector when they are projected onto the null space of S_W. We also proposed an alternative

algorithm for obtaining common vectors based on the subspace methods and the Gram-Schmidt orthogonalization procedure, which avoids handling large matrices and improves the stability of the computation. Using common vectors also leads to an increased computational efficiency in pattern recognition tasks in high-dimensional spaces. Optimal projection vectors are found by using the common vectors, and the discriminative common vectors are determined by projecting any sample from each class onto the span of optimal projection vectors. There is no loss of information content in our method in the sense that the method has 100% recognition rate for linearly independent training set data. Experimental results show that the proposed method is superior to other methods in terms of accuracy, real-time performance, storage requirements, and numerical stability.

CHAPTER IV

NONLINEAR FEATURE EXTRACTION METHODS

This chapter presents a general introduction to nonlinear feature extraction methods employing kernel functions. The kernel trick concept has been introduced here, and this trick is applied to the linear DCV Method to make it a nonlinear method. Then, a large scale comparison of linear and nonlinear feature extraction methods has been carried out and its results are examined. Finally, we draw our conclusions based on those experimental results at the end of the chapter.

4.1 An Introduction to Kernel Feature Extraction Methods

Sometimes linear methods may not provide sufficient nonlinear discriminant power for classification of linearly non-separable classes (e.g., exclusive-or problem). Thus, kernel methods have been proposed to overcome this limitation. The basic idea of these methods is first to transform the data samples into a higher-dimensional space $\Im$ via nonlinear mapping $\phi(.)$, and then apply the linear methods in this space. More formally, we apply the mapping $\phi : R^d \rightarrow \Im$, $x \mapsto \phi(x)$ to all the data samples. The motivation behind this process is to transform linearly non-separable data samples into a higher-dimensional space where the data samples are linearly separable as illustrated in Figure 4.1, which is adopted from [101]. Since the mapped space is nonlinearly related to the original sample space, nonlinear decision boundaries between classes can be obtained for classification. This approach seems to contradict the curse of dimensionality phenomenon since it increases the dimensionality of

the sample space for a fixed number of available training set samples. A satisfactory explanation for this dilemma lies in statistical learning theory. This theory tells us that learning in high-dimensional space can be simpler if one uses low complexity, i.e., a simple class of decision rules such as linear classifiers [89]. In other words, it is not the dimensionality but the complexity of the function that matters.

In some recognition tasks we may have sufficient knowledge about the problem and can choose $\phi(.)$ by hand. If the mapping is not too complex and $\Im$ is not too high-dimensional, we can explicitly apply this mapping as happens in Radial Basis Networks or Boosting Algorithms. However, in most cases we may not have sufficient prior knowledge to design $\phi(.)$, or the mapping of the data samples into a higher-dimensional space explicitly cannot be intractable. In such cases, we utilize kernel functions to circumvent these limitations.

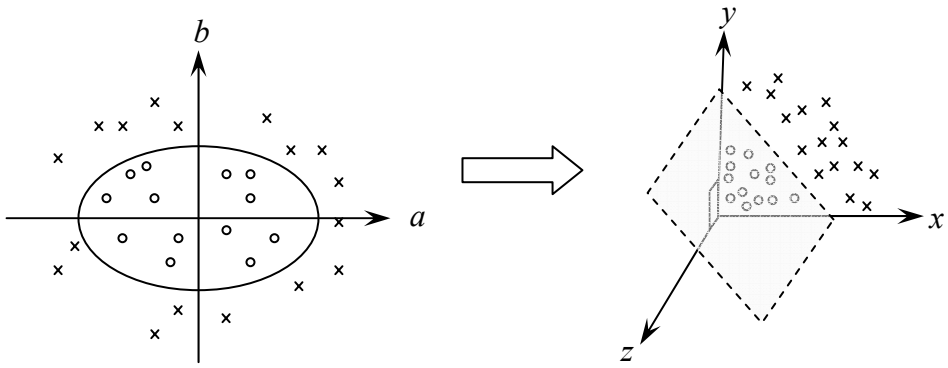


Fig 4.1: Kernel (nonlinear) mapping of 2-dimensiona data into 3-dimensional space by polynomial kernel function.

4.2 Kernel Functions and Feature Spaces Induced by Kernels

Utilizing kernel functions allows us to compute the dot products of the mapped samples in the higher-dimensional space, $\Im$. Therefore, we must first formulate the pattern recognition

methods in terms of dot products of the mapped samples. Then, we use kernel functions to compute dot products in $\Im$ such that

$$k(x, y) =< \phi(x), \phi(y) >, \tag{4.1}$$

where $< . >$ represents the dot product. In this case, we do not need to carry out the mapping of $\phi(.)$ explicitly, which makes the application of linear algorithms in higher-dimensional spaces feasible. Furthermore, this allows us to know only kernel function *k*, not the mapping $\phi(.)$. As a result, any linear algorithm that only uses scalar products can be executed in $\Im$ via kernel functions using the data samples in the original sample space $\Re^d$. This is also known as the *kernel trick*.

For example, in a polynomial case, the dot product of $\phi(x)$ and $\phi(y)$ can be found by the kernel function

$$k(x, y) = (< x, y >)^n, \tag{4.2}$$

where ϕ maps any sample vector *x* in *d*-dimensional space to the vector $\phi(x)$ whose entries are all possible *n*-th degree ordered products of the entries of *x*. Note that for small dimensional spaces and for small degrees *n*'s, this scalar product can be computed explicitly. However, if the dimensionality of the sample space or the degree is high, the computation of the scalar product explicitly becomes intractable since there exist

$$d_F = \frac{(d+n-1)!}{n!(d-1)!} \tag{4.3}$$

different monomials comprising a feature space $\Im$ of dimensionality d_F.

In general, any kernel function corresponds to a dot product in some higher-dimensional space $\Im$ if it satisfies the conditions given in the following propositions [101].

Proposition 4.1: If *k* is a continuous symmetric kernel of a positive integral operator *K*, i.e.,

$$(Kf)(y) = \int_C k(x,y) f(x) dx \tag{4.4}$$

with

$$\int_{CxC} k(x,y) f(x) f(y) dx dy \geq 0 \tag{4.5}$$

for all $f \in L^2(C)$ (C being a compact subset of $\Re^d$), it can be expanded in a uniformly convergent series (on CxC) in terms of eigenfunctions ψ_j and positive eigenvalues λ_j,

$$k(x,y) = \sum_{j=1}^{d_F} \lambda_j \psi_j(x) \psi_j(y), \tag{4.6}$$

where $d_F \leq \infty$.

Proposition 4.2: If k is a continuous kernel of a positive integral operator (conditions as in Proposition 4.1), one can construct a mapping ϕ into a space where k acts a dot product, $< \phi(x), \phi(y) >= k(x,y)$.

In practice, we are given a finite amount of data sample vectors. The following proposition explains how we can choose the kernel functions corresponding to dot products of the mapped samples for some real pattern recognition applications without analytically analyzing a given kernel.

Proposition 4.3: Suppose the data sample vectors $x_1, ..., x_M$ and the kernel k are such that the matrix $K = (K_{ij})_{\substack{i=1,...,M \\ j=1,...,M}}$ is a positive semi-definite matrix, where each element of the matrix is defined as

$$K_{ij} = k(x_i, x_j). \tag{4.7}$$

Then it is possible to construct a map $\phi(.)$ into some high-dimensional space $\Im$ such that

$$k(x_i, x_j) =< \phi(x_i), \phi(x_j) >. \tag{4.8}$$

Conversely, for a map $\phi(.)$ into some high-dimensional space, $\Im$, the matrix K is positive semi-definite.

4.3 The Kernel Principal Component Analysis Method

The basic idea of the Kernel PCA Method is first to map the data samples into a higher-dimensional space $\Im$ via nonlinear mapping, and then apply the linear PCA Method in this higher-dimensional space. Let $\phi(x_1^1),\phi(x_2^1),...,\phi(x_{N_1}^1),\phi(x_1^2),...,\phi(x_{N_C}^C)$ represent the mapped samples in $\Im$. The within-class scatter matrix S_W^Φ, the between-class scatter matrix S_B^Φ, and the total scatter matrix S_T^Φ in $\Im$ are given by,

$$S_W^\Phi = \sum_{i=1}^{C}\sum_{m=1}^{N_i}(\phi(x_m^i)-\mu_i^\Phi)(\phi(x_m^i)-\mu_i^\Phi)^T = (\Phi-\Phi G)(\Phi-\Phi G)^T, \quad (4.9)$$

$$S_B^\Phi = \sum_{i=1}^{C} N_i(\mu_i^\Phi-\mu^\Phi)(\mu_i^\Phi-\mu^\Phi)^T = (\Phi U-\Phi L)(\Phi U-\Phi L)^T, \quad (4.10)$$

and

$$S_T^\Phi = \sum_{i=1}^{C}\sum_{m=1}^{N_i}(\phi(x_m^i)-\mu^\Phi)(\phi(x_m^i)-\mu^\Phi)^T = (\Phi-\Phi 1_M)(\Phi-\Phi 1_M)^T = S_W^\Phi + S_B^\Phi, \quad (4.11)$$

where μ^Φ is the mean of all mapped samples, μ_i^Φ is the mean of mapped samples in the *i*-th class, and Φ is the matrix whose columns are the mapped training set samples in $\Im$. Here $G = diag[G_1...G_C] \in \Re^{MxM}$ is a block diagonal matrix, and $G_i \in \Re^{N_i x N_i}$ is a matrix whose all entries are $1/N_i$; $U = diag[u_1...u_C] \in \Re^{MxC}$ is a block diagonal matrix and $u_i \in \Re^{N_i x 1}$ is a vector whose entries are all $1/\sqrt{N_i}$; $L = [l_1...l_C] \in \Re^{MxC}$ is matrix where $l_i \in \Re^{Mx1}$ is a vector whose all entries are $\sqrt{N_i}/M$; $1_M \in \Re^{MxM}$ is a matrix whose all entries are $1/M$.

The principal components are computed by solving the eigenvalue problem,

$$\lambda w = S_T^{\Phi} w . \tag{4.12}$$

All eigenvectors w corresponding to the nonzero eigenvalues lie in the span of $\{ \phi(x_1^1) - \mu^{\Phi}, \phi(x_2^1) - \mu^{\Phi}, ..., \phi(x_{N_1}^1) - \mu^{\Phi}, \phi(x_1^2) - \mu^{\Phi}, ..., \phi(x_{N_C}^C) - \mu^{\Phi} \}$. This can be written as

$$w = \sum_{i=1}^{C} \sum_{m=1}^{N_i} \alpha_{ij} (\phi(x_m^i) - \mu^{\Phi}) = (\Phi - \Phi 1_M)\alpha . \tag{4.13}$$

If we multiply (4.12) with $(\Phi - \Phi 1_M)^T$ from left and substitute (4.13) in this equation, we obtain

$$\lambda \widetilde{K} \alpha = \widetilde{K}^2 \alpha \Rightarrow \lambda \alpha = \widetilde{K} \alpha , \tag{4.14}$$

where $\widetilde{K} = K - 1_M K - K 1_M + 1_M K 1_M \in \Re^{MxM}$ and $K \in \Re^{MxM}$ is given by,

$$K = \Phi^T \Phi = (K_{mn}^{ij} =< \phi(x_m^i), \phi(x_n^j) >= k(x_m^i, x_n^j))_{\substack{i,j=1,...,C \\ m=1,...,N_i; n=1,...,N_j}} . \tag{4.15}$$

There are at most M-1 eigenvectors corresponding to nonzero eigenvalues of $\widetilde{K}$. Since $\{w_1, ..., w_{M-1}\}$ must be an orthonormal set, the vectors α_j must be normalized such that,

$$< w_j, w_j >= \alpha_j^T (\Phi - \Phi 1_M)^T (\Phi - \Phi 1_M)\alpha_j = \alpha_j^T \widetilde{K} \alpha_j = \lambda_j < \alpha_j, \alpha_j >= 1 . \tag{4.16}$$

Then we select the most significant n $(1 \le n \le M-1)$ eigenvectors for feature extraction. A test sample feature vector is obtained by the equation $\Omega_{test} = W^T (\phi(x_{test}) - \mu^{\Phi})$, where W is the matrix of the projection vectors w_j $(j = 1, ..., n)$. In this case each element of Ω_{test} can be found by

$$< w_j, \phi(x_{test}) - \mu^{\Phi} >=< w_j, \phi(x_{test}) - \Phi 1_M^{'} >= \alpha_j^T (K_{test} - K 1_M^{'} - 1_M K_{test} + 1_M K 1_M^{'}) , \tag{4.17}$$

where $1_M^{'} \in \Re^{Mx1}$ is a vector with all terms equal to $1/M$, and $K_{test} \in \Re^{Mx1}$ is a vector with entries $< \phi(x_m^i), \phi(x_{test}) >_{\substack{i=1,...,C \\ m=1,...,N_i}}$.

We do not need to map the samples into $\Im$ explicitly if we use the kernel functions in (4.15). Therefore all calculations can be done using the data samples in the original sample space, $\Re^d$. The extracted features are nonlinear in the original sample space since the mapped space is nonlinearly related to the original sample space. Thus, if we apply a linear classifier to the extracted features, we can get non-convex decision regions in the original sample space, which in turn may increase the flexibility and the accuracy of the linear classifiers.

In the PCA Method we can extract $n = \min(d, M-1)$ features. In the Kernel PCA Method, since the dimensionality d_F of the mapped space is typically too large, we can extract at most M-1 features. Therefore, if the training set size is larger than the dimensionality of the sample space d, the number of the extracted features obtained by the Kernel PCA can exceed the original dimension of the sample space. The linear PCA and the Kernel PCA also differ from the reconstruction point of view. Although it is possible to reconstruct the training set samples by using all principal components in the linear PCA Method, this is not possible in the Kernel PCA case.

4.4 The Kernel Linear Discriminant Analysis Methods

The Kernel PCA Method is an unsupervised technique that aims to extract features for optimal reconstruction in the mapped space. Thus, the extracted features may not be optimal from the discrimination point of view. Therefore, discriminant analysis techniques utilizing kernels have been recently proposed [4], [86], [126]. Similar to the Kernel PCA, these methods also employ kernel functions to project data samples into a higher-dimensional space via a nonlinear kernel mapping, and then the Linear Discriminant Analysis (LDA) is

performed in this higher-dimensional space. However, the singularity problem of the matrices is encountered in these techniques. Two different approaches are adopted to solve this problem. Mika *et al*. use the original FLDA criterion in the nonlinearly mapped space; they solve the singularity problem by adding a small perturbation matrix which makes the singular matrix become nonsingular [86]. Yang *et al*. use the modified FLDA criterion instead of the original FLDA criterion in the mapped space [126]. They first project the data onto the range space of the total scatter matrix of mapped samples through Kernel PCA, and then they apply the LDA Method which maximizes the modified FLDA criterion in this reduced space. The first approach is called the Kernel Fisher's Discriminant Analysis (Kernel FDA) Method, and the latter approach is called the Kernel PCA+LDA (KPCA+LDA) Method.

4.4.1 The Kernel Fisher's Discriminant Analysis Method

This method aims to maximize the FLDA criterion $J^{\Phi}_{FLDA}(W_{opt}) = \max \frac{|W^T S^{\Phi}_B W|}{|W^T S^{\Phi}_W W|}$ in the mapped space $\Im$. The projection vectors that maximize this criterion are obtained by solving the equation

$$\lambda S^{\Phi}_W w = S^{\Phi}_B w. \tag{4.18}$$

To compute the projection vectors, we must first formulate (4.18) in terms of dot products of mapped samples which we then replace with kernel functions. Similar to the Kernel PCA Method, we can write the projection vectors *w* as in (4.13). In equation (4.18), $\lambda S^{\Phi}_W w$ term can be written as

$$\lambda S^{\Phi}_W w = \lambda(\Phi - \Phi G)(\Phi - \Phi G)^T (\Phi - \Phi 1_M)\alpha. \tag{4.19}$$

By multiplying (4.19) with $(\Phi - \Phi 1_M)^T$ from left we obtain,

$$\lambda \widetilde{K}_W \widetilde{K}_W{}^T \alpha, \tag{4.20}$$

where $\widetilde{K}_W = K - KG - 1_M K + 1_M KG$.

By following the same approach for $S_B^\Phi w$ term we get,

$$(\Phi - \Phi 1_M)^T S_B^\Phi w = \widetilde{K}_B \widetilde{K}_B{}^T \alpha, \tag{4.21}$$

where $\widetilde{K}_B = (K - 1_M)(U - L)$. By combining equations (4.20) and (4.21) we obtain,

$$\lambda \widetilde{K}_W \widetilde{K}_W^T \alpha = \widetilde{K}_B \widetilde{K}_B^T \alpha. \tag{4.22}$$

Thus, the original problem reduces to solving the following

$$J_{FLDA}^K(\alpha) = \max \frac{\alpha^T \widetilde{K}_B \widetilde{K}_B{}^T \alpha}{\alpha^T \widetilde{K}_W \widetilde{K}_W{}^T \alpha}. \tag{4.23}$$

However the matrix $\widetilde{K}_W \widetilde{K}_W{}^T \in \Re^{MxM}$ is rank deficient since its rank cannot be larger than *M-C*. Therefore a small diagonal perturbation matrix Δ is added to $\widetilde{K}_W \widetilde{K}_W{}^T \in \Re^{MxM}$ in order to make it nonsingular. Then, the vectors α_j are chosen as the eigenvectors that correspond to the nonzero eigenvalues of $(\widetilde{K}_W \widetilde{K}_W{}^T + \Delta)^{-1} \widetilde{K}_B \widetilde{K}_B{}^T$. After α_j's are computed, they are normalized as given in the previous section. Finally, the feature vectors are obtained by using equation (4.17).

4.4.2 The Kernel PCA+LDA Method

This method aims to maximize the modified FLDA criterion $J_{MFLDA}^\Phi(W_{opt}) = \max |W^T S_B^\Phi W| / |W^T S_T^\Phi W|$ in the mapped space, $\Im$. By following the same

steps given in Kernel Fisher's Discriminant Analysis, this problem is converted to the solving the following problem

$$J^{K}_{MFLDA}(\alpha) = \max \frac{\alpha^T \widetilde{K}_B \widetilde{K}_B{}^T \alpha}{\alpha^T \widetilde{K}\widetilde{K}^T \alpha}. \tag{4.24}$$

The matrix $\widetilde{K}\widetilde{K}^T \in \Re^{MxM}$ is a singular matrix since its rank cannot be larger than *M*-1. To circumvent this problem, all training set samples are first projected onto the range space of S_T^{Φ} through the Kernel PCA where the new total scatter matrix is nonsingular. Then, the Linear Discriminant Analysis which maximizes the modified FLDA criterion is applied in this reduced space. Thus, the Kernel PCA+LDA Method is equal to applying the Kernel PCA Method followed by the Linear Discriminant Analysis [126].

4.5 The Kernel Discriminative Common Vector Method

Sometimes discriminative common vectors obtained by the linear DCV Method are not distinct in the original sample space. In such cases, one can map the original sample space to a higher-dimensional space where the new discriminative common vectors in the mapped space are distinct among themselves. This is because the mapping function, $\phi : R^d \rightarrow \Im$, can map two vectors that are linearly dependent in the original sample space onto two vectors that are linearly independent in $\Im$. Note that the mapped space could have arbitrarily large, possibly infinite dimensionality, which turns out to be a perfect environment for the application of the DCV Method. Tsuda proved that if the kernel matrix *K* given in (4.15) is strictly positive definite, then all mapped samples are linearly independent [111]. Therefore, even though the data samples are linearly dependent in the original sample space, the distinctness of the discriminative common vectors is satisfied in $\Im$ by choosing a kernel

function which makes K a positive definite matrix. Therefore, a 100% recognition accuracy rate can be obtained for linearly non-separable classes by applying the linear DCV Method in $\Im$.

In the transformed space, S_W^Φ is typically singular due to the high dimensionality of the mapped space. Thus, the optimal projection vectors that maximize the modified FLDA criterion are in the intersection of the null space $N(S_W^\Phi)$ of S_W^Φ and the range space $R(S_T^\Phi)$ of S_T^Φ. Similar to the linear case, there are mainly two approaches to compute these optimal projection vectors. We can either first project the training set samples onto $N(S_W^\Phi)$ and then apply PCA, or we can first apply PCA to project the training set samples onto $R(S_T^\Phi)$ and then find an orthonormal basis for the new null space of the within-class scatter matrix of the transformed samples. However, the first approach is not feasible since the algorithms that follow this approach use the mapping function $\phi(.)$ explicitly. Therefore, it is better to follow the second approach. The training set samples can be easily projected onto $R(S_T^\Phi)$ through the Kernel PCA. Then we can find the vectors that span the null space of the within-class scatter matrix of the transformed samples. After this operation, we obtain the discriminative common vectors that represent each class. The algorithm for this method can be summarized as follows:

Step 1: Project the training set samples onto $R(S_T^\Phi)$ through the Kernel PCA. Let

$$\widetilde{K} = K - 1_M K - K 1_M + 1_M K 1_M = U \Lambda U^T \in \Re^{MxM} \tag{4.25}$$

where the diagonal elements of Λ are nonzero and $K \in \Re^{MxM}$ is given in (4.15). There are at most M-1 nonzero eigenvalues. The matrix that transforms the training set samples onto

$R(S_T^{\Phi})$ is $(\Phi - \Phi 1_M)U\Lambda^{-1/2}$. Then the new total and the within-scatter matrices in the reduced space will be

$$\begin{aligned} \widetilde{S}_T^{\Phi} &= ((\Phi - \Phi 1_M)U\Lambda^{-1/2})^T S_T^{\Phi} (\Phi - \Phi 1_M)U\Lambda^{-1/2} \\ &= \Lambda^{-1/2}U^T U\Lambda U^T U\Lambda U^T U\Lambda^{-1/2} = \Lambda \end{aligned} \tag{4.26}$$

and

$$\begin{aligned} \widetilde{S}_W^{\Phi} &= ((\Phi - \Phi 1_M)U\Lambda^{-1/2})^T S_W^{\Phi} (\Phi - \Phi 1_M)U\Lambda^{-1/2} \\ &= \Lambda^{-1/2}U^T \widetilde{K}_W \widetilde{K}_W{}^T U\Lambda^{-1/2}, \end{aligned} \tag{4.27}$$

where $\widetilde{K}_W = K - KG - 1_M K + 1_M KG = (K - 1_M K)(I - G)$.

Step 2: Find vectors that span the null space of $\widetilde{S}_W^{\Phi}$. This can be performed by an eigen-decomposition. The normalized eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_W^{\Phi}$ form an orthonormal basis for the null space of $\widetilde{S}_W^{\Phi}$. Let V be a matrix whose columns are the computed eigenvectors corresponding to the zero eigenvalues such that,

$$V^T \widetilde{S}_W^{\Phi} V = 0. \tag{4.28}$$

Step 3 (optional): Remove the null space of $V^T \widetilde{S}_B^{\Phi} V$, if it exists and rotate the projection directions so that the new total and between-scatter matrices are diagonal (i.e., the scatter matrices of the feature vectors of the training set samples are uncorrelated). That is,

$$V^T \widetilde{S}_B^{\Phi} V = V^T \widetilde{S}_T^{\Phi} V = V^T \Lambda V = L\widetilde{\Lambda} L^T. \tag{4.29}$$

Then the final projection matrix W will be

$$W = (\Phi - \Phi 1_M)U\Lambda^{-1/2} VL. \tag{4.30}$$

There are at most *C-1* projection vectors. The feature vector of a test sample is obtained by the equation

$$\Omega_{test} = W^T(\phi(x_{test}) - \mu^{\Phi}), \tag{4.31}$$

where W is the matrix of the projection vectors w_j. Then each element of the feature vector of the test sample can be obtained by

$$< w_j, \phi(x_{test}) - \mu^{\Phi} >=< w_j, \phi(x_{test}) - \Phi 1'_M >= (P\Lambda^{-1/2}VL)^T (K_{test} - K1'_M - 1_M K_{test} + 1_M K1'_M), \tag{4.32}$$

where $1'_M \in \Re^{Mx1}$ is a vector with all terms equal to $1/M$, and $K_{test} \in \Re^{Mx1}$ is a vector with entries $< \phi(x_m^i), \phi(x_{test}) >_{\substack{i=1,...,C \\ m=1,...,N_i}}$. The terms including K can be removed in (4.32) since they do not depend on the test vector.

All mathematical properties of the linear DCV carry over to the Kernel DCV Method with the modifications which now apply to the mapped vectors, $\phi(x_m^i)$, $i = 1,...,C,\ m = 1,...,N_i$, in $\Im$. After performing the feature extraction, all training set samples in each class usually produce a distinct discriminative common vector of that class. Therefore, similar to the linear DCV case, 100% recognition accuracy with respect to the training data is also guaranteed for this method. If, in practice, we cannot easily find kernel functions which guarantee the distinctness of the discriminative common vectors in $\Im$, we can add new projection vectors from outside the optimal discriminant subspace as described previously in Chapter 3 of this study. However, in all our experience, it was very rare that any of the kernels ever exhibited this problem, and in particular the Gaussian kernels were never observed to have this problem.

As we stated previously, the KPCA+LDA Method is equivalent to applying the Kernel PCA Method followed by the Linear Discriminant Analysis [126]. After this operation, we also obtain projection vectors that give rise to discriminative common vectors for classes. Therefore this method also guarantees a 100% recognition accuracy if the discriminative

common vectors are distinct in the mapped space. It should be noted that the discriminative common vectors obtained by the KPCA+LDA are different from the ones obtained by the proposed method since the projection vectors of the proposed method are orthonormal, i.e., $w_i^T w_j = \delta_{ij}$. Additionally, the projection vectors are orthogonal with respect to S_T^Φ and S_B^Φ if the optional step 3 is carried out in the Kernel DCV algorithm. More formally,

$$W^T S_T^\Phi W = W^T S_B^\Phi W = \widetilde{\Lambda}, \tag{4.33}$$

where $\widetilde{\Lambda}$ is the diagonal matrix given in (4.29). On the other hand, the projection vectors of the KPCA+LDA are not necessarily orthogonal. This property of the existence of such discriminative common vectors for the KPCA+LDA does not seem to have been noticed in the literature. Thus, the feature vector of a test sample must be compared only to the discriminative common vector of each class during classification, which makes the Kernel DCV and the KPCA+LDA methods practical for real-time applications. Note that these methods do not offer any advantages over other competing methods during the computation of the feature vectors of a test sample. Thus, if one uses a single representative prototype feature vector (e.g., mean of the feature vectors) for each class during classification of a kernel method, the real time performance of this method will be similar to the Kernel DCV and the KPCA+LDA methods.

4.5.1 Comparison of the Linear DCV and Kernel DCV Methods

Mapping samples to a higher-dimensional space via nonlinear mapping $\phi(.)$ yields some advantages for the proposed method over the linear DCV Method. The differences between these two methods can be summarized as follows:

i) The DCV Method extracts linear features from the original sample space, and the dimension of the null space of the within-class scatter matrix must be large for good recognition rates. However, the Kernel DCV Method extracts features from an implicit higher-dimensional space. It is possible to extract nonlinear features since the mapped space is nonlinearly related to the original sample space. Also, one can obtain good recognition rates by the Kernel DCV Method even though the original null space of the within-class scatter matrix is small or trivial since the null space of the within-class scatter matrix in $\mathfrak{I}$ is typically huge. Additionally, we have the flexibility of creating different nonlinear decision boundaries by simply changing the kernel functions. However, these improvements are achieved at the expense of more computations.

ii) The DCV Method can be applied only to data sets with the small sample size problem (the data sets in which the dimensionality of the original sample space is larger than the rank of the within-class scatter matrix). However, this limitation does not apply to the proposed kernel method. We can apply the Kernel DCV to these data sets even if the number of the samples is larger than the dimensionality of the sample space because of the high dimensionality of the mapped space.

4.6 Other Kernel Approaches for Pattern Recognition

There are some other kernel methods that apply linear methods in the mapped space $\mathfrak{I}$ such as the Kernel Direct-LDA Method [79] and the Support Vector Machines [26]. We will not examine these methods since studying all kernel methods is beyond the scope of this study.

4.7 Experimental Results

All supervised linear and kernel feature extraction methods discussed so far can be classified in two groups. The methods in the first group (FLDA, Direct-LDA, and Kernel FDA) use the projection directions coming from $R(S_W)$ or $R(S_W^{\Phi})$ for feature extraction, i.e., the projection vectors satisfy $W^T S_W W \neq 0$ for linear methods and they satisfy $W^T S_W^{\Phi} W \neq 0$ for nonlinear methods. On the other hand, the projection vectors of the methods in the second group (DCV, PCA+Null Space, Kernel DCV, and KPCA+LDA) come from $N(S_W)$ or $N(S_W^{\Phi})$ and they satisfy $W^T S_W W = 0$ or $W^T S_W^{\Phi} W = 0$. As explained before, projection directions of the methods of the second category come from the optimal discriminant subspace, and under certain conditions all training set samples can be classified correctly by using these projection directions for feature extraction. However, the goal of a recognition method is not only to classify all training data themselves, but also to classify well the test data samples that are not used for training. In other words, we want the recognition method to generalize well. In our experiments, we first tested the generalization abilities of those methods coming from the two different general categories separately, and then we investigated whether the performance of the methods from the second category can be improved by adding some projection directions from $R(S_W)$ or $R(S_W^{\Phi})$. In addition to the supervised feature extraction methods, we also tested the unsupervised feature extraction methods, the PCA and the Kernel PCA, to give a better assessment of the recognition accuracy of the proposed kernel method. The nearest-neighbor (NN) and the nearest-mean (NM) algorithms were employed using the Euclidean distance for classification of feature extraction methods, except for the methods that employ the discriminative common vectors (DCV, Kernel DCV, and KPCA+LDA), in which case the feature vector of the test sample

was compared only to the discriminative common vectors by using the Euclidean distance for those methods.

The dimensionality of the sample space and the size of the training set are two important factors that affect the recognition rates of the methods [59]. Therefore, experiments were performed on data sets from two different populations with different training set sizes and dimensionalities. We have selected two databases from the first population and one database from the second population. The size of the training set is larger than the dimensionality of the sample space for the databases from the first population, unlike the case of the second population. Therefore, S_W is nonsingular for the data sets from the first population and singular for the data set of the second population. In the first group of experiments, since S_W is nonsingular, we cannot apply the linear DCV Method. However, it is possible to apply the Kernel DCV Method since, as we noted, the training set samples are first transformed into a higher-dimensional space for which S_W^{Φ} is singular. For the second group of experiments, the FLDA Method cannot be applied directly. Therefore, we applied the approach suggested by Swets and Weng in which the training set samples were first projected onto an *M-C* dimensional space through PCA, for which S_W is nonsingular [104]. Then, the FLDA Method was applied to the projected samples. For the linear PCA and the Kernel PCA methods, the most significant eigenvectors were chosen in such a way that the corresponding eigenvalues contain 95% of the total energy.

An appropriate selection of kernel functions for special tasks is still an open problem since different kernel functions give rise to different constructions of the implicit feature space [94]. We have used polynomial kernels $k(x, y) = (< x, y >)^n$, with degrees $n = 2,3$ and the Gaussian kernel $k(x, y) = \exp(-\| x - y \|^2 / \gamma)$ for all data sets. We have employed a

small set of randomly created training and test sets to compute the best Gaussian parameters, γ, for each database. We first computed the minimum and the maximum values of Gaussian parameters that produce acceptable recognition rates by globally searching over a wide range of the parameter space. Then, we linearly divided the interval determined by the minimum and the maximum values of parameters into subintervals and computed the recognition rates. Finally, we carried out a local search in the neighborhood of the Gaussian parameter that yielded to the best recognition rate and computed the final best Gaussian parameter. This process was repeated for every method.

4.7.1 Experiments with Large Number of Training Samples

In this group of experiments we tested the proposed algorithm with two databases. The first database is the well-known Fisher's Iris database [36], and the second database is the digit data set consisting of handwritten numerals (0-9) extracted from a collection of utility maps [11]. The number of samples is larger than the dimensionality of the sample space for both databases.

Experiments on the Fisher's Iris Database

The Iris flower database contains four measurements on 50 Iris specimens for each of three species: Iris setosa, Iris versicolor, and Iris virginica for a total of 150 samples in the database. It was reported that the first class is linearly separable from the other two classes and that the latter two are not linearly separable from each other. We first conducted experiments to visualize the extracted features. We applied the proposed method and the other feature extraction methods discussed in the paper to this database and plotted the

extracted features. The data samples were centered before the feature extraction. We used the Gaussian kernel with $\gamma = 0.7$ for all the kernel methods. For the linear PCA and the Kernel PCA methods, we chose the most significant two eigenvectors for feature extraction. The feature vectors obtained by the linear feature extraction algorithms are illustrated in Figure 4.2, and the feature vectors obtained by the kernel methods are illustrated in Figure 4.3. As can be seen in the figures, all samples are separable for the supervised kernel methods whereas they are not separable for the linear methods and the Kernel PCA Method.

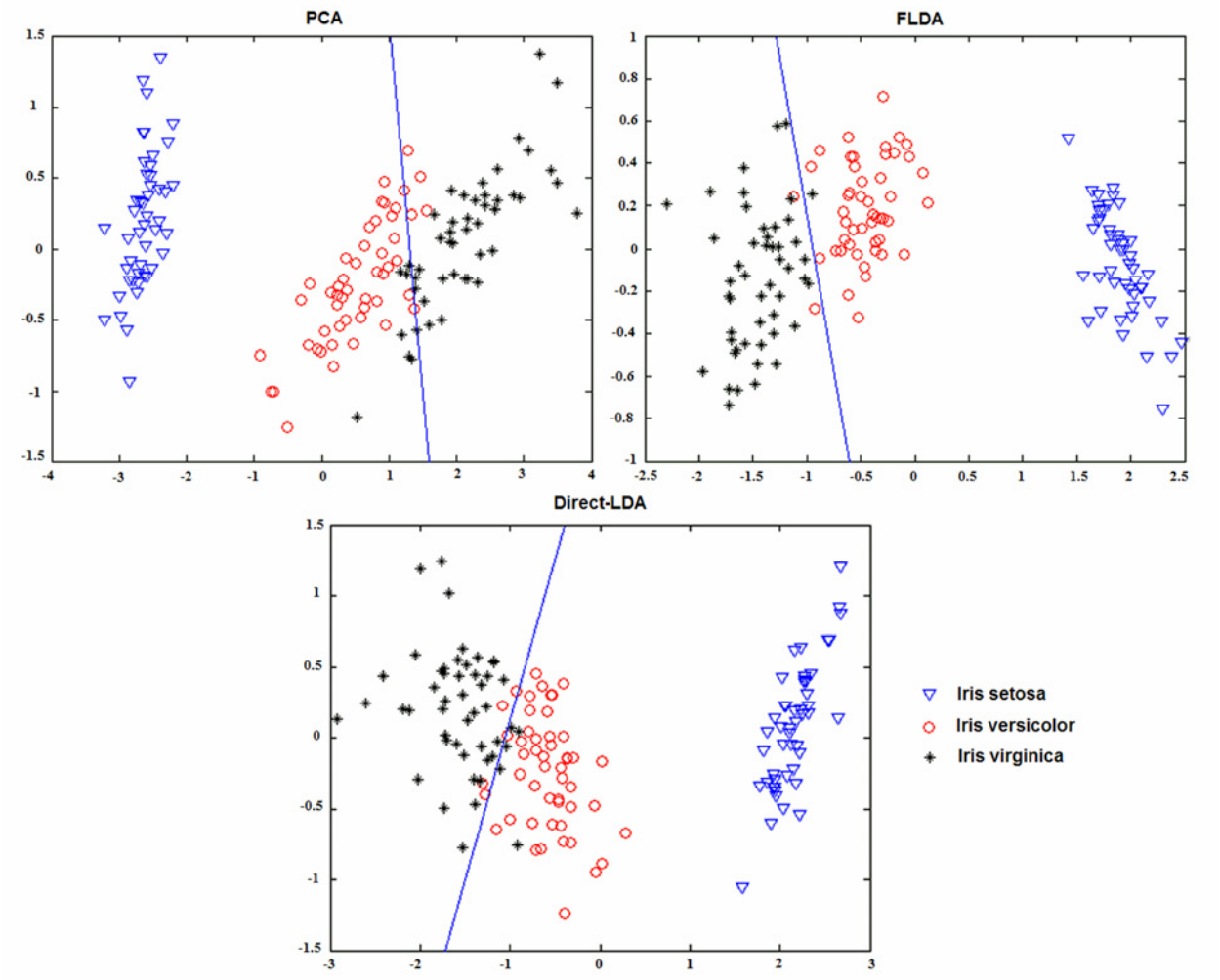


Figure 4.2: Feature vectors obtained by the linear feature extraction methods. The lines represent the decision boundaries of nonseparable classes obtained by the nearest-mean classifiers.

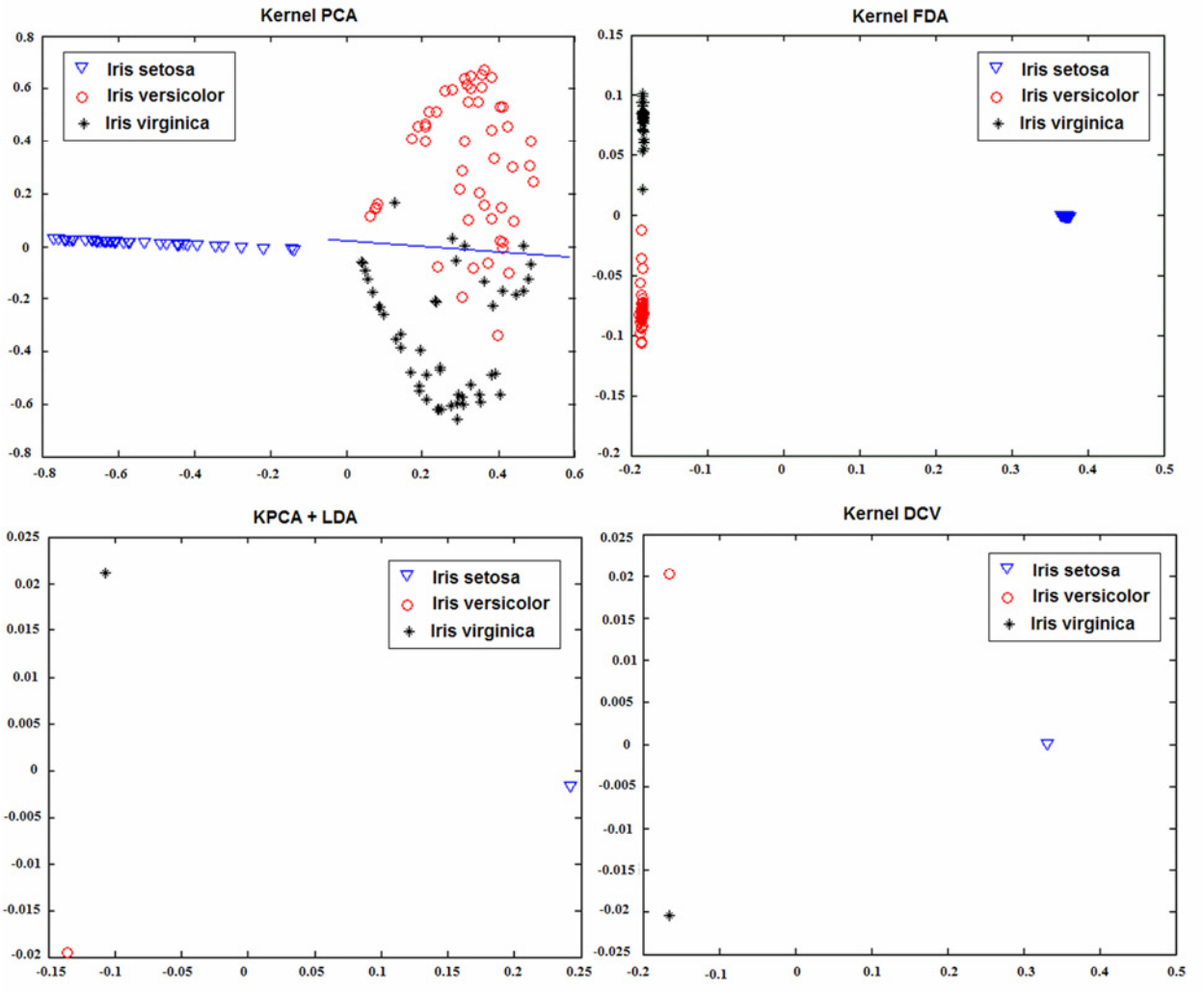


Figure 4.3: Feature vectors obtained by the kernel feature extraction methods. The line represents the decision boundary of nonseparable classes obtained by the nearest-mean classifier.

TABLE 4.1
Recognition Rates of Methods on the Fisher's Iris Database

Methods & Gaussian Kernel Parameters	**Recognition Rates (%)**	
	NN	**NM**
PCA	96	90
FLDA	96.67	**98**
Direct-LDA	92.67	94
Kernel PCA, $\gamma = 0.9$	96	96
Kernel FDA, $\gamma = 0.7$	95.33	95.33
KPCA+LDA, $\gamma = 0.2$	94.67	
Kernel DCV, $\gamma = 0.1$	96	

In the second set of experiments, we tested the generalization performances of the methods by adopting the leave-one-out strategy [39]. The recognition rates and the Gaussian parameters, which were found by the search procedure described previously, are given in Table 4.1. Note that we used only the Gaussian kernel since the small sample size does not occur for the polynomial kernel functions. In this case, the Kernel DCV and the KPCA+LDA methods cannot be used for recognition.

In terms of classification performance, the linear FLDA Method followed by the NM classifier achieves the best recognition rates among all methods for the Iris database. The proposed method achieves the best recognition rate among the kernel methods. Only the Kernel PCA Method shows an improvement over its linear counterpart.

Experiments on the Digit Dataset of Handwritten Numerals

This database includes $C = 10$ classes, each having 200 patterns. Sample patterns are available in the form of binary images. These characters are represented in terms of different feature sets. In our experiments we used only a subset of the original data set consisting of 76 Fourier coefficients and 240 pixel averages.

We have randomly chosen 100 samples from each class for training; the rest are used for testing. Thus, a training set of $M = 1000$ samples and a test set of 1000 samples were created for each database. This process was repeated 25 times, and 25 different training and test sets were created. The first 5 data sets were used for parameter selection and the rest were used for performance evaluation. Thus, the final recognition rates for the experiment were found by averaging these 20 rates obtained in each trial. The means and the standard deviations of computed recognition rates on these databases are given in Table 4.2 and Table 4.3.

As can be seen from Table 4.2, the best recognition rate among the linear methods was obtained by the PCA Method followed by the NN classifier for the Fourier Coefficients database. The proposed method using the Gaussian kernel achieved the highest recognition rate method over all methods. Although the Kernel PCA Method did not outperform the classical linear counterpart for the test sets, both the Kernel FDA and the KPCA+LDA methods outperformed the FLDA for all the kernel functions used here.

TABLE 4.2
Recognition Rates of Methods on the 76 Fourier Coefficients Database

Linear Methods	**Recognition Rates (%) and Standard Deviations**					
	NN			NM		
PCA	**82.50**, $\sigma = 1.02$			77.67, $\sigma = 0.94$		
FLDA	80.24, $\sigma = 0.81$			80.16, $\sigma = 0.83$		
Direct-LDA	81.12, $\sigma = 0.94$			79.47, $\sigma = 0.87$		
Kernel Methods & Gaussian Kernel Parameters	**Recognition Rates (%) and Standard Deviations**					
	Polynomial kernel functions with different degrees				Gaussian kernel function	
	$n = 2$		$n = 3$			
	NN	NM	NN	NM	NN	NM
Kernel PCA, $\gamma = 5.77e+7$	82.06, $\sigma = 0.87$	77.65, $\sigma = 1.05$	81.84 $\sigma = 0.87$	76.13, $\sigma = 1.02$	82.50 $\sigma = 1.02$	77.67, $\sigma = 0.94$
Kernel FDA $\gamma = 0.46$	82.30, $\sigma = 0.83$	**82.66**, $\sigma = 0.77$	83.35, $\sigma = 0.88$	**83.77**, $\sigma = 0.91$	84.60, $\sigma = 0.95$	84.98, $\sigma = 0.78$
KPCA+LDA $\gamma = 0.38$	80.62, $\sigma = 1.07$		81.96, $\sigma = 0.95$		84.82, $\sigma = 0.85$	
Kernel DCV $\gamma = 0.46$	82.35, $\sigma = 0.88$		82.84, $\sigma = 0.83$		**85.01**, $\sigma = 0.63$	

TABLE 4.3
Recognition Rates of Methods on the 240 Pixel Averages Database

Linear Methods	**Recognition Rates (%) and Standard Deviations**	
	NN	NM
PCA	**97.07**, $\sigma = 0.47$	91.63, $\sigma = 0.72$
FLDA	93.98, $\sigma = 0.69$	94.53, $\sigma = 0.65$
Direct-LDA	95.85, $\sigma = 0.61$	93.17, $\sigma = 0.63$

Kernel Methods & Gaussian Kernel Parameters	**Recognition Rates (%) and Standard Deviations**					
	Polynomial kernel functions with different degrees				Gaussian kernel function	
	$n = 2$		$n = 3$			
	NN	NM	NN	NM	NN	NM
Kernel PCA, $\gamma = 4.5e4$	96.95, $\sigma = 0.39$	91.88, $\sigma = 0.56$	96.57, $\sigma = 0.47$	91.61, $\sigma = 0.54$	97.05, $\sigma = 0.43$	91.74, $\sigma = 0.72$
Kernel FDA, $\gamma = 1200$	97.83, $\sigma = 0.34$	97.83, $\sigma = 0.34$	98.04, $\sigma = 0.36$	98.04, $\sigma = 0.36$	98.15, $\sigma = 0.30$	98.08, $\sigma = 0.34$
KPCA+LDA, $\gamma = 1200$	97.7, $\sigma = 0.41$		98.05, $\sigma = 0.32$		98.14, $\sigma = 0.31$	
Kernel DCV, $\gamma = 1200$	**98.01**, $\sigma = 0.22$		**98.10**, $\sigma = 0.32$		**98.16**, $\sigma = 0.31$	

We also performed statistical significance tests to evaluate the differences between the recognition rates of the proposed method and the other competing methods from Table 4.2. This test is a null hypothesis statistical test. If the resulting significance is below the desired significance level, the null hypothesis is rejected and the performance difference between two methods is considered to be statistically significant. The details of the test can be found in the Appendix. The results of testing for significance (with significance level of 0.05) in the observed recognition rates are given in Table 4.4 for the Fourier Coefficients database. We compared only the proposed method to the other kernel methods and to the linear method that achieved the best recognition rate among the linear methods. In terms of recognition performance, the term 0 implies the two methods are statistically the same; 1 implies the

proposed method performs better; and -1 implies the proposed method is worse than the compared method in the table. The recognition rates obtained by using the Gaussian kernels were generally observed to be the best overall. With regard to the Gaussian kernels, the proposed method was found to be significantly better than the Kernel PCA and all linear methods with a significance level 0.05 since the PCA Method performed the best out of all linear methods on this database.

Similar to the previous case, the best recognition rate among the linear methods was obtained by the PCA Method followed by the NN classifier for the Pixel Averages Database. The proposed method achieved the highest recognition rates in all cases. Both the Kernel FDA and the KPCA+LDA methods outperformed the FLDA Method whereas the Kernel PCA Method did not outperform the classical linear counterpart. Additionally, we performed statistical significance tests to evaluate the differences between the recognition rates of the proposed method and the other competing methods on the Pixel Averages database. The results of the significance test are given in Table 4.5. The results show that the proposed method significantly outperforms the Kernel PCA and all linear methods in all cases with a significance level of 0.05 on the Pixel Averages database.

TABLE 4.4
Statistical Significance Comparison of Recognition Performances on the Fourier Coefficients Database

Kernel Functions	**KDCV/KPCA**		**KDCV/KFDA**		**KDCV/KPCA+LDA**	**KDCV/PCA**
	NN	**NM**	**NN**	**NM**		
$n = 2$	0	1	0	0	1	0
$n = 3$	1	1	0	-1	1	0
GK	1	1	0	0	0	1

TABLE 4.5
Statistical Significance Comparison of Recognition Performances on the Pixel Averages Database

Kernel Functions	**KDCV/KPCA**		**KDCV/KFDA**		**KDCV/KPCA+LDA**	**KDCV/PCA**
	NN	**NM**	**NN**	**NM**		
$n = 2$	1	1	0	0	1	1
$n = 3$	1	1	0	0	0	1
GK	1	1	0	0	0	1

In general, the test results show that the proposed method generalizes well compared to other kernel approaches for data sets with large number of samples studied here since for both data sets, the proposed method achieves either competitive or the best recognition results. We also conducted some experiments to observe if the recognition performance of the Kernel DCV Method can be increased by incorporating some projection directions from outside the optimal discriminant subspace into the Kernel DCV framework. Only one randomly created training and test set were used for both data sets in these experiments. We used the Gaussian kernels, with the parameters as given in the tables, since these yielded the highest recognition rates. A variation of the PCA+Null Space Method from [129] was employed to add the projection directions coming from outside the optimal discriminant subspace. We split the new within-class scatter matrix, $\widetilde{S}_W^{\Phi}$ (the within-class scatter matrix of the samples obtained after the Kernel PCA process), into its null space $N(\widetilde{S}_W^{\Phi}) = span\{\xi_{r+1},...,\xi_t\}$ and orthogonal complement (i.e., range space) $R(\widetilde{S}_W^{\Phi}) = span\{\xi_1,...,\xi_r\}$ (where r is the rank of $\widetilde{S}_W^{\Phi}$, and $t = rank(S_T^{\Phi})$ is the dimension of the reduced space after Kernel PCA step). Subsequently, all the projection vectors maximizing the between-class scatter in the null space are chosen. These projection vectors

are from the optimal discriminant subspace and there are 9 of them. Then, beginning with these optimal projection vectors, we gradually added new projection vectors from the range space until we reached to the number of $t = 998$ projection vectors, and we computed the corresponding recognition rates. The results for the training and test sets are illustrated in Figure 4.4. As can be seen from the figure, adding new projection directions from outside the optimal discriminant subspace does not increase the performance; in fact the performance can be seen to degrade. Adding projection directions from outside the optimal discriminant subspace also degrades the real-time performance since the added projections no longer produce a unique discriminative common vector for each class. As a result, if one does not utilize a single representative prototype feature vector for each class during classification, the comparisons must be made over all feature vectors of the training set, rather than just over a much smaller number of discriminative common vectors, leading to an increase in the computational cost.

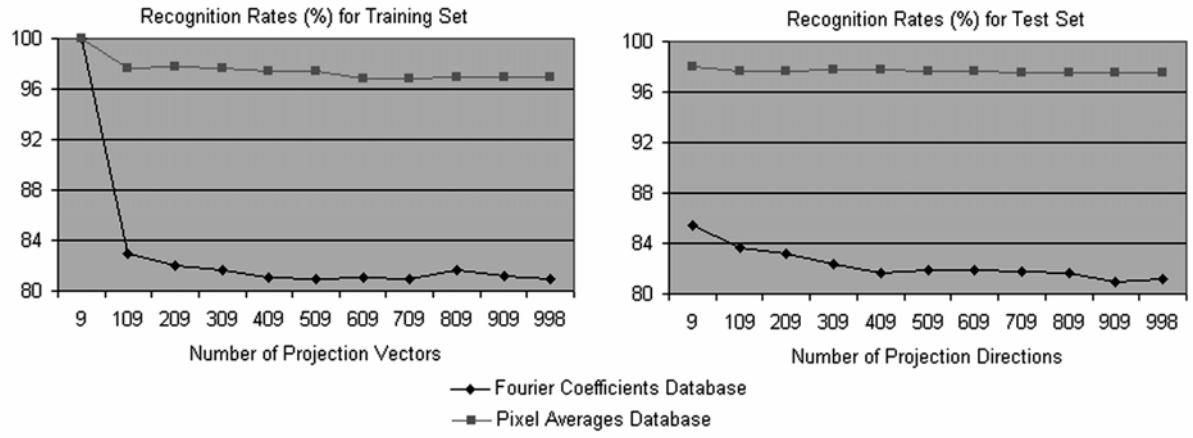


Figure 4.4: Recognition rates as a function of projection vectors that are used for feature extraction.

4.7.2 Experiments with High-Dimensional Sample Spaces

In this group of experiments, we used the ORL (Olivetti-Oracle Research Lab) face database. The ORL face database contains $C = 40$ individuals with 10 images per person. The images are taken at different times with varying lighting conditions, facial expressions, and facial details. All individuals are in an up-right, frontal position (with tolerance for some side movement). The size of the each image is 92x112 pixels. Some individuals from the ORL face database are shown in Figure 3.6.

We randomly selected $N = 3,5,7$ samples from each class for training and the remaining $(10 - N)$ samples of each class were used for testing. This process was repeated 25 times, and 25 different training and test sets were created. The first 5 data sets were used for parameter selection and the rest were used for performance evaluation. We did not apply any pre-processing to the images. The recognition rates for the experiment were found by averaging the recognition rates of each trial. The computed recognition rates and standard deviations for the linear and kernel methods are given in Table 4.6 and Table 4.7, respectively. The best recognition was obtained by the DCV Method among the linear methods in all cases. The recognition performance of the DCV Method is especially superior to the other linear methods when $N = 3$ samples are used for training. As the number of training samples is increased, the difference between the recognition rates of the DCV Method and other linear methods decreases. Similarly, the best recognition results among the kernel methods were obtained by the Kernel DCV Method for all cases.

TABLE 4.6
Recognition Rates of Linear Methods on the ORL Face Database

Number of training samples in each class	**Recognition Rates (%) & Standard Deviations**						
	PCA		**FLDA**		**Direct-LDA**		**DCV**
	NN	**NM**	**NN**	**NM**	**NN**	**NM**	
$N = 3$	86.82, $\sigma = 2.99$	84.78, $\sigma = 3.04$	86.35, $\sigma = 2.91$	86.01, $\sigma = 3.57$	85.48, $\sigma = 3.11$	84.85, $\sigma = 2.58$	**90.60,** $\sigma = 2.58$
$N = 5$	93.75, $\sigma = 1.5$	90.45, $\sigma = 2.16$	92.10, $\sigma = 2.66$	92.47, $\sigma = 2.22$	95.70, $\sigma = 1.37$	95.00, $\sigma = 1.61$	**95.95,** $\sigma = 1.60$
$N = 7$	96.29, $\sigma = 1.78$	92.41, $\sigma = 2.26$	94.33, $\sigma = 2.38$	94.79, $\sigma = 2.24$	97.58, $\sigma = 1.45$	97.29, $\sigma = 1.70$	**97.74,** $\sigma = 1.38$

Similar to the large sample size case, we also performed statistical significance tests to evaluate the differences between the recognition rates of the proposed method and the other competing methods for the ORL face database. The results are given in Table 4.8. Although the proposed method either matches or significantly outperforms the other kernel methods, it does not offer any improvement over the linear methods. In fact, it statistically performs worse than the linear DCV Method for the polynomial kernel with degree 3 for $N = 3$. This can be attributed to the nature of the face images in the database. The images of individuals are mostly in frontal position and the lighting conditions are similar. Therefore the face images in the database are linearly separable. In such cases, using higher order correlations via kernels may degrade the performance as in our case since the problem is close to linearly separable.

TABLE 4.7
Recognition Rates of Kernel Methods on the ORL Face Database

Number of training samples	Kernel functions	Classifier	Recognition Rates (%) & Standard Deviations			
			Kernel PCA $\gamma = 1.06e12$	**Kernel FDA** $\gamma = 3.18e7$	**KPCA+LDA** $\gamma = 3.18e7$	**Kernel DCV** $\gamma = 1.06e8$
$N = 3$	$n = 2$	NN	85.91, $\sigma = 2.94$	89.37, $\sigma = 2.74$	86.78, $\sigma = 3.49$	**90.46**, $\sigma = 2.58$
		NM	83.81, $\sigma = 3.13$	89.37, $\sigma = 2.74$		
	$n = 3$	NN	84.39, $\sigma = 2.81$	87.12, $\sigma = 3.24$	85.05, $\sigma = 3.74$	**88.78**, $\sigma = 3.00$
		NM	81.51, $\sigma = 2.82$	87.12, $\sigma = 3.24$		
	GK	NN	86.82, $\sigma = 2.99$	90.12, $\sigma = 2.46$	91.14, $\sigma = 2.69$	**91.17**, $\sigma = 2.44$
		NM	84.78, $\sigma = 3.04$	89.35, $\sigma = 2.59$		
$N = 5$	$n = 2$	NN	93.20, $\sigma = 1.50$	95.25, $\sigma = 1.69$	93.55, $\sigma = 1.67$	**96.12**, $\sigma = 1.48$
		NM	90.32, $\sigma = 2.41$	95.25, $\sigma = 1.69$		
	$n = 3$	NN	92.57, $\sigma = 1.67$	94.37, $\sigma = 1.51$	92.20, $\sigma = 1.93$	**95.37**, $\sigma = 1.57$
		NM	88.65, $\sigma = 2.85$	94.37, $\sigma = 1.51$		
	GK	NN	93.75, $\sigma = 1.50$	96.32, $\sigma = 1.34$	96.42, $\sigma = 1.31$	**96.55**, $\sigma = 1.17$
		NM	90.45, $\sigma = 2.16$	95.57, $\sigma = 1.57$		
$N = 7$	$n = 2$	NN	95.87, $\sigma = 1.95$	97.08, $\sigma = 1.78$	96.08, $\sigma = 1.87$	**97.66**, $\sigma = 1.70$
		NM	92.70, $\sigma = 2.24$	97.08, $\sigma = 1.78$		
	$n = 3$	NN	95.58, $\sigma = 1.69$	96.54, $\sigma = 1.60$	95.33, $\sigma = 1.99$	**97.41**, $\sigma = 1.73$
		NM	91.16, $\sigma = 2.69$	96.54, $\sigma = 1.60$		
	GK	NN	96.41, $\sigma = 1.93$	98.16, $\sigma = 1.52$	97.83, $\sigma = 1.30$	**98.25**, $\sigma = 1.32$
		NM	92.20, $\sigma = 2.34$	98.04, $\sigma = 1.24$		

TABLE 4.8
Statistical Significance Comparison of Recognition Performances on the ORL Face Database

Number of training samples	Kernel functions	KDCV/KPCA		KDCV/KFDA		KDCV/ KPCA+LDA	KDCV/DCV
		NN	NM	NN	NM		
$N = 3$	$n = 2$	1	1	0	0	1	0
	$n = 3$	1	1	0	0	1	-1
	GK	1	1	0	1	0	0
$N = 5$	$n = 2$	1	1	0	0	1	0
	$n = 3$	1	1	1	1	1	0
	GK	1	1	0	1	0	0
$N = 7$	$n = 2$	1	1	0	0	1	0
	$n = 3$	1	1	0	0	1	0
	GK	1	1	0	0	0	0

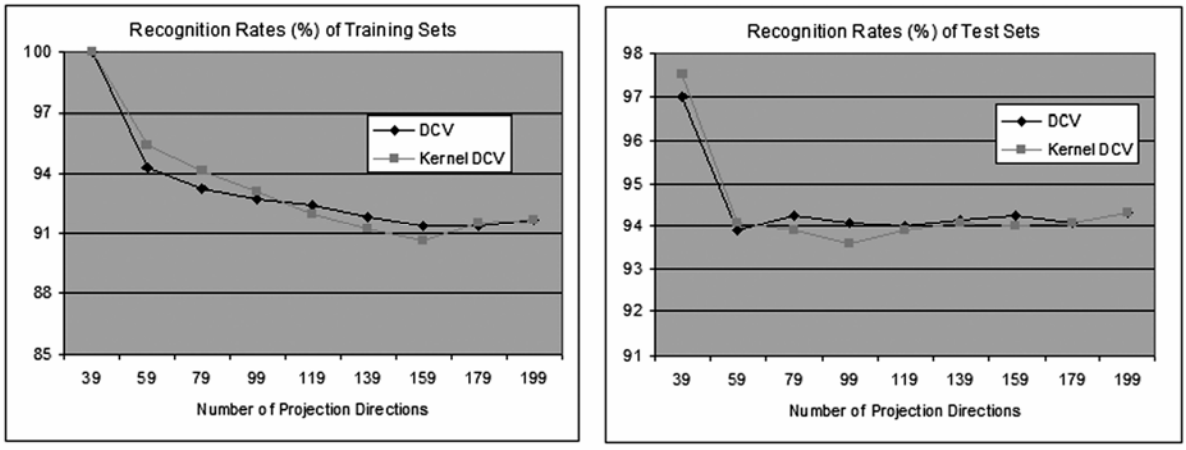


Figure 4.5: Recognition rates as a function of projection vectors that are used for feature extraction.

Finally, we carried out experiments in order to observe whether the performance of the DCV and the Kernel DCV methods can be increased by adding projection directions from outside the optimal discriminant subspace. The same procedure was followed as in the previous subsection. These experiments were performed on the data set using $N = 5$ samples

for training. The Gaussian kernel with parameter $\gamma = 1.06e8$ was used for the Kernel DCV Method. For both methods, starting with 39 optimal projection vectors, we gradually added new projection vectors from outside the optimal discriminant subspace until we reached the number $t = 199$ of projection vectors. The results are given in Figure 4.5. As can be seen, adding new projection vectors degraded the performance of the method similar to the large sample size case.

In general, these results show that the proposed kernel method leads to a reliable input-output mapping for the data sets with a high-dimensional space by using only a few training set samples.

4.8 Discussion

We have seen in the described experiments that when the dimension of the sample space was smaller than the size of the training set, the kernel methods typically produced better results than the linear methods. Although the Kernel PCA did not improve the classical PCA Method significantly, the supervised kernel approaches, the Kernel FDA and the KPCA+LDA methods, outperformed the FLDA Method significantly. In many cases the proposed method outperformed the other kernel methods. Unlike the results obtained for the data sets from the first population, there is not a significant difference between the recognition rates of the linear and the kernel methods for the face database since the face samples were linearly separable. The DCV Method outperformed all other linear methods in all cases. Similarly, the Kernel DCV Method outperformed all other kernel methods in all cases. The Kernel DCV Method may improve the recognition results of the linear DCV Method on different face databases having nonlinear and complex distributions.

The recognition results of the kernel methods may be improved for different kernels that fulfill Mercer's theorem. However, we did not attempt to find better kernels since our aim here was to compare the accuracy of the Kernel DCV Method with other kernel techniques. The test results show that the projection vectors coming from the optimal discriminant subspace are the best suited set of projection directions for feature extraction. Another advantage of the Kernel DCV Method is its real-time performance. The proposed method and the KPCA+LDA Method yield the highest real-time efficiency among the kernel methods. In these methods, after a test image is projected onto the (C-1) optimal projection vectors, the feature vector of the test sample is compared to C discriminative common vectors only, in sharp contrast to all other methods, where it must be compared to all training set feature vectors if the nearest neighbor algorithm is used. Thus, if we assume that each class has N samples and each kernel method uses (C-1) projection vectors for feature extraction, then the computational complexity of the other kernel approaches will be N times greater than the computational complexity of the Kernel DCV and the KPCA+LDA methods.

4.9 Conclusion

In this chapter we proposed a new nonlinear method that uses kernel functions for recognition. The proposed method combines kernel-based methodologies with the optimal discriminant subspace concept and finds the projection vectors coming from the optimal discriminant subspace in the nonlinearly mapped higher-dimensional space. Under certain conditions, all training set samples in each class produce a distinct vector called the discriminative common vector representing that class. Thus a 100% recognition rate is guaranteed for the training set samples even though they are not linearly separable in the

original sample space. To assess the performance of the proposed method, we performed several tests. First, we compared the proposed method with the methods that use projection directions from outside the optimal discriminant subspace. The proposed method outperformed other kernel feature extraction methods in most of the cases. Then, we generated a new set of projection vectors by adding new projection vectors from outside the optimal discriminant subspace to the optimal projection vectors. We then used these new vectors for feature extraction. However, this process degraded the performance of the method presented. The results show that the generalization ability of the proposed method is comparable to all tested kernel approaches. Also the fact that the test sample feature vectors are compared only to the discriminative common vectors, as opposed to all training set sample feature vectors, makes the proposed method ideal for real-time applications.

CHAPTER V

LINEAR AND NONLINEAR SUBSPACE CLASSIFIERS

Most of the classifiers that carry out computations at full dimensionality may not deliver the advantages of high-dimensional sample spaces if there are insufficient training sample patterns. However, unlike other classifiers, subspace classifiers are shown to work well in recognition tasks with high-dimensional sample spaces. Most of the assumptions upon which the subspace classifiers are founded, hold in high-dimensional sample spaces. Therefore, this chapter is devoted to linear and nonlinear subspace classifier methods. In addition, based on our findings in Chapter 3, we propose a variation of a linear subspace classifier here. Then, this method is generalized to the nonlinear case by utilizing the kernel trick. Finally, we provide experimental results and our conclusions at the end of the chapter.

5.1 An Introduction to the Linear Subspace Classifiers

Subspaces were originally introduced for compression and optimal reconstruction of multi-dimensional data. Watanabe *et al*. proposed the first subspace method of pattern recognition to classify and represent the multi-dimensional pattern vectors [117]. The motivation behind the introduction of subspace classifiers is that each class has its own set of representative features differing from those of the other classes [90]. Therefore, the most conspicuous features are extracted from each class by using the corresponding training samples in the hope that those features also carry the most important discriminatory information.

In subspace methods, it is assumed that each class corresponds to a lower-dimensional subspace of the original feature space. The subspaces representing classes are defined in terms of basis vectors that are linear combinations of the sample vectors of corresponding classes. For this reason, basis vectors spanning these subspaces must be computed first. Then, an unknown test sample vector is classified based on the length of the projections of that sample onto each of the subspaces or, alternatively, on the distances of the test vector from these subspaces.

5.2 Bases and Decision Rules

Suppose there are C classes denoted by $\omega^{(1)}, \omega^{(2)}, ..., \omega^{(C)}$ where the i-th class contains N_i samples. Let $x_m^i \in \Re^d$ be the m-th sample of the i-th class. Let $L^{(1)}, L^{(2)}, ..., L^{(C)}$ are the subspaces representing classes. Suppose each subspace is spanned by l_i orthonormal basis vectors $\{w_1^i, ..., w_{l_i}^i\}$ in $\Re^d$. Let $W^{(i)}$ be the matrix whose columns are the orthonormal basis vectors spanning $L^{(i)}$, i.e.,

$$L^{(i)} = \{x^i \mid x^i = \sum_{j=1}^{l_i} \zeta_j w_j^i, \zeta \in \Re\}. \tag{5.1}$$

Then, the projection matrix (or orthogonal projection operator) $P^{(i)} \in \Re^{dxd}$ of any subspace $L^{(i)}$ can be computed by

$$P^{(i)} = \sum_{j=1}^{l_i} w_j^i (w_j^i)^T = W^{(i)} W^{(i)^T}. \tag{5.2}$$

Thus, the projection matrix $P^{(i)}$ of $L^{(i)}$ is symmetric. Note also that although the basis is not unique for a subspace, the projection matrix $P^{(i)}$ is unique and completely defines the subspace $L^{(i)}$. If the projection matrix is given, the basis vectors can be found by an eigen-

decomposition of the projection matrix. The eigenvectors corresponding to the eigenvalues of $P^{(i)}$, which are equal to 1, form an orthonormal basis for $L^{(i)}$.

Projection matrices have two important properties. First, the projection of any *d*-dimensional vector x_{test} onto $L^{(i)}$ can be found by

$$\hat{x}_{test}^{i} = P^{(i)} x_{test}. \tag{5.3}$$

Second, the projection of any vector from $L^{(i)}$ is equal to itself. Thus, if we combine these two characteristics, it implies that $P^{(i)^2} = P^{(i)}$, which also means that $P^{(i)}$ is idempotent. The projection matrix $\overline{P}^{(i)}$ of the orthogonal complement of $L^{(i)}$ (denoted by $L^{(i)^{\perp}}$) can be found by

$$\overline{P}^{(i)} = (I - P^{(i)}), \tag{5.4}$$

where $I \in \Re^{dxd}$ is the identity matrix. Therefore, the projection of x_{test} onto $L^{(i)^{\perp}}$ can be computed by

$$\tilde{x}_{test}^{i} = (I - P^{(i)}) x_{test}, \tag{5.5}$$

which implies that

$$x_{test} = \hat{x}_{test}^{i} + \tilde{x}_{test}^{i}. \tag{5.6}$$

The vector $\tilde{x}_{test}^{i}$ is also called the residual. Figure 5.1, which is adopted from [90], illustrates the projection of a 3-dimensional vector onto a 2-dimensional subspace. In addition, the projection matrices $P^{(i)}$ and $\overline{P}^{(i)}$ of two orthogonal subspaces fulfill the condition

$$P^{(i)} \overline{P}^{(i)} = \overline{P}^{(i)} P^{(i)} = 0. \tag{5.7}$$

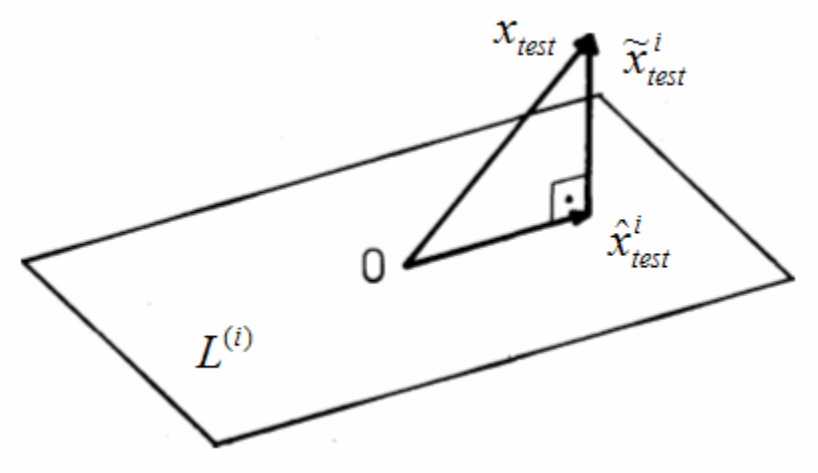


Figure 5.1: Projection $\hat{x}^i_{test}$ of x_{test} on $L^{(i)}$ and the orthogonal residual $\tilde{x}^i_{test}$.

It is not practical to store and use the projection matrices explicitly for computations, especially if the number of the samples in each class is smaller than the dimensionality of the sample space *d*. Instead, we use the basis vectors for all computations. However, we will use projection matrices for the purposes of abbreviation.

In subspace methods, a test sample is classified according to the following classification rule:

$$\text{If } x^T_{test} P^{(i)} x_{test} > x^T_{test} P^{(j)} x_{test}, j \neq i, \text{ then assign } x_{test} \text{ to the class } \omega^{(i)}. \tag{5.8}$$

Since $P^{(i)}$ is idempotent we can rewrite $x^T_{test} P^{(i)} x_{test}$ as

$$x^T_{test} P^{(i)} x_{test} = x^T_{test} P^{(i)} P^{(i)} x_{test} = \| P^{(i)} x_{test} \|^2. \tag{5.9}$$

We know that $P^{(i)} x_{test}$ is the orthogonal projection of x_{test} onto $L^{(i)}$. Thus, the classification rule can also be written as

$$\text{If } \| P^{(i)} x_{test} \|^2 > \| P^{(j)} x_{test} \|^2, j \neq i, \text{ then assign } x_{test} \text{ to the class } \omega^{(i)}. \tag{5.10}$$

In other words, we assign x_{test} to the class where the length of the projection of x_{test} is maximum. Thus, we need only compute the norm of the vector $\| \hat{x}^i_{test} \|$ for each class during classification. The squared norm can be computed more efficiently by using the basis vectors as follows:

$$\| \hat{x}^i_{test} \|^2 = \sum_{j=1}^{l_i} ((w^i_j)^T x_{test})^2 = \| W^{(i)^T} x_{test} \|^2 . \quad (5.11)$$

Then the classification rule becomes

$$\text{if } \| W^{(i)^T} x_{test} \|^2 > \| W^{(j)^T} x_{test} \|^2, j \neq i, \text{ then assign } x_{test} \text{ to the class } \omega^{(i)}, \quad (5.12)$$

or

$$\text{if } \| \hat{x}^i_{test} \|^2 > \| \hat{x}^i_{test} \|^2, j \neq i, \text{ then assign } x_{test} \text{ to the class } \omega^{(i)} . \quad (5.13)$$

Since $\hat{x}^i_{test}$ and $\tilde{x}^i_{test}$ are orthogonal we can alternatively use the following decision rule,

$$\text{if } \| \tilde{x}^i_{test} \|^2 < \| \tilde{x}^i_{test} \|^2, j \neq i, \text{ then assign } x_{test} \text{ to the class } \omega^{(i)} . \quad (5.14)$$

All these decision rules show that the length of the input vector x_{test} does not contribute to the classification decision which implies that the subspace classifier is invariant to the input vector length. Therefore, without a loss of generality, we can set the norms of pattern vectors to 1 by normalization before classification.

5.3 The Class Featuring Information Compression (CLAFIC) Method

The CLAFIC Method was first proposed by Watanabe *et al*. for classification of multi-dimensional data [117]. It aims to maximize the ratio

$$\sum_{i=1}^{C} E(x^T P^{(i)} x \mid x \in \omega^{(i)}) = \sum_{i=1}^{C} E(\sum_{j=1}^{l_i} (x^T (w^i_j)^T)^2 \mid x \in \omega^{(i)}), \quad i = 1,...,C, \quad j = 1,...,l_i , \quad (5.15)$$

subject to $\| w_j^i \| = 1$, where $E(.)$ represents the expectation operator. This method employs the PCA or the Karhunen-Loeve transform to compute the basis vectors $\{w_1^i, ..., w_{l_i}^i\}$ spanning the subspaces $L^{(i)}$. The basis vectors are computed through eigen-decomposition of class correlation matrices R_i defined as

$$R_i = \frac{1}{N_i} \sum_{m=1}^{N_i} x_m^i (x_m^i)^T = \frac{1}{N_i} X^{(i)} X^{(i)^T}, \quad i = 1, ..., C, \tag{5.16}$$

where $X^{(i)}$ is the matrix whose columns are the samples of the *i*-th class. Note that the mean vectors μ_i of classes are not subtracted from the data samples. The matrix R_i is a positive semi-definite matrix; hence, all the eigenvalues are larger than or equal to 0. The l_i eigenvectors corresponding to the largest eigenvalues of R_i are chosen as basis vectors. There are various strategies for choosing the subspace dimensions l_i. One way is to set all the l_is to be equal to a fixed value *l*. Then, the optimal value of *l* can be chosen from the error curves [71]. The second way employs eigenvalues for choosing the dimensions of subspaces. Let the eigenvalues of R_i be ordered as $\lambda_1^i \geq \lambda_2^i \geq ... \geq \lambda_{r_i}^i > 0$, where r_i is the rank of the matrix R_i. The dimension of $L^{(i)}$ is selected as the value by which the ratio of cumulative sums $\kappa = \sum_{j=1}^{l_i} \lambda_j^i / \sum_{j=1}^{r_i} \lambda_j^i$ exceeds a threshold. Typical values of the threshold lie between $0.9 \leq \kappa \leq 1$.

The algorithm of the CLAFIC Method for high dimensional databases with the small sample size problem can be summarized as follows:

Step 1: For each class, compute the nonzero eigenvalues and corresponding eigenvectors of $R_i \in \Re^{dxd}$ by using the smaller matrix, $X^{(i)^T} X^{(i)} \in \Re^{N_i x N_i}$, where $R_i = \frac{1}{N_i} X^{(i)} X^{(i)^T}$. There are at most N_i nonzero eigenvalues for each class.

Step 2: Select the most significant l_i eigenvectors by employing one of the procedures described above and form the matrices

$$W^{(i)} = [w_1^i \quad w_2^i \quad ... \quad w_{l_i}^i], \quad i = 1,...,C. \tag{5.17}$$

The columns of each matrix $W^{(i)}$ form a basis for the corresponding class.

Step 3: To classify a test sample, x_{test} compute the squared norms of vectors $\| \hat{x}_{test}^i \|^2 = \| W^{(i)^T} x_{test} \|^2$ for each class and assign the test sample to the class which gives the maximum value.

Iijima *et al*. introduced a variation of the CLAFIC known as the Multiple Similarity Method (MSM) in which individual weights were used for all basis vectors during the computation of the squared norms of the projected vectors [56]. Each basis vector was weighted by its corresponding eigenvalues as follows:

$$\| \hat{x}_{test}^i \|^2 = \sum_{j=1}^{l_i} \frac{\lambda_j^i}{\lambda_1^i} ((w_j^i)^T x_{test})^2, \quad i = 1,...,C. \tag{5.18}$$

Wold used the so-called SIMCA Method which uses linear regression models for the representation of classes [121].

All these methods can easily be applied to high-dimensional databases with the small sample size problem since the eigenvalues and corresponding eigenvectors can be computed by using smaller matrices. However, the following methods discussed below are not suitable for databases with high-dimensional sample spaces.

5.4 Other Subspace Classifier Methods

The methods CLAFIC, MSM, and SIMCA discussed above, each has one serious drawback [71], [90]. They optimize problematic criterion functions which lead to representation of each class independently of the other classes. Therefore these methods do not necessarily find the optimal solution for the classification of data samples. In order to solve this problem, the following criterion was proposed

$$\min \sum_{\substack{j=1 \\ j\neq i}}^{C} E(x^T P^{(i)} x \mid x \in \omega^{(i)}) - E(P^{(i)} x \mid x \in \omega^{(i)}), \qquad i = 1,...,C. \tag{5.19}$$

This criterion function is optimized if the eigenvectors corresponding to the smallest eigenvalues of $\sum_{\substack{j=1 \\ j\neq i}}^{C} R_j - R_i$ are chosen as the basis vectors for $L^{(i)}$ [90]. However, this method could not improve the recognition rates significantly compared to the CLAFIC Method. Fukunaga and Koontz proposed a new method known as the Generalized Fukunaga-Koontz Method for the two-class problem, which enabled the selection of the basis vectors in such a way that the projections onto the so-called rival subspaces are minimized [40]. It was generalized to the multi-class case by Kittler [67]. This method suggests choosing the eigenvectors corresponding to the largest eigenvalues of the generating matrix

$$\Psi^{(i)} = R_i + \sum_{\substack{j=1 \\ j\neq i}}^{C} (I - R_j), \quad i = 1,...,C \tag{5.20}$$

for the basis vectors of the subspace $L^{(i)}$.

In general the CLAFIC Method may produce overlapping and non-orthogonal subspaces. This is problematic since the discrimination between classes weakens if the subspace dimensions are small. On the other hand, if we increase the subspace dimensions the classification decisions may be dominated by the less robust directions. This problem can be

avoided by removing the intersections of subspaces and orthogonalizing the remaining subspaces. Watanabe *et al*. introduced the Method of Orthogonal Subspaces (MOSS) to accomplish this task [118]. Orthogonalization process can be accomplished by employing Observation 3.1 such that the basis vectors of each class are chosen as the eigenvectors corresponding to the eigenvalues of 1 of the following matrix

$$\Psi^{(i)} = a_i P^{(i)} + \sum_{\substack{j=1 \\ j \neq i}}^{C} a_j (I - P^{(j)}), \qquad i = 1,...,C, \tag{5.21}$$

where $\sum_{i=1}^{C} a_i = 1$.

Lastly, iterative learning subspace methods, capable of learning in a decision fashion have been proposed [69], [70]. These methods modify the basis vectors of classes in order to diminish the number of misclassifications during the training phase. It was reported that they produce better recognition results compared to other subspace classifiers. However, all these methods discussed above are not applicable to high-dimensional databases with the small sample size problem. In particular, they require the use of large class correlation matrices or class projection matrices explicitly, and the smaller matrices to compute the basis vectors as in CLAFIC cannot be used.

5.5 The Common Vector Method

Linear subspace methods are based on the assumption that the most representative features of each class carry the most discriminatory information for discrimination; hence, these methods typically try to optimize criterion functions that may not be compatible with classification purposes. Therefore, Gulmezoglu *et al*. proposed the Common Vector (CV) Method using a different criterion for classification tasks where the number of samples in

each class is smaller than or equal to the dimensionality of the sample space [44], [45]. The CV Method aims to extract features that are common for all samples in each class. In order to accomplish its goal, the method eliminates features that are in the direction of the eigenvectors corresponding to the nonzero eigenvalues of the covariance matrices (or scatter matrices) of classes. It has been demonstrated that these features also carry the most discriminatory information for classification of samples. The CV Method has been successfully applied to isolated word and face recognition problems [13], [45].

The scatter matrix of each class is defined as

$$\begin{aligned} S_i &= \sum_{m=1}^{N_i} (x_m^i - \mu_i)(x_m^i - \mu_i)^T \\ &= A_i A_i^T, \quad i = 1,...,C, \quad m = 1,...,N_i, \end{aligned} \tag{5.22}$$

where μ_i is the mean of the samples in the *i*-th class and $A_i \in \Re^{dxN_i}$ is given by

$$A_i = [(x_1^i - \mu_i) \quad ... \quad (x_{N_1}^i - \mu_i)]. \tag{5.23}$$

Each sample in the training set is represented as,

$$x_m^i = x_{m,dif}^i + x_{com}^i + \varepsilon_m^i, \quad i = 1,...,C, \quad m = 1,...,N_i, \tag{5.24}$$

where x_{com}^i is a unique vector representing the *i*-th class, and ε_m^i is the error vector term. The CV Method aims to minimize the criterion given below for each class,

$$F_i = \sum_{m=1}^{N_i} \| \varepsilon_m^i \|^2 = \sum_{m=1}^{N_i} \| x_m^i - x_{m,dif}^i - x_{com}^i \|^2, \; i = 1,...,C. \tag{5.25}$$

It was shown that if the common vector x_{com}^i is chosen as

$$x_{com}^i = x_m^i - x_{m,dif}^i, \quad i = 1,...,C, \quad m = 1,...,N_i, \tag{5.26}$$

then F_i is minimized such that $F_i = 0$, where $x_{m,dif}^i$ represents the projection of x_m^i onto the range space of the scatter matrix of the *i*-th class [45]. This projection can be computed by

$$x_{m,dif}^{i} = P^{(i)} x_{m}^{i}, \quad i = 1,...,C, \quad m = 1,...,N_i, \tag{5.27}$$

where $P^{(i)}$ is the orthogonal projection operator of the range space of S_i. Thus, equation (5.26) can also be written as,

$$x_{com}^{i} = (I - P^{(i)}) x_{m}^{i} = \overline{P}^{(i)} x_{m}^{i}, \quad i = 1,...,C. \tag{5.28}$$

As can be seen in equations (5.26) and (5.28), x_{com}^{i} is unique for each class and does not depend on the choice of the sample vector (i.e., x_{com}^{i} is independent of the sample index *m*) [45]. Since the projection of x_{m}^{i} onto the range space $R(S_i)$ of S_i is removed in order to compute the common vectors, each common vector, x_{com}^{i}, is a linear combination of the eigenvectors corresponding to the zero eigenvalues of S_i. That is, the CV Method employs the directions coming from the null space $N(S_i)$ of S_i for representation of each class.

To recognize a test sample x_{test}, the test sample is first projected onto the null space of the scatter matrix of each class separately; then, the projected vector is compared to the common vector of each class using the Euclidean distance. The unknown test sample is assigned to the class which gives the minimum distance.

The method described above can be summarized as follows:

Step 1: Compute the nonzero eigenvalues and the corresponding eigenvectors of the scatter matrix S_i of each class using the matrix $A_i^T A_i \in \Re^{N_i x N_i}$, where $S_i = A_i A_i^T \in \Re^{dxd}$, and A_i is given by (5.23). Normalize the computed eigenvectors and set $U^{(i)} = [u_1^i \quad ... \quad u_{r_i}^i]$, where r_i is the rank of S_i.

Step 2: Project any sample from each class onto the null space of S_i and compute the common vector of each class by,

$$x_{com}^{i} = x_{m}^{i} - P^{(i)} x_{m}^{i} = x_{m}^{i} - U^{(i)} U^{(i)^T} x_{m}^{i}, \quad i = 1,...,C, \quad m = 1,...,N_i \tag{5.29}$$

Note that common vectors x_{com}^{i}, $i = 1,...,C$, are unique for each class and independent of the sample index *m*.

Step 3: Project a test sample onto the null spaces of S_i to obtain the feature vectors by

$$\Omega_{test}^{i} = x_{test} - P^{(i)} x_{test}, \; i = 1,...,C. \tag{5.30}$$

Compute the Euclidean distance between the test sample feature vector and the common vector of each class by,

$$\kappa_i = \| \Omega_{test}^{i} - x_{com}^{i} \|, \; i = 1,...,C. \tag{5.31}$$

Assign the test sample to the class which produces the minimum distance.

5.5.1 Computing Common Vectors by Using the Difference Subspace and the Gram-Schmidt Orthogonalization Procedure

The algorithm described above uses the eigenvectors of the scatter matrices of classes to compute an orthonormal basis for $R(S_i)$. There are more efficient ways to compute an orthonormal basis for each class using the Gram-Schmidt orthogonalization procedure as described below.

To compute common vectors, we first choose any of the sample vectors from each class as the subtrahend vector and then compute the difference vectors b_j^i ($j = 1,...,N_i - 1$). Then, assuming that the first sample of each class is taken as the subtrahend vector, we have

$$b_j^i = x_{j+1}^i - x_1^i, \;\; i = 1,...,C, \;\; j = 1,...,N_i - 1. \tag{5.32}$$

Each subspace, which is spanned by these difference vectors, is called the *difference subspace* of the corresponding class, and it is represented by B_i. From Theorem 3.3, we

know that the difference subspace B_i and $R(S_i)$ are same, i.e., $B_i = R(S_i)$. As described previously, assuming that the difference vectors are linearly independent, the orthogonal projection operator of spaces B_i and $R(S_i)$ can be computed by,

$$P^{(i)} = D^{(i)}(D^{(i)^T} D^{(i)})^{-1} D^{(i)^T}, \quad i = 1,...,C, \tag{5.33}$$

where $D^{(i)} = [b_1^i \quad b_2^i \quad ... \quad b_{N_i-1}^i]$. However, the direct computation of the projection matrix is not practical since the size of the projection matrix is very large for high-dimensional sample spaces. However, we can compute projections efficiently by using the basis vectors. Therefore, linearly independent difference vectors are orthonormalized by using the Gram-Schmidt orthogonalization procedure to obtain an orthonormal basis for each class. Let $\widetilde{U}^{(i)} = [\beta_1^i \quad \beta_2^i \quad ... \quad \beta_{N_i-1}^i]$ be the matrix whose columns are the computed orthonormal basis vectors after applying the Gram-Schmidt orthogonalization procedure. Then, the common vector of each class can be obtained using the formula,

$$x_{com}^i = x_m^i - \widetilde{U}^{(i)}\widetilde{U}^{(i)^T} x_m^i, \quad i = 1,...,C, \quad m = 1,...,N_i. \tag{5.34}$$

The common vectors obtained through this procedure are the same as those obtained by using the eigenvectors of scatter matrices since the projection matrices satisfy the relation

$$P^{(i)} = U^{(i)}U^{(i)^T} = \widetilde{U}^{(i)}\widetilde{U}^{(i)^T}, \quad i = 1,...,C. \tag{5.35}$$

The algorithm described above can be summarized as follows:

Step 1: Find the linearly independent vectors b_j^i that span the difference subspace B_i, and set $B_i = span\{b_1^i,....,b_{r_i}^i\}$ for each class. There are totally r_i linearly independent vectors for each class, where r_i is at most $N_i - 1$.

Step 2: Apply the Gram-Schmidt orthogonalization procedure to obtain an orthonormal basis $\beta_1^i,....,\beta_{r_i}^i$ for B_i and set $\widetilde{U}^{(i)} = [\beta_1^i \quad .... \quad \beta_{r_i}^i]$.

Step 3: Choose any sample from each class and project it onto B_i to obtain common vectors by using (5.34).

Step 4: Project a test sample onto the null spaces of S_i to obtain the feature vectors by

$$\Omega_{test}^i = x_{test} - \widetilde{U}^{(i)}\widetilde{U}^{(i)^T} x_{test}, \quad i = 1,..., C. \tag{5.36}$$

Compute the Euclidean distance between the test sample feature vector and the common vector of each class by,

$$\kappa_i = \| \Omega_{test}^i - x_{com}^i \|, \quad i = 1,..., C. \tag{5.37}$$

Assign the test sample to the class which produces the minimum distance.

5.6 A Variation of the Common Vector Method

In Lemma 3.1, we showed that the null space of the total scatter matrix of the pooled data does not contain any discriminative information for classification of data samples. Therefore, this subspace can be discarded from our consideration in the CV Method. Then, the new subspace representing each class will be defined as the intersection of the null space of that class' scatter matrix and the range space of the scatter matrix of the pooled data. As shown in Theorem 5.1 below, the projection matrix of the null space $N(S_i)$ of the scatter matrix of the *i*-th class and the projection matrix of the range space $R(S_T)$ of the scatter matrix of the pooled data, commute in the sense that

$$P^{(i)}P = PP^{(i)}, \tag{5.38}$$

where $P^{(i)}$ is the projection matrix of $N(S_i)$, and P is the projection matrix of $R(S_T)$. Therefore, the projection matrix $P_{\text{int}}^{(i)}$ of the intersection $N(S_i) \cap R(S_T)$ for each class can be found as

$$P_{\text{int}}^{(i)} = P^{(i)}P = PP^{(i)}, \quad i = 1,...,C. \tag{5.39}$$

Theorem 5.1: Let P and $P^{(i)}$ be the projection matrices of the subspaces $R(S_T)$ and $N(S_i)$, $i = 1,...,C$, respectively. Then P and $P^{(i)}$ commute, i.e.,

$$P^{(i)}P = PP^{(i)}, \quad i = 1,...,C. \tag{5.40}$$

Proof: Let $L^{(1)} = R(S_T)$ and, for any fixed *i*, let $L^{(2)} = N(S_i)$. Clearly, $L^{(1)^\perp} = N(S_T)$ and $L^{(2)^\perp} = R(S_i)$. By Lemma 3.3,

$$\begin{aligned} N(S_T) &= N(S_B + S_1 + ... + S_C) \\ &= N(S_B) \cap N(S_1) \cap ... \cap N(S_C), \end{aligned} \tag{5.41}$$

and, in particular, $N(S_T) \subset N(S_i)$, which, together with the fact that $N(S_i) \perp R(S_i)$, shows that

$$N(S_T) \perp R(S_i) \text{ or } L^{(1)^\perp} \perp L^{(2)^\perp}. \tag{5.42}$$

The assertion of the theorem now follows from Lemma 3.2. □

The basis vectors spanning each mentioned intersection space $P_{\text{int}}^{(i)}$ can be found by using an eigen-decomposition. More precisely, the eigenvectors corresponding to the eigenvalues 1 of $P_{\text{int}}^{(i)}$ span the intersection subspaces representing the classes of interest. However, this approach is not always practical since the size of the projection matrices can be very large. On the other hand, since the projection matrices commute, we can first project the samples onto $R(S_T)$ and then find the null spaces of the classes in the transformed space, so as to

compute basis vectors of the intersection subspaces. The algorithm that implements this idea can be summarized as follows:

Step 1: *Projection of the training set samples onto* $R(S_T)$*:*

i) Compute the nonzero eigenvalues and corresponding eigenvectors u_k of S_T using the matrix $A_T^T A_T \in \Re^{MxM}$, where $S_T = A_T A_T^T \in \Re^{dxd}$ and A_T is given by (3.6). Set $U = [u_1 \quad ... \quad u_r]$, where *r* is the rank of S_T.

ii) Project the training set samples onto $R(S_T)$ by

$$\tilde{x}_m^i = U^T (x_m^i - \mu), \quad i = 1,..., C, \qquad m = 1,..., N_i. \tag{5.43}$$

Step 2: *Finding the null spaces of classes in the transformed space:* In the transformed space, the new scatter matrices of the classes will be

$$\tilde{S}_i = U^T S_i U, \qquad i = 1,..., C. \tag{5.44}$$

Apply eigen-decomposition to each covariance matrix, $\tilde{S}_i \in \Re^{rxr}$. Let q_k^i be the eigenvectors corresponding to the zero eigenvalues of $\tilde{S}_i$. Set $Q^{(i)} = [q_1^i \quad ... \quad q_{n_i}^i]$, where n_i is the dimensionality of $N(\tilde{S}_i)$.

Step 3: *Computation of the final basis vectors of the intersection space* $N(S_i) \cap R(S_T)$*:* The final basis vectors spanning the intersection subspaces will be

$$W^{(i)} = UQ^{(i)}, \qquad i = 1,..., C. \tag{5.45}$$

Note that the basis vectors span the intersection subspace $N(S_i) \cap R(S_T)$ and therefore the following holds:

$$P_{\text{int}}^{(i)} = W^{(i)} W^{(i)^T}, \qquad i = 1,..., C. \tag{5.46}$$

When the samples of each class are projected onto their corresponding intersection subspace, the feature vector $\Omega_{com}^{i} = [< x_m^i, w_1^i > \quad .... \quad < x_m^i, w_{n_i}^i >]^T$ of each sample is the same for all samples in that class. These feature vectors are called the common vectors, as in the CV Method. To recognize a test sample, we compute the Euclidean distance between the test sample feature vector and the common vector of each class

$$\kappa_i = \| \Omega_{test}^{i} - \Omega_{com}^{i} \|, \; i = 1,..., C . \tag{5.47}$$

Then we assign the test sample to the class that minimizes this distance. This method produces the same results as in the CV Method; however, the training phase requires more computations compared to the CV Method. Although this method may not appear useful at first glance, this idea enables us to extend the common vector idea to the nonlinear case. In the following sections, we propose a new nonlinear method by incorporating the kernel trick into the procedure introduced here.

5.7 An Introduction to the Nonlinear Subspace Classifiers

As we discussed earlier, recognition performance of linear subspace classifiers may be degraded because of overlapping subspaces. The MOSS Method, which has been proposed to avoid this problem, reduces the dimensionality of the subspaces by removing the overlapping regions. However, the worst effects of overlapping subspaces can be avoided by increasing the dimensionality of the sample space as opposed to reducing it. The original sample space can be mapped nonlinearly to some higher-dimensional feature space by using the kernel trick explained in Chapter 4. Increasing the dimensionality of the sample space spreads the data over a greater volume. This process reduces the overlap between the subspaces and enhances the potential for discrimination. Since the transformed space is nonlinearly related

to the original sample space, these approaches assume samples of each class lie in a nonlinear subspace.

The samples in a high-dimensional space with the small sample size problem are usually independent. Thus, the overlapping problem is not typical for the databases with high-dimensional spaces. However, this problem occurs when the number of samples in the training set is larger than the dimensionality of the sample space. It has been reported that the recognition rates were significantly improved by using nonlinear subspace classifiers over the linear subspace classifiers [3], [110]. In the following sections, we examine these nonlinear subspace classifiers and propose a new nonlinear subspace classifier that applies the variation of the CV Method in the nonlinearly mapped space.

5.8 The Kernel CLAFIC Method

The Kernel CLAFIC Method was proposed by Tsuda [110] and Balachander [3] at the same time. The method employs the Kernel PCA Method for computing the subspaces that represent the classes. However, it uses the correlation matrix of the mapped samples as opposed to the Kernel PCA, which uses the scatter matrix of samples. The correlation matrix of each class in $\Im$ can be expressed as

$$R_i^{\Phi} = \frac{1}{N_i} \sum_{m=1}^{N_i} \phi(x_m^i)\phi(x_m^i)^T = \frac{1}{N_i} \Phi^{(i)} \Phi^{(i)T}, \qquad i = 1,...,C, \tag{5.48}$$

where $\Phi^{(i)}$ is the matrix whose columns are the mapped samples of the *i*-th class in $\Im$. The rank of each matrix is determined by the number of samples in each class. Since samples in the mapped space are typically linearly independent, the rank of each matrix R_i^{Φ} is N_i.

The algorithm for the Kernel CLAFIC Method can be summarized as follows:

Step 1: Find the eigenvalues and corresponding eigenvectors of each matrix $K^{(i)} \in \Re^{N_i x N_i}$, which is defined as

$$K^{(i)} = \Phi^{(i)^T} \Phi^{(i)} = (K^i_{mn} =< \phi(x^i_m), \phi(x^i_n) >= k(x^i_m, x^i_n))_{m=1,...,N_i; n=1,...,N_i}, \quad i = 1,...,C. \quad (5.49)$$

The matrix $K^{(i)}$ is typically a full rank matrix; thus, all the eigenvalues are positive.

Step 2: Choose the dimensionality l_i of each subspace by using one of the procedures given previously. Form the matrix $U^{(i)} = [u^i_1 \quad u^i_2 \quad ... \quad u^i_{l_i}]$ whose columns are the most significant eigenvectors corresponding to the largest eigenvalues of $K^{(i)}$. Let $\Lambda^{(i)} = diag(\lambda^i_1, \lambda^i_2, ..., \lambda^i_{l_i})$ be a diagonal matrix whose diagonal elements are the largest eigenvalues of $K^{(i)}$.

Step 3: The final basis vectors spanning the subspaces will be the normalized eigenvectors such that

$$W^{(i)} = \Phi^{(i)} U^{(i)} (\Lambda^{(i)})^{-1/2}, \quad i = 1,...,C. \quad (5.50)$$

In this case, the length of the projection of a new test sample x_{test} can be computed by

$$\| W^{(i)^T} \phi(x_{test}) \|^2 = \| (\Lambda^{(i)})^{-1/2} (U^{(i)})^T K^i_{test} \|^2, \; i = 1,...,C, \quad (5.51)$$

where $K^i_{test} \in \Re^{N_i x 1}$ is a vector with entries $< \phi(x^i_m), \phi(x_{test}) >_{m=1,...,N_i}$. Then we assign the test sample to the class which gives the maximum value.

5.9 The Kernel Common Vector Method

This method consists of mapping the given training set samples to an implicit higher-dimensional space $\Im$ using a nonlinear kernel mapping and applying the variation of the linear CV Method in the transformed space.

Our aim is to find basis vectors for the intersection subspaces $N(S_i^\Phi) \cap R(S_T^\Phi)$, for each class. Here, S_i^Φ represents the scatter matrix of the *i*-th class in $\Im$. To find these basis vectors, we follow the steps given in the previous section: We first project all training samples onto $R(S_T^\Phi)$ and then find the null spaces of the classes in the transformed space. The projection of training set samples onto $R(S_T^\Phi)$ can be done easily by employing the Kernel PCA Method. The algorithm for the Kernel Common Vector (Kernel CV) Method can be summarized as follows:

Step 1: Project the training set samples onto $R(S_T^\Phi)$ using the Kernel PCA. Let

$$\widetilde{K} = K - 1_M K - K 1_M + 1_M K 1_M = U \Lambda U^T \in \Re^{MxM}, \tag{5.52}$$

where the diagonal elements of Λ are nonzero and $K \in \Re^{MxM}$ as in (4.15). The matrix that transforms the training set samples onto $R(S_T^\Phi)$ is $(\Phi - \Phi 1_M) U \Lambda^{-1/2}$. The new scatter matrix $\widetilde{S}_i^\Phi \in \Re^{rxr}$ (r is the rank of $R(S_T^\Phi)$ and cannot be larger than M-1) of each class in the reduced space becomes

$$\begin{aligned} \widetilde{S}_i^\Phi &= ((\Phi - \Phi 1_M) U \Lambda^{-1/2})^T S_i^\Phi (\Phi - \Phi 1_M) U \Lambda^{-1/2} \\ &= \Lambda^{-1/2} U^T \widetilde{H}^{(i)} \widetilde{H}^{(i)T} U \Lambda^{-1/2}, \quad i = 1,...,C. \end{aligned} \tag{5.54}$$

Here, the matrix $\widetilde{H}^{(i)} \in \Re^{MxN_i}$ is given by

$$\begin{aligned} \widetilde{H}^{(i)} &= H^{(i)} - H^{(i)} G^{(i)} - 1_M H^{(i)} + 1_M H^{(i)} G^{(i)} \\ &= (H^{(i)} - 1_M H^{(i)})(I - G^{(i)}) \end{aligned}, \tag{5.55}$$

where $G^{(i)} \in \Re^{N_i xN_i}$ is a matrix whose elements are all $1/N_i$, $1_M \in \Re^{MxM}$ is a matrix all of whose entries are $1/M$, and the matrix $H^{(i)} \in \Re^{MxN_i}$ is given by $H^{(i)} = \Phi^T \Phi^{(i)} = (H^{(i)j})_{j=1,...,C} \in \Re^{MxN_i}$, where each matrix $H^{(i)j} \in \Re^{N_j xN_i}$ is defined as

$$H^{(i)j} = (k_{mn}^{(i)j})_{\substack{m=1,..,N_j \\ n=1,...,N_i}} =< \phi(x_m^j), \phi(x_n^i) >= k(x_m^j, x_n^i)_{\substack{m=1,...,N_j \\ n=1,...,N_i}}. \quad (5.56)$$

Step 2: For each class, find a basis of the null space of $\widetilde{S}_i^{\Phi}$. This can be done by eigen-decomposition. The normalized eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_i^{\Phi}$ form an orthonormal basis for the null space of $\widetilde{S}_i^{\Phi}$. Let $Q^{(i)}$ be a matrix whose columns are the computed eigenvectors corresponding to the zero eigenvalues, such that

$$Q^{(i)^T} \widetilde{S}_i^{\Phi} Q^{(i)} = 0, \quad i = 1,..., C. \quad (5.57)$$

Step 3: The basis vector matrix $W^{(i)}$ whose columns span the intersection subspace of the *i*-th class is

$$W^{(i)} = (\Phi - \Phi 1_M) U \Lambda^{-1/2} Q^{(i)}, \quad i = 1,..., C. \quad (5.58)$$

The number of basis vectors spanning the intersection subspaces is determined by the dimensionality of $N(\widetilde{S}_i^{\Phi})$ for each class. After performing feature extraction, all training set samples in each class give rise to the common vector of that class. Therefore, similarly to the linear CV case, a 100% recognition accuracy is also guaranteed with this method. Moreover, to recognize a given test sample, we compare the Euclidean distances between the common vectors and the feature vector of the test sample for each class using (5.43), and we assign the test sample to the class that minimizes the distance.

5.10 Experimental Results

Experimental studies performed in this chapter can be classified into two groups. In the first set of experiments, we compared the CV Method to the DCV Method in terms of recognition accuracy, training cost, storage requirements, and real-time performance. In the second set of experiments, we tested the recognition accuracies of the subspace classifiers. In all

experiments we used the ORL face database to test the proposed method. The ORL face database contains C = 40 individuals with 10 images per person. The images are taken at different time instances with slightly varying lighting conditions, facial expressions, and facial details. The size of each image is 92x112. Some individuals from the ORL face database are shown in Figure 3.2.

5.10.1 Comparison of the CV and DCV Methods

In these experiments, we first randomized the samples in the ORL face database and then selected $N = 3,5,7,9$ from each class for training; and the remaining $(10 - N)$ samples of each class were used for testing. We did not apply any pre-processing to the images. Then recognition rates were computed. Euclidean distance was used to compute the distances between the sample feature vectors of test set and the common vectors for the CV Method; similarly, the same metric was used to compute the distances between the feature vectors of test samples and the discriminative common vectors. This process was repeated seven times, and the recognition rates were found by averaging the recognition rates of seven trials. The recognition rates for the test sets are given in Table 5.1. The recognition rates of training set are not given since they are 100% for both methods.

Some of the common vectors obtained by the CV and the DCV methods are plotted in Figure 5.2. Figure 5.2 displays the absolute values of the common vectors obtained by the CV Method in image form. On the other hand, to display the common vectors obtained by the DCV Method, we took the logarithm of the values after taking the absolute values since common vectors displayed by taking only the absolute values were mostly dark.

Figure 5.2: Common vectors obtained by the CV and the DCV methods. The first row shows some individuals from the ORL face database and the second and the third rows show the corresponding common vectors obtained by the CV and the DCV methods, respectively.

TABLE 5.1
Recognition Rates of Methods on the ORL face database

Number of training samples in each class	**Methods**	
	CV	**DCV**
$N = 3$	88.82%, $\sigma = 3.73$	**91.02%,** $\sigma = 1.89$
$N = 5$	95.78%, $\sigma = 1.41$	**96.92%,** $\sigma = 1.30$
$N = 7$	97.97%, $\sigma = 1.16$	**98.21%,** $\sigma = 1.39$
$N = 9$	**99.28%,** $\sigma = 1.21$	**99.28%,** $\sigma = 1.21$

Recognition accuracy, training cost, storage requirements, and real-time performance are some factors that may be used to evaluate a method. We discuss here the differences among these factors between the CV and the DCV methods.

As can be seen in Table 5.1, the DCV Method tends to yield better results compared to the CV Method. The results reveal the important fact that there is a relationship between the number of training samples N in each class and the difference between the recognition rates

of the CV and the DCV methods. As the number of training set samples is increased, the difference between the recognition rates decreases and finally becomes zero in this example. These observations somewhat support the hypothesis that the variations among the face samples of each class are similar. Therefore, we can assume that the scatter matrices of each face class are identical, and that we can replace them with the within-class scatter matrix. A similar assumption is made in the Fisher's Linear Discriminant Analysis approach. For this reason we obtained better results for the DCV Method in the case of having only a few training vectors in each class. As explained previously, the CV Method first models the variations in each class and removes them from the samples in order to obtain the common vectors. If this variation is modeled correctly, all samples are classified correctly. The low recognition rates of the CV Method for small numbers of training set samples show that the number of training samples in each class is not sufficient to obtain a good model of the variations. On the other hand, the DCV Method does a better job with a small number of training set samples since it makes use of all of the pattern samples from all classes and does not perform a separate analysis on each class by itself. Some of the variations emerging from the test samples of one class may be captured by the variations between the training set samples of one or more other classes.

Training cost is the amount of computations required to find the projection vectors and the sample feature vectors of the training set samples. We compared the training cost of the methods based on their computational complexities (number of flops). The CV Method yields higher efficiency in terms of computation complexity since the DCV Method includes an additional step of applying PCA to the common vectors.

The DCV Method requires less storage space than the CV Method. If we assume that all training set sample vectors are linearly independent, then the CV Method requires us to store (M-C) d-dimensional projection vectors and C d-dimensional common vectors. However, we need only store (C-1) d-dimensional projection vectors and C (C-1)-dimensional discriminative common vectors for the DCV Method. Therefore, if we assume that each class has N samples, the storage space of the CV method is approximately N times the storage space of the DCV method.

The real-time performance of a method is determined by the time that is required to classify a new test image. To do this, we need to compute the feature vector of the test sample and compare it to the feature vectors of the training set. We compared testing times based on computational complexities here. The DCV Method is more efficient than the CV Method in terms of testing time. For the CV Method, we had to project our test sample onto (M-C) d-dimensional vectors to obtain feature vectors and compute the distances between the d-dimensional common vector and the feature vectors. On the other hand, we had to project our test sample onto only (C-1) d-dimensional vectors to obtain the feature vector of the test sample and compare it to the C (C-1)-dimensional vectors. Assuming d>>(C-1), the difference between the testing times of the methods is determined by the number of computations required to project a test sample onto (M-2C+1) d-dimensional vectors.

5.10.2 Testing Generalization Performance of Subspace Classifiers

In this set of experiments we tested the generalization performances of the linear and nonlinear subspace classifiers. We have experimented with the polynomial kernel $k(x, y) = (< x, y >)^2$ of degree 2 and the Gaussian kernel $k(x, y) = \exp(-\| x - y \|^2 / \gamma)$, for

all data sets. The parameter γ was chosen as 1.06e8, based on empirical observations. Beside the subspace classifier methods proposed in this chapter, we also tested the linear CLAFIC and the Kernel CLAFIC methods. Class correlation matrices were used for finding the basis vectors spanning the subspaces of classes for the CLAFIC and the Kernel CLAFIC methods. For both methods, the dimension of each subspace was determined by the rank of the corresponding correlation matrix since there are only a few training samples in each class. In particular, the dimension of each subspace was equal to the number of samples in each class. We randomly selected five samples from each class for training; the remaining samples were used for testing. We did not apply any pre-processing to the images. Then, recognition rates were computed, and this process was repeated five times. The recognition rates were found by averaging the recognition rates in each run. The computed recognition rates are shown in Table 5.2.

TABLE 5.2
Recognition Rates of Methods on the ORL Face Database

Linear Methods	**Recognition Rates(%) & Standard Deviations**	
CLAFIC	95.3, $\sigma = 1.68$	
Variation of CV	**96,** $\sigma = 1.58$	
Nonlinear Methods	Polynomial Kernel	Gaussian Kernel
Kernel CLAFIC	95.3, $\sigma = 1.85$	**95.9**, $\sigma = 1.67$
Kernel CV	**96,** $\sigma = 1.83$	95.8, $\sigma = 1.68$

As can be seen from the results, although there is not a significant difference between the recognition rates, the variation of CV Method outperforms the CLAFIC Method and similarly, the Kernel CV Method outperforms the Kernel CLAFIC Method. However, the kernel methods do not offer any recognition improvements over the linear methods. This can

be attributed to the linear distribution of image classes. Since the problem is close to linearly separable, using nonlinear methods does not improve the recognition rates. However, the nonlinear subspace classifiers may improve the recognition accuracies of the linear subspace classifiers on different databases having nonlinear and complex distributions.

5.11 Conclusion

In this chapter we proposed a variation of a subspace classifier. Then, this method was generalized to the nonlinear case by employing kernel functions. The proposed methods employ the intersection subspace of the null space of a class' covariance matrix and the range space of the covariance matrix of pooled data to represent each class. When the training set samples are projected onto these intersection subspaces, all training set samples in each class give rise to a unique vector, called the common vector. Thus, a 100% recognition rate is typically guaranteed for the training set samples. Then, we compared the proposed linear subspace classifier to the linear DCV Method. After comparing the CV and the DCV methods, we arrived at the following conclusions:

i) The DCV Method is more efficient than the CV Method in terms of recognition accuracy, storage requirements, and real-time performance for face recognition tasks. However, the training cost of the CV Method is lower than the DCV Method.

ii) The CV Method is expected to perform well if the variations among the test samples of a class are similar to the variations among the training samples of that class.

iii) The DCV Method performs well if the variations among the samples of classes are similar. This enables us to classify the test samples more accurately even if they are not similar to the ones used for training.

These results show that the subspace classifiers are not suitable for all classification tasks. In particular, if there are a few samples in each class, the estimation of basis vectors spanning subspaces may not be reliable. In such cases, it is better to use the DCV Method. Also, the dimensionality of the sample space must be large enough to ensure that the pattern classes are distributed in a lower-dimensional subspace of the original sample space.

We later compared the proposed subspace classifiers to other subspace classifiers. Our test results show that the generalization ability of the proposed method competes with the other subspace classifiers. Therefore, we conclude that the basis vectors, which span the intersection of the null space of a class' covariance matrix and the range space of the covariance matrix of pooled data, carry important discriminatory information for classification.

APPENDIX A

Statistical Significance Test Involving Differences of Means and Proportions

Consider the two classes, X_1 and X_2 come from two populations having means, $\overline{X}_1$, $\overline{X}_2$ and standard deviations σ_1, σ_2 obtained by N_1 and N_2 trials, respectively. Then, we have to decide between two hypotheses:

H_0 : $\mu_1 = \mu_2$, and the difference is merely due to chance.

H_1 : $\mu_1 \neq \mu_2$, and there is a significant difference between classes.

Under hypothesis H_0, both classes come from the same population. The mean and standard deviation of the difference in means are given by,

$$\mu_{\overline{X}_1 - \overline{X}_2} = 0 \text{ and } \sigma_{\overline{X}_1 - \overline{X}_2} = \sqrt{\sigma_1^2 / N_1 + \sigma_2^2 / N_2} .$$

Then,

$$z = (\overline{X}_1 - \overline{X}_2) / \sigma_{\overline{X}_1 - \overline{X}_2} .$$

For a two-tailed test, the results are significantly different at a 0.05 level if *z* lies outside the range -1.96 to 1.96. Hence, we conclude that the difference in performance of the two methods is significantly different if *z* lies outside the range -1.96 to 1.96 with a significance level of 0.05.

REFERENCES

[1] Adini, Y., Moses, Y. and Ullman, S. (1997) Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19, 721-732.

[2] Ariki, Y. and Motegi, Y. (1995) Segmentation and recognition of hand written characters using subspace method. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1, 120-123.

[3] Balachander, T. and Kothari, R. (1999) Kernel based subspace pattern classification. In *International Joint Conference on Neural Networks*, 5, 3119-3122.

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

[5] Baudat, G. and Anouar, F. (2001) Kernel-based methods and function approximation. In *International Joint Conference on Neural Networks*, 1244-1249.

[6] Baudat G. and Anouar, F. (2003) Feature vector selection and projection using kernels. *Neurocomputing*, 55, 21-38.

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

[8] Bellman, R. (1961) *Adaptive Control Processes: A Guided Tour*. New Jersey: Princeton University Press.

[9] Bing, Y., Lianfu, J. and Ping, C. (2002) A new LDA-based method for face recognition. In *Proceedings of 16th International Conference on Pattern Recognition*, 1, 168-171.

[10] Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*. Oxford University Press.

[11] Breukelen, M. V., Duin, R.P.W., Tax, D.M.J. and Hartog, J. E. D. (1998) Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4) ,381-386.

[12] Brown, W. C. (1991) *Matrices and Vector Spaces*. New York, Marcel Dekker, Inc.

[13] Cevikalp, H., Barkana, B. and Barkana, A. (2005) A comparison of the common vector and the discriminative common vector methods for face recognition. In *the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, USA, to appear.

[14] Cevikalp, H. and Neamtu, M. (2005) Nonlinear common vectors for pattern classification. In *the 13th European Signal Processing Conference*, Antalya, Turkey, to appear.

[15] Cevikalp, H., Neamtu, M. and Wilkes, M. (2005) Nonlinear discriminative common vector method. In *the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, USA, to appear.

[16] Cevikalp, H., Neamtu, M., Wilkes, M. and Barkana, A. (2004) A novel method for face recognition (Turkish). In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, 579-582.

[17] Cevikalp, H., Neamtu, M., and Wilkes, M. (2005) Discriminative common vector method with kernels. *IEEE Transaction on Neural Networks*, in review.

[18] Cevikalp, H., Neamtu, M. and Wilkes, M. (2005) Nonlinear discriminant common vectors (Turkish). In *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference*, Kayseri, Turkey, to appear.

[19] Cevikalp, H., Neamtu, M. Wilkes, M. and Barkana, A. (2005) Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 27, 4-13.

[20] Cevikalp, H. and Wilkes, M. (2004) Face recognition by using discriminative common vectors. In *Proceedings of the 17th International Conference on Pattern Recognition*, 1, 326-329.

[21] Chellappa, R., Wilson, C.L., and Sirohey, S. (1995) Human and machine recognition of faces: a survey. In *Proceedings of the IEEE*, 83, 705-740.

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

[23] Chen, X., Kwong, S. and Lu, Y. (2000) Human facial expression recognition based on learning subspace method. In *IEEE International Conference On Multimedia And Expo*, 1, 403-406.

[24] Cheng, Y. Q., Zhuang, Y. M. and Yang, J. Y. (1992) Optimal fisher discriminant analysis using the rank decomposition. *Pattern Recognition*, 25, 101-111.

[25] Cooke, T. (2002) Two variations on Fisher's linear discriminant for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 268-273.

[26] Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge: Cambridge University Press.

[27] Cristianini, N. and Shawe-Taylor, J. (2004) *Kernel Methods for Pattern Analysis.* Cambridge: Cambridge University Press.

[28] Dai, D. Q. and Yuen, P. C. (2003) Regularized discriminant analysis and its application to face recognition. *Pattern Recognition*, 36, 845-847.

[29] Demir, E., Akarun, L. and Alpaydin, E. (2000) Two-stage approach for pose invariant face recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6, 2342-2344.

[30] Devroye, L., Gyorfi, L. and Lugosi, G. (1996) *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag.

[31] Duchene, J. and Leclercq, S. (1988) An optimal transformation for discriminant and principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 978-983.

[32] Duda, R. O., Hart, P. E. and Stork, D. G. (2001) *Pattern Classification*. 2nd edition, John Wiley & Sons, Inc.

[33] Duin, R. P. W. (2000) Classifiers in almost empty spaces. In *Proceedings of the 15th International Conference on Pattern Recognition*, 2, 1-7.

[34] Duin, R. P. W., Loog, M. and Haeb-Umbach, R. (2000) Multi-class linear feature extraction by nonlinear PCA. In *Proceedings of the 15th International Conference on Pattern Recognition*, 2, 398-401.

[35] Er, M. J., Wu, S., Lu, J. and Toh, H. L. (2002) Face recognition with radial basis function (RBF) neural networks. *IEEE Transactions on Neural Networks*, 13, 697-710.

[36] Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. In *Annals of Eugenics*, 7, 179-188.

[37] Foley, D. H. and Sammon, J. W. (1975) An optimal set of discriminant vectors. *IEEE Transactions on Computers*, C-24, 281-289.

[38] Frey, B. J., Colmenarez, A. and Huang, T. S. (1998) Mixtures of local linear subspaces for face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA., 32-37.

[39] Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition.* 2nd edition, New York: Academic Press.

[40] Fukunaga, K. and Koontz, W. L. (1970) Application of the Karhunen-Loeve expansion to feature selection and ordering. *IEEE Transactions on Computers*, C-19(4), 311-318.

[41] Golub, G. H. and Van Loan, C. F. (1996) *Matrix Computations.* 3rd edition, Baltimore, Maryland, The Johns Hopkins University Press.

[42] Gu, Y., Zhang, Y. and Zhang, J. (2002) A kernel based nonlinear subspace projection method for reduction of hyperspectral image dimensionality. In *International Conference on Image Processing*, 2, 357-360.

[43] Gulmezoglu, M. B., Cevikalp, H. and Barkana, A. (2004) A new point of view to common vector approach. In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, 732-734.

[44] Gulmezoglu, M. B., Dzhafarov, V., Keskin, M. and Barkana, A. (1999) A novel approach to isolated word recognition. *IEEE Transactions on Speech and Audio Processing*, 7, 620-628.

[45] Gulmezoglu, M. B., Dzhafarov, V. and Barkana, A. (2001) The common vector approach and its relation to principal component analysis. *IEEE Transactions on Speech and Audio Processing*, 9, 655-692.

[46] Guo, Y.-F., Li, S.-J., Yang, J.-Y., Shu, T.-T. and Wu, L.-D. (2003) A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition. *Pattern Recognition Letters*, 24, 147-158.

[47] Gupta, H., Agrawal, A. K., Pruthi, T., Shekhar, C. and Chellappa, R. (2002) An experimental evaluation of linear and kernel-based methods for face recognition. In *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision*, 13-18.

[48] Halmos, P. R. (1948) *Finite Dimensional Vector Spaces.* Princeton University Press.

[49] Hamamoto, Y., Kanaoka, T. and Tomita, S. (1993) On a theoretical comparison between the orthonormal discriminant vector method and discriminant analysis. *Pattern Recognition*, 26, 1863-1867.

[50] Hamamoto, Y., Matsuura, Y., Kanaoka, T. and Tomita, S. (1991) A note on the orthonormal discriminant vector method for feature extraction. *Pattern Recognition*, 24, 681-684.

[51] Haykin, Simon. (1999) *Neural Networks A Comprehensive Foundation.* 2nd edition, New Jersey, Prentice-Hall, Inc.

[52] Heisele, B., Verri, A. and Poggio, T. (2002) Learning and vision machines. In *Proceedings of IEEE*, 90, 1164-1177.

[53] Hong, Z.-Q. and Yang, J. Y. (1991) Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24, 317-324.

[54] Huang, J., Yuen, P. C., Chen, W.-S. and Lai, J. H. (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 327-332.

[55] Huang, R., Liu, Q., Lu, H. and Ma, S. (2002) Solving the small size problem of LDA. In *Proceedings of 16th International Conference on Pattern Recognition*, 3, 29-32.

[56] Iijima, T., Genchi, H. and Mori, K. (1973) A theory of character recognition by pattern matching method. In *Proceedings of the 1st International Joint Conference on Pattern Recognition.* 50-56.

[57] Jain, A. K., Duin, R. P. W. and Mao, J. (2000) Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22, 4-37.

[58] Jauch, T. W. (1998) Optimal subspace metric design for classification and regression through nonparametric crossvalidation objective function optimization. In *IEEE International Conference on Systems, Man, and Cybernetics*, 4, 3614-3622.

[59] Jimenez, L. O. and Landgrebe, D. A. (1998) Supervised classification in high dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews,* 28(1), 39-54.

[60] Jin, Z., Yang, J.-Y., Hu, Z.-S. and Lou, Z. (2001) Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34, 1405-1416.

[61] Jin, Z., Yang, J.-Y., Tang, Z.-M. and Hu, Z.-H. (2001) A theorem on the uncorrelated optimal discriminant vectors. *Pattern Recognition*, 34, 2041-2047.

[62] Kato, N. and Nemoto, Y. (1996) Large scale hand-written character recognition system using subspace method. In *IEEE International Conference on Systems, Man, and Cybernetics*, 1, 432-437.

[63] Kim, H.-C., Kim, D. and Bang, S. Y. (2002) Face recognition using LDA mixture model. In *Proceedings of 16th International Conference on Pattern Recognition,* 2, 486-489.

[64] Kim, K. I., Jung, K. and Kim, H. J. (2002) Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9, 40-42.

[65] Kim, S.-W. and Oommen, B. J. (2003) On using prototype reduction schemes to optimize kernel-based nonlinear subspace methods. *Pattern Recognition*, 37, 227-239.

[66] Kim, S.-W. and Oommen, B. J. (2005) On utilizing search methods to select subspace dimensions for kernel-based nonlinear subspace classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27, 136-141.

[67] Kittler, J. (1978) The subspace approach to pattern recognition. In *Progress in Cybernetics and systems Research*. Hamisphere Pub. Co.

[68] Kittler, J., Li, Y. P. and Matas, J. (2000) On matching scores for LDA-based face verification. In *British Machine Vision Conference*, 42-51.

[69] Kohonen, T., Nemeth, G., Bry, K. J., Jalanko, M. and Riittinen, H. (1979) Spectral classification of phonemes by learning subspaces. In *Proceedings of the 5th International Conference on Acoustics, Speech and Signal Processing*, 97-100.

[70] Kuusela, M. and Oja, E. (1982) The averaged learning subspace method for spectral pattern recognition. In *Proceedings of the 6th International Conference on Pattern Recognition,* 134-137.

[71] Laaksonen, J. (1997) *Subspace Classifiers in Recognition of Handwritten Digits*. Ph. D. thesis, Helsinki University of Technology, Finland.

[72] Landgrebe, D. (2002) Hyperspectral image data analysis. In *Special Issue of. the IEEE Signal Processing Magazine*, 19(1), 17-28.

[73] Lawrance, S., Yianilos, P., and Cox, I. (1997) Face recognition using mixture-distance and raw images. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 3, 2016-2021.

[74] Liu, K., Cheng, Y.-Q. and Yang, J.-Y. (1992) A generalized optimal set of discriminant vectors. *Pattern Recognition*, 25, 731-739.

[75] Liu, K., Cheng, Y.-Q. and Yang, J.-Y. (1993) Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition*, 26, 903-911.

[76] Liu, Q., Huang, R., Lu, H. and Ma, S. (2002) Kernel-based optimized feature vectors selection and discriminant analysis for face recognition. In *Proceedings of the 16th International Conference on Pattern Recognition*, 2, 362-365.

[77] Liu, Q., Lu, H. and Ma, S. (2004) Improving kernel Fisher discriminant analysis for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14, 42-49.

[78] Lotlikar, R. and Kothari, R. (2000) Adaptive linear dimensionality reduction for classification. *Pattern Recognition*, 33, 185-194.

[79] Lu, J., Plataniotis, K. N. and Venetsanopoulos, A. N. (2003) Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks,* 14, 117-126.

[80] Lu, J., Plataniotis, K. N. and Venetsanopoulos, A. N. (2003) Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, 14, 195-200.

[81] Lu, J., Plataniotis, K. N. and Venetsanopoulos, A. N. (2003) Regularized discriminant analysis for the small sample size problem in face recognition. *Pattern Recongition Letters*, 24, 3079-3087.

[82] Maeda, E. and Murase, H. (1999) Multi-category classification by kernel based nonlinear subspace method. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2, 1025-1028.

[83] Martinez, A. M. and Benavente, R. (1998) The AR face database. *CVC Tech. Report #24*.

[84] Martinez, A. M. and Kak, A. C. (2001) PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 23, 228-233.

[85] Mika, S. (2002) *Kernel Fisher Discriminants.* Ph. D. thesis, Informatik der Technischen Universität, Berlin.

[86] Mika, S., Rätsch, G., Weston, J., Schölkopf, B. and Müller, K. R. (1999) Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX,* Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. Piscataway, NJ: IEEE, pp. 41-48.

[87] Moghaddam, B., Jebara, T. and Pentland, A. (2000) Bayesian face recognition. *Pattern Recognition*, 33, 1771-1782.

[88] Moghaddam, B. and Pentland, A. (1998) Probabilistic matching for face recognition. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, 30-35.

[89] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K. and Schölkopf, B. (2001) An introduction to kernel-based learning algorithms. *IEEE Transaction on Neural Networks,* 12, 181-201.

[90] Oja, E. (1983) *Subspace Methods of Pattern Recognition.* Letchworth, UK: Research Studies Press.

[91] Oja, E. and Kohonen, T. (1988) The subspace learning algorithm as a formalism for pattern recognition and neural networks. In *IEEE International Conference on Neural Networks*, 1, 277-284.

[92] Okada, T. and Tomita, S. (1984) An optimal orthonormal system for discriminant analysis. *Journal of Pattern Recognition*, 18, 139-144.

[93] Pentland, A., Moghaddam, B. and Starner, T. (1994) View-based and modular eigenspaces for face recognition. In *Proceedings of IEEE Conference on Computer Vision & Pattern Recognition*, Seattle,WA., 84-91.

[94] Perez-Cruz, F. and Bousquet, O. (2004) Kernel methods and their potential use in signal processing. *IEEE Signal Processing Magazine,* 21, 57-65.

[95] Perlibakas, V. (2004) Distance measures for PCA-based face recognition. *Pattern Recognition Letters*, 25, 711-724.

[96] Phillips, P. J., Moon, H., Rizvi, S. A. and Rauss, P. J. (2000) The Feret evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 1090-1104.

[97] Ripley, B. D. (1996) *Pattern Recognition and Neural Networks.* Cambridge University Press.

[98] Roth, D., Yang, M.-H. and Ahuja, N. (2000) Learning to recognize objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1, 724-731.

[99] Schiele, B. and Pentland, A. (1999) Probabilistic object recognition and localization. In *Proceedings of International Conference on Computer Vision*, 1, 177-182.

[100] Schölkopf, B. (1997) *Support Vector Learning.* Ph. D. thesis, Informatik der Technischen Universität, Berlin.

[101] Schölkopf, B. and Smola, A. J. (2002) *Learning with Kernels.* MIT Press.

[102] Schölkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997) Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45, 2758-2765.

[103] Sierra, A. (2002) High-order Fisher's discriminant analysis. *Pattern Recognition*, 35, 1291-1302.

[104] Swets, D. L. and Weng, J. (1996) Using discriminant eigenfeatures for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18, 831-836.

[105] Tan, Y. and Wang, J. (2004) A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension. 16, 385-395.

[106] Theodoridis, S. and Koutroumbas, K. (1999) *Pattern Recognition*. Academic Press.

[107] Therrien, C. W. (1975) Eigenvalue properties of projection operators and their applications to the subspace method of feature extraction. *IEEE Transactions on Computers*, 24(9), 944-948.

[108] Tian, Q., Barbero, M., Gu, Z. H. and Lee, S. H. (1986) Image classification by the Foley-Sammon transform. *Opt. Eng*., 25, 834-840.

[109] Tou, J. T. and Gonzalez, R. C. (1974) *Pattern Recognition Principles.* Addison-Wesley Publishing Company, Inc.

[110] Tsuda, K. (1999) Subspace classifier in reproducing kernel Hilbert space. In *International Joint Conference on Neural Networks*, 5, 3054-3057.

[111] Tsuda, K. (1999) Subspace classifier in the Hilbert space. *Pattern Recognition Letters*, 20, 513-519.

[112] Turk, M. (2001) A random walk through eigenspace. *IEICE Trans. Inf. & Syst.,* E84-D, 1586-1695.

[113] Turk, M. and Pentland, A. P. (1991) Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3, 71-86.

[114] Turk, M. and Pentland, A. P. (1991) Face recognition using eigenfaces. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 586-591.

[115] Vapnik, V. N. (1995) *The Nature of Statistical Learning Theory*. 2nd edition, New York, Springer-Verlag.

[116] Watanabe, H. and Katagiri, S. (1995) Discriminative subspace method for minimum error pattern recognition. In *Proceedings of 1995 IEEE Workshop on Neural Networks for Signal Processing*, 77-86.

[117] Watanabe, S., Lambert, P. F., Kulikowski, C. A., Buxton, J. L. and Walker, R. (1967) Evaluation and selection of variables in pattern recognition. In *Computer and Information Sciences II*. New York: Academic Press.

[118] Watanabe, S. and Pakvasa, N. (1973) Subspace method in pattern recognition. In *Proceedings of the 1st International Conference on Pattern Recognition*, Washington, D.C.

[119] Watanabe, H., Yamaguchi, T. and Katagiri, S. (1995) A novel approach to pattern recognition based on discriminative metric design. In *Proceedings of 1995 IEEE Workshop on Neural Networks for Signal Processing*, 48-57.

[120] Webb, A. (1999) *Statistical Pattern Recognition.* New York: Oxford University Press.

[121] Wold, S. (1976) Pattern recognition by means of disjoint principal components models. 8, 127-39.

[122] Xu, J. and Zikatanov, L. (2002) The method of alternating projections and the method of subspace corrections in Hilbert space. *Journal of the American Mathematical Society*, 15, 573-597.

[123] Xu, Y., Yang, J.-Y. and Jin, Z. (2003) Theory analysis on FSLDA and ULDA. *Pattern Recognition*. 36, 3031-3033.

[124] Xu, Y., Yang, J.-Y. and Jin, Z. (2004) A novel method for Fisher discriminant analysis. *Pattern Recognition*, 37, 381-384.

[125] Xu, Y., Yang, J.-Y. and Yang, J. (2003) A reformative kernel Fisher discriminant analysis. *Pattern Recognition*, 37, 1299-1302.

[126] Yang, J., Frangi, A. F., Jin, Z. and Yang, J.-Y. (2004) Essence of kernel Fisher discriminant: KPCA plus LDA. *Pattern Recognition,* 37, 2097-2100.

[127] Yang, J. and Yang, J.-Y. (2003) Why can LDA be performed in PCA transformed space. *Pattern Recognition*, 36, 563-566.

[128] Yang, J, Yang, J.-Y. and Zhang, D. (2002) What's wrong with Fisher criterion. *Pattern Recognition*, 35, 2665-2668.

[129] Yang, J., Zhang, D. and Yang, J.-Y. (2003) A generalised K-L expansion method which can deal with small sample size and high-dimensional problems. *Pattern Analysis & Applications*, 6, 47-54.

[130] Yang, M.-H. (2002) Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, 215-220.

[131] Yang, M.-H., Ahuja, N. and Kriegman, D. (2000) Face recognition using kernel eigenfaces. In *Proceedings of the IEEE International Conference on Image Processing*, 1, 37-40.

[132] Yankun, Z. and Chongqin, L. (2002) Face recognition using kernel principal component analysis and genetic algorithms. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing,* 337-343.

[133] Yu, H. and Yang, J. (2001) A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34, 2067-2070.

[134] Zhao, W. (1999) Subspace methods in object/face recognition. In *International Joint Conference on Neural Networks*, 5, 3260-3264.

[135] Zhao, W., Chellappa, R. and Krishnaswamy, A. (1998) Discriminant analysis of principal components for face recognition. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, 336-341.

[136] Zhao, W. and Nandhakumar, N. (1998) Linear discriminant analysis of MPF for face recognition. In *Proceedings of 14th International Conference on Pattern Recognition*, 1, 185-188.

[137] Zhao, W., Chellappa, R., Rosenfeld, A. and Phillips, P. J. (2000) Face recognition: a literature survey. *Technical Report CAR-TR-948*, University of Maryland, College Park.

[138] Zheng, W., Zhao, L. and Zou, C. (2003) An efficient algorithm to solve the small sample size problem for LDA. *Pattern Recognition*, 37, 1077-1079.

[139] Zheng, W., Zhao, L. and Zou, C. (2005) Foley-Sammon optimal discriminant vectors using kernel approach. *IEEE Transactions on Neural Networks*, 16, 1-9.