Discriminant Feature Analysis for Pattern Recognition

Huang Dong

Department of Electrical & Computer Engineering

National University of Singapore

A thesis submitted for the degree of

Doctor of Philosophy (PhD)

May 7, 2010

Abstract

Discriminant feature analysis is crucial in the design of a satisfactory pattern recognition system. Usually it is problem dependent and requires specialized knowledge of the specific problem itself. However, some of the principles of statistical analysis may still be used in the design of a feature extractor, and how to develop a general procedure for effective feature extraction always remains an interesting and also challenging problem.

In this thesis we have investigated the limitations of traditional feature extraction algorithms like Fisher's linear discriminant (FLD) and devised new methods that overcome the shortcomings of FLD. The new algorithm termed recursive cluster-based Bayesian linear discriminant (RCBLD) has a number of advantages: it has a Bayesian criterion function in the sense that the Bayes error is confined by a coherent pair of error bounds and the maximization of the criterion function is equivalent to minimization of one of the error bounds; it can deal with complex class distributions as unions of Gaussian distributions; it also has no feature number limitation and can fully extract all discriminant information available; the solution of the algorithm can be easily obtained without resorting to some gradient-based methods.

Since the proposed algorithms are designed as general-purpose feature extraction tools, they have been applied to a wide variety of pattern classification problems such as face recognition and brain-computer-interface (BCI) applications. The experimental results have verified the effectiveness of the proposed algorithms.

I would like to dedicate this thesis to my loving parents, for all the unconditional love, guidance, and support.

Acknowledgements

I would like to formally thank:

Dr. Xiang Cheng, my supervisor, for his hard work and guidance throughout my Ph.D candidature and for believing in my abilities. I have learned so much, and without him, this would not have been possible. Thank him so much for a great experience.

Dr. Sam Ge Shuzhi, my co-supervisor, for his insight and guidance throughout the past four years.

My fellow graduate students, for their friendships and support. The last four years have been quite an experience and it is a memorable time of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Pattern recognition is the study of how machines can learn to observe the environment, distinguish patterns of interest from others, and make sound and reasonable decisions about the category of the pattern.

Automatic (machine) recognition of patterns is an important subject in a variety of engineering and scientific disciplines such as biology, psychology, marketing, computer vision, and artificial intelligence. From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification, and much more, it is clear that reliable and accurate pattern recognition by machine would be immensely useful. Moreover, by designing systems for accomplishing such tasks, we gain deeper understanding and appreciation for pattern recognition systems in the natural world—most particularly in humans. For some problems, such as speech and visual recognition, our design efforts may in fact be influenced by knowledge of how these are solved in nature, both in the algorithm we employ and in the design of special-purpose hardware.

As the task of a pattern recognition system is to observe the environment and distinguish patterns of interest, a complete pattern recognition system typically includes four main stages: sensing, pre-processing, feature extraction and classification. This conceptual decomposition of a pattern recognition system is illustrated in Figure 1.1. The sensor captures the input, which are a set of measurements or observations of the environment, which are referred to as the input patterns. Pre-processing is sometimes performed on the input pattern, e.g., low-pass-filtering of a signal, image segmentation, etc. The input pattern is then usually represented as a d-dimensional feature vector. Feature extraction does discriminant analysis and extracts discriminant information from the input features and classifier does the actual job of labeling the input patterns with one of the possible classes, relying on the set of extracted features. Usually, the type of sensors are determined by the application and the initial pre-processing and feature vector representation is defined by the designer taking into account the characteristics of the sensor. In such cases, the pattern recognition process starts with feature extraction task and may be considered as a direct application of machine learning or statistics methods. The design of the classifier is closely tied to the feature extraction stage. A good classifier should be designed such that it can effectively exploit the embedded information in the extracted features and make sensible decisions. The arrows linking the various components of the pattern recognition system in Figure 1.1 indicate that these components are not independent in the design of the whole system. Depending on the results, one may go back to re-design other components in order to improve the overall performance. Also note that the conceptual boundary between pre-processing and feature extraction, and between feature extraction and classification is somewhat arbitrary. For instance, an ideal feature extractor would yield a representation that makes the job of the classifier trivial; conversely, an omnipotent classifier would not need the help of a sophisticated feature extractor. This thesis focuses on the feature extraction component of the system, or in other words, discriminant feature analysis for pattern recognition.

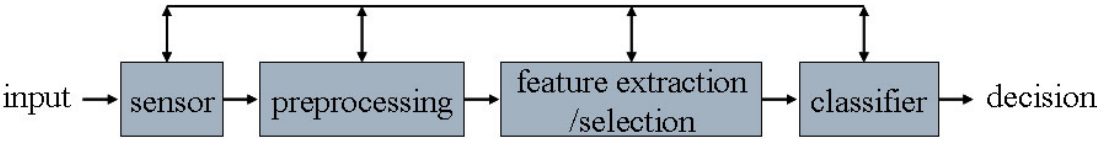


Figure 1.1: The basic components of a typical pattern recognition system -

1.2 Discriminant Feature Analysis for Pattern Recognition

Discriminant feature analysis plays a crucial role in the design of a satisfactory pattern recognition system. Although the original d-dimensional input feature vector captured by the sensor could be directly fed into a classifier, it is usually not the case. Instead, discriminant feature analysis is performed on the raw features due to several compelling reasons. First of all, discriminant feature analysis could improve the performance of the system by extracting useful information and discarding irrelevant information such as noise from the set of input features. Second, the efficiency of the system can be greatly improved. Discriminant feature analysis reduces the feature dimension and allows subsequent processing of features to be done efficiently. For instance, Gaussian maximum-likelihood classification time increases quadratically with the dimension of feature vectors. Increasing the dimension of feature vectors leads to a disproportionate increase in cost. Therefore, the reduction of dimension by discriminant feature analysis could save the computational and memory cost significantly. For applications involving high-dimensional features, such as hyper-spectral imaging, and bioinformatics etc, analysis of high-dimensional data is often computationally and memory too expensive to be practically feasible. Discriminant feature analysis is an indispensable step for such applications. Third, discriminant feature analysis reduces the complexity of the classification model and thus it can potentially improve the classification accuracy in the lower-dimensional space. Due to the small sample size and curse of dimensionality problem as discussed below, an over-complex model may be selected as a result of over-training. The complexity of the classification model could strongly affect its stability and performance on new test data. By reducing the number of features and removing noises from the features, the performance of the classification model can be more robust with a reduced complexity. Because the decision of the classifier is based on the set of features provided by the feature extractor, discriminant feature analysis is crucial for the performance of the whole pattern recognition system.

1.2.1 The Issues in Discriminant Feature Analysis

In practice, the issues we encounter in designing the feature extraction component is usually domain or problem-specific, and their solution will depend upon the knowledge and insights about the particular problem. Nevertheless, there are some problems that may be commonly-encountered, difficult, and important. Some of the important issues regarding discriminant feature analysis are presented below.

1.2.1.1 Noise

For pattern recognition, the term "noise" may refer generally to any form of component in the sensed pattern that is not generated from the true underlying model of the pattern. All pattern recognition problems involve noise in some form. An important problem is knowing somehow whether the variation in some signal is noise or instead because of the complex underlying model. How then can we use this information to improve the classification performance?

1.2.1.2 The Problem of Sample Size

The small sample size (SSS) problem is encountered when there are only limited number of training samples compared to the high dimension of the input patterns. The small sample size problem is almost always encountered due to the fact of limited samples for real-world applications. Due to insufficiency of samples, the estimated models may be far from the true underlying models. Also the evaluation of the system's performance based on a small set of samples is not reliable. One technique for the SSS problem is to incorporate knowledge of the problem domain.

1.2.1.3 The Problem of Dimension

The problem of dimension involves learning from few data samples in a high-dimensional feature space. Therefore, this problem is coupled with the SSS problem. Intuitively one may think that, the more features we have, the better we can make the system's performance, since more information is present. However, it has been observed in practice that addition of features beyond a certain point may actually lead to a higher probability of error, as indicated in [14]. This behavior is known in pattern recognition as the curse of dimensionality [14, 32, 61, 62], and it is caused by the finite number of samples. The curse of dimensionality requires the number of training samples to be an exponential function of the feature dimension.

Therefore, a feature extraction/selection stage is needed to reduce the number of features. The extraction/selection of relevant features for classification is crucial for a successful pattern recognition system.

1.2.1.4 Model Selection

In the designing of a pattern recognition system, we often need to use some models to describe the objects of interest, for example, a particular form of distribution of a class, or a particular form of representation of a pattern. If the models we selected to use differs significantly from the true model, we can't expect good performance from the resulting system.

Traditionally, the performance of a pattern recognition system is affected from the data modeling perspective by the interplay between size of training set, dimension of feature vector, and complexity of model. In building a pattern recognition system, one may be tempted to increase the complexity of the model to obtain good performance on the set of training data. For example, the decision boundary of a classifier can be made arbitrary complex so that all the training samples are correctly classified. Obviously, this model is too complex compared to the true underlying model.

Conventional wisdom holds that simpler models built from larger sets of training data, while usually less accurate on the training data, are better able to maintain their training data level of performance when subjected to new test data. It is a well-understood phenomenon that a prediction model built from large number of features and a relatively small sample size can be quite unstable [53]. This paradoxical relationship between the model complexity and performance is well known, appearing in things ranging from simple regression analysis (the linear function, while hitting none of the given training points, far better predicts the new points than some high-degree polynomial specifically designed to pass through the training points) to modern neural network analysis (where performance drop-off on test data due to complexity, overtrained models is a major problem).

The complexity of model thus should be selected by considering factors including the sample size, the feature dimension, and also the nature of the problem. One of the most important areas of research in statistical pattern classification is determining how to adjust the complexity of the model — not so simple that it cannot explain the differences between the categories, yet not so complex as to give poor classification on novel patterns. Simple models are often favored, especially for cases where sample size is small. Complex models are only advisable for situations where there are sufficient training data.

1.2.1.5 Generalization and Overfitting

In building a pattern recognition system, the system is trained to classify accurately a set of known samples, or training samples. However, the final goal of a pattern recognition system is to be able to classify a *novel* pattern correctly. The ability of the system to be able to correctly classify novel patterns by training on a set of known patterns is called the *generalization* ability of the system.

Apparently, one wants to design a pattern recognition system that can perform well on the training data as well as the test data. Without a good performance on the training data, there is no chance of descent performance in the real world. The system should also be able to transfer, or *generalize* its performance on training data to novel data in the real world.

As a result, the performance of a pattern recognition system can be measured by two different accuracies: training accuracy and test accuracy. Training accuracy is obtained on the training samples, which are known to the system and are used to tune the parameters of the system. Test accuracy is a measure of the system's ability to correctly classify new test samples which are not known to the system. The goal of the designer is to make the two accuracies as high as possible.

However, these two accuracies are usually conflicting with each other. For instance, if the decision boundary of a classifier is overly complex, it seems to be "tuned" to the particular training samples, rather than the true underlying characteristics. This situation is known as *overfitting*. As discussed above, it is usually the case that very simple models perform poorly on training data but have good generalization ability, while complex models perform well on training data but are more likely to suffer from poor generalization to test data.

1.2.1.6 Computational Complexity

Computational complexity is one of the major concerns in real-time applications. In some cases we know we can design an excellent recognizer, but the recognizer may not be practically feasible due to high computational complexity. One may also be concerned how the computational complexity of an algorithm scales as a function of the feature dimension, the size of training data, or the number of classes. In practice, we often need to face tradeoff between computational cost and performance. We are typically less concerned with the complexity of learning, which is done in the laboratory, than with the complexity of classification, which is done with the fielded application.

1.3 Scope and Organization

My research work has been primarily focused on discriminant feature analysis in the feature extraction component for a pattern recognition system. The thesis contains two parts: algorithm development and applications.

The first part describes the algorithmic development for discriminant feature extraction. First, background review of some popular discriminant feature analysis techniques is given in Chapter 2. The proposed algorithms, termed recursive modified linear discriminant (RMLD), recursive cluster-based linear discriminant (RCLD), and recursive Bayesian linear discriminant (RBLD), are presented in Chapter 3, 4, and 5, respectively. The advantages of these three methods are then integrated and the new algorithm is named recursive cluster-based Bayesian linear discriminant (RCBLD), which is described in Chapter 6. The new algorithms are proposed to overcome some of the drawbacks of existing algorithms described in Chapter 2 and address some of the common issues in designing a pattern recognition system as identified above.

The second part tests the effectiveness of the proposed algorithms on various pattern recognition tasks: a range of patten recognition problems from the UCI Machine Learning Repository in Chapter 7, face recognition problems in Chapter 8, and brain signal analysis problems in Chapter 9.

At last, some conclusions are drawn in Chapter 10.

Part I

Algorithm Development

Chapter 2

Background Review

Discriminant feature analysis plays an important role in pattern recognition. As discussed in Chapter 1, it can reduce the complexity of the classification model and potentially improve the classification performance by obtaining discriminant features and discarding useless components like noise from an input feature vector. It also saves computational load and memory requirement for subsequent processing. The problem of "curse of dimensionality" is alleviated and the underlying models or parameters can be simplified and estimated more accurately which may lead to better classification performance. Reduction of dimension is sometimes a necessary step for problems with high dimensional samples and for hardware implementation of a pattern recognition system.

Although there is some extra computational effort spent for discriminant feature analysis, this extra computational effort mainly reside in the training stage, which can be done off-line. Once the training is done, the classification can be performed with very little additional computation.

Many algorithms have been proposed for feature extraction. In the following, some popular feature extraction algorithms are briefly introduced.

2.1 Principal Component Analysis (PCA)

One of the earliest methods used for feature extraction is principal component analysis (PCA). PCA was invented in 1901 by Karl Pearson [57] and has become a popular technique in pattern recognition to reduce feature dimension. Depending on the field of application, it is also named the discrete Karhunen-Loève transform (KLT), the Hotelling transform.

PCA is a feature extraction method that is best for representation in the sense of minimal squared reconstruction error. It is an unsupervised linear feature extraction method that is largely confined to dimension reduction.

Suppose that we have a set of N d-dimensional samples $x_1, \ldots, x_N$ belonging to C different classes with N_i samples in the subset D_i labeled ω_i, $i = 1, \cdots, C$.

PCA seeks a projection matrix W that minimizes the squared error function:

$$J_{PCA}(W) = \sum_{k=1}^{N} ||x_k - y_k||^2 \tag{2.1}$$

where $y_k = W(W^T x_k)$ is obtained after projection of x_k by W. The solution is the eigenvector of the total scatter matrix defined as:

$$S_T = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T \tag{2.2}$$

where μ is the mean of all the samples:

$$\mu = \frac{1}{N} \sum_{k=1}^{N} x_k. \tag{2.3}$$

The main properties of PCA are: approximate reconstruction, orthonormality of the basis, and decorrelated principal components. That is to say,

$$x \approx Wy \tag{2.4}$$

$$W^T W = I \tag{2.5}$$

$$YY^T = D \tag{2.6}$$

where Y is a matrix whose kth column is y_k, and D is a diagonal matrix.

Usually, the columns of W associated with significant eigenvalues, called the principal components (PCs), are regarded as important, while those components with the smallest variances are regarded as unimportant or associated with noise.

2.2 Fisher's Linear Discriminant (FLD)

Although PCA is efficient for data representation, it may not be good for class discrimination. Fisher's linear discriminant (FLD) has recently emerged as a more efficient approach for many pattern classification problems than traditional PCA. Although FLD is not as popular as PCA for extracting discriminating features until late 90s, FLD is by no means a new technique. On the contrary, it is a "classical" technique whose history can be traced back to as early as 1936 when Fisher first suggested it to deal with the taxonomic problems [20]. The original FLD was proposed to deal with two-class problems and was naturally generalized to deal with multi-class problems that is well described in various standard textbooks on pattern classification such as [14, 23, 52]. Many interesting applications of FLD have also appeared in the literature. Cheng and co-workers suggested a method of applying FLD for face recognition where features were acquired from polar quantization of the shape [10], while Cui and colleagues applied it to hand sign recognition [12]. A theory on pattern rejection was developed by Baker and Nayar based upon the two-class linear discriminant [2]. And around the same year of 1997, comparison studies between FLD and PCA on face recognition problem were reported independently by numerous authors including Belhumeur, Hespanha and Kriegman [3], Etemad and Chellappa [16], and Swets and Weng [73]. It was consistently demonstrated that FLD outperforms PCA significantly for face recognition problems. These successful applications of FLD have drawn lots of attention on this subject and ensuing years witnessed a burst of research activities on this issue [8, 47, 47, 51, 77, 85].

To find a feature vector w that separates classes, FLD maximizes the following criterion function,

$$J_{FLD}(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.7}$$

where the between-class scatter matrix S_B, and the within-class scatter matrix S_W are defined as follows:

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T = \frac{1}{N}\sum_{i<j} N_i N_j(\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{2.8}$$

$$S_W = \sum_{i=1}^{C} S_i, \quad \mathrm{where\ S_i} = \sum_{\mathrm{x \in D_i}} (\mathrm{x} - \mu_{\mathrm{i}})(\mathrm{x} - \mu_{\mathrm{i}})^{\mathrm{T}} \tag{2.9}$$

where μ_i is the sample mean of class i.

It is easy to show that a vector w that maximizes (2.7) must satisfy

$$S_B w = \lambda S_W w \tag{2.10}$$

If S_W is non-singular we can obtain a conventional eigenvalue problem by writing

$$S_W^{-1} S_B w = \lambda w \tag{2.11}$$

Unfortunately, in real applications, S_W is very often singular because the number of training samples is much smaller than the dimension of the samples. This problem is called the small sample size problem and is very common for pattern recognition problems. To address this issue, a typical approach [3] is to employ PCA to reduce the feature dimension so that S_W is non-singular.

It is obvious that the at most $C - 1$ features may be extracted from above procedure simply because the rank of S_B is at most $C - 1$.

2.3 Other Variants of FLD

Although most of the research results have consistently established the superiority of FLD over PCA for extracting features for pattern classification problems, there are some drawbacks and limitations of FLD and various variants of FLD have been proposed to improve its performance. The following sub-sections describe some of these variants.

2.3.1 Recursive FLD (RFLD)

One serious limitation of FLD is that the total number of features available from FLD is limited to $C - 1$, where C is the number of classes. This cap on the total number of features is rooted in the mathematical treatment of FLD. The number of non-zero eigenvectors of (2.11) is equal to the rank of S_B, which is at most $C - 1$. If the number of classes is large as is the case for face identity recognition problems, this limitation may not arise as a visible obstacle. However, it may pose as a bottleneck if the number of classes is small. For instance, for the glasses-wearing recognition problem treated in [3], the number of classes is two, and hence the number of features resulting from FLD is only one. Although it was demonstrated there that even one FLD feature could beat PCA for this particular case, it may not be the case for other two-class classification problems since it is too naive to believe that only one FLD feature would suffice for all. Therefore it is essential to eliminate this constraint completely if possible such that FLD can be applied to a much wider class of pattern classification problems. It is for this purpose that recursive FLD (RFLD) was proposed by Xiang, et al. [81] to overcome the feature number constraint using a recursive procedure.

The basic idea of RFLD may be roughly described as follows. The first feature extracted from RFLD is exactly the same as that of the FLD, but the procedure of calculating other features by RFLD, as well as the resulting feature vectors will be significantly different from FLD. While the feature vectors can be computed from a conventional eigenvalue problem once and for all by FLD, the feature vectors will be obtained recursively, step by step, by RFLD, i.e., at every step, the calculation of a new feature vector will be based upon all the feature vectors obtained from earlier iterations. More specifically, at each step when a new feature vector is computed, the training data has to be pre-processed such that all the information represented by those "old" features extracted previously will be eliminated. And then the problem of extracting the new feature most efficient for classification based upon the pre-processed database will be formulated in the same fashion as that of FLD.

Because only one feature is extracted per iteration, RFLD has the drawback of high computational complexity compared to traditional approaches.

2.3.2 LDA Based on Null Space of S_W

Another drawback of FLD is that it cannot extract discriminatory information from null space of S_W due to the non-singular requirement for S_W. From (2.11), we can see that if S_W is singular, then its inverse does not exist and the solution to FLD is not well posed. To make S_W non-singular, a typical approach is to use PCA to reduce the feature dimension, which means that the null space of S_W is discarded before FLD is applied. However, this null space also contains discriminatory information as S_B is non-zero in this subspace. To utilize information from the null space of S_W, LDA based on null space of S_W was proposed by Chen et al. [8]. Let F denote the feature space which is spanned by all feature samples. And we use $\bar{F}$ to denote the null space of the feature space. In practice, F can be estimated by the subspace spanned by the non-trivial eigenvectors of the total scatter matrix S_T, which is the sum of between-class scatter matrix S_B and within-class scatter matrix S_W: $S_T = S_B + S_W$. Let F_W denote the principal subspace of S_W, which is spanned by the non-trivial eigenvectors of S_W. The feature space can be decomposed as $F = F_W \cup \bar{F}_W$, where $\bar{F}_W$ is called the null space of S_W. LDA based on null space of S_W maximizes between-class scatter in the space $\bar{F}_W$, as the most discriminatory information is contained in this subspace. The shortcoming of this method is that it can only utilize information from $\bar{F}_W$.

In order to use all the discrimination information available, Fisher's criterion was extended to MFLD [35] as shown below.

2.3.3 Modified Fisher Linear Discriminant (MFLD)

MFLD modifies the Fisher's criterion function by replacing S_W in the denominator of (2.7) by S_T. The modified criterion function is

$$J(w) = \frac{w^T S_B w}{w^T S_T w} = \frac{w^T S_B w}{w^T S_B w + w^T S_W w} \tag{2.12}$$

It is easy to prove that the modified criterion (2.12) is equivalent to the original criterion (2.7) in the case that S_W is nonsingular. However, if S_W is singular, then all the vectors from $\bar{F}_W$ would maximize criterion (2.12) giving the maximal possible value of one to $J(w)$. This implies that all information from both F_W and $\bar{F}_W$ may be possibly utilized by MFLD. Unfortunately, the maximal number of features can be extracted by MFLD is also $C - 1$ due to the same reason as for FLD. Note that the dimension of $\bar{F}_W$ is $C - 1$ and features from $\bar{F}_W$ are most discriminant, the $C-1$ features extracted by MFLD actually span $\bar{F}_W$. Therefore, MFLD is only able to utilize information from $\bar{F}_W$ and fails to take advantage of F_W.

We can conclude that:

- in the case of singular S_W (small sample size), MFLD actually fails to utilize information from F_W. It only uses information from $\bar{F}_W$, as LDA based on null space of S_W does.
- in the case of non-singular S_W (sample size is large compared to feature dimension), MFLD is equivalent to FLD.

2.3.4 Direct FLD (DFLD)

Previously, the feature space F is decomposed as $F = F_W \cup \bar{F}_W$, another way to decompose F is $F = F_B \cup \bar{F}_B$, where F_B and $\bar{F}_B$ denote the principal subspace of S_B and its complementary null space. DFLD [85] is based on the idea that since different classes are not separated in $\bar{F}_B$, $\bar{F}_B$ contains no discriminatory information for classification. Therefore, instead of discarding $\bar{F}_W$, which contains the most discriminative information, DFLD discards $\bar{F}_B$. DFLD then searches a W from F_B that minimizes the within-class scatter.

Although the basic idea of DFLD – $\bar{F}_B$ contributes nothing to the separability of classes and thus should be discarded – seems correct, but actually it is not. To illustrate this point, a two-class problem with idealized distribution is shown in Figure 2.1.

In the figure, the means of class 1 and class 2 are at $(3, 0)$ and $(-3, 0)$. DFLD would discard the projection vector along y-axis and retain only x-axis, since S_B is zero along y-axis. However, the best projection axis that separates these two classes is along the line $y = -x$, which contains component from null space of S_B. From this simple example, we can see that although the null space does not have any information about class separability, it does help to separate classes by reducing the within-class scatter.

2.3.5 Regularized LDA

Linear discriminant analysis (LDA) like FLD has been applied for applications where the sample sizes are small and the number of measurement variables is large. One drawback of FLD that has been recognized is that it requires relatively

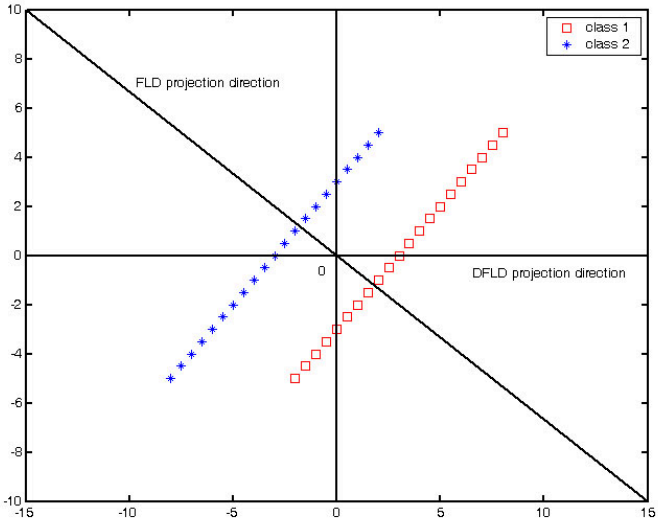


Figure 2.1: A simple examples that illustrates the deficiency of DFLD. -

large training sample size per class, compared to PCA, for good generalization [51], a typical symptom of over-fitting.

One remedy to alleviate this over-fitting problem is first proposed by Friedman [22]. For applications with small sample size and high-dimensional samples, the estimation of the within-class scatter matrix S_W by maximum-likelihood estimates incurs large variance, especially for the low-variance subspace spanned by small eigenvalues of S_W. This low-variance subspace is strongly affected by noise. By introducing a small bias, called the regularization term, the variance can be significantly reduced and the performance of LDA may be improved significantly:

$$S_W^* = S_W + \gamma I \tag{2.13}$$

where γ is a real scalar and I is the identity matrix.

2.3.6 Chernoff-based Discriminant Analysis

Although FLD and its many extensions have demonstrated their success in various applications [3, 8, 16, 35, 46, 47, 51, 73, 77, 81, 85], FLD may not deal well with data having very different covariance matrices for different classes because of the homoscedastic property of FLD.

Chernoff distance provides a measure of class separability and takes into consideration the orientation mismatch between the classes, which Mahalanobis distance based FLD fails to do. Chernoff bound forms a tight upper bound on the Bayes error for two-class problems:

$$P_e^* \leq P_2^s P_1^{1-s} \exp(-d_{ch}(f_1, f_2; s)), \quad 0 \leq s \leq 1 \tag{2.14}$$

where P_e^* is the minimal probability of error, or the Bayes error, for two classes with a priori probabilities P_1 and P_2, and conditional probability density functions f_1 and f_2; and

$$d_{ch}(f_1, f_2; s) = -\log \int f_2^s(x) f_1^{1-s}(x) dx \tag{2.15}$$

is the Chernoff distance between f_1 and f_2. Some people refer to (2.15) as Chernoff distance only when s maximizes (2.15). If $s = 1/2$, Chernoff distance given by (2.15) becomes Bhattacharya distance.

If we assume the data are Gaussian with the PDF given by

$$f(x|\mu_i, \Sigma_i) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right) \tag{2.16}$$

for class i, we can obtain the Chernoff distance between two Gaussian classes in closed-form:

$$d_{ch}(f_1, f_2; s) = \frac{s(1-s)}{2} \Delta\mu^T \Sigma^{-1} \Delta\mu + \frac{1}{2} \log\left(\frac{|\Sigma|}{|\Sigma_1|^s |\Sigma_2|^{1-s}}\right), \tag{2.17}$$

Chernoff-based discriminant analysis algorithms aim to minimize the Chernoff bound in (2.14). Loog and Duin (LD) developed an FLD like criterion based on Chernoff distance in the original space [48]. The LD criterion function for two-class case is:

$$\begin{aligned} J_{LD2}(W) &= (W^T S_W W)^{-1} \\ &\left\{ W^T \left[S_B - S_W^{\frac{1}{2}} \frac{P_1 \log(S_W^{-\frac{1}{2}} S_1 S_W^{-\frac{1}{2}}) + P_2 \log(S_W^{-\frac{1}{2}} S_2 S_W^{-\frac{1}{2}})}{p_1 p_2} S_W^{\frac{1}{2}} \right] W \right\}. \end{aligned} \tag{2.18}$$

The generalization of (2.18) to multi-class case is as follows:

$$\begin{aligned} J_{LD}(W) = &\sum_{i<j} P_i P_j (W^T S_W W)^{-1} \\ &\left\{ W^T S_W^{\frac{1}{2}} \left[(S_W^{-\frac{1}{2}} S_{Wij} S_W^{-\frac{1}{2}})^{-\frac{1}{2}} S_W^{-\frac{1}{2}} S_{Eij} S_W^{-\frac{1}{2}} (S_W^{-\frac{1}{2}} S_{Wij} S_W^{-\frac{1}{2}})^{-\frac{1}{2}} + \right.\right. \\ \frac{1}{\pi_i \pi_j} &\left.\left. \left(\log(S_W^{-\frac{1}{2}} S_{Wij} S_W^{-\frac{1}{2}}) - \pi_i \log(S_W^{-\frac{1}{2}} S_i S_W^{-\frac{1}{2}}) - \pi_j \log(S_W^{-\frac{1}{2}} S_j S_W^{-\frac{1}{2}}) \right) \right] S_W^{\frac{1}{2}} W \right\}, \end{aligned} \tag{2.19}$$

where $S_{Eij} = (\mu_i - \mu_j)(\mu_i - \mu_j)^T$, $\pi_i = \frac{P_i}{P_i + P_j}$, and $S_{Wij} = \pi_i S_i + \pi_j S_j$.

Rueda and Herrera (RH) proposed a criterion function that incorporates Chernoff distance in the transformed space [63]:

$$\begin{aligned} J_{RH2}(W) = &P_1 P_2 W^T S_B W (W^T S_W W)^{-1} + \\ &\log(W^T S_W W) - P_1 \log(W^T S_1 W) - P_2 \log(W^T S_2 W) \end{aligned} \tag{2.20}$$

The generalization of (2.20) to multi-class case is done in the same way as LD method:

$$J_{RH}(W) = \sum_{i<j} d_{ch}(f_i, f_j; \pi_i). \tag{2.21}$$

For RH methods and its extensions, a gradient descent algorithm is employed to seek the optimal solution to the criterion function [63, 75].

2.4 Nonparametric Discriminant Analysis (NDA)

As FLD calculates the between class scatter matrix by the means of every classes, it implicitly makes the assumption that the underlying distributions of each class are uni-modal, which is often not the case for real-world problems. This problem is due to the parametric nature of FLD. To overcome this problem, a nonparametric approach, named nonparametric discriminant analysis (NDA), was first proposed by K. Fukunaga in [23] for the case of two-class problems. NDA is generalized for multi-class problems by Bressan and Vitria in [6], and Li and his colleagues in [45]. It is worth mentioning that NDA does not have the constraint of total number of features available.

NDA also uses Fisher's criterion function as defined above in (2.7), but it re-defines S_B in a nonparametric way. For FLD, S_B is defined using the mean of each class as a representative of that class. This kind of definition for S_B is only suitable if the distributions of classes are uni-modal Gaussian. The nonparametric definition for S_B was first proposed by K. Fukunaga [23] for two-class problems. It is defined as:

$$S_B^N = \sum_{i=1}^{n} W(i)(x_i - \overline{m}_k(x_i))(x_i - \overline{m}_k(x_i))^T \tag{2.22}$$

where x_i is the ith data sample, $\overline{m}_k(x_i)$ denotes the mean of the k nearest neighbors of x_i that doesn't belong to the class of x_i, $W(i)$ is the weight of x_i defined by

$$W(i) = \frac{\min\{d^\alpha(x_i, m_k(x_i)), d^\alpha(x_i, \overline{m}_k(x_i))\}}{d^\alpha(x_i, m_k(x_i)) + d^\alpha(x_i, \overline{m}_k(x_i))} \tag{2.23}$$

where $m_k(x_i)$ denotes the mean of the k nearest neighbors of x_i that are from the same class as x_i. $d(v_1, v_2)$ is the distance between two vectors v_1 and v_2. α is a control parameter that can be selected between zero and infinity. This sample weight is introduced in order to emphasize samples near class boundaries. The weight has a property that for samples near class boundaries it approaches 0.5 and drops off to zero if the samples are far away from the boundaries.

To generalize to multi-class case, Li et al. [45] used the following definition:

$$S_B^N = \sum_{i=1}^{c} \sum_{j=1, j\neq i}^{c} \sum_{t=1}^{N_i} W(i,j,t)(x_t^i - m_j(x_t^i))(x_t^i - m_j(x_t^i))^T \tag{2.24}$$

and the sample weight is changed accordingly

$$W(i,j,t) = \frac{\min\{d^\alpha(x_t^i, m_k(x_t^i)), d^\alpha(x_t^i, m_k^j(x_t^i))\}}{d^\alpha(x_t^i, m_k(x_t^i)) + d^\alpha(x_t^i, m_k^j(x_t^i))} \tag{2.25}$$

where $m_k^j(x_t^i)$ is the mean of the k nearest neighbors of x_t^i that are from class j.

Bressan and Vitria [6] used a different definition for S_B^N. For each sample, all samples that are not from the same class as that sample are pulled together and treated as a single class. Thus, the multi-class problem could be treated as a 2-class problem and then definition of S_B^N in (2.22) for 2-class problem is used.

When the number of considered neighbors reaches the total number of available class samples and the sample weights are ignored, the definition of S_B by NDA is essentially the same as that of FLD. So NDA can be considered as a nonparametric extension of FLD. Notice that by the nonparametric definition of S_B, NDA is able to perform well for multi-modal class distributions and it captures the boundary structure of classes effectively. It also breaks the feature number limitation of FLD as S_B^N is generally full rank.

2.5 Locality Preserving Projection (LPP)

LPP [28] is an unsupervised learning algorithm but seems to have discriminating power. It aims to find a linear subspace that best preserves local structure and detects the essential face manifold structure. The objective function of LPP is as follows:

$$\min \sum_{ij} (y_i - y_j)^2 S_{ij} \tag{2.26}$$

where y_i is the one-dimensional representation of x_i and S_{ij} is similarity matrix, which can be defined by:

$$S_{ij} = \begin{cases} \exp(-||x_i - x_j||^2/t), & ||x_i - x_j||^2 < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

or

$$S_{ij} = \begin{cases} \exp(-||x_i - x_j||^2/t), & \text{if } x_i \text{ is among } k \text{ nearest neighbors of } x_j \\ & \text{or } x_j \text{ is among } k \text{ nearest neighbors of } x_i \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

where ϵ is small positive value, and t is some suitable constant. Here, ϵ defines the radius of the local neighborhood. In other words, ϵ defines the “locality”. The objective function with the symmetric weights S_{ij} incurs a heavy penalty if neighboring points x_i and x_j are mapped far apart, i.e., if $(y_i - y_j)^2$ is large. Therefore, minimizing it is an attempt to ensure that, if x_i and x_j are close, then y_i and y_j are close as well. After some simple algebraic manipulations, the transformation vector w that minimizes the objective function is given by the minimum eigenvalue solution to the following generalized eigenvalue problem:

$$XLX^T w = \lambda XDX^T w \tag{2.29}$$

where $X = [x_1, x_2, \cdots, x_n]$, and D is a diagonal matrix; its entries are column (or row since S is symmetric) sums of S. $L = D - S$ is the Laplacian matrix [11]. D provides a natural measure on the data points. The bigger the value D_{ii} is, the more important is y_i.

The overall procedure of the LPP algorithm is stated as follows:

1. Dimension reduction by PCA. The original high dimension of image sample vectors is reduced to a lower dimension by throwing away principal components whose corresponding eigenvalues are zero, as these components don't carry any information about the sample distributions.

2. Constructing the nearest-neighbor graph. Let G denote a graph with each node represents a sample image. We put an edge between two nodes if they are close, i.e. S_{ij} is not equal to zero.

3. Choosing the weights. If node i and j are connected, put

   $$S_{ij} = \exp(-||x_i - x_j||^2/t)$$

   Otherwise, put $S_{ij} = 0$.

4. Eidgemap. Compute the eigenvectors for the generalized eigenvector problem of (2.29). The projection vectors extracted by LPP are the set of eigenvectors corresponding to the smallest eigenvalues.

Notice that as D is full rank, and L is generally full rank. So the two matrices XLX^T and XDX^T are also generally full rank. Hence, LPP does not have the feature number limitation problem as FLD.

Chapter 3

Recursive Modified Linear Discriminant (RMLD)

In the previous chapter, a few popular feature extraction algorithms have been presented. Among them, FLD has gained its popularity probably due to its relevance to classification: it extracts features that maximize the between-class scatter and meanwhile minimize the within-class scatter. However, FLD also suffers several major limitations. And a number of enhanced or improved versions of FLD have been proposed in the past to overcome the limitations of FLD, for example, RFLD, MFLD, and DFLD, as discussed in the previous chapter. However, there are still some issues that need to be addressed. In this chapter and the rest chapters of the first part of my thesis, I will try to identify these issues and propose new algorithms that conquer them. The first algorithm, which is described in this chapter, is termed recursive modified linear discriminant (RMLD).

3.1 Objectives of RMLD

RMLD is proposed to overcome two shortcomings of FLD: 1) feature number limitation; and 2) utilize discriminant information from both F_W (principal subspace of S_W) and $\bar{F}_W$ (null space of S_W). These two shortcomings have also been attempted by RFLD and MFLD, respectively, as discussed in Chapter 2.

3.2 RMLD Algorithm

To fulfill the objectives, RMLD optimizes the criterion function of MFLD and employs a recursive strategy which is similar to RFLD. However, RMLD differs from RFLD by the following two points:

- RMLD uses the modified Fisher's criterion as defined in (2.12) in order to utilize discriminant information from both F_W and $\bar{F_W}$. Notice that MFLD actually fails to utilize discriminant information from both F_W and $\bar{F_W}$ although it also uses the modified Fisher's criterion. Nevertheless, RMLD is truly able to extracting discriminant information from both subspaces since it can extract more than $C - 1$ features by using more than one iteration.

- RMLD extracts $C - 1$ features per iteration rather than just one feature as RFLD, thus reducing the computational load significantly.

For a training set of N independent d-dimensional samples ($N \ll d$), the intrinsic dimensionality or degree-of-freedom is $N - 1$ after the mean is subtracted. In other words, the sample distribution resides in a $N - 1$-dimensional subspace. Dimensionality reduction techniques like PCA can be used to save computational load and memory requirement while ensuring it is information lossless if all non-trivial principal components are retained. As RMLD aims to utilize all the information contained in the training sample set, it first uses PCA to reduce the dimension of the samples from d to $N - 1$ such that no information is lost and the intrinsic structure of the training samples is not changed. Notice that in the case of FLD (or RFLD), the dimension of samples have to be reduced to at least $N - c$ instead of $N - 1$ in order to make S_W non-singular. The dimension reduction to $N - c$ or less implies that the distribution of the training samples is modified and some information is lost. More specifically, information from $\bar{F_W}$ is discarded after dimension reduction for FLD.

After the dimension reduction, S_T is non-singular. And RMLD can extract the first set of $C - 1$ discriminant features in the same way as MFLD. As what we have already showed in the subsection on MFLD, these $C - 1$ features constitute the null space of S_W. After the first iteration, information already extracted, which constitute the null space of S_W, is discarded and then another set of $C-1$ features are extracted. For subsequent iterations, all information extracted by previous iterations will be eliminated before going to the next iteration, just as the procedure of RFLD. The features extracted from the second iteration onwards are from the principal space of S_W. Thus, RMLD can extract discriminant features from both the null and principal space of S_W. Because there are $C-1$ features extracted at each iteration and all the extracted information are removed before going to the next iteration, the rank of S_T is reduced by $C-1$ after every iteration. So PCA is employed to reduce the dimension of the sample space by $C-1$ at each iteration so that the re-calculated S_T based on the reduced subspace is non-singular. The algorithm for RMLD is outlined as follows.

1. Use PCA to reduce the dimension of the original sample space to $n-1$, so that S_T is non-singular.

2. For the first iteration, use MFLD to extract the first $C-1$ discriminative feature vectors.

3. Discard the extracted information from all samples, i.e., the projections of the sample vectors on those "old" features will be eliminated.

   $$x_i^{(k)} = x_i^{(k-1)} - (W_{k-1}^T x_i^{(k-1)})W_{k-1} \tag{3.1}$$

   where the superscript of x_i and the subscript of W denote which iteration x_i and W come from, and W is the transformation matrix whose columns are the projection vectors extracted by each iteration. PCA is then employed to reduce the dimension of the sample space by $C-1$. Re-calculate S_B and S_T.

4. Use MFLD to extract another set of $C-1$ discriminative feature vectors.

5. If needed, go through the iteration from step 3 again to extract more feature vectors.

The dimension reduction by PCA and re-calculation of S_B and S_T in step 3 are computationally expensive. A much more efficient way is to use the null space of the extracted feature vectors and the revised algorithm for RMLD is as follows:

1. Use PCA to reduce the dimension of the original sample space to $n - 1$, so that S_T is non-singular.

2. For the first iteration, use MFLD to extract the first $C - 1$ discriminative feature vectors. Denote this set of $C - 1$ features by W_1.

3. For the kth iteration, get the null space of the extracted feature vectors, denoted as $\overline{W}_{k-1}$.

4. Discard information from extracted features by projecting S_B and S_T into the null space $\overline{W}_{k-1}$

$$S'_B = \overline{W}^T_{k-1} S_B \overline{W}_{k-1} \tag{3.2}$$

$$S'_T = \overline{W}^T_{k-1} S_T \overline{W}_{k-1} \tag{3.3}$$

   where S'_B and S'_W represents the new version of S_B and S_W.

5. Use MFLD to extract another set of $C - 1$ discriminative feature vectors, denoted by w_k.

6. Concatenate the newly extracted set of features w_k with the previous features: $W_k = [W_{k-1}, \overline{W}_{k-1} \times w_k]$.

7. If needed, go through steps 3-6 for one more iteration to extract more feature vectors. The recursive procedure terminates when desired number of features have been extracted.

3.3 Summary

In summary, RMLD uses the modified criterion function of MFLD and a novel recursive strategy to over come the feature number limit and exploit discriminant information from both F_W (principal subspace of S_W) and $\overline{F}_W$ (null space of S_W). The novel recursive strategy extracts a set of $C - 1$ features instead of only 1 feature per iteration. Thus it requires less number of iteration to extract the desired number of features. The novel recursive strategy of RMLD removes the extracted information by projecting S_B and S_W into the null space $\overline{W}_k$ of the concatenated extracted features W_k. This avoids the re-computation of S_B and S_W after projecting all samples by the null space. Due to the computational efficiency, this novel recursive strategy of RMLD is employed in my other proposed algorithms presented in the following chapters.

Chapter 4

Recursive Cluster-based Linear Discriminant (RCLD)

One major problem with traditional FLD is that it makes an implicit assumption that the underlying distribution for each class is uni-modal. This implicit assumption is made due to the mathematical formulation for S_B as defined in (2.8) and S_W as defined in (2.9), where class means μ_i are used as representatives of their respective classes. This parametric definition of S_B and S_W assumes that classes have a uni-modal distribution. However, the uni-modal assumption is often too strong to fit the real situation. For example, in the case of identity recognition, the variations of a person's image may be caused by illumination, pose and expression etc., and the distribution for one person probably contains multiple clusters, with each cluster corresponding to one particular variation. The situation of multiple clusters in each class is especially true for other face recognition tasks like facial expression recognition and glasses-wearing recognition, where each class contains images from different persons and the images from the same person are very likely to cluster together. In general, it is not unusual that the underlying classes may have a complex distribution function rather than an ideal Gaussian distribution.

It is not surprising that FLD cannot perform well if the true underlying distributions of samples are more complex than uni-modal Gaussian. This problem of FLD with multi-modal distribution of underlying classes is demonstrated by a simple 2D example as shown in Figure 4.1. In the example, there are two classes and class 2 has three clusters. The direction extracted by PCA maximizes the variances but one cluster of class 2 is mixed with class 1 after projection. FLD also fails to separate the two classes as it treats class 2 as a single cluster.

Instead of a simple uni-modal Gaussian distribution, a complex distribution function can be more appropriately approximated as a union of Gaussian distributions, or multi-modal Gaussian distributions. Therefore, instead of treating each class as a single entity, a cluster-based approach (CLD) is developed in [9, 80]. However, it will be shown later that the cluster-based approach in [9, 80] is appropriate only for cases where clusters are well formed. In the following, we propose a fuzzy-cluster-based approach, which also takes into account cases where clusters are not well-formed. The proposed fuzzy-cluster-based approach is able to perform well no matter how well clusters are formed.

4.1 Objectives of the Cluster-based Approach

Since it is more appropriate to model real-world class distribution as a union of Gaussian clusters, the objectives of the cluster-based approach, as defined in [9, 80], are to:

- maximize the distances between clusters belonging to different classes;
- minimize the distances of samples within the same clusters to keep clusters compact;
- put no constraint on clusters belonging to the same class.

To realize the above objectives of the cluster-based approach, the form of the Rayleigh quotient of S_B and S_W of FLD can be used, but the definition of the two scatter matrices should be modified to be cluster-based. There are two important steps that need to be implemented: (a) a cluster-based definition of S_B and S_W, and (b) determination of clusters.

4.2 Cluster-based Definition of S_B and S_W

First, there is one or more than one clusters in a class and we assume that we know the cluster where each sample belongs to. The definition of S_B and S_W should now be changed to take into account the relationship between different clusters as well as different classes so that the objectives of the cluster-based approach stated above can be achieved.

$$S_{B\_CLD} = \frac{1}{N} \sum_{i=1}^{C-1} \sum_{l=i+1}^{C} \sum_{j=1}^{C_i} \sum_{h=1}^{C_l} N_{ij} N_{lh} (\mu_{ij} - \mu_{lh})(\mu_{ij} - \mu_{lh})^T \tag{4.1}$$

$$S_W = \sum_{i=1}^{C} \sum_{j=1}^{C_i} \sum_{s} (x_s - \mu_{ij})(x_s - \mu_{ij})^T \tag{4.2}$$

where μ_{ij} is the mean of the jth cluster in the ith class, N_{ij} is the number of samples in the jth cluster of the ith class, C_i is the number of clusters in the ith class, and N is the total number of training samples. One point to note is that the definition for S_B above is not the same as the original one in [9], which was defined as,

$$S_{B\_CLD} = \sum_{i=1}^{C-1} \sum_{l=i+1}^{C} \sum_{j=1}^{C_i} \sum_{h=1}^{C_l} (\mu_{ij} - \mu_{lh})(\mu_{ij} - \mu_{lh})^T \tag{4.3}$$

The reason for adding the weighting element $N_{ij}N_{lh}/N$ as shown in (4.1) is to take into account the different sizes of the clusters.

The effectiveness of the cluster-based approach is illustrated in Figure 4.1. In the 2D example, CLD (the cluster-based approach) works as it takes care of the existence of multiple clusters in a class.

4.3 Determination of Clusters

The calculation of cluster-based S_B and S_W requires that the number of clusters for each class and the cluster membership of each sample to be known beforehand. So a pre-requisite for this cluster-based approach is clustering analysis. There are a variety of clustering methods. Generally speaking, the various clustering methods available can be broadly put into two categories: crisp clustering and

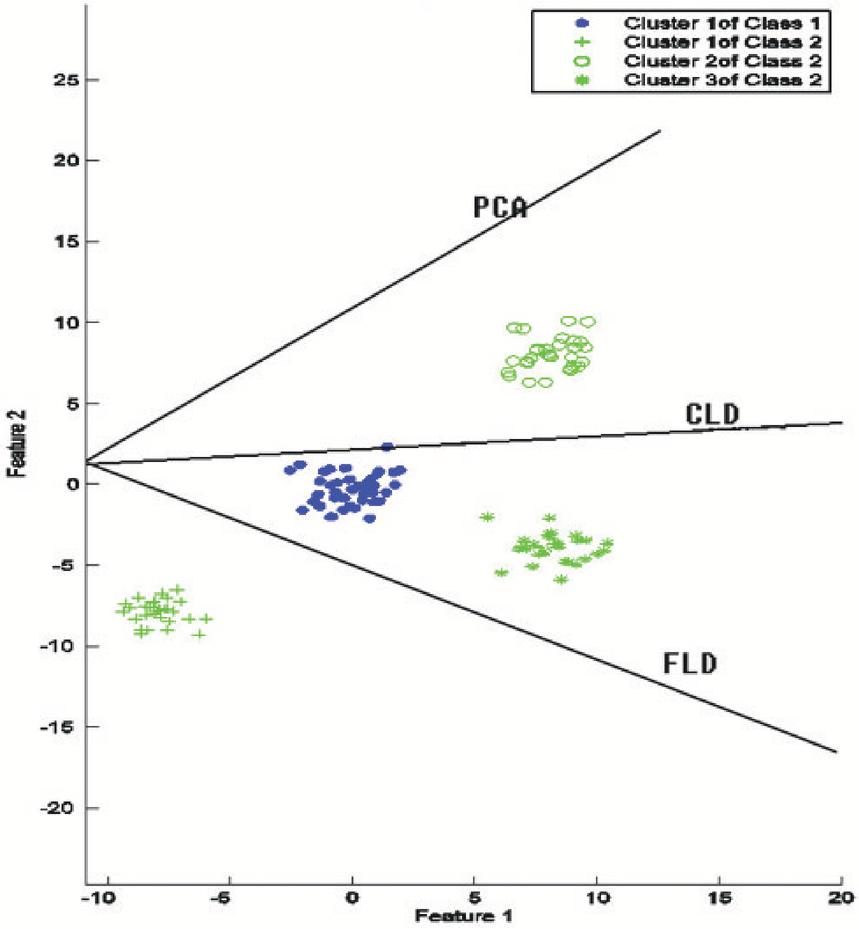


Figure 4.1: Comparison of different projection directions extracted by: PCA, FLD (or RFLD) and the cluster-based approach (CLD). -

fuzzy clustering. In crisp clustering, every sample is assigned to exactly one cluster. On the other hand, in fuzzy clustering, samples are assigned with a gradual membership to the clusters.

The ideas of crisp clustering and fuzzy clustering can be taken into account by the definition of S_W. We modify the definition of the cluster-based S_W and use S_{WW} to denote this new definition of S_W hereafter.

$$S_{WW} = \sum_{i=1}^{C} \sum_{j=1}^{C_i} \sum_{n=1}^{N} m_{ij}^n (x_n - \mu_{ij})(x_n - \mu_{ij})^T \tag{4.4}$$

where m_{ij}^n denotes the relationship of sample x_n to cluster j of class i, which is represented by its mean μ_{ij}, defined by

$$\mu_{ij} = \sum_{n=1}^{N} m_{ij}^n x_n. \tag{4.5}$$

For crisp clustering, m_{ij}^n is a binary function:

$$m_{ij}^n = \begin{cases} 1 \text{ if } x_n \in X_{ij}. \\ 0, \text{ otherwise.} \end{cases} \tag{4.6}$$

where X_{ij} denotes the set of samples that comprise the jth cluster of the ith class.

For fuzzy clustering, m_{ij}^n's are no longer constrained to be equal to 0 or 1. Instead, they can take any value in the interval $[0, 1]$. m_{ij}^n indicates the degree to which sample x_n belongs to the cluster X_{ij}. The greater the m_{ij}^n, the larger the degree of the belongness of x_n to X_{ij}. Clustering is done for each class separately such that

$$\sum_{j=1}^{C_i} m_{ij}^n = 1 \text{ for } L(x_n) = i, \text{ and } m_{ij}^n = 0 \text{ for } L(x_n) \neq i \tag{4.7}$$

where $L(x_n)$ means the class label of sample x_n.

With fuzzy clustering, samples close to the center of a cluster have a higher weight. A proper selection of the degree of fuzziness is important for a good performance of the cluster-based approach. If clusters are well formed, then less fuzziness should be selected. But if clusters are not well formed, then more fuzziness should be used. With high degree of fuzziness used for the clustering process, m_{ij}^{n}'s will tend to be close to each other for all samples of the same class. This means that S_{WW} will be close to the S_W of traditional FLD. Correspondingly, if less fuzziness is chosen for the clustering process, then m_{ij}^{n} will be close to either 1 or 0, and S_{WW} will be close to S_{WW} of crisp clustering.

Fuzzy clustering is more advantageous compared to crisp clustering especially in the case where clusters are not well formed. To confirm this, simple experiments can be designed. Two toy data sets are created. To make it easy to analyze and visualize, the samples in the two data sets are 2D samples, and there are only 2 classes. For the first data set, clusters are close to each other. For the second data set, clusters are far from each other, i.e., clusters are well formed. The results of applying fuzzy clustering based approach, crisp clustering based approach, and traditional FLD, are plotted in Figure 4.2 and Figure 4.3.

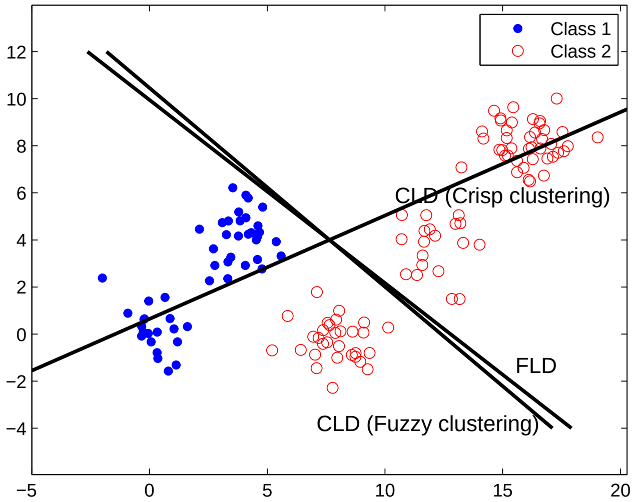


Figure 4.2: Comparison of different projection directions extracted by FLD, crisp clustering based approach and fuzzy clustering based approach on toy data set 1. -

From the two figures, we can see that crisp clustering based approach could not extract good feature for data set 1 and FLD does not perform well for data set 2. This means that crisp clustering based approach could not perform well

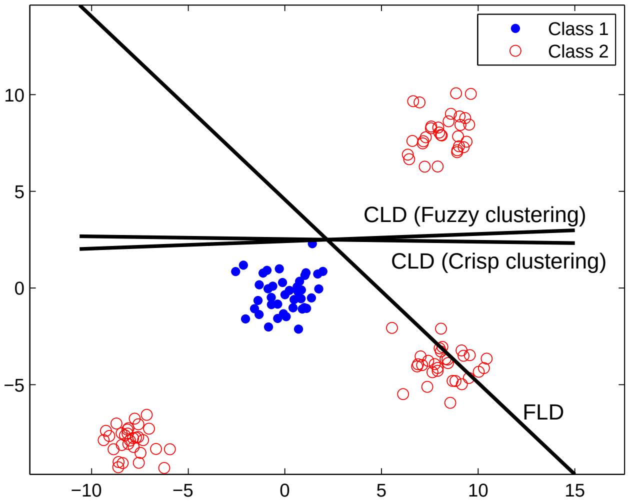


Figure 4.3: Comparison of different projection directions extracted by FLD, crisp clustering based approach and fuzzy clustering based approach on toy data set 2. -

when clusters are not well formed, and FLD does not perform well when there are well formed clusters in a class. Fuzzy clustering based approach can extract good features for both data sets. It exhibits a more robust performance compared to FLD and crisp clustering based approach.

4.4 Determination of Cluster Number

Most clustering algorithms require the number of clusters to be known beforehand. In our experiments, we used K-means clustering and fuzzy C-means clustering [82] for crisp and fuzzy clustering, respectively. Both of them require the number of clusters as an input parameter. The number of clusters specified affects the clustering process and therefore also affects the performance of CLD. One straightforward way to determine the number of clusters for every class is to project the samples into a 2D or 3D space, where the scattering of samples can be visualized. The low dimensional space can be determined by PCA. After projecting to the low dimensional space, the number of clusters can be visually inspected.

The drawback of the above method is that the 2D or 3D PCA subspace is often too low to adequately represent the scattering of the data samples. Furthermore, the subspace extracted by CLD is very different from the PCA subspace. In addition to these adversities, the number of clusters may not be easily determined by subjective visual assessment when clusters are not well separated. This problem is illustrated in Figure 4.4 for facial expression recognition problem on Yale face database. From the figure, it is very hard to tell how many clusters are contained in each class. Take the class on the bottom right of the figure for example, it seems that no well-separated cluster is formed and the number of clusters could not be well determined. So this PCA subspace method may not be appropriate for the determination of cluster numbers.

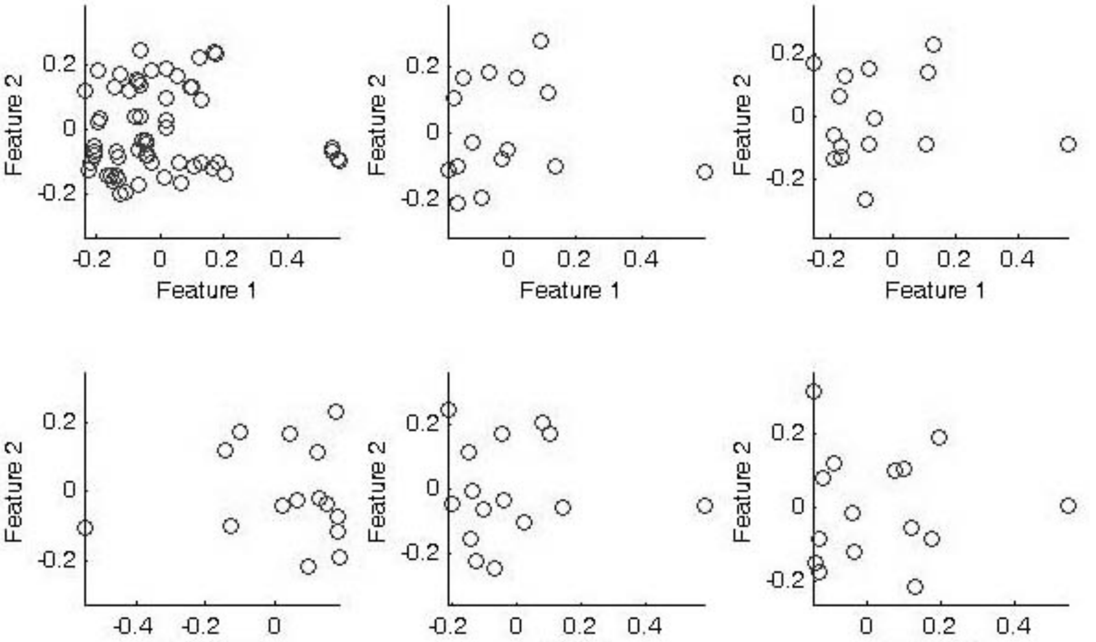


Figure 4.4: Sample distribution of Yale database in the 2D principal subspace extracted by PCA. From left to right, up to down, the distributions correspond to facial expressions: normal, wink, happy, sad, sleepy, and surprise. -

A more effective way using self-organizing map (SOM) [40, 65] to determine the cluster number is proposed here. All data samples from all classes in the training set are used to train the SOM network. After training, close samples within a class will be form clusters in the transformed space defined by SOM. Therefore, the number of clusters in a class is the number of clusters of that class in the trained SOM.

An example of using SOM to determine the number of clusters in each class is shown by Figure 4.5. The example is a facial expression recognition problem. The samples used for training the SOM is from The Japanese Female Facial Expression (JAFFE) Database [49]. There are seven expressions, i.e., seven classes, in this database. Figure 4.5 shows the resulted structure of the trained SOM. In the figure, there is a number on each unit of the trained SOM. This number is the class number that the respective unit is assigned during a labeling procedure after training is completed. This type of training is called supervised training of SOM [39, 40, 41]. The number of clusters for each class is obtained by counting the number of clusters in the trained SOM. For example, in the figure there are two clusters for class 1. So the number of clusters for class 1 is two.

According to our experience, cluster numbers determined using SOM are very close to the optimal number of clusters which results in the highest classification accuracy. The number of clusters determined using SOM can serve as an starting point. Further fine-tuning of this starting point by one or two number of clusters can be done to obtain better performance.

4.5 Incorporation of a Recursive Strategy

To relax the constraint on the number of features, we apply the recursive strategy of RMLD. RCLD adopts the redefined formula (4.1) and (4.4) for S_B and S_W by doing clustering analysis first and obtains $S_T = S_B + S_W$. Then it follows the same procedure as that of RMLD.

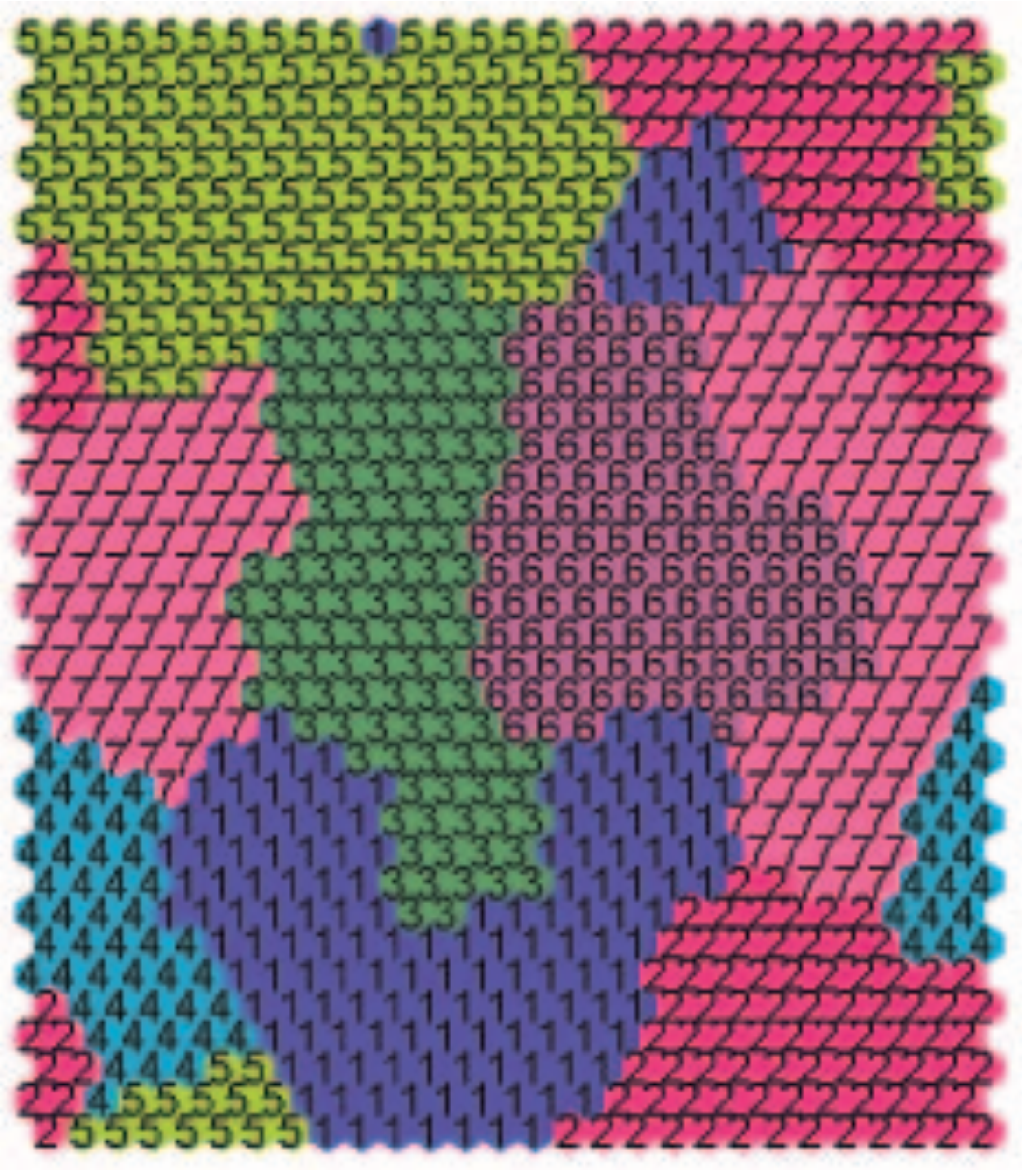


Figure 4.5: Determination of cluster number by SOM. After training of the SOM, the number of clusters in a class is the number of clusters of that class in the trained SOM. -

Chapter 5

Recursive Bayesian Linear Discriminant (RBLD)

In the previous two chapters, we have addressed several obvious limitations of FLD. However, there is another more subtle issue which is related to the relation between the criterion function and the classification error.

Since the goal of a pattern recognition system is to recognize a pattern correctly, an intuitive measure of "goodness" of the extracted features is the probability of classification error, i.e. the extracted set of features should be the one with which the classification result is as close to the minimum probability of classification error, or the Bayes error, as possible.

However, popular feature extraction algorithms do not extract features based on a criterion that is directly related to the probability of classification error. For example, PCA extracts features that are most efficient for representation, FLD maximizes the between-class scatter and meanwhile minimizes the within-class scatter, and ICA extracts statistically independent features. Although FLD is more pertinent to classification, its criterion function is not directly related to the classification performance and the maximization of its criterion function only leads to the minimal classification error under very special conditions, which are going to be shown later in this chapter.

The novel linear discriminant, coined Recursive Bayesian Linear Discriminant (or RBLD), is devised to aim at approaching the Bayes error. In the following, the new criterion function is first derived for two-class problems. Then it is generalized to multi-class problems. It will be shown that the maximization of the Bayesian criterion function is equivalent to the minimization of one of two coherent error bounds that confine the Bayes error, under certain assumptions and approximations.

5.1 The Criterion Based on the Bayes Error

The probability of classification error, P_e, can be expressed as:

$$P_e = \sum_{i=1}^{C} P_i e_i \tag{5.1}$$

where P_i is the a priori probability of class i, and e_i is the probability of error from class i, defined as:

$$e_i = \int_{\bar{R}_i} p_i(x)dx \tag{5.2}$$

where $\bar{R}_i$ is the region assigned to all other classes except class i, and $p_i(x)$ is the conditional probability density function of class i.

To derive our Bayesian criterion function, we first derive the functional form of Bayes error for the simplest case: two homoscedastic normally distributed classes with equal a priori probabilities. The two-class Bayes criterion function is then extended for general multi-class problems.

5.1.1 Two-class Bayes criterion function

Figure 5.1 shows the probability density functions of two normal classes with equal covariance matrices and equal a priori probabilities after projection onto feature vector direction w, $y = w^T x$. The probability of classification error after projection onto w can be expressed as follows:

$$F(w) = P_1 \int\limits_{\frac{y_0-\mu'_1}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2})dy + P_2 \int\limits_{\frac{\mu'_2-y_0}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{y^2}{2})dy \tag{5.3}$$

where μ'_i and σ^2 is the mean and variance of class i after projection onto w:

$$\mu'_i = w^T \mu_i \tag{5.4}$$

$$\sigma^2 = w^T \Sigma w \tag{5.5}$$

where μ_i and Σ are the mean and covariance matrix of class i in the original space. Note that we used μ'_i and σ instead of $\mu'_i(w)$ and $\sigma(w)$ in order to make the notation as simple as possible, although they are dependent on w. For the same reason, we used the notation y_0 and Σ' instead of $y_0(w)$ and $\Sigma'(w)$ later on although the notations appended with "(w)" explicitly indicate the dependence of the variable on w. Without loss of generality, we assumed $\mu'_1 \leq \mu'_2$ in (5.3).

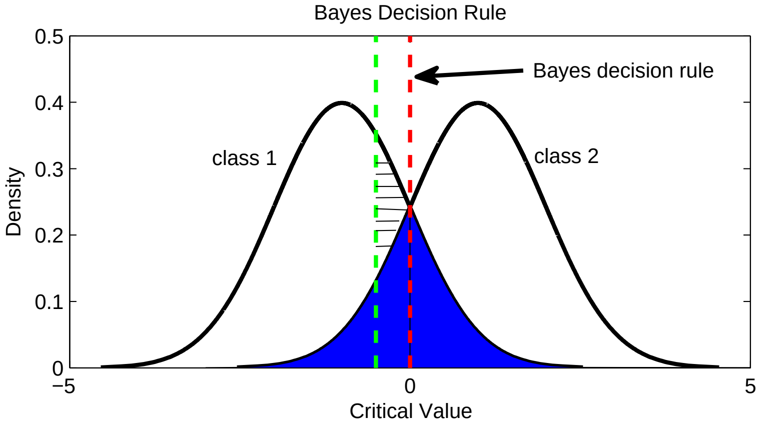


Figure 5.1: Minimum classification error by Bayes rule for the simplest case: two normal classes with equal covariance and equal a priori probabilities. -

From Figure 5.1, it is obvious that classification error depends on the position of the decision boundary y_0. From Bayesian decision theory [14], y_0 that minimizes (5.3) is determined by:

$$P_1 \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}\frac{(y_0 - \mu'_1)^2}{\sigma^2}) = P_2 \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}\frac{(y_0 - \mu'_2)^2}{\sigma^2}) \tag{5.6}$$

which can be simplified to

$$y_0 = \frac{\mu'_1 + \mu'_2}{2} - \frac{\sigma^2 \ln \frac{P_2}{P_1}}{(\mu'_2 - \mu'_1)} \tag{5.7}$$

Introducing (5.7) for y_0 into (5.3), the Bayes error $F(w)$ can then be written in the following form:

$$F(w) = \frac{1}{2} - \frac{1}{2}\left\{ P_1 erf\left(\frac{\mu'_2 - \mu'_1}{\sqrt{8}\sigma} - \frac{\sigma \ln(P_2/P_1)}{\sqrt{2}(\mu'_2 - \mu'_1)} \right) + P_2 erf\left(\frac{\mu'_2 - \mu'_1}{\sqrt{8}\sigma} + \frac{\sigma \ln(P_2/P_1)}{\sqrt{2}(\mu'_2 - \mu'_1)} \right) \right\} \tag{5.8}$$

where $erf(\cdot)$ is the error function of the normal distribution and is defined as:

$$erf(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-t^2} dt \tag{5.9}$$

If we let

$$J(w) = 2(1 - F(w)) - 1, \tag{5.10}$$

minimizing the Bayes error $F(w)$ in (5.8) is thus equivalent to maximizing $J(w)$, which can be written as

$$J(w) = P_1 erf\left(\frac{\mu'_2 - \mu'_1}{\sqrt{8}\sigma} - \frac{\sigma \ln(P_2/P_1)}{\sqrt{2}(\mu'_2 - \mu'_1)}\right) + P_2 erf\left(\frac{\mu'_2 - \mu'_1}{\sqrt{8}\sigma} + \frac{\sigma \ln(P_2/P_1)}{\sqrt{2}(\mu'_2 - \mu'_1)}\right) \tag{5.11}$$

where $J(w) + 1$ is actually two times the probability of correct classification. Therefore, the criterion function $J(w)$ represents a measure of the probability of correct classification.

Assuming equiprobable classes ($P_1 = P_2$), eq. (5.11) gives

$$J(w) = erf\left(\frac{\mu'_2 - \mu'_1}{\sqrt{8}\sigma}\right) \tag{5.12}$$

which can be written as

$$J(w) = erf\left(\frac{h'_{12}}{\sqrt{8}}\right), \tag{5.13}$$

where h'_{12} is the Mahalanobis distance between the two class means in the projection subspace

$$h'_{12} = \sqrt{(w^T\mu_1 - w^T\mu_2)(w^T\Sigma w)^{-1}(w^T\mu_1 - w^T\mu_2)} = \frac{\mu'_2 - \mu'_1}{\sigma}. \tag{5.14}$$

5.1.1.1 Comments

While some high dimensional data may not be Gaussian, often its low dimensional projection may become more Gaussian by the virtue of the central limit theorem. For situations where the covariance matrices of the two classes are different, the decision boundary is a curve in the original space by Bayes decision theory, and it is not easy to derive simple closed-form expression like (5.8) or (5.13). If the covariance matrices of the two classes do not differ a lot, which is true for some real-world applications, the assumption of equal covariances could still be appropriate and $F(w)$ in (5.8) (or equivalently $J(w)$ in (5.13)) is an approximation of the Bayes error that has a reasonable credit.

5.1.2 Multi-class Generalization of the Bayes Criterion Function

The generalization of the criterion function $J(w)$ to multi-class is not trivial because the probability of error P_e is not simply equal to the sum of the errors generated by each pair of classes when there are more than two classes. The situation of multi-class problems is usually too complex to derive a nice and simple expression like (5.13) for the Bayes error. To solve this problem, we first consider two extreme scenarios where simple expressions can be derived. These two extreme scenarios are then used to derive lower and upper bounds of the Bayes error since real situations are usually in between of these two extreme scenarios. The two extreme scenarios and the derivation of the lower and upper error bounds are described in the following.

Let $R_{j|i}$ denote the region that is assigned to class j by the Bayes rule when considering only the two classes i and j, $e_{j|i}$ denote the probability of samples from class i being misclassified to class j by the Bayes decision rule when considering only the two classes i and j. For example, $R_{2|1}$ and $e_{2|1}$ for the two classes in Figure 5.1 is the region on the right side of y_0 and the probability of samples from class 1 being misclassified to class 2 by Bayes rule for the two classes, respectively.

Lemma 1. *If* $\bar{R}_i = \sum_{j\neq i} R_{j|i}$ *then* $e_i = \sum\limits_{j\neq i} e_{j|i}$.

Proof. Because $\bar{R}_i = \sum_{j\neq i} R_{j|i}$ and $e_{j|i} = \int_{R_{j|i}} p_i(x)dx$, we have

$$e_i = \int_{\bar{R}_i} p_i(x)dx = \sum_{j\neq i} \int_{R_{j|i}} p_i(x)dx = \sum_{j\neq i} e_{j|i} \tag{5.15}$$

□ □

Since $\bar{R}_j = \cup_{j\neq i} R_{j|i}$, it is evident that $\bar{R}_i = \sum_{j\neq i} R_{j|i}$ is equivalent to $R_{j|i} \cap R_{k|i} = \phi \quad \forall k \neq j$, i.e., there is no intersection between the regions where class i is misclassified to other classes. Under this condition, the probability of error from class i is equal to the sum of the probability of class i being classified to every other class individually.

Let $P_{j|i}$ denote the a priori probability of class j after class i is taken out of consideration, i.e., $P_{j|i} = \frac{P_j}{1-P_i}$ since all classes are assumed to be independent of each other. We have the following lemma.

Lemma 2. *If $\bar{R}_i = R_{j|i} \forall j \neq i$, then $e_i = \sum\limits_{j \neq i} P_{j|i} e_{j|i}$.*

Proof. Because $\bar{R}_i = R_{j|i} \forall j \neq i$, we have $\bar{R}_i = \sum\limits_{j \neq i} P_{j|i} R_{j|i}$. It follows that

$$e_i = \int_{\bar{R}_i} p_i(x)dx = \sum_{j \neq i} P_{j|i} \int_{R_{j|i}} p_i(x)dx = \sum_{j \neq i} P_{j|i} e_{j|i} \tag{5.16}$$

□ □

Since $\bar{R}_i = R_{j|i} \forall j \neq i$ is equivalent to $R_{j|i} = R_{k|i} \quad \forall k \neq j$, it means the region where class i is misclassified to class j overlaps completely with misclassified regions for every other class k. Under this condition, the probability of error from class i is equal to the sum of the probability of class i being classified to every other class j multiplied with $P_{j|i}$.

The above two lemmas describe two extreme scenarios: no overlapping and complete overlapping. In real situations, partial overlapping is most likely.

Lemma 3. *If $\bar{R}_i \neq \sum_{j \neq i} R_{j|i}$ and $\bar{R}_i \neq R_{j|i} \; \exists j \neq i$, then $\sum\limits_{j \neq i} P_{j|i} e_{j|i} < e_i < \sum\limits_{j \neq i} e_{j|i}$.*

Proof. For partial overlapping of $R_{j|i}$'s, we have $\sum\limits_{j \neq i} P_{j|i} R_{j|i} < \bar{R}_i < \sum\limits_{j \neq i} R_{j|i}$. Therefore $\sum\limits_{j \neq i} P_{j|i} e_{j|i} < e_i = \int_{\bar{R}_i} p_i(x)dx < \sum\limits_{j \neq i} e_{j|i}$ □ □

Then P_e in (5.1) is bounded as follows:

$$\sum_i \sum_{j \neq i} P_i P_{j|i} e_{j|i} \leq P_e \leq \sum_i \sum_{j \neq i} P_i e_{j|i} \tag{5.17}$$

Using the fact that $\sum\limits_i \sum\limits_{j \neq i} a_{ij} = \sum\limits_{i<j} (a_{ij} + a_{ji})$, (5.17) can be rewritten as

$$\sum_{i<j} (P_i P_{j|i} e_{j|i} + P_j P_{i|j} e_{i|j}) \leq P_e \leq \sum_{i<j} (P_i e_{j|i} + P_j e_{i|j}). \tag{5.18}$$

Denoting the Bayes error from classifying the two classes i and j by $e_{i,j}$, it is

$$e_{i,j} = \frac{P_i}{P_i + P_j} e_{j|i} + \frac{P_j}{P_i + P_j} e_{i|j}. \tag{5.19}$$

We have

$$(P_i + P_j)e_{i,j} = P_i e_{j|i} + P_j e_{i|j}, \tag{5.20}$$

and

$$P_i P_j e_{i,j} = \frac{P_i^2 P_j}{P_i + P_j} e_{j|i} + \frac{P_i P_j^2}{P_i + P_j} e_{i|j} < \frac{P_i^2 P_j}{P_i - P_i^2} e_{j|i} + \frac{P_i P_j^2}{P_j - P_j^2} e_{i|j} = P_i P_{j|i} e_{j|i} + P_j P_{i|j} e_{i|j}. \tag{5.21}$$

Using (5.18), (5.20), and (5.21), it follows that

$$\sum_{i<j} P_i P_j e_{i,j} < P_e \leq \sum_{i<j} (P_i + P_j) e_{i,j}. \tag{5.22}$$

Since the two-class Bayes error $e_{i,j}$ in (5.22) is estimated by $F(w)$ defined in (5.8) or equivalently $(1 - J(w))/2$ from (5.13), the minimization of the two error bounds is equivalent to the maximization of the following two criterion functions:

$$J(w) = \sum_{i<j} P_i P_j erf\left(\frac{h'_{ij}}{\sqrt{8}}\right) \tag{5.23}$$

or

$$J(w) = \sum_{i<j} (P_i + P_j) erf\left(\frac{h'_{ij}}{\sqrt{8}}\right) \tag{5.24}$$

5.1.2.1 Comments

We have derived lower and upper bounds of Bayes error in eq. (5.22) for multi-class problems, under certain assumptions and approximations. One natural question is "how 'tight' are the bounds given in (5.22), taken into account the further weakening in (5.21)?" Typically one would like to choose bounds that are asymptotic or very tight. Unfortunately, it is very difficult to know whether if the two bounds in (5.22) are asymptotic or to evaluate their tightness. However, if we compare the two bounds, we can see that they have similar forms with the only difference being the multiplier before $e_{i,j}$. The multiplies $P_i P_j$ and $P_i + P_j$ results in a nice property between the two error bounds: the two bounds are coherent, in the sense that a decrease in the value of one bound usually couples with a decrease in the value of the other bound. The minimization of one bound probably makes the other bound near its minimal value. In the special case of

$P_i = P_j$ for all $j \neq i$, these bounds become equivalent to each other, with the minimization of one bound ensuring the minimization of the other bound. Similar observations can be made on criterion functions (5.23) and (5.24).

Although it is a common way to choose the upper error bound for minimization, we argue that the lower bound can also be used to make P_e small since the coherence property as discussed above. One can choose either one of the two criterion functions (5.23) and (5.24). In this paper, we deliberately selected (5.23) for the derivation of our Bayesian criterion function. The reason we selected (5.23) instead of (5.24), which may be the common choice, is that we will show later FLD can been seen as a special case of our Bayesian criterion function if (5.23) is used.

5.2 Maximization of the Bayesian Criterion Function

In the sequel, the a priori probability of class i is estimated by N_i/N where N_i and N is the number of samples in class i and the total number of samples, respectively. To maximize (5.23), we take the derivative $\frac{\partial J(w)}{\partial w}$ and set it equal to 0. Thus it is

$$\begin{aligned}\frac{\partial J(w)}{\partial w} &= \sum_{i<j} \frac{\frac{N_i N_j}{N^2} \partial erf(\frac{h'_{ij}}{\sqrt{8}})}{\partial w} = \sum_{i<j} \frac{2}{\sqrt{\pi}} \frac{N_i N_j}{N^2} \frac{\partial}{\partial w} \int_0^{\frac{h'_{ij}}{\sqrt{8}}} e^{-x^2} dx \\ &= \frac{1}{\sqrt{2\pi} N^2} \sum_{i<j} (N_i N_j) e^{-\frac{h'^2_{ij}}{8}} \frac{\partial h'_{ij}}{\partial w} = \frac{1}{\sqrt{8\pi} N^2} \sum_{i<j} e^{-\frac{h'^2_{ij}}{8}} (h'_{ij})^{-1} \frac{(N_i N_j) \partial h'^2_{ij}}{\partial w} = 0\end{aligned} \tag{5.25}$$

Since it is very difficult to derive a closed-form solution for (5.25), the following approximation is used:

$$\frac{1}{\sqrt{8\pi} N^2} \sum_{i<j} e^{-\frac{h'^2_{ij}}{8}} (h'^{-1}_{ij}) \frac{(N_i N_j) \partial h'^2_{ij}}{\partial w} \approx \frac{1}{\sqrt{8\pi} N^2} \sum_{i<j} e^{-\frac{h^2_{ij}}{8}} (h^{-1}_{ij}) \frac{(N_i N_j) \partial h'^2_{ij}}{\partial w} = 0 \tag{5.26}$$

where the projected Mahalanobis distance h'_{ij} in the coefficients of the derivatives is replaced by the original Mahalanobis distance h_{ij}.

To avoid problems related to limited sample size, it is a common practice to estimate a single common covariance matrix for all classes instead of C different covariance matrices, one for each class. The covariance matrix estimated by samples from all classes is called pooled covariance matrix. The pooled covariance matrix is actually the same as S_W up to a scaling factor, i.e., $\Sigma = S_W/N$. So the squared Mahalanobis distance $h_{ij}'^2$ after projection onto w can be estimated as

$$\begin{aligned}(N_iN_j)h_{ij}'^2 = (N_iN_j)(\mu_i' - \mu_j')(w^T\Sigma w)^{-1}(\mu_i' - \mu_j') = \\ \frac{w^T(N_iN_j)(\mu_i - \mu_j)(\mu_i - \mu_j)^T w}{w^T(S_W/N)w} = N\frac{w^T S_{ij} w}{w^T S_W w}\end{aligned} \tag{5.27}$$

where S_{ij} is given by

$$S_{ij} = N_iN_j(\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{5.28}$$

So (5.26) can be written as:

$$\frac{1}{\sqrt{8\pi}N^2}\sum_{i<j} e^{-\frac{h_{ij}^2}{8}}(h_{ij}^{-1})\frac{N\partial\left(\frac{w^T S_{ij} w}{w^T S_W w}\right)}{\partial w} = \frac{1}{\sqrt{8\pi}}\frac{\partial\left(\frac{w^T S_{B\_RBLD} w}{w^T S_W w}\right)}{\partial w} = 0 \tag{5.29}$$

where

$$S_{B\_RBLD} = \frac{1}{N}\sum_{i<j} S_{Bij} \ , \tag{5.30}$$

and

$$S_{Bij} = e^{-\frac{h_{ij}^2}{8}}(h_{ij}^{-1})S_{ij} \ . \tag{5.31}$$

and S_{ij} is defined above in (5.28).

The solution of (5.29) is obvious. It is similar to the one for the FLD criterion function. The Bayesian approach maximizes

$$J(w) = \frac{w^T S_{B\_RBLD} w}{w^T S_W w}. \tag{5.32}$$

5.2.1 Comparison of RBLD to FLD

Compared to FLD, RBLD's criterion function (5.32) have the same form as (2.7). The difference lies in the definition for S_B:

For FLD,

$$S_B = \frac{1}{N}\sum_{i<j} S_{ij} \tag{5.33}$$

For RBLD,

$$S_{B\_RBLD} = \frac{1}{N}\sum_{i<j} S_{Bij} = \frac{1}{N}\sum_{i<j} e^{-\frac{h_{ij}^2}{8}}(h_{ij}^{-1})S_{ij}. \tag{5.34}$$

Comparing the formulation of S_B of BLD to that of FLD, we can observe that BLD puts a weighting factor for S_{ij}. The weighting factor has the property that it decreases as the Mahalanobis distance between class centers h_{ij} increases, as can be seen in Figure 5.2(b). This means that the weighting factor suppresses the influence of far distant classes, or in other words, they put more emphasis on close classes. This makes sense intuitively since close classes are more likely to generate classification errors and therefore require more attention than distant classes.

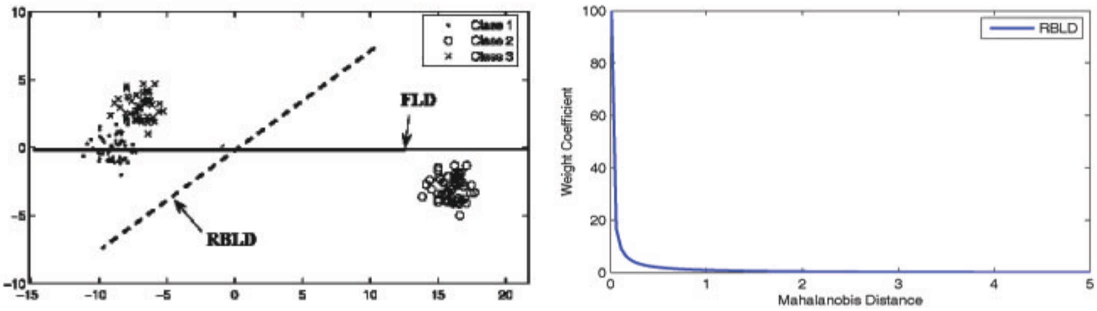
To see the effect of the weighting factor, a simple 2D classification problem is shown in Figure 5.2(a). This simple 2D classification problem illustrates that features extracted by FLD is over-influenced by far apart classes, while the Bayesian linear discriminant pays more attention on close classes.

Mathematically speaking, FLD maximizes the sum of squared Mahalanobis distances between class means in the transformed feature space. Hence, the feature directions w extracted by FLD are over-influenced by far apart classes. In contrast, BLD finds w that minimizes one of the two error bounds. We can see that FLD is a special case of the Bayesian linear discriminant: for two-class problems, FLD is equivalent to BLD; for multi-class problems, FLD is equivalent to BLD only when all classes are equally separated (The Mahalanobis distances h_{ij} are the same for all classes).

5.2.2 Summary

To make clear the assumptions and approximations we have made in the derivation of the Bayesian criterion function (5.32), we summarize them here:

- $P_i = P_j$. It is a valid assumption when the a priori probabilities of different classes are not very much different.



(a) (b)

Figure 5.2: (a) Left: A simple 2D example that shows: FLD is over-influenced by class pairs that are far apart; RBLD is able to extract good features by paying more attention to close classes; (b) Right: The weighting factor as a decreasing function of Mahalanobis distance.

- Classes have equal covariance matrices in the transformed feature space.
- The Bayesian criterion function (5.32) is derived from the minimization of the error bound in (5.22).
- The mahalanobis distance in the original space is used in place of that in the transformed feature space in (5.26) to derive a closed-form solution to (5.25).

The validity of the assumption "classes are normal with equal a priori probabilities and equal covariance matrices." depends on the specific application at hand. The Bayesian criterion function (5.32) is derived by minimizing one of the two error bounds that are coherent. The mahalanobis distance in the original space is used as a rough approximation of that in the transformed feature space. Usually classes that are closer in the original space remain closer in the transformed space. This means the use of original mahalanobis distance may still be a good approximation as more attention (larger weights) are paid to closer classes. In spite of the weakening of BLD due to the assumptions and approximations, it is still possible for BLD to achieve good results even for applications with strong violation of assumption due to two reasons: (1) the summation in the criterion function may cancel out the adverse effect of each individual deviation from the assumption; (2) the number of samples available for training is usually quite limited and as a result simple models with less parameters are usually favored. The assumption of equal covariance matrices may lead to improved results for some applications since there are less parameters to estimate. The above two reasons are discussed extensively in the literature for algorithms like the naive Bayes classifier to account for its superior performance in spite of its strong assumptions [27].

5.3 Incorporation of a Recursive Strategy

To conquer the feature number limitation inherent in FLD, the same recursive strategy as described in Chapter 3, where a set of $C-1$ features can be extracted per iteration, is adopted.

Chapter 6

Recursive Cluster-based Bayesian Linear Discriminant (RCBLD)

Since the cluster-based approach and the Bayesian approach have been described in the previous chapters, we are now ready to integrate the idea of the cluster-based approach and the Bayesian approach so that the resulted algorithm aims at approaching minimal classification error and is capable of handling complex class distributions. This integration and the resulted algorithm, called cluster-based Bayesian linear discriminant (CBLD), is shown in Section 6.1. After the presentation of CBLD, the integration of CBLD with RMLD is then brought up in Section 6.2 and the new algorithm is termed recursive CBLD (RCBLD).

6.1 Cluster-based Bayesian Linear Discriminant (CBLD)

In order to have a Bayesian criterion function, the idea of weighting factor by the Bayesian approach for each pair of classes is adapted to each pair of clusters, and the resulting criterion function is as follows:

$$S_{B\_CBLD} = \frac{1}{N}\sum_{i=1}^{C-1}\sum_{l=i+1}^{C}\sum_{j=1}^{C_i}\sum_{h=1}^{C_l} S_{ijlh} \qquad (6.1)$$

where S_{ijlh} is defined as:

$$S_{ijlh} = e^{-\frac{h_{ijlh}^2}{8}} h_{ijlh}^{-1} N_{ij} N_{lh} (\mu_{ij} - \mu_{lh})(\mu_{ij} - \mu_{lh})^T \tag{6.2}$$

where h_{ijlh} is the Mahalanobis distance between the means of cluster j of class i and cluster h of class l:

$$h_{ijlh} = N(\mu_{ij} - \mu_{lh})^T S_{WW}^{-1} (\mu_{ij} - \mu_{lh}) \tag{6.3}$$

where S_{WW}/N is the pooled covariance matrix of the clusters and is estimated using the within-cluster scatter matrix (defined in (4.4)) divided by the total number of samples N.

The definition for S_{WW} remains the same as that of RCLD, which is defined in (4.4).

Note that if we simply use the within-cluster scatter matrix S_{WW} in the denominator of the criterion function, S_{WW} must be non-singular so that its inverse exists. As a result, discriminant information from the null space of S_{WW} can not be extracted. To overcome this limitation, S_{WW} can be replaced by S_T, as in the case of RMLD discussed previously in Chapter 3. However, S_T can not be simply calculated in the traditional way as in (6.4) for FLD:

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T. \tag{6.4}$$

This is because S_{B_CBLD} and S_{WW} are calculated with weights. We can use the decomposition of total scatter matrix S_{T_CBLD} for CBLD as follows:

$$\begin{aligned} S_{T\_CBLD} &= S_{B\_CBLD} + S_{WB} + S_{WW} \\ &= \frac{1}{N} \sum_{i<l} \sum_{j=1}^{C_i} \sum_{h=1}^{C_l} e^{-\frac{h_{ijlh}^2}{8}} h_{ijlh}^{-1} N_{ij} N_{lh} (\mu_{ij} - \mu_{lh})(\mu_{ij} - \mu_{lh})^T \\ &+ \frac{1}{N} \sum_{i=1}^{C} \sum_{1 \le j < h \le C_i} e^{-\frac{h_{ijih}^2}{8}} h_{ijih}^{-1} N_{ij} N_{ih} (\mu_{ij} - \mu_{ih})(\mu_{ij} - \mu_{ih})^T \\ &+ \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} \sum_{n=1}^{N} m_{ij}^n (x_n - \mu_{ij})(x_n - \mu_{ij})^T \end{aligned} \tag{6.5}$$

where S_{WB} denotes the within-class between-cluster scatter that is not contained in S_{B_CBLD} or S_{WW}.

Now the criterion function of CBLD has the form

$$J(w) = \frac{w^T S_{B\_CBLD} w}{w^T S_T w} = \frac{w^T S_{B\_CBLD} w}{w^T (S_{B\_CBLD} + S_{WB} + S_{WW}) w} \tag{6.6}$$

where

$$S_{WB} = \frac{1}{N} \sum_{i=1}^{C} S_{WBi} \tag{6.7}$$

and

$$S_{WBi} = \sum_{1 \leq j < h \leq C_i} e^{-\frac{h_{ijih}^2}{8}} h_{ijih}^{-1} N_{ij} N_{ih} (\mu_{ij} - \mu_{ih})(\mu_{ij} - \mu_{ih})^T \tag{6.8}$$

The denominator of $J(w)$ has been changed from S_{WW} to a sum which also takes into account the within-class between-cluster scatter S_{WB}.

In our implementation, PCA is employed to reduce the sample dimension to $N - 1$ instead of $N - C$, as typically done to make S_W non-singular. This is possible as the denominator of our new criterion function (6.6) is non-singular with dimension $N - 1$. The dimension reduction to $N - 1$ for N samples is information loss-less. Discriminant information from both inside and outside the null space of S_W can be extracted now.

We also adopted a regularization technique proposed by Friedman [22] to alleviate over-learning due to the small sample size problem. An identity matrix multiplied with a small value γ, called the regularization term, is added to S_{T_CBLD}. And the feature vector w can be found by solving the following eigenvalue problem:

$$(S_{B\_CBLD} + S_{WB} + S_{WW} + \gamma I)^{-1} S_{B\_CBLD} w = \lambda w. \tag{6.9}$$

The value of γ is usually empirically determined. One can also use a validation set and select the value that gives the best result on the validation set.

6.2 Recursive CBLD (RCBLD)

To remove the feature number constraint, RMLD is integrated with CBLD, and the resulting algorithm is coined recursive cluster-based Bayesian linear discriminant, or RCBLD. As more than $C' - 1$ features can be extracted by RCBLD, where C' is the number of clusters, the modified RCBLD is now truly able to extract discriminant information from both F_W and $\overline{F}_W$.

The steps of RCBLD are described as follows:

1. The first iteration is exactly the same as CBLD and extracts $C'-1$ features and C' is the total number of clusters from all classes. Let W_1 be the set of $C' - 1$ features.

2. For the kth iteration, extract the null space of W_{k-1}, denoted as $\overline{W}_{k-1}$.

3. Discard information from extracted features by projecting S_{B_CBLD}, S_{WB}, and S_{WW} into the null space $\overline{W}_{k-1}$.

$$S'_{B\_CBLD} = \overline{W}^T_{k-1} S_{B\_CBLD} \overline{W}_{k-1} \tag{6.10}$$
$$S'_{WB} = \overline{W}^T_{k-1} S_{WB} \overline{W}_{k-1} \tag{6.11}$$
$$S'_{WW} = \overline{W}^T_{k-1} S_{WW} \overline{W}_{k-1} \tag{6.12}$$

   where S'_{B_CBLD}, S'_{WB}, and S'_{WW} represent the new version of S_{B_CBLD}, S_{WB}, and S_{WW}.

4. Extract a new set of $C'-1$ number of features using CBLD, denoted as w_k.

5. Concatenate the new extracted features into the previous features by $W_k = [W_{k-1}, \overline{W}_{k-1} \cdot w_k]$

6. Go through Steps 2-5 for one more iteration to extract another set of $C' - 1$ features. The recursive procedure terminates when desired number of features have been extracted.

The determination of the desired number is usually done in the training stage by selecting the one leading to the minimal classification error on a validation set.

6.3 Summary

The main characteristics of RCBLD can be briefly summarized as follows:

- RCBLD maximizes a Bayesian criterion function (6.6), which aims at approaching the minimal classification error, the Bayes error.
- RCBLD works with complex class distributions, which are modeled as a union of Gaussian distributions, or multi-modal Gaussian distribution.
- The estimation of S_{WW}, S_{WB}, and S_{B_CBLD} requires clustering analysis. Fuzzy clustering analysis is preferred over crisp clustering for the estimation of S_{WW}. The number of clusters can be selected by supervised training of SOM.
- A recursive approach is used to extract as many features as desired. In each iteration, a set of $C' - 1$ features are extracted, where C' is the total number of clusters from all classes. More features can be extracted by going through more than one iteration.

Although RCBLD relaxes the requirement of uni-modal class distribution for RBLD, it still suffers from limitation inherited from RBLD due to its strong assumptions: (1) equal a priori probabilities of clusters; (2) equal covariances of clusters. These two assumptions are very strong and are violated in almost any real-world applications. But we believe that RCBLD can still lead to good results for situations that do not deviate a lot from the two assumptions. And for the same reason as explained for RBLD in Section 5.2.2, RCBLD may still be able to achieve good performance even when the assumptions are severely violated.

In the following sections of this thesis, we are going to assess the performance of the proposed algorithms. To evaluate the applicability of the algorithms, we have selected various pattern recognition problems. In Chapter 7 a range of different pattern recognition problems from the UCI Machine Learning Repository [56] are selected; in Chapter 8, different face recognition tasks including identity recognition and facial expression recognition are experimented; and in Chapter 9 an application for brain-computer-interface (BCI) problem is tested.

Part II

Applications

Chapter 7

Experiments on UCI Machine Learning Repository

To evaluate the performance of the proposed algorithms, we first selected databases from the UCI Machine Learning Repository, which contains databases for various pattern recognition problems and has been widely used by the machine learning and pattern recognition community. The databases are intentionally selected with sizes ranging from about 100 samples to more than 5,000 samples. Before presenting the experimental work on the UCI databases, we will first give a brief description of the selected UCI databases.

7.1 UCI Databases

The UCI Machine Learning Repository [56] is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.

There are 187 data sets in the UCI repository. Among these data sets, we have intentionally selected 7 multi-class databases with various sizes ranging from small to large to test different algorithms' performance on databases with varying sizes. The 7 databases chosen are *wine*, *vehicle*, *glass*, *optdigits*, *segmentation*, *zoo* and *iris*. The *iris* database is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper [20] is a classic in the field and is referenced frequently to this day. (See [14], for example.) The number of classes varies from 3 to 10. The sizes of each data set are listed in Table 7.1. In the table, the number of attributes (or feature dimensions), and the number of classes are also listed. [56] has more detailed information on each data set and the machine learning repository.

7.2 Experimental Setup

Except for data sets *segmentation* and *optdigits*, which contain separated training and test set, the other data sets have only one single set. The division of one data set into training and test set have been somewhat arbitrary by different researchers. In our experiments, the "leave-one-out" strategy [3, 14] is employed for *wine, zoo* and *iris* databases; "stratified 5-fold cross validation" for glass, and "stratified 9-fold cross validation" for *vehicle*. In "leave-one-out", each time one sample is taken out as the test sample and the rest used to train the system. Every sample is used as the test sample once and the classification error rate is the ratio of the misclassified samples over the total number of samples. In "k-fold cross validation", the whole data set is divided into k subsets of equal size. Each subset is chosen once as the test set to test the system while the rest used to train the system. The classification error rate is the average of the error rates over the k subsets. Stratification ensures that each class is represented with approximately equal proportions in training and test sets. For *segmentation* and *optdigits*, as there are two separated training and test set, we just used the training set for training and test set for performance evaluation.

7.2.1 Classifier

Because the objective of the experiments is to evaluate the ability of our algorithms to extract discriminatory features, a simple classifier is selected such that the classification performance is determined by the feature extraction algorithm as much as possible. If the selected classifier is very powerful, good performance may still be achieved even when the feature extraction algorithm does not do well. Due to this consideration, we used the nearest-neighbor classifier with Euclidean distance as the similarity metric in our experiments. Our proposed algorithms can be readily combined with other more advanced classifiers such as neural networks or SVM to achieve better classification performance.

7.3 Experimental Results

The classification error rates are tabulated in Table 7.1. Note that the results on FLD were not listed in the table, this is because the first iteration of RFLD is actually FLD, and hence the results of RFLD are always superior or at least equal to that of FLD. It is thus not necessary to list the results of both FLD and RFLD in the table.

Table 7.1: Classification error rates on 7 UCI data sets(%). The last three columns are some characteristics of the data sets: N_C is the number of Classes, N_F the number of features, and N the number of samples.

Databases	RFLD	RBLD	RCBLD	N_C	N_F	N
wine	1.1	1.1	0	3	13	178
zoo	3.0	1.0	0	7	16	101
iris	3.3	3.3	0	3	4	150
vehicle	21.9	21.9	19.6	4	8	946
glass	40.2	40.2	33.2	6	9	214
optdigits	7.0	6.6	2.2	10	64	5620
segmentation	11.9	11.0	7.0	7	19	2310

7.3.1 Discussion of Results

Comparing the results of the three different algorithms, we can see that RBLD generally outperforms RFLD. This fact verifies the effectiveness of the Bayesian criterion. It can also be observed that RCBLD achieves the best results for all the seven databases. This means that the combination of the cluster-based approach and the Bayesian criterion does improve the classification performance further. The results on the UCI databases verify that RCBLD has superior performance over RFLD as well as RBLD on databases with a range of sizes varying from small database with about 100 samples to large databases with over 1000 samples.

In the following, the performance of RCBLD is also compared to other state-of-the-art algorithms that have reported results on one or more than one of the 7 data sets in recent years.

7.3.1.1 Discussion of Results on Wine Database

Each pattern in the *wine* database describes 13 chemical constituents found in each of the three types of wines.

The *wine* data was used in [1] for comparing various classifiers: Only RDA (regularized linear discriminant analysis) has achieved 0% classification error rate; QDA (quadratic discriminant analysis) achieves 0.6%, FLD 1.1%, and 1NN (1-nearest-neighbor) 3.9%. In comparison, RCBLD also achieves 0% error rate. This shows that only RCBLD and RDA succeed in classifying the linearly separable problem.

In [34], 10 runs of 10-fold cross validation is performed with random partitions to evaluate kNN classifiers for which the training data is edited by neural network ensembles. The error rate on *wine* database is 3.95%.

7.3.1.2 Discussion of Results on Zoo Database

Zoo is a simple database containing 17 Boolean-valued attributes.

Frank et al. [21] used ensembles of nested dichotomies for multi-class problems. They showed that ensembles of nested dichotomies produce more accurate classifiers than pairwise classification if both techniques are used with C4.5 as base learners, and comparable results for logistic regression. The classification performance is estimated based on 50 runs of the stratified hold-out method, in each run using 66% of the data for training and the rest for testing. They achieved 6.69% by their END (ensembles of nested dichotomies) with C5.4 as base learner and 4.75% with logistic regression as base learner.

Jiang et al. [34] also selected *zoo* for their experiments with experimental setups as described in discussion on *wine* database. They achieved 5.52% error rate, which is also higher than mine although the experimental setup is different.

7.3.1.3 Discussion of Results on Iris Database

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

In [15] Genetic Programming is used to evolve decision trees for data classification. The lowest error rate achieved in [15] is 2.1%, which is higher than 0% of mine. However, in their experiments they used "10-fold cross validation" instead of "leave-one-out".

Frank [21] achieved 6.04% with C5.4 and 4.27% with logistic regression as base learner. As mentioned in the discussion on zoo database, their experimental setup is different from mine.

Jiang et al. [34] also selected *iris* for their experiments with experimental setups as described in discussion on *wine* database. They achieved 4.53% error rate, which is also higher than ours although the experimental setup is different.

7.3.1.4 Discussion of Results on Vehicle Database

Each pattern in the database is a set of features extracted from a given silhouette used to classify a given silhouette as one of four types of vehicle: Opel, Saab, Bus, and Van.

[74] incorporates the inter-class relationships as relevance weights into the estimation of the overall within-class scatter matrix in order to improve the performance of the basic FLD method and some of its improved variants. [74] used "10-fold cross validation" instead of "9-fold cross validation". Only a subset of 846 samples out of the total 946 samples are used in their experiments. The lowest classification error rates achieved are 21.75% for FLD, 21.64% for both WLDR (relevance-weighted linear dimension reduction algorithm) and EWLDR (evolution-based WLDR). Compared to our results, it can be seen that result of

Tang's FLD (21.75%) is slightly better than mine 21.9%. Despite that the result of our version of FLD is slightly worse than theirs, RCBLD can improve the performance to 19.6%, which outperforms that of WLDR and EWLDR (21.64%).

Frank [21] achieved 26.52% with C5.4 and 19.97% with logistic regression as base learner. As mentioned in the discussion on zoo database, their experimental setup is different from mine.

7.3.1.5 Discussion of Results on Glass Database

Motivated by criminological investigation, the type of glass is to be classified for the database.

Frank [21] achieved 29.33% with C5.4 and 35.81% with logistic regression as base learner. As mentioned in the discussion on *zoo* database, their experimental setup is different from mine.

Jiang et al. [34] also selected *glass* for their experiments with experimental setups as described in discussion on *wine* database. They achieved 32.06% error rate.

7.3.1.6 Discussion of Results on Optdigits Database

The patterns in *optdigits* database are obtained from a total of 43 people, 30 contributed to the training set and different 13 to the test set.

Frank [21] achieved 2.76% with C5.4 and 3.0% with logistic regression as base learner. As mentioned in the discussion on *zoo* database, their experimental setup is different from mine.

[74] also did experiments on *optdigits* data set. Classification error rates of 6.12% for FLD, 6.07% for WLDR, and 5.9% for EWLDR, are achieved in [74]. Compared to our results, it can be observed that the performance of Tang's implementation of FLD (error rate = 6.12%) achieves better result than our implementation (RFLD's error rate = 7.0%). Despite the better implementation of FLD by Tang, his best result for EWLDR 5.9% is worse than our 2.2% by RCBLD.

7.3.1.7 Discussion of Results on Image Segmentation Database

The *image segmentation* database consists of 2310 patterns, each corresponding to a 3×3 region drawn randomly from a database of 7 outdoor images. It has 19 continuous attributes. The problem is classify the pattern into one of the seven classes: brickface, cement, foliage, grass, path, sky, and window. There are 210 patterns in the training set and 2100 patterns in the test set (each class has 300 test patterns). In [84], Kwok extended the use of moderated outputs to SVM by making use of a relationship between SVM and the evidence framework. In his experiments, the error rate of nearest-neighbor classifier is 12.3%; the error rate by maximum a posteriori (MAP) decision rule is 9.8%; and the error rate by moderated SVM is 8.6%. In our experiments, the same experimental setup is used and a better error rate of 7.0% is obtained.

Chapter 8

Applications to Face Recognition

To assess the performance of the proposed algorithms for more challenging pattern recognition tasks, we applied our algorithms to face recognition problems. We selected face recognition to test our algorithms because face recognition has become one of the hottest research topics in pattern recognition community and its difficulty is well acknowledged. In the following, we will first give an overview of face recognition. And next, the databases chosen for the experiments are described. Finally the experimental setup and results are given.

8.1 Overview of Face Recognition

Face perception is an important part of the capability of human perception system and is a routine task for humans, while building a similar machine system is still an on-going research area. The research on face recognition has an interdisciplinary nature, tied to many research fields, such as pattern recognition, image processing, computer vision, computer graphics, statistical computing, and machine learning. In addition, automatic face recognition designs are often guided by the psychophysical and neural studies.

The earliest work on face recognition can be traced back at least to the 1950s in psychology [7] and to the 1960s in the engineering literature [5]. During the early and mid-1970s, geometrical feature based approaches, which use measured attributes of features (e.g., the distances between important points) in faces or face profiles, were used [36, 37]. During the 1980s, work on face recognition remained largely dormant. Since the early 1990s, research interest in face recognition has grown significantly [3, 8, 16, 31, 42, 47, 81, 83]. One main reason that accounts for the increased interest in face recognition is the wide range of commercial and law enforcement applications. For example, at present, one needs to create and remember a password to get cash from an ATM, to log into a computer, to access the internet, and so on. Although very reliable methods of biometric personal identification exists, for example, fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of face images is often effective without the participant's cooperation or knowledge. Some of the advantages/disadvantages of different biometrics are described in Philips et al [60].

8.1.1 Face Recognition Problems

Depending on the nature of the applications, there are various types of face recognition problems, such as identity recognition, facial expression recognition, gender recognition, race recognition, and glass-wearing recognition, etc. we have applied the proposed feature extraction algorithms on three types of face recognition problems: identity recognition, facial expression recognition, and glass-wearing recognition. The problem of identity recognition can be stated as: given an input image, either in the form of a static image like a photo, or image sequences from a video, the task is to identify the person in the image. On the other hand, the task of facial expression recognition is to identify the type of facial expressions that the person in the image possesses. The task of glass-wearing recognition is a two-class problem: whether the subject is wearing glasses or not. If not specified, the term "Face Recognition" usually refers to "identity recognition", as it is the most commonly encountered task. Here, the term "identity recognition" is used to differentiate it from other face recognition problems.

Generally speaking, automatic face recognition is a difficult task, which is afflicted by the usual difficulties faced in pattern recognition and computer vision tasks, coupled with face specific problems. Although a fully automatic face recognition system typically involves tasks including face detection, segmentation, normalization, feature extraction, and recognition, our work mainly focuses on extracting discriminant features for the problem of recognizing identities and facial expressions of faces in still images.

8.1.2 Holistic (Global) Matching and Component (Local) Matching

A wide range of techniques from image processing, computer vision, and pattern recognition, have been applied on face recognition applications. One can generally put a face recognition system into one of the two categories: holistic matching methods and component matching methods. The two categories are sometimes referred to as global and local matching methods. In holistic/global matching the whole face region is used as a single entity for analysis. On the contrary, component/local matching methods first locate several facial features (components), and then classify the faces by comparing and combining the corresponding local statistics. Careful comparative studies of different options in a holistic recognition system have been reported in [64]. A similar comparative study for local matching approach is given in [88]. Heisele et al. compared component (local) and global (holistic) approaches in [29].

Although several psychophysical experiments suggest that human face recognition is a holistic process, some researchers e.g. Zou [88] believe that at the current state of the art, local region matching is more appropriate for machine face recognition. The main advantage of local matching approach is its robustness to pose variation and partial occlusion. However, the improved performance of local matching approach requires reliable detection and selection of local facial features, which are challenging issues by itself. For example, Feng et al. obtained an error rate of 23% [17] for facial expression recognition with LBP features from a manually selected set of fiducial points and a coarse-to-fine classification scheme. The error rate rises to 30.1% when the feature points are automatically located by a modified Active Appearance Model (AAM) [18]. The reliability of the extraction of local features has a significant influence on the performance of the local matching methods. Unfortunately, treatment of local facial feature detection is still rudimentary.

8.1.3 Feature Extraction for Face Recognition

Whether it's global or local matching approach, discriminant feature analysis is usually employed to extract discriminant features for the succeeding classifier. In global matching, a single set of discriminant features is extracted from the whole face region, whereas in local matching, an individual set of local features is usually extracted from each individual component (or local patch). There are two schemes to combine the extracted local features to reach a final decision: (1) put all the local features into a single feature vector and then classify it by a single classifier; (2) classify each set of local features by a base classifier and then combine all the decisions from all base classifier to determine the final class label of the input pattern. One can select either different algorithms or a single general feature extraction algorithm for the discriminant analysis of global and local matching approaches.

To test the effective of the proposed algorithms, we employed the global matching scheme and applied our algorithms to identity and facial expression recognition problems. One reason for me to select the global matching scheme is that most feature extraction algorithms have been applied with the global matching scheme. The most well known example could be eigenfaces [38, 76]and fisherfaces [3, 16], which have been proved to be effective in experiments with large databases. Many local matching approaches are extensions of their corresponding global approaches. For instance, Pentland extended the eigenface to eigenmodules, such as eigeneyes, eigennoses, and eigenmouths [58]. Another reason for selecting global approach is that the implementation of global matching is relatively simple and straight forward. It is obvious that one should select some well known benchmark algorithms for comparative analysis. And should the selected benchmark algorithms be simple to implement, the comparison between different algorithms could be as fair as possible. This is because implementation details can affect the results of a face recognition system significantly. Experimental results of the same algorithm on the same database could vary significantly due to different implementations. For example, different implementations of a PCA-based face recognition algorithm were compared in [54]. This effect of implementation details will also be experimentally demonstrated later in section 8.4.

8.2 Databases for Face Recognition

Four publicly available databases are used in our experiments to evaluate the performance of different feature extraction algorithms: Yale [3], Yale B [25], ORL (Olivetti Research Laboratory), and JAFFE (Japanese Female Facial Expression) databases [49]. Yale, Yale B, and ORL databases were used for identity recognition. Yale and JAFFE were selected for facial expression recognition because these two databases pose the problem of recognizing expressions against variations of different face appearance, illumination conditions, and face accessories etc. with limited training sample size. For glass-wearing recognition, Yale and ORL databases were used. So for each type of face recognition problem, there are at least two different databases used to test the performance of various algorithms.

8.2.1 Yale Face Database and Its Pre-processing

There are 165 images in Yale database, which is made up of 15 different persons (14 males and 1 female) with 11 images for each person. The 11 images of each person are labeled by facial expressions, lighting conditions or whether wearing glasses or not: “normal”, “happy”, “sad”, “sleepy”, “surprise”, “wink”, “left light”, “central light”, “right light”, “without glasses”, and “with glasses”. There are 6 facial expressions for Yale database: “normal”, “happy”, “sad”, “sleepy”, “surprise”, and “wink”. For those images not labeled by expression, their expressions are usually “normal”. Images from Yale database are cropped manually to eliminate most of the background and some part of hair and chin. The size of images changes from 320 × 243 to 124 × 147. Figure 8.1 shows images of one person from Yale face database.

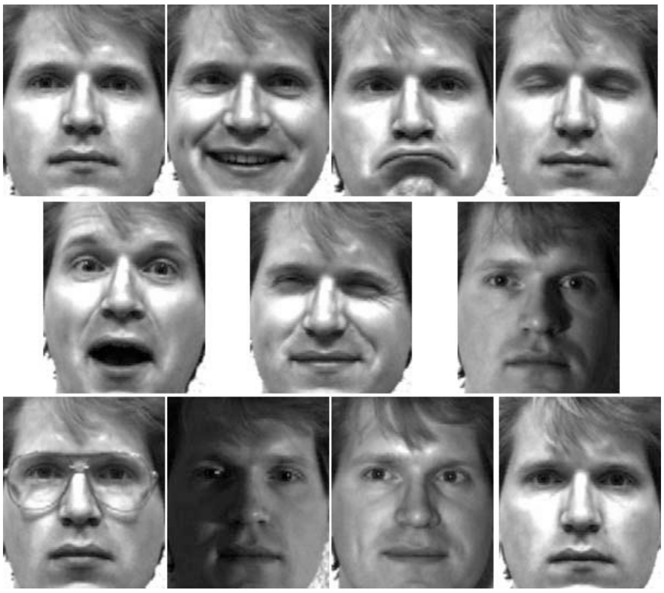


Figure 8.1: Sample images of one person from Yale face database. -

8.2.2 Yale B Face Database and Its Pre-processing

The Yale face database B consists of 5760 single light source images of 10 subjects with varying pose and illuminations and is built to test the performance of identity recognition algorithms against illumination and pose variations. Images of each individual were acquired under 576 viewing conditions: 64 different lighting conditions in 9 poses (a frontal pose, five poses at 12°, and three poses at 24° from the camera axis). Of the 64 images per person in each pose, 45 were used in our experiments. In other words, our experiments used 4050 images from the database. The images from each pose were divided into 4 subsets (12°,25°,50°, and 77°) according to the angle that the light source makes with the camera's axis. Subset 1 (respectively, 2,3,4) contains 70 (respectively, 120,120,140) images per pose. Figure 8.2 shows sample images with frontal illumination and frontal pose of the 10 persons from the Yale B database. Figure 8.3 shows the 9 poses of a person under frontal illumination. And Figure 8.4 shows 4 sample images of a person under different illuminations with frontal pose. During the use of this database, we have also found some 'bad' quality images, an example of which are shown in Figure 8.5.

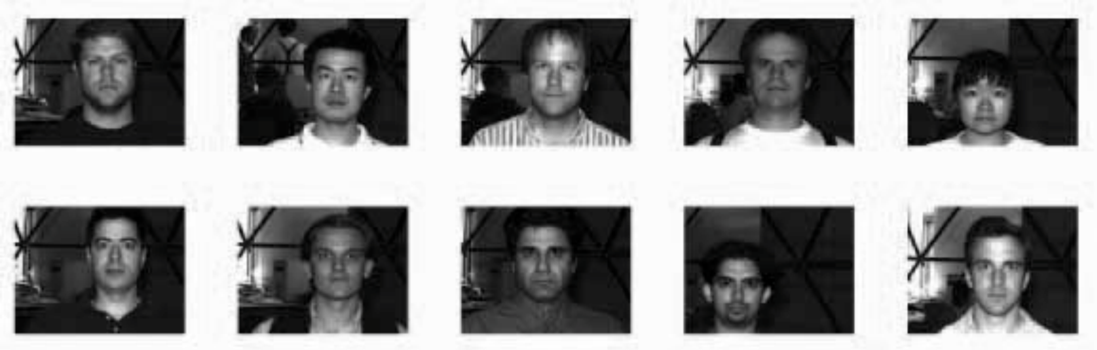


Figure 8.2: Sample images with frontal illumination and frontal pose of the 10 persons from the Yale face database B. -

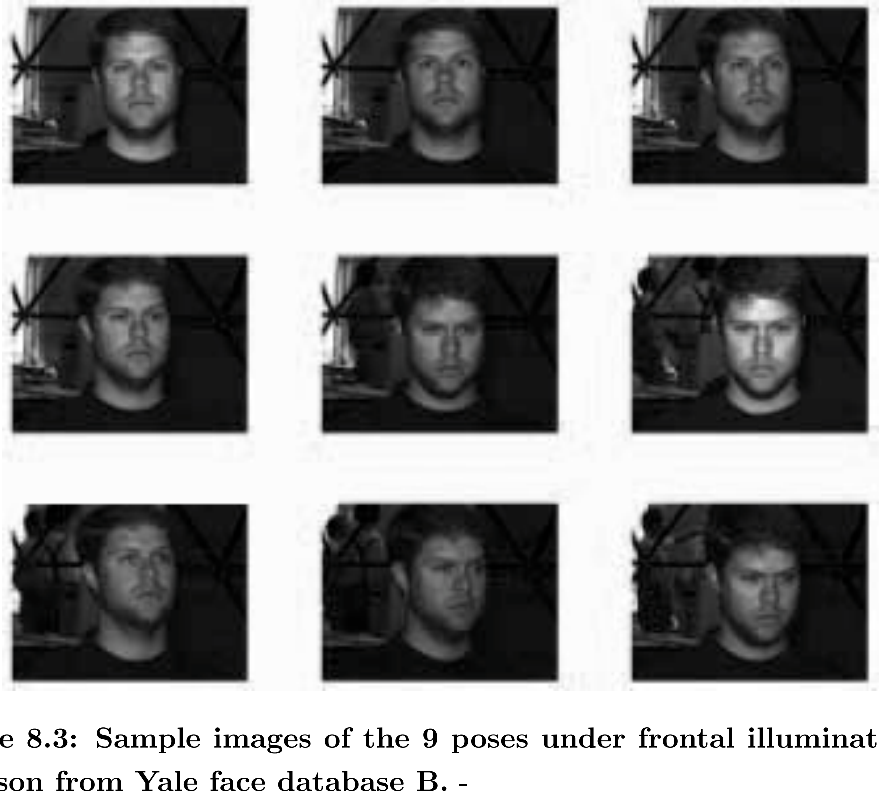


Figure 8.3: Sample images of the 9 poses under frontal illumination of a person from Yale face database B. -

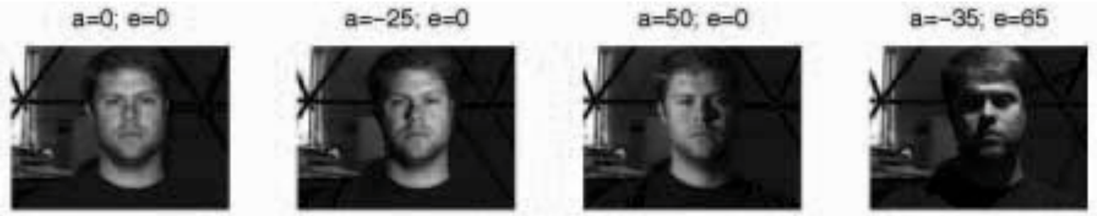


Figure 8.4: Sample images under 4 illuminations of frontal pose of a person from Yale face database B. - a and e above each figure represents the azimuth and elevation angle of the light source under which the respective photo was taken.

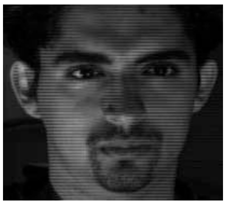


Figure 8.5: A sample image of 'bad' quality from the Yale face database B. - This image is corrupted by gray strips.

The original size of the images is 640 × 480. In the experiments, all images were manually cropped to 270 × 280 to include only the face region with as little hair and background as possible. Each frontal pose image was aligned by an affine transformation so that the eyes lie at a fixed distance apart (equal to four sevenths of the cropped window width) and on an imaginary horizontal line. Furthermore, the face was centered along the vertical direction so that two imaginary horizontal lines passing through the eyes and mouth were equidistant from the center of the cropped window. This alignment was performed in order to remove any bias from the recognition results due to the association of a particular scale, position, or orientation to a particular face. Only the frontal pose images were aligned because the eyes' and mouth coordinates information are given for the frontal images only. The images in the other eight poses were only aligned by the two eyes' coordinates, since only the eyes' coordinates are available. After cropping, the images are down-sampled by 2 and have a resolution of 135 × 140.

Besides spatial normalization, i.e., cropping and alignment, gray level normalization is also performed using histogram equalization. Each face image is first divided equally into two: the left half and the right half. Histogram equalization is then performed separately on every half face image. This separate histogram equalization of left and right half image is used due to the symmetrical nature of a face image under frontal illumination condition. The left and right half of a face image would have the same gray level histogram under frontal illumination condition, but this is not, in general, the case for lighting conditions that are not frontal. Obviously, histogram equalization on the whole face image could not correct this problem. Figure 8.6, 8.7, and 8.8 show the sample images corresponding to Figure 8.2, 8.3, and 8.4 after this histogram equalization scheme.

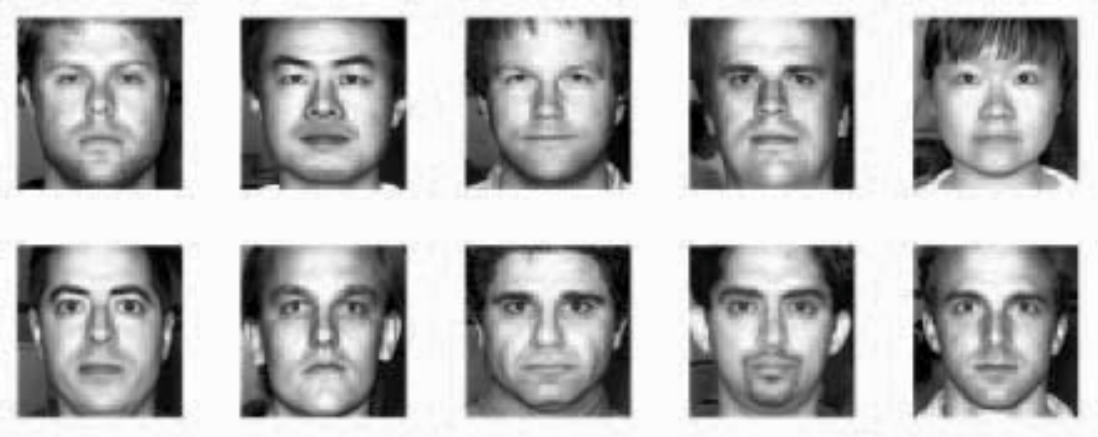


Figure 8.6: Histogram equalized sample images with frontal illumination and frontal pose of the 10 persons from the Yale face database B. -

8.2.3 ORL Face Database and Its Pre-processing

ORL database consists of 40 different individuals with 10 images for each individual. The images from ORL database were also cropped from 112×92 to 81×72. Some sample images from the ORL database are shown in Figure 8.9

8.2.4 JAFFE Face Database and Its Pre-processing

JAFFE database comprises images of 10 Japanese females. Each person has 7 facial expressions: “happy”, “sad”, “surprise”, “angry”, “disgust”, “fearful”, and “neutral”. There are 3 or 4 images for each facial expression of each person.

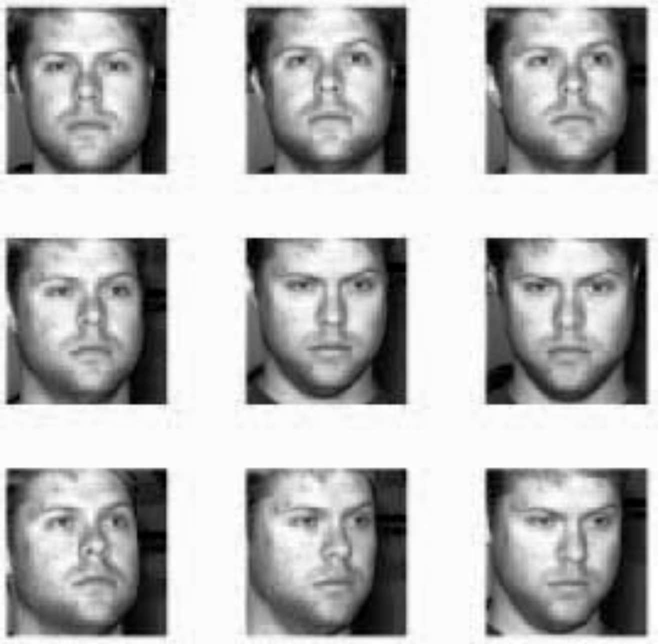


Figure 8.7: Histogram equalized sample images of the 9 poses under frontal illumination of a person from Yale face database B. -

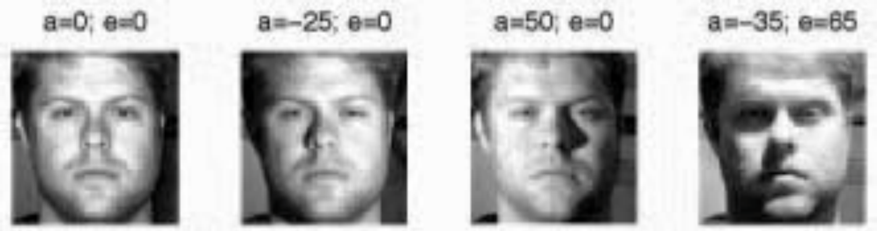


Figure 8.8: Histogram equalized sample images under 4 illuminations of frontal pose of a person from Yale face database B. -

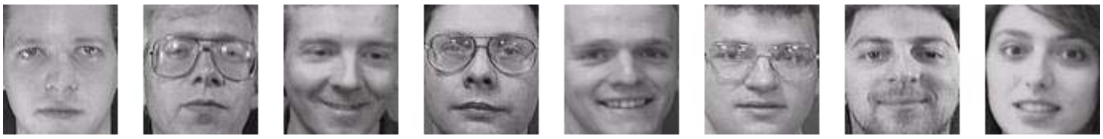


Figure 8.9: Some sample images from ORL face database. -

The resolution of each image is 256 × 256. Some sample images are shown in Figure 8.10. The JAFFE images are not cropped to remove background and hair region, i.e., original images are used. We did this on purpose in order to check the robustness of the algorithms in the adversity of background disturbance and imperfect face alignment.

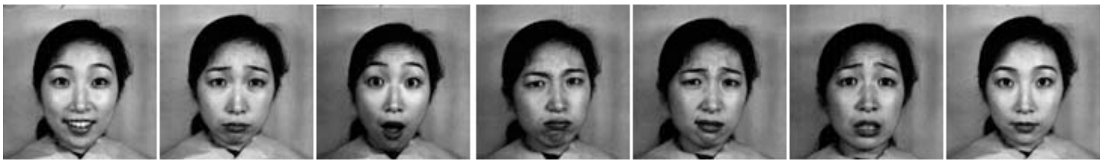


Figure 8.10: Sample images of one subject from JAFFE face database. - The seven expressions from left to right are "happy", "sad", "surprise", "angry", "disgust", "fearful", and "neutral".

8.3 Experimental Setup for Training and Testing

The identity recognition error rate is determined by "leaving-one-out" strategy [3, 14]: to classify one particular image, all the rest of the images are pooled together to form the training data set. Each image is used as the test image once and the error rate is computed as the ratio of misclassified images over the total number of images in the database.

To evaluate the performance of the different feature extraction algorithms for facial expression recognition, a person independent cross validation strategy was adopted. The images from the database is partitioned into groups by identities so that each group consists of images from one person. The evaluation is carried out by taking one identity out as the test set, and all other identities as the training set each time. This process is repeated over all the identities so that each group is used as the test set for one time. The recognition error rate is then averaged over all groups.

This kind of "leave-one-person-out" cross validation was adopted in the experiments so that the recognition of an expression is face appearance independent.

In other words, the facial expression recognition system does not have any image of the person from the test set, and therefore the classification of the expression of the test image is not affected by the appearance of the face. This kind of scheme for training and testing tries to evaluate objectively test the ability of a classification system to recognize an expression.

There are 6 facial expressions for Yale database: “normal”, “happy”, “sad”, “sleepy”, “surprise”, and “wink”. As mentioned before, there are 11 images for each person in Yale database. They are labeled by facial expressions, lighting conditions or whether wearing glasses or not: “normal”, “happy”, “sad”, “sleepy”, “surprise”, “wink”, “left light”, “ center light”, “right light”, “without glasses” and “with glasses”. For those images not labeled by expression, their expressions are usually “normal”. Thus all images are used for facial expression recognition for Yale database, instead of just a subset of the database.

For JAFFE database, there are 7 expressions: “happy”, “sad”, “surprise”, “angry”, “disgust”, “fearful”, and “wink”. All the 7 expressions are included in the experiments.

Like facial expression recognition, the “leave-one-person-out” cross validation is adopted in the experiments for glasses-wearing recognition.

8.3.1 Classifiers

Because the objective here is to evaluate the ability of our algorithms to extract discriminatory features in comparison with other peer feature extraction algorithms, we selected a simple classifier such that the classification performance is determined by the feature extraction algorithm as much as possible. If the selected classifier is very powerful, good performance may still be achieved even when the feature extraction algorithm does not do well. Due to this consideration, we used the nearest-neighbor classifier with Euclidean distance as the similarity metric in our experiments. Our proposed algorithms can be readily combined with other more advanced classifiers such as neural networks or SVM to achieve better classification performance.

8.4 Experimental Results

8.4.1 Experimental Results on RMLD

To make comparative analysis, RMLD and six other feature extraction algorithms have been implemented and compared:

- The first method uses PCA to reduce the high-dimensional images into lower-dimensional ones, but no discriminant analysis is performed afterwards.

- The second method is FLD. To solve the small sample size problem, PCA is used first to reduce the sample dimension so that the within-class scatter matrix S_W is non-singular.

- The third method, Enhanced FLD Model (EFM) [47], is the same as FLD except that EFM selects a different sub-eigenspace, which is more optimal for subsequent FLD process. EFM aims to seek a proper number of PCA features that balance between the need to keep enough spectral energy of raw data and the requirement that the eigenvalues of within-class scatter in the reduced PCA space are not too small, for the tiny eigenvalues are associated with noise that make FLD over-fitting while exposed to new data. Unfortunately, no quantitative criterion for measuring the adequacy of energy and the smallness of eigenvalues of within-class scatter is currently available and hence the cut-off point for the number of PCA components to retain has to be obtained through trial and error. In our experiments, the optimal number of PCA features is the one leads to the lowest error rate, and is found through simple exhaustive search rather than analyzing the spectrum of the eigenvalues as suggested in [47].

- The fourth method is RFLD. Like FLD, RFLD also employs PCA to reduce the sample dimension so that S_W is non-singular.

- The fifth method is RMLD which uses the full eigenspace extracted by PCA as discussed before.

- The sixth feature extraction method compared is Nonparametric Discriminant Analysis (NDA) which is also free of the feature number limitation and supposed to deal with multi-modal class distributions. As described in Chapter 2, NDA is very similar to FLD except that it adopts a nonparametric definition for S_B. The implementation of NDA, as suggested in [45], is more straightforward, and hence adopted in the experiments for the comparative studies.
- The last method is Locality Preserving Projection (LPP), which is also described in Chapter 2. LPP is not an extension of FLD. Instead, it is an unsupervised learning algorithm that aims to find a linear subspace that best preserves local structure and detects the essential face manifold structure.

The lowest recognition error rates achieved by these methods are shown in Table 8.1, 8.2, and 8.3 for identity recognition, facial expression recognition, and glass-wearing recognition respectively. We can observe several interesting points by comparing the experimental results of these different methods for the three face recognition problems:

- The performance of PCA and LPP is generally much worse than other methods. This is not surprising since PCA and LPP are unsupervised learning algorithms which do not utilize class information to extract discriminant features.
- NDA achieves comparable performance as compared to FLD.
- RFLD improves the recognition performance of FLD by going through more than one iteration to extract more features.
- The performance of RMLD is generally better than that of RFLD because RMLD can extract discriminatory features from the null space $\bar{F}_W$.
- EFM generally achieves good results. However, it requires exhaustive search of the optimal cut-off point for the number of PCA components. RMLD can obtain comparable results without the exhaustive search for optimal PCA components.

Table 8.1: Comparative experiments for RMLD: identity recognition results

Methods	Yale Database		ORL Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
PCA	17.6	20	8.5	59
FLD	0.6	14	4.3	39
EFM	0	14	1.5	39
RFLD	0.6	14	2.0	41
RMLD	0	31	1.5	69
NDA	0.6	14	4.3	39
LPP	15.2	149	8.8	325

Table 8.2: Comparative experiments for RMLD: facial expression recognition results

Methods	Yale Database		JAFFE Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
PCA	50.9	38	66.7	62
FLD	35.8	5	54.3	6
EFM	30.3	5	51.4	6
RFLD	32.7	32	50.5	9
RMLD	32.1	29	49.5	10
NDA	32.7	71	54.3	9
LPP	43.0	143	55.7	97

Table 8.3: Comparative experiments for RMLD: glasses-wearing recognition results

Methods	Yale Database		ORL Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
PCA	29.7	16	39	57
FLD	17.0	1	16.3	1
EFM	13.3	1	15.3	1
RFLD	13.3	2	16.3	1
RMLD	14.0	3	16.3	1
NDA	16.4	16	16.3	1
LPP	23.6	96	27.8	35

Table 8.4 compares classification performance of RMLD on original and normalized data for identity and facial expression recognition problems. Each face image is represented as a matrix of intensity values and this matrix can be concatenated into a feature vector. The normalized is done to make the vector have unit magnitude. This normalization reduces variations caused by different illumination. The results in Table 8.4 show that implementation details could affect the algorithm's performance.

Table 8.4: Results of RMLD on original and normalized data. The number in the bracket indicates the number of features corresponding to the respective error rate.

Method	Recognition Task	Identity Recognition		Expression Recognition	
	Databases	Yale	ORL	Yale	Jaffe
RMLD	Original	0 (31)	1.5 (69)	32.1 (29)	49.5 (10)
	Normalized	0 (21)	1.5 (49)	31.5 (27)	48.1 (8)

Table 8.5: Comparative experiments for RBLD: identity recognition results

Methods	Yale Database		ORL Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
EFM	0	14	1.5	39
RMLD	0	31	1.5	69
RBLD	0	36	1.0	20

Table 8.6: Comparative experiments for RBLD: facial expression recognition results

Methods	Yale Database		JAFFE Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
EFM	30.3	5	51.4	6
RMLD	32.1	29	49.5	10
RBLD	30.9	26	49.0	8

8.4.2 Experimental Results on RBLD

Table 8.5 and 8.6 compare the recognition results of RBLD to RMLD on identity and facial expression recognition problems. The results of EFM are also listed in the two tables. The results show that RBLD improves the performance of RMLD.

Table 8.7 compares classification performance of RBLD on original and normalized data for identity and facial expression recognition problems. The results in Table 8.7 again show that implementation details could affect algorithms' performance.

8.4.3 Experimental Results on RCBLD

8.4.3.1 Identity Recognition on Yale Face Database B

To evaluate the performance of RCBLD for identity recognition on Yale B database, the classification performance of RCBLD in comparison with PCA, RMLD, and

Table 8.7: Results of RBLD on original and normalized data. The number in the bracket indicates the number of features corresponding to the respective error rate.

Methods	Recognition Task	Identity Recognition		Expression Recognition	
	Databases	Yale	ORL	Yale	Jaffe
RBLD	Original	0 (14)	1.3 (38)	30.9 (26)	49.5 (8)
	Normalized	0 (21)	1.5 (21)	30.3 (7)	47.6 (8)

RBLD are listed in Table 8.8. The results in the table show that all the tested feature extraction algorithms can achieve perfect recognition result on subset 2. This suggests that subset 2 is rather easy a classification task, which matches the fact that subset 2 is the most similar set to the training set. Obviously, subset 2 is too easy for the purpose of comparing the strength of different feature extraction algorithms. The difference with respect to the training set increases for Subset 3 and 4. Correct classification of these two sets are then more difficult. From the table, we see that there are some difference between the performance of different algorithms on subset 3, and the difference is significant on subset 4. This result suggests that the performance of PCA is significantly affected by illumination variation between the training and test set. RMLD, RBLD, and RCBLD are more robust to illumination variation. The result of RBLD is better than RMLD, which confirms the effectiveness of the Bayesian criterion function. From the table, we can also see that RCBLD outperforms all other algorithms, which shows that RCBLD further improves the results of RMLD and RBLD by integrating the strength of the Bayesian criterion and the cluster-based approach. We can also observe that the lowest error rates on subset 4 is obtained with 277 features. This shows that recognition performance can be improved by extracting more discriminating features. This is also shown in Figure 8.11 which plots the classification error rates of RCBLD with respect to the number of features extracted. The error rate decreases with more numbers of features and reaches minimum at 277 features.

Besides the classification error rate, the cumulative matching score with the number of features corresponding to the lowest error rate on subset 4 is also plotted in Figure 8.12. From the cumulative matching score, we can see that

RCBLD always achieves better accuracy with different ranks. It can achieve perfect recognition with rank 2.

Table 8.8: Identity recognition results on Yale face database B

Methods	Subset 2		Subset 3		Subset 4	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
PCA	0	27	6.4	475	34.0	369
RFLD	0	5	0.4	9	6.2	10
RBLD	0	7	0.1	14	6.0	17
RCBLD	0	12	0	62	1.4	277

Note that the error rates reported in Table 8.8 are for illumination subsets of all poses. Figure 8.13 and Table 8.9, on the other hand, show the break-down of the results of RCBLD on Subset 4 for different poses.

Table 8.9: Identity recognition results of RCBLD on Yale face database B subset 4

Poses	1	2	3	4	5	6	7	8	9	2-6	7-9	all poses
Error rates	0	0	2.9	2.1	1.8	2.1	1.4	0	3.6	1.6	1.7	1.4

Discussion on Identity Recognition Experiments

The Yale face database B was constructed and first used by Georghiades, et al. in [25]. We compare our results to those reported in [25]. Although the same set of 4050 images out of the 5760 images were used in their experiments, the experimental framework is different. Besides, the pre-processing methods of face images are also not exactly the same. So the comparison cannot be completely fair. In order to make the comparison as fair as possible, we will compare only the results where the experimental setup is similar to ours. In their experiments, they performed two sets of experiments against variation in illumination and pose. In

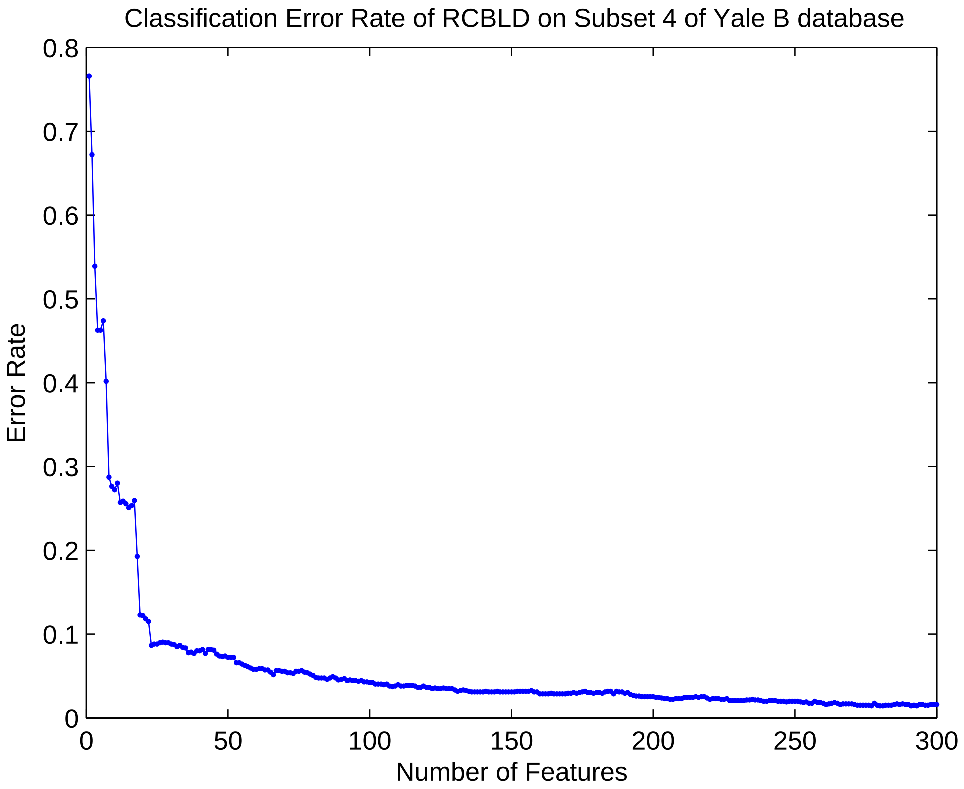


Figure 8.11: Classification error rates of RCBLD on subset 4 of Yale face database B. -

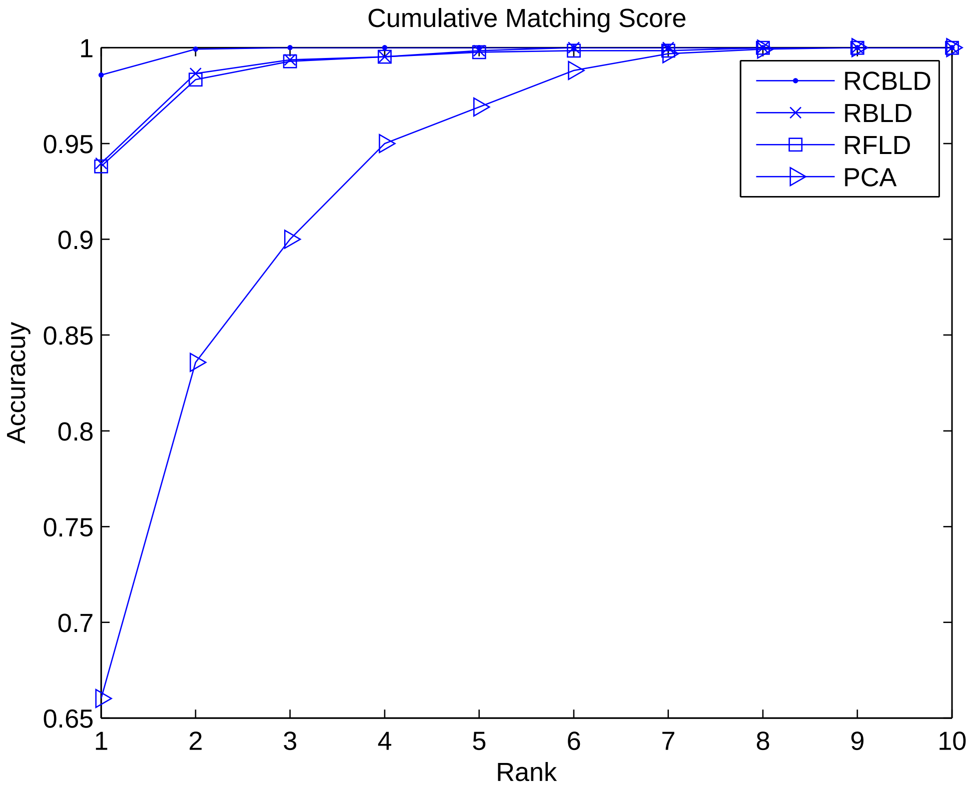


Figure 8.12: Cumulative matching score of RCBLD with the number of features corresponding to the lowest error rate on subset 4. -

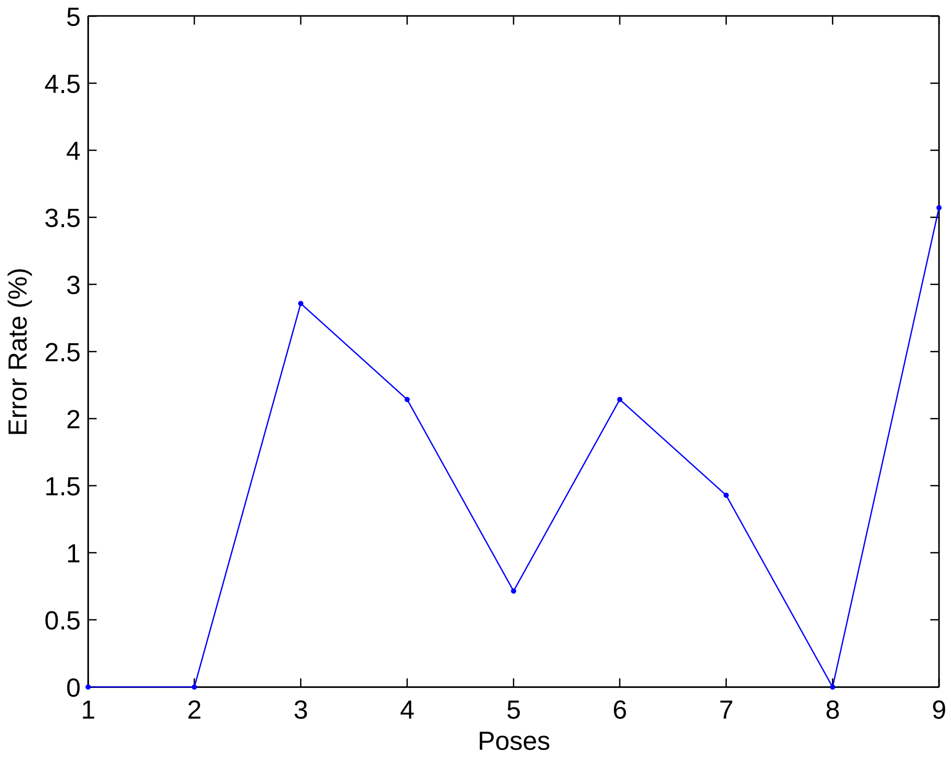


Figure 8.13: Decomposition of classification error rates of RCBLD on subset 4 of Yale face database B. Frontal pose: pose 1; 12°**: poses 2-6;** 24°**: poses 7-9. -**

the first set of experiments, extrapolation in illumination is tested, where only the 450 frontal pose images (10 faces x 45 illuminations) are used for training and testing. The lowest error rates are all 0 for the 3 subsets. In comparison with our method, the lowest error rates are 0 for the first 2 subsets, but 1.4, which is a bit higher than 0, for subset 4. However, our experiments used all the 9 poses, including 4050 images instead of 450 images. So a higher error rate is expected with more poses and images included. If we decompose the results for subset 4 into poses, the error rate for frontal pose on Subset 4 is also 0. In the second set of experiments by Georghiades, all 9 poses are used to test recognition performance under variation in pose and lighting. Their proposed method, called Cones Approximation, achieves 2.9%, 7.4%, and 12.6% on Subset 4 for frontal pose, $12°(poses 2, 3, 4, 5, 6)$, and $24°(poses 7, 8, 9)$, respectively. In comparison, the error rates of RCBLD are 0%, 1.6%, and 1.7%, as shown in Table 8.9, which are substantially better.

8.4.3.2 Facial Expression Recognition

For facial expression recognition, radial encoding [55] is applied on all face images from Yale and JAFFE databases as a pre-processing technique for representing the face image. We selected radial encoding prior to the feature extraction stage to emulate the retina sampling in the human vision system. Another desirable characteristic of radial encoding is that it under-samples face images. The encoded face images usually have a much lower dimension than the original images.

The mechanism of radial encoding is illustrated in Figure 8.14. It converts the traditional discretization of an image in Euclidean coordinate system into a discretization in polar coordinate system. In the experiments, each image is divided into 30×10 regions (30 angular, 10 radial divisions). The average of gray levels of one region is used to represent the gray value of that region.

Table 8.10 lists the lowest recognition error rates of RCBLD and several related feature extraction algorithms. Experiments on both Yale and JAFFE databases show that RCBLD achieves superior recognition results compared to other methods. The lowest recognition error rate of 26.7% and 37.1% for Yale and JAFFE

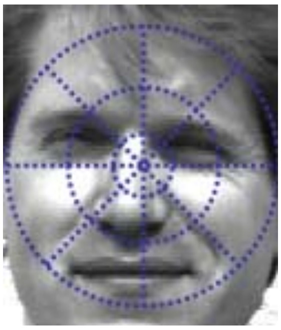


Figure 8.14: Radial encoding of the face image. - The face image is divided by a radial grid and the average of gray levels of each region is used to represent the gray level of that region.

databases by RCBLD is obtained with 55 and 10 number of features, respectively. This indicates again that classification performance can be improved by extracting more features for discrimination by the recursive approach. The experimental results obtained from the two databases on facial expression recognition unanimously confirm the advantage of RCBLD.

Table 8.10: Facial expression recognition results: comparative experiments for RCBLD.

Methods	Yale Database		JAFFE Database	
	Lowest Error Rate (%)	Number of Features	Lowest Error Rate (%)	Number of Features
PCA	50.9	38	66.7	62
RMLD	32.1	29	49.5	10
RCLD	30.4	31	45.2	88
RCBLD	26.7	55	37.1	10

Discussion on Facial Expression Recognition Experiments

We searched the literature for reported experiments that also used the same Yale and JAFFE databases in a similar way to ours and compared them to our results here. Jerez et al. used a four layer neural net that combined a local receptive field with a modified Hebbian rule and a modular network [33]. An error rate of 17.1% is obtained. Although the error rate is considerably lower than our 26.7%, they tested on a subset of Yale database: 14 faces and 4 expressions: neutral, happy, sad and surprise. In our experiments, we used all the 15 faces and 6 expressions.

The JAFFE database has been a popular database for evaluating the performance of facial expression recognition systems. But the way the database is used is different for different researchers. There are mainly three ways. The first way is to divide the whole database randomly into several equal-sized groups, and then use cross validation [19, 70, 86]. The second way is to take one image as the test set and all other images as the training set each time. It repeats over all images and takes the average as the recognition accuracy. This way of using JAFFE database is called "leave-one-image-out" [87]. The third way is to divide the database into groups corresponding to identities and is called "leave-one-person-out". The "leave-one-person-out" strategy is used in our experiments and its detail is described above. We adopted this strategy because it assesses how system generalizes on new faces. To compare the results on JAFFE database using the "leave-one-person-out" strategy, one needs to take note that only a subset of the database was used in some results. For example, Lyons et al. [49, 50] and Shinohara et al. [71] used only 9 faces and reported an error rate of 25% and 30.6%, respectively. The face whose expressions are most difficult to tell is excluded from the experiments. Feng et al. used the same subset of 9 faces, and obtained an error rate of 23% [17] with LBP features from a manually selected set of fiducial points and a coarse-to-fine classification scheme. The error rate rises to 30.1% when the feature points are automatically located by a modified Active Appearance Model (AAM) [18]. This shows that pre-processing of face images by some manual assistance affects the performance of the facial expression recognition system significantly. [87] and [24] used only 6 expressions, excluding the "neutral" expression, and reported error rates of 22.95% and 37.22%, respectively. In [87] manual selection of facial geometric points was required. Horikawa [30] used the full database, i.e., 10 faces and 7 expressions, and reported an error rate of 33.0%. But note that they manually take the center region of 200×200 pixels of the face region and then resized it to 20×20 in the pre-processing stage. A linear normalization is also carried out to make the 20×20 pixel data have a zero mean and unit standard deviation. In contrast, our method was applied directly on the 256×256 full face images with some background included. Besides the feature extraction method, the performance of a facial expression recognition system is also significantly affected by a careful design of the pre-processing technique, the classifier and classification scheme, and the implementation detail. Considering that only simple pre-processing technique and the simplest 1-nearest-neighbor classifier is used in our facial expression recognition system, we think that the performance of our method is comparable to the aforementioned recent results on JAFFE database. The performance of our method can be boosted by carefully designing the pre-processing stage (manual face cropping, alignment, normalization, Gabor wavelet decomposition etc), more advanced classifiers (SVM, neural net), and a more complicated classification scheme that takes into account face-specific properties.

Chapter 9

Application to Brain Computer Interface

Since our proposed algorithms are developed as general-purpose feature extraction algorithms, they are also applied to brain-computer interface problem, which will be described in the following sections.

9.1 Introduction

A brain-computer interface (BCI), sometimes called a direct neural interface or a brain-machine interface, is a direct communication pathway between a human or animal brain (or brain cell culture) and an external device.

Began in the 1970s, BCI research has attracted a surge of interest in recent years due to advances in computer technology and neuroscience [13, 44, 68, 79]. Since 2001 there have been four BCI competitions that aim to validate signal processing and classification methods for BCI systems [4, 66]. Many people who suffer from amyotrophic lateral sclerosis, cerebral palsy, spinal cord injury and other diseases will disrupt the neuromuscular channels where the brain communicates with the external environment. The main focus for BCI research is to fulfill the potential of BCI systems which is to provide assistance to people with these disabilities.

BCI systems can be broadly classified into three types based on the placement of the electrodes used to detect and measure neurons firing in the brain: invasive, partially-invasive, and non-invasive.

9.1.1 Invasive BCIs

Invasive techniques require recording electrodes to be implanted either in the cerebral cortex (microelectrode arrays or neurotropic electrode) or on the cortical surface (electrocorticography or ECoG). As they rest in the grey matter, invasive devices have the characteristics such as stability of location, freedom from muscle movement artifacts, higher signal-to-noise ratio, and better spatial resolution and produce the highest quality signals of BCI devices. But as probes are implanted into the brain, there are risks related to surgery. Furthermore, they are prone to human immune responses, tissue encapsulation and the structural changes in vivo, causing the signal to become weaker or even lost as the body reacts to a foreign object in the brain [67, 69].

9.1.2 Partially-invasive BCIs

Partially invasive BCI devices are implanted inside the skull but rest outside the brain rather than within the grey matter. They produce better resolution signals than non-invasive BCIs (see below) where the bone tissue of the cranium deflects and deforms signals and have a lower risk of forming scar-tissue in the brain than fully-invasive BCIs [69, 72].

One partially-invasive technique is Electrocorticography (ECoG), which measures the electrical activity of the brain taken from electrodes that are embedded in a thin plastic pad and placed above the cortex, beneath the dura mater. ECoG technologies were first tried in humans in 2004 by Eric Leuthardt and Daniel Moran from Washington University in St Louis. In a later trial, the researchers enabled a teenage boy to play Space Invaders using his ECoG implant. This research indicates that it is difficult to produce kinematic BCI devices with more than one dimension of control using ECoG.

9.1.3 Non-invasive BCIs

Non-invasive techniques detects the brain signals from the surface of the skull. Although they are easy to wear, non-invasive implants produce low signal-to-noise ratio and poor spatial resolution because the skull attenuates the signals, dispersing and blurring the electromagnetic waves created by the neurons. Although the waves can still be detected it is more difficult to determine the area of the brain that created them or the actions of individual neurons. Extensive training is usually required for non-invasive BCI systems.

Electroencephalography (EEG) is the most studied potential non-invasive interface, mainly due to its fine temporal resolution, ease of use, portability and low set-up cost. It measures electrical potentials on the scalp and generates a record of the electrical activity of the brain. The electrical activity measured may be from the firing of the neurons of the brain due to the subject performing a task or thinking of performing a task (mental task) [13]. With this thought in mind, the EEG can be used in a number of systems and devices with the intention to provide motor or sensory function.

The theoretical basis for BCI devices such as EEG and ECoG is dependent on how well we are able to detect the neural signals and translate these signals into something we can understand. Firstly, one has to realize that every 'action' results in a pattern in the neural signal and this pattern have to be recognized by the BCI system. It is only after this pattern is recognized that it can be used as a control signal for external devices including computers, robotic arms, and other complex machines. EEG and ECoG signals contain transient, time-domain signals phase-locked to events such as the P300 and motor potentials. These field potentials contain many frequency-domain signals such as the μ rhythm and they can help in classifying the type of task being performed by a subject [72].

In 1997, Pfurtscheller et al. demonstrated the feasibility of using EEG to differentiate between imagination of left and right hand movement [59]. Recently, motor imagery has become the focus of BCI research. In the following, a data set on motor imagery is also selected for evaluation of the performance of our algorithms.

9.2 Experiments

To detect and translate the brain signal into something we can understand, a typical BCI system needs to include four stages: signal acquisition, pre-processing, feature extraction, and classification.

9.2.1 Experimental Data

The data selected to evaluate the applicability of our feature extraction algorithms for BCI applications is data set I from BCI competition III, 'Motor imagery in ECoG recordings, session-to-session transfer' [78]. During the BCI experiment, a subject had to perform imagined movements of either left small finger or the tongue. The time series of the electrical brain activity was picked up during these trials using a 8×8 ECoG platinum electrode grid which was placed on the contralateral (right) motor cortex. Every trial was recorded for 3 seconds duration with a sampling rate of 1000Hz. A detailed description of the data collection can be found in [43]. Training data set and test data set were recorded with about 1 week in between. There are 278 trials in training set, and 100 trials in test set.

The measured brain signals are usually high-dimensional and the activities specific to the tasks (left or right finger movement) are usually overwhelmed by spontaneous EEG and other non-task activities. Proper method to extract the important information for recognition is therefore necessary and crucial for good performance. In the following two different approaches for classifying the brain signals are presented: The first approach, called channel-based approach, analyzes individual channels separately. Pre-processing and feature extraction methods discussed preciously are applied on each channel separately. The recognition of the brain activities is by the use of a single channel; Another approach determines the type of mental activities based on discriminatory information extracted from all channels. It needs to pre-filter (or select) useful time-frequency components from all channels of the signal and then concatenate all the selected components, after which feature extraction is applied on the concatenated components. The details of the two approaches and their experimental results are described below.

9.2.2 Classification Based on Single Channel

Since only a small part of the brain cortex are associated with task, signals from only a small number of channels are actually useful for recognition. The problem is then to identify the useful channels and use only them for the recognition task. The channel-based approach classifies signals based on each individual channel and selects the one that gives rise to the best recognition result.

9.2.2.1 Pre-processing and Feature Extraction

Low-pass-filtering and down-sampling It is believed that important information about cognitive activity of the brain mainly reside in these frequency bands: $\theta(4-8Hz)$,$\alpha(8-12Hz)$, $\beta(12-16Hz)$, and $\gamma(30-44Hz)$ [26]. The range of the frequency bands are in the range of 0 to 50Hz. However, the original sampling frequency is 1000Hz, which is more than enough for interpreting the brain signal. Low-pass-filtering (LPF) is employed to reduce the sampling rate and also remove high-frequency noise. Therefore the original signal is low pass filtered at 50Hz and down-sampled at 100Hz. Figure 9.1 shows the LPF filter used prior to the down-sampling. After down-sampling, the power spectral density (PSD) is estimated. An example of the down-sampled signal in time and frequency domain is shown in Figure 9.2.

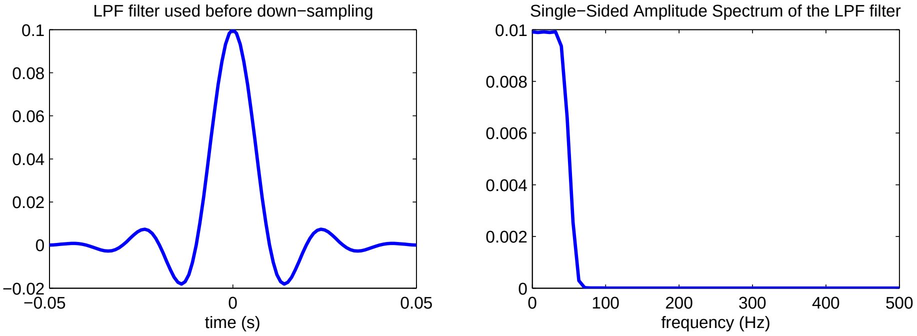


Figure 9.1: Low pass filter used before down-sampling. - Left:The impulse response of the LPF filter; Right: The frequency response of LPF filter.

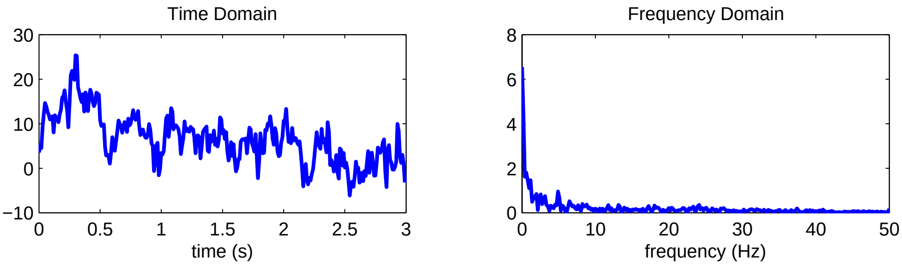


Figure 9.2: One sample of the signal from dataset I of BCI competition III after down-sampling - Left: the signal in time domain; Right: the power spectral density of the signal.

Normalization After down-sampling, three different methods are tried to reduce the difference between different samples: normalization of each channel, normalization of each trial, common average referencing (CAR). The purpose of normalization of each channel is to reduce difference between different channels. Each channel is normalized to be zero mean and unit variance. Normalization of each trial does the same normalization, i.e., zero mean and unit variance, but on each trial instead of each channel. In CAR, the mean of all channels is subtracted from each channel. These normalization methods are tried on both time and frequency representations of the signals. Experimental results were obtained with both time domain and frequency domain.

Feature extraction In the first part, RMLD is selected as the feature extraction method for all pre-processed data both in time and frequency domain: down-sampled data, channel-normalized data, trial-normalized data, and CAR data. The pre-processing method that gives the best result is then selected for further experiments. RMLD is selected for its relative simplicity compared to RCBLD and superiority in performance compared to FLD. As for the application on face recognition, the simple nearest-neighbor classifier with Euclidean norm as the similarity measurement is employed.

It will be presented in the following that channel-normalized data in the frequency domain results in the best classification performance with RMLD. Therefore, the channel-normalized data in the frequency domain is selected in the subsequent experiment to evaluate the performance of RCBLD in comparison to RMLD. The experimental results are shown below.

9.2.2.2 Experimental Results

For each pre-processed data in time and frequency representations, the classification accuracy is obtained based on each individual channels. That is to say, for each channel, the classification system is trained with signals of that channel, and classification accuracy is obtained for test samples of the same channel. This training and testing are repeated over all channels and the channel with lowest classification error rate is selected. The lowest classification error rates with different pre-processing techniques are shown in Table 9.1. In the table, the corresponding error rates of FLD are also given for comparison. It can be observed that lowest classification error rate of FLD is 28%, while the lowest error rate of RMLD is 14% by extracting one more feature. The improvement by RMLD over FLD is significant.

Table 9.1: Lowest classification error rates (%) for data with different normalization methods in both time and frequency domain.

	Time Domain			
	down-sampled	channel normalized	trial normalized	CAR
FLD	36	38	35	38
RMLD	31	31	33	31
Channel	14	14	14	4

	Frequency Domain			
	down-sampled	channel normalized	trial normalized	CAR
FLD	29	28	19	32
RMLD	18	14	19	19
Channel	29	40	38	21

Comparing the classification error rates for time domain and frequency domain, one can observe that the error rates for frequency domain are significantly lower than those for time domain for all the pre-processed data. This means classification of motor imagery ECoG signals is better dealt with in frequency domain rather than in time domain. It seems that frequency description is more revealing for the characteristics of motor imagery ECoG signals. Comparing results for different pre-processed data in frequency domain, channel-normalized data results in the lowest error rate. Therefore, it is selected in subsequent experiment to test the performance of RCBLD in comparison to RMLD.

The experimental results of RCBLD as well as FLD and RMLD are shown in Table 9.2. From the table, one can see that RCBLD outperforms RMLD. The results for RBLD is not listed here because RBLD is actually the same as RMLD for 2-class problems. The experimental results here again confirm the advantage of RCBLD over FLD and RMLD.

Table 9.2: Lowest classification error rates (%) based on channel-normalized data in frequency domain.

	FLD	RMLD	RCBLD
Lowest error rate (%)	28	14	12
Channel	40	40	38

To further improve the recognition performance, another classifier is applied in place of nearest-neighbor classifier on the best channel with best pre-processing method, that is, channel 38 after channel-normalization in frequency domain. One major shortcoming of nearest-neighbor classifier is its susceptibility to noisy attributes and noisy instances. One remedy to this is to take a majority vote over the k nearest neighbors, and the resulting classifier is termed k-NN (k-nearest-neighbor) classifier. However, one major issue of k-NN classifier is the selection of k. One solution is to weight the vote of each instance by the distance of that instance to the test sample. The classifier can be defined as follows

$$L(x) = \max_i \{ \sum_{x_j \in i} f(d(x, x_j)) \} \tag{9.1}$$

where $L(x)$ is the assigned class label of test sample x, and $f(d)$ is the weighted vote from sample x_j, which is a decreasing function of distance between two instances x and x_j, e.g., $f(d) = 1/d^2$ or $f(d) = \exp(-d)$. Note that all samples from the ith class are used in the summation for determining the class label of the test sample. This way there is no need to select a suitable k , as for the k-NN classifier. With $f(d) = 1/d^2$ adopted as the weight function, the lowest error rates are further reduced as shown in Table 9.3.

Table 9.3: Lowest classification error rates (%) obtained by nearest-neighbor classifier and weighted k-NN classifier based on channel 38 (channel-normalization & frequency domain).

classifier	RCBLD
nearest-neighbor	12
weighted k-NN	10

9.2.3 Classification Based on All Channels

One drawback of the channel-based approach is that exhaustive search is required to find the channel and the pre-processing method that give rise to the best performance. In real-world applications one does not know which channel and which pre-processing method to choose in order to get the best performance possible. One can solve this problem by collecting a validation set and use it for the selection of best channels and pre-processing methods. Another approach is to treat all the channels as a single entity, which does not require beforehand channel-selection. In the following a method based on all channels is presented. The new method first selects "useful" time and frequency components from all the channels, using some objective measurements. The selection process can be done manually or automatically. After the selection process, discriminant features are extracted from the set of "useful" time-frequency components selected from all the channels. Finally, classification is performed using the extracted features.

9.2.3.1 Spectrogram

To select useful time-frequency components, the spectrogram for each channel of each sample is first estimated by short time Fourier transform (STFT)). A spectrogram is 2D PSD map $P(f,t)$ that decomposes a temporal signal along time and frequency axes. The window size used in STFT is 0.5s with 0.25s overlap. As a result, the number of time and frequency components are:

$$\begin{aligned} N_T &= (3s - 0.25s)/(0.5s - 0.25s) = 11 \\ N_F &= (0.5s \times 100Hz)/2 + 1 = 26 \end{aligned}$$

The spectrogram of one channel is visualized below in Figure 9.3. It is also shown in Figure 9.4 in dB scale. The two sub-figures on the left show two training samples, while the other two sub-figures on the right show two test samples. In the figure, color is used to indicate the magnitude of the spectrogram. Cold colors like blue indicate small values while warm colors like red indicate big values. The colorbar used for Figure 9.3 is shown in Figure 9.5.

9.2.3.2 Quantitative Measure of Discrimination Power

Before one can select the time-frequency component from the spectrogram of a signal, the discrimination power of the time-frequency component should be measured so that components with high discrimination power could be selected. One quantitative measure of the discrimination power is the Fisher ratio, defined as

$$r = \frac{tr\{S_B\}}{tr\{S_W\}} \tag{9.2}$$

Fisher-ratio map describes the discrimination power of each time-frequency component of a signal. The Fisher-ratio maps for the training and test data are computed respectively. The Fisher-ratio maps of one channel is visualized in Figure 9.6 and in Figure 9.7 in dB scale. The Fisher-ratio map on the right is computed on the test samples. It is shown in the figure just for comparison to that for training samples. Only the Fisher-ratio map for the training samples are used for the selection of time-frequency in the experiments. One can see that frequency components around 10Hz are more discriminative in general.

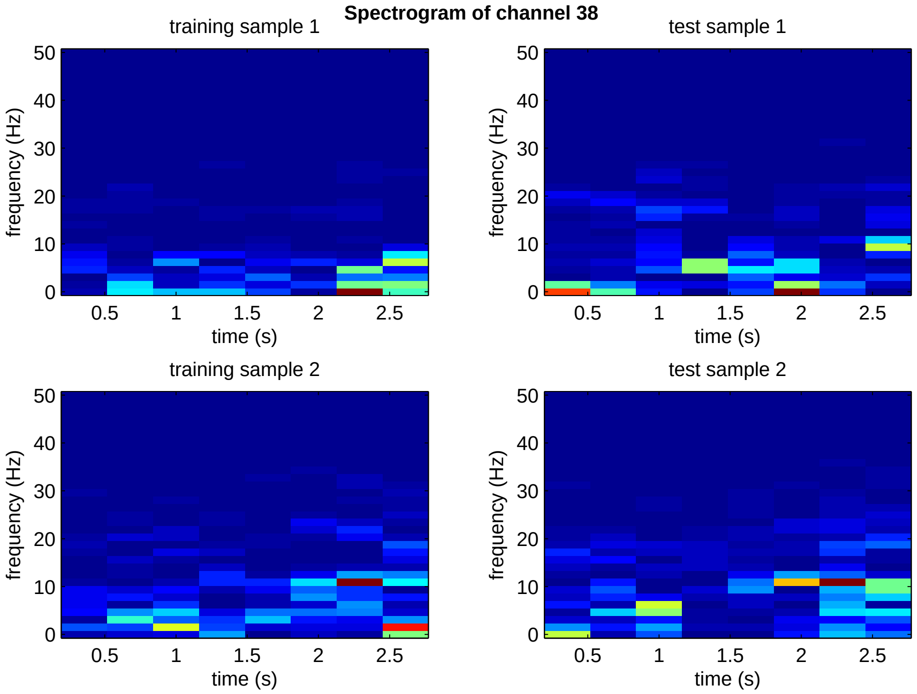


Figure 9.3: Spectrogram of a Channel -

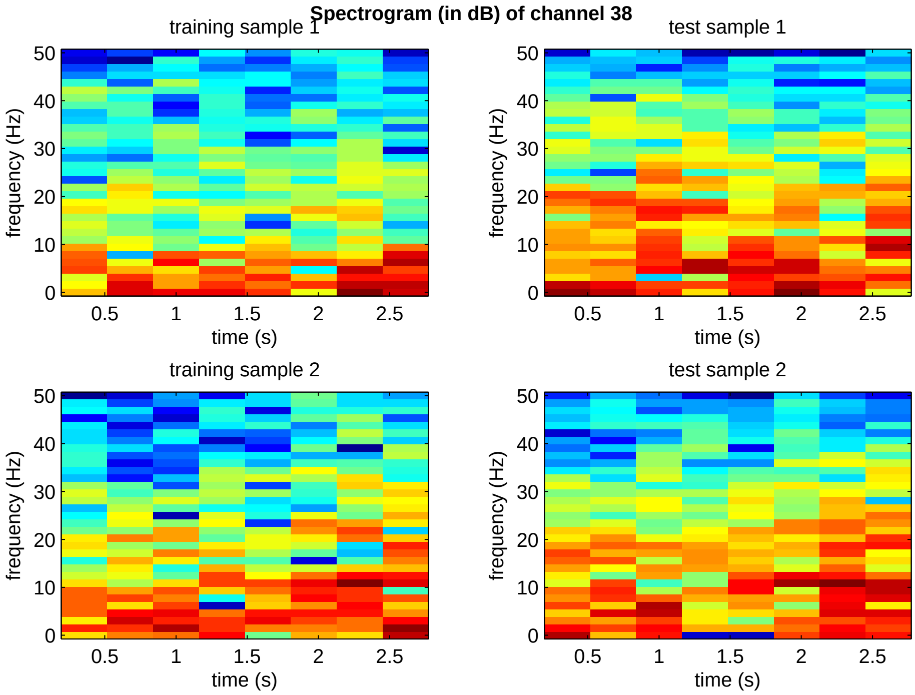


Figure 9.4: Spectrogram of a Channel in dB scale -

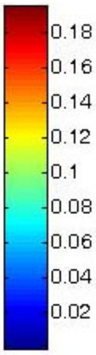


Figure 9.5: Colorbar used for the spectrum as shown in Figure 9.3 -

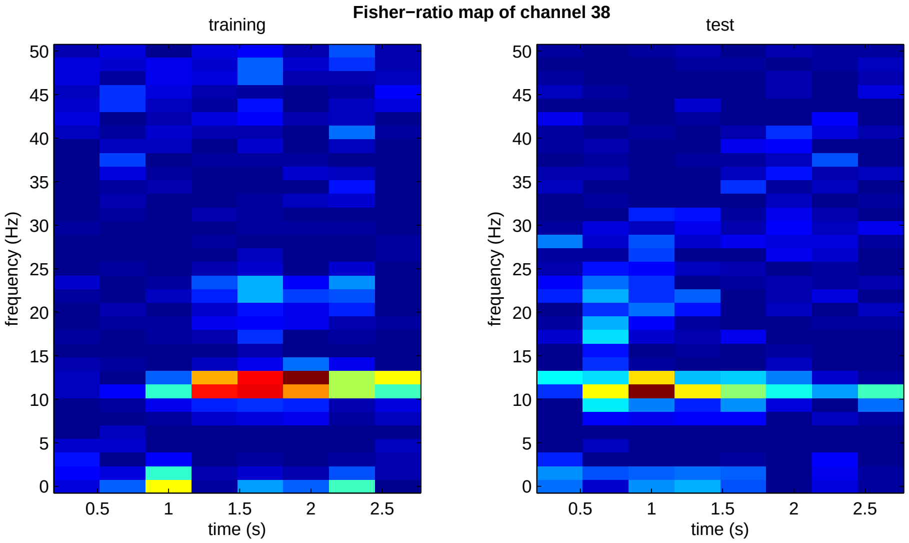


Figure 9.6: Fisher-Ratio Map of a Channel. - Left: Fisher-ratio map computed on the training data; Right: Fisher-ratio map computed on the test data.

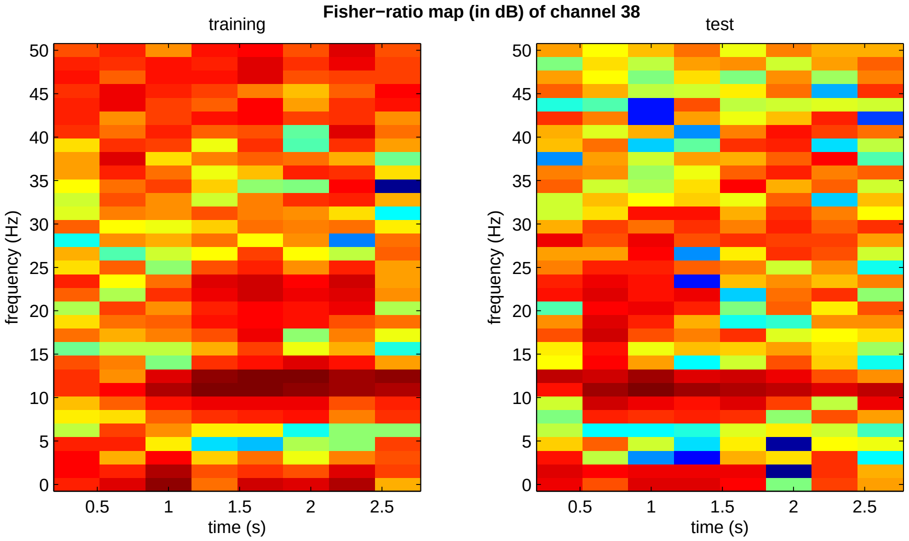


Figure 9.7: Fisher-Ratio Map of a Channel in dB Scale. - Left: Fisher-ratio map computed on the training data; Right: Fisher-ratio map computed on the test data.

The histogram of the Fisher-ratios of time-frequency components from all channels and all data samples are plotted in Figure 9.8. One can see from the figure that most of time-frequency components have very small discrimination power, and are not related to the task. Only a small portion of the time-frequency component should be selected for the classification task.

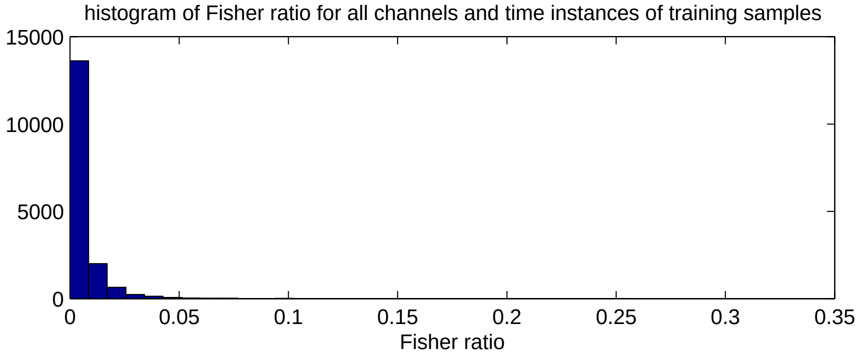


Figure 9.8: Histogram of Fisher-ratio values of all time-frequency components from all channels and all data samples. -

9.2.3.3 Time-frequency Component Selection from All Channels

Time-frequency components are selected in blocks which contains high discrimination power. The time-frequency components are selected independently for each individual channel. Therefore, each channel has its very own selected time-frequency blocks. Some channel may have a block with size 0×0. This means that no component of the channel is considered to be useful for classification and therefore this channel is dismissed from the classification task.

One can select the time-frequency blocks manually. Another way is to find proper time-frequency blocks automatically. The following algorithm is devised to automatically find the time-frequency blocks:

1. Threshold the Fisher-ratio map by the median value
2. Find connected objects. If the area of an object is small, discard it.
3. Find bounding box of each object. Check the 4 sides of the bounding box, if most of parts of one side are below the threshold, remove that side. Repeat

this process until no side needs to be removed or the area of the box is too small.

The time-frequency blocks selected automatically by the above algorithm for training data are shown in Figure 9.9, 9.10, 9.11, and 9.12. For visual comparison, the time-frequency blocks selected by the same process for test samples are also shown in Figure 9.13, 9.14, 9.15, and 9.16. The location of the blocks should be similar for training and test samples if good classification performance is expected.

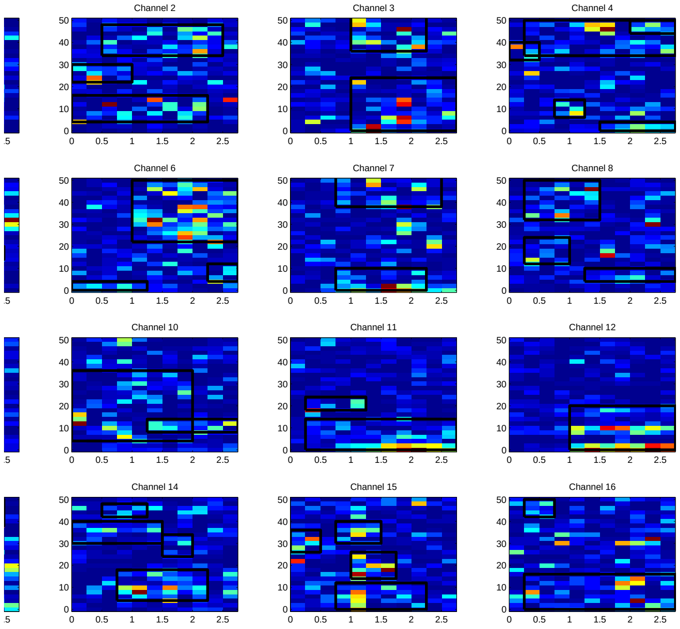


Figure 9.9: Automatically selected time-frequency blocks for channels 1-16 for training samples -

9.2.3.4 Experimental Results

The same feature extraction methods and classifiers, as in the experiments based on single channels, are applied on the set of selected time-frequency components. The lowest classification error rates of RMLD on data by manual and automatic selection of time-frequency blocks with different pre-processing techniques are shown in Table 9.4.

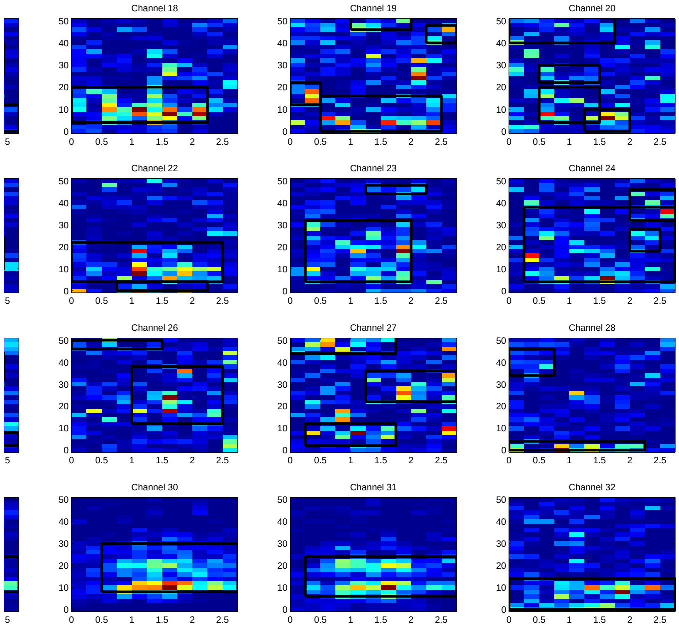


Figure 9.10: Automatically selected time-frequency blocks for channels 17-32 for training samples -

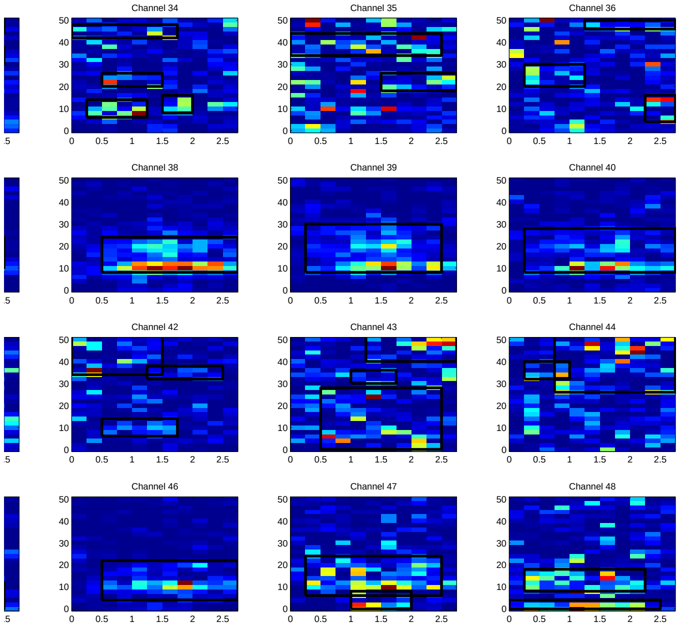


Figure 9.11: Automatically selected time-frequency blocks for channels 33-48 for training samples -

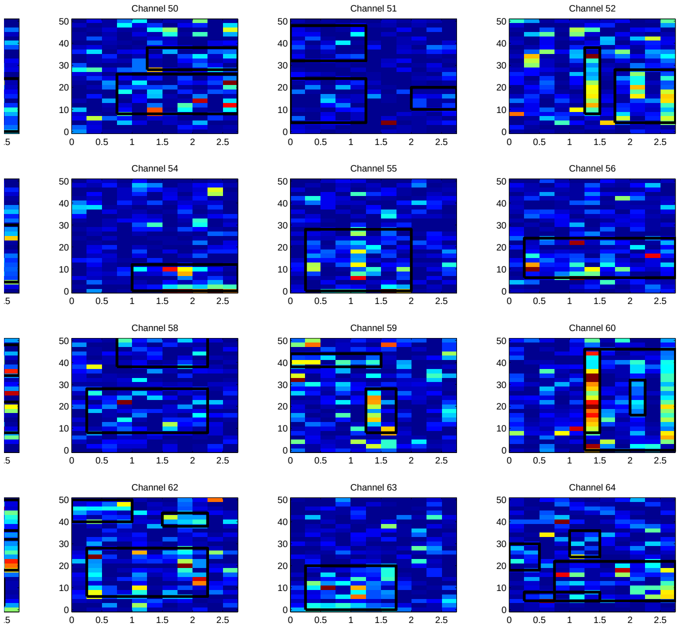


Figure 9.12: Automatically selected time-frequency blocks for channels 49-64 for training samples -

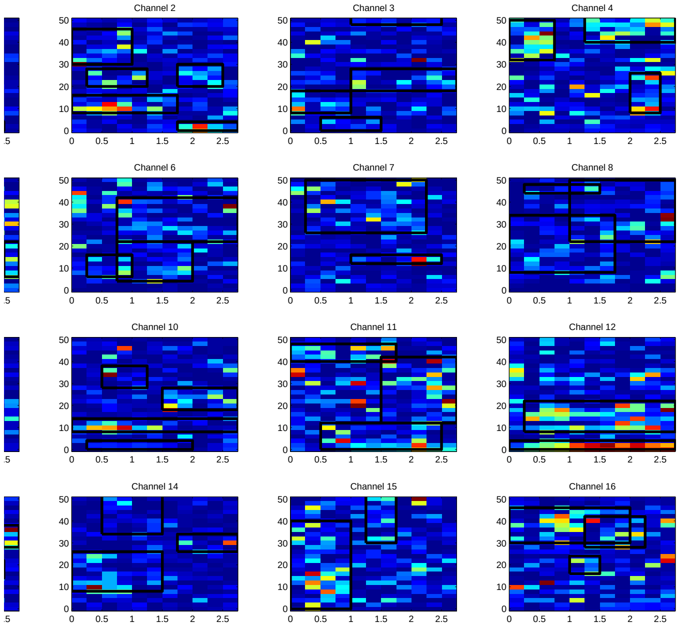


Figure 9.13: Automatically selected time-frequency blocks for channels 1-16 for test samples -

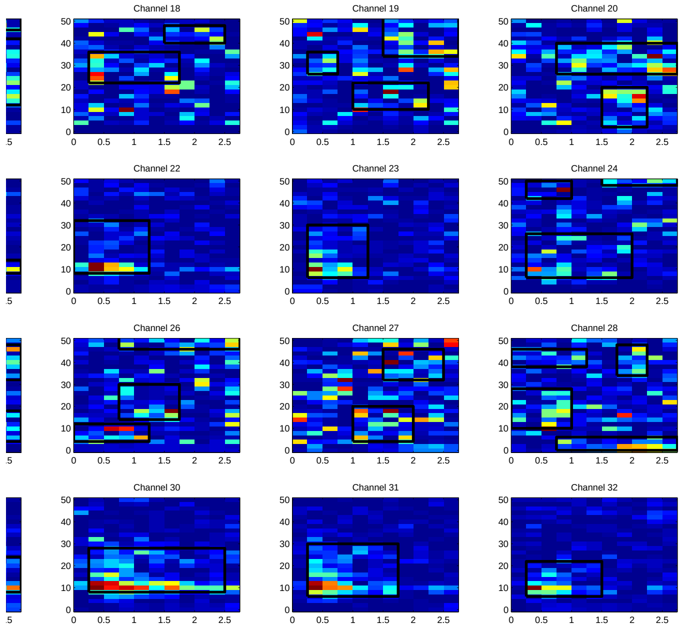


Figure 9.14: Automatically selected time-frequency blocks for channels 17-32 for test samples -

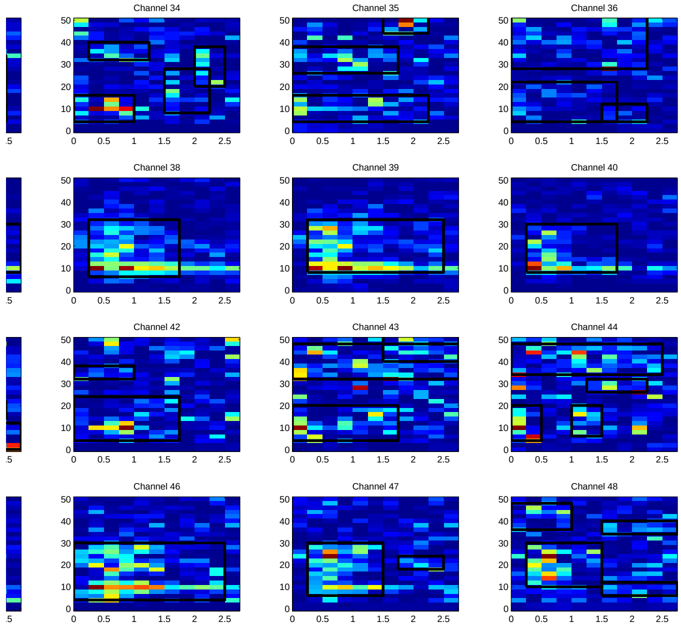


Figure 9.15: Automatically selected time-frequency blocks for channels 33-48 for test samples -

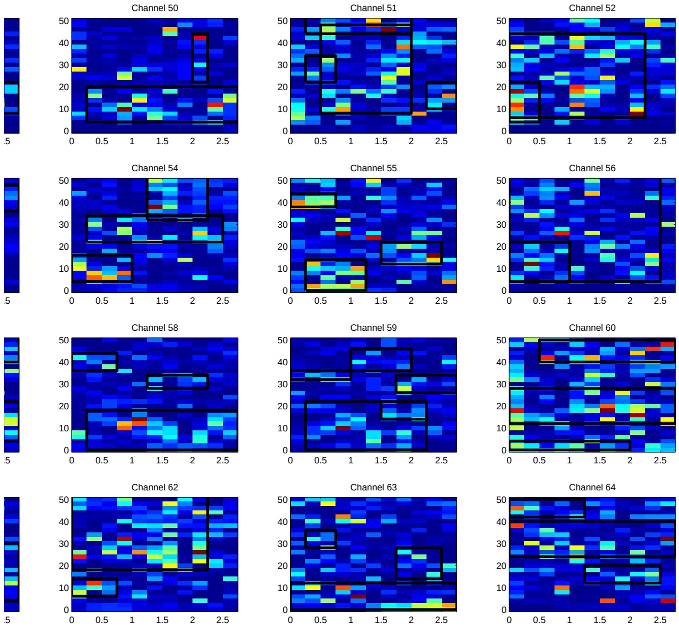


Figure 9.16: Automatically selected time-frequency blocks for channels 49-64 for test samples -

Table 9.4: Lowest classification error rates (%) of RMLD on data obtained by manual and automatic selection time-frequency components for different pre-processing methods.

Block Selection	down-sampled	channel normalized	trial normalized	CAR
manual	30	12	19	30
auto	35	12	22	26

Comparing the error rates by manual and automatic selection, one can see that their performances are comparable with the lowest error rates being 12% for both selection methods. This indicates that the automatic process does well in finding good time-frequency blocks.

Another interesting fact one can observe from the comparison is that the pre-processing method that gives rise to the best classification performance is normalization of channel for both block-selection methods. This observation also conforms with results from the channel-based approach.

To compare the performance of RCBLD and RMLD, channel-normalization and automatic time-frequency selection are selected as the pre-processing method and the experimental results are list in Table 9.5. From the table, one can see that RCBLD achieves better results than RMLD with either nearest-neighbor or weighted k-NN classifier.

Table 9.5: Lowest classification error rates (%) on data pre-processed by channel-normalization and automatic time-frequency selection.

classifier	RMLD	RCBLD
nearest-neighbor	12	10
weighted k-NN	12	10

Comparing the lowest error rate of the channel-based approach and the automatic time-frequency selection approach, one can see that the time-frequency selection from all channels approach achieves the same level of performance on channel-normalized data. However, the channel-based approach requires the knowledge of the usefulness of individual channels beforehand. The advantage of time-frequency selection approach is that it can be done automatically.

The competition results on data set I is available at [78]. Compared to others' results, the lowest error rate of our system are nearly the same as the winner's 9% error rate.

Chapter 10

Conclusion

Automatic (machine) recognition of patterns is an important task in a wide variety of real-world applications. The designing of a satisfactory pattern recognition system usually requires a good feature extraction algorithm, which plays a crucial role for the performance the pattern recognition system. It is often problem dependent and requires specialized knowledge of the specific problem itself to devise a competent feature extraction algorithm and the development of a general procedure for effective feature extraction always remains an interesting and also challenging problem.

This dissertation focuses on one of the most important problems in the research field of pattern recognition: discriminant feature analysis for pattern recognition. The objective of this thesis is to develop general-purpose feature extraction tools that could be applied to a wide variety of pattern recognition problems.

The algorithmic development is presented in Part I of this thesis. Before introducing the proposed algorithms for discriminant feature extraction, a number of popular feature extraction algorithms are briefly reviewed in Chapter 2. Among the various feature extraction algorithms, FLD has probably become one of the most popular feature extraction algorithms due to its relevance to classification: it finds features that maximize the between-class scatter and meanwhile minimize within-class scatter. However, FLD also suffers from several major limitations. The limitations or shortcomings of FLD that are analyzed and identified in the chapters of Part I are listed below:

- The total number of features that can be extracted by FLD is at most $C-1$, where C is the number of classes.

- Discriminant information from $\overline{F}_W$, the null space of S_W, cannot be exploited by FLD, as FLD requires S_W to be non-singular in the computation of its solution.

- FLD implicitly assumes uni-modal Gaussian distributions for the underlying class. This is due to its parametric formulation for the between-class and within-class scatter matrices. The assumption is often too strong to fit the real-world applications.

- Although FLD extracts discriminating information by maximizing the between-class scatter and minimizing the within-class scatter at the same time, the criterion function it optimizes is not directly related to the classification performance. The optimization of its criterion function thus does not necessarily mean a good classification performance.

In Chapter 3, RMLD is proposed to use a recursive strategy and the modified criterion function of MFLD to eliminate the feature number constraint and extract discriminant information from both the principal space of S_W and the null space of S_W. The recursive method used by RMLD is, however, computationally more efficient than the one used by RFLD. RMLD avoids the re-computation of S_B and S_W by projecting them into the null space $\overline{W}_k$ and extracts $C-1$ features instead of only one feature per iteration.

In Chapter 4, RCLD is proposed to handle complex class distributions that cannot be well approximated as uni-modal Gaussian distributions. Due to the parametric definition of S_B and S_W, FLD implicitly assumes a uni-modal Gaussian distribution for the underlying classes. Thus it may not work well when the underlying class distributions cannot be well approximated by uni-modal Gaussian distributions. To solve this problem, RCLD employs a cluster-based approach to approximate complex class distributions as unions of uni-modal Gaussian distributions. A fuzzy-clustering based RCLD works well no matter how well the clusters are formed.

The issue of selecting proper number of clusters and degree of fuzziness of clusters for each class is essential for achieving good performance with RCBLD. We proposed a way of determining cluster numbers using SOM. The selection of degree of fuzziness for fuzzy clustering is problem dependent and has been carried out by trial and error in our experiments.

In Chapter 5, RBLD is proposed to relate the criterion function to the classification performance. The Bayesian criterion function of RBLD is derived as an approximation of one of the two coherent error bounds that confine the Bayes error. The optimization of the criterion function would make the two coherent error bands small and as a result the classification error small. The solution to the approximated Bayesian criterion function is obtained without resorting to some gradient-based method.

In Chapter 6, the ideas of RMLD, RCLD, and RBLD are integrated and the resulted algorithm, termed RCBLD, combines the different strength of the Bayesian criterion function of RBLD, the cluster-based idea of RCLD, and the recursive procedure of RFLD. It has following main advantages over FLD and its variations:

- It has a Bayesian criterion function in the sense that the Bayes error is confined by a coherent pair of error bounds and the maximization of the criterion function is equivalent to minimization of one of the error bounds. Compared to FLD, RCBLD's criterion function is not dominated by far apart classes. Instead, it pays more attention to close classes.

- The solution of the Bayesian criterion function can be easily obtained without resorting to gradient-based methods.

- Capability of handling complex class distributions as unions of Gaussian distributions.

- Use of fuzzy clustering based definition of S_W which makes the algorithm performs well no matter how well clusters are formed.

- Elimination of feature number constraint by adopting a recursive procedure.

- Less computational expensive than RFLD by calculating $C' - 1$ features at each iteration instead of only one, where C' corresponds to the total number of clusters. Computational cost is also reduced by use of the null space $\overline{W}_k$ to avoid the re-computation of S_B and S_W, as required by RFLD.

- Full utilization of all discriminant information available by replacing within-class scatter matrix by the total scatter matrix in the criterion function.

In spite of the strong assumptions of equal a priori probability and equal covariances, RCBLD may still be able to obtain good results due to two reasons: (1) the summation in the criterion function may cancel out the adverse effect of each individual deviation from the assumptions; (2) the number of samples available for training is usually quite limited and as a result simple models with less parameters are usually favored.

Part II of this thesis presents the experimental work that assesses the performance of the proposed algorithms. Since the new algorithms are designed as general feature extraction tools, they have been applied to various pattern classification problems from UCI Machine Learning Repository in Chapter 7, face recognition problems in Chapter 8, and BCI applications in Chapter 9.

In Chapter 7, 7 multi-class databases with sizes ranging from about 100 samples to more than 5,000 samples are selected to test the performance of the proposed algorithms in dealing with different pattern recognition problems with different training sample size.

To test the algorithms' ability in classifying more challenging pattern recognition problems, different face recognition tasks including identity recognition, facial expression recognition, and glass-wearing recognition have been experimented in Chapter 8. Although only simple pre-processing techniques and simple classifiers like nearest neighbor classifier are used in our system, our proposed algorithms demonstrate classification performance comparable to some recently reported results.

To further test the algorithms' ability as a general-purpose feature extraction methods, they are applied to BCI applications in Chapter 9. Two different approaches have been tried: one based on single channel; the other based on all channels. The approach based on single channel requires the selection of channels beforehand. It can also be used to identify the region of cortex that is related to the mental activity. The approach based on automatic selection of time-frequency components from all channels does not require any expertise or user intervention.

The experimental results have verified the effectiveness of the new algorithms. It is my strong belief that improvement can also be expected for other pattern recognition problems such as iris recognition, hand gesture recognition, etc.

One price paid for the superior performance of RCBLD is that it is computationally more intensive. However, the training stage of RCBLD is done off-line and therefore is not critical for some applications.

There are several directions that the proposed RCBLD method can be extended:

- The method can be extended to be nonlinear by adopting a kernel approach, or by a hybrid network where the first hidden layer implements the nonlinear transformation and the second hidden layer implements the RCBLD method.

- Chernoff distance can be used instead of Mahalanobis distance for the criterion function such that better results may be achieved for heteroscedastic normal distributions.

- Classifier other than the nearest-neighbor classifier can be used with the proposed method, which is likely to improve the classification performance.

References

[1] S. Aeberhard, D. Coomans, and O. de Vel. Comparison of classifiers in high dimensional settings. Tech. Rep. 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992. 59

[2] S. Baker and S. Nayar. Pattern rejection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 544–549, 1996. 12

[3] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, July 1997. 12, 13, 14, 17, 57, 64, 66, 67, 73

[4] B. Blankertz, K. R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer. The bci competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomed. Eng.*, 51(6):1044–1051, 2004. 89

[5] W. W. Bledsoe. The model method in facial recognition. PRI 15, Panoramic research Inc., Palo Alto, CA, 1964. 63

[6] M. Bressan and J. Vitria. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24:2743–2749, 2003. 19, 20

[7] I. Bruner and R. Tagiuri. The perception of people. In L. G., editor, *Handbook of Social Psychology*, volume 2, pages 634–654. Addison-Wesley, 1954. 63

[8] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33: 1713–1726, 2000. 12, 15, 17, 64

[9] X. Chen and T. Huang. Facial expression recognition: A clustering-base approach. *Pattern Recognition Letters*, (24):1295–1302, 2003. 29, 30

[10] Y. Cheng, K. Liu, J. Yang, Y. Zhuang, and N. Gu. Human face recognition method based on the statistical model of small sample size. In *SPIE Proc. Intelligent Robots and Computer Vision X: Algorithms and Technology*, pages 85–95, 1991. 12

[11] F. R. K. Chung. Spectral graph theory. In *Proc. Regional Conf. Series in Math.*, volume 92, 1997. 22

[12] Y. Cui, D. Swets, and W. J. Learning-based hand sign recognition using shoslif-m. In *Int'l Conf. on Computer Vision*, pages 631–636, 1995. 12

[13] E. A. Curran and M. J. Stokes. Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems. *Brain Cogn.*, 51:326–336, 2003. 89, 91

[14] R. Duda, P. E. Hart, and D. Stork. *Pattern Classification.* Wiley, New York, 2nd edition, 2001. ISBN 0471056693. 5, 12, 40, 56, 57, 73

[15] J. Eggermont, J. N. Kok, and W. A. Kosters. Genetic programming for data classification: partitioning the search space. In *SAC*, page 1001, 2004. 60

[16] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *J. Opt. Soc. Am. A*, 14(8):1724–1733, Aug 1997. 12, 17, 64, 66

[17] X. Feng, A. Hadid, and M. Pietikäinen. A coarse-to-fine classification scheme for facial expression recognition. In *The First International conference on Image Analysis and Recognition*, pages 668–675, Porto, Portugal, 2004. 65, 87

[18] X. Feng, B. Lv, Z. Li, and J. Zhang. Automatic facial expression recognition with AAM-based feature extraction and SVM classifier. In *MICAI 2006: Advances in Artificial Intelligence. 5th Mexican International Conference on Artificial Intelligence*, pages 726–33, Apizaco, Mexico, 2006. Springer-Verlag. 65, 87

[19] X. Feng, M. Pietikäinen, and A. Hadid. Facial expression recognition based on local binary patterns. *Pattern Recognition and Image Analysis*, 17(4):592–8, 2007. 87

[20] R. A. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936. 12, 56

[21] E. Frank and S. Kramer. Ensembles of nested dichotomies for multi-class problems. In *In Proc 21st International Conference on Machine Learning*, pages 305–312. ACM Press, 2004. 59, 60, 61

[22] J. H. Friedman. Regularized discriminant analysis. *J. Amer. Statist. Assoc.*, (84): 165–175, 1989. 17, 52

[23] K. Fukunaga. *Statistical Pattern Recognition.* Adcademic Press, 1990. 12, 19, 20

[24] X. Geng and Y. Zhang. Facial expression recognition based on the difference of statistical features. In *International Conference on Singal Processing*, pages 16–20, Guilin, China, 2006. 87

[25] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. 67, 81

[26] A. Gevins, M. Smith, L. McEvoy, and D. Yu. High resolution EEG mapping of cortical activation related to working memory: Effects of task difficulty, type of processing, and practice. *Cerebral Cortex*, 7:374–385, 1997. 93

[27] D. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–399, 2001. 49

[28] X. He, S. Yan, Y. Hu, and H. Zhang. Learning a locality preserving subspace for visual recognition. In *Proc. of the IEEE International Conference on Computer Vision*, volume 1, pages 385–392, 2003. 21

[29] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: Component-based versus global approaches. *Comput. Vis. Image Understand.*, 91(1):6–12, 2003. 65

[30] Y. Horikawa. Facial expression recognition using KCCA with combining correlation kernels and kansei information. In *Fifth International Conference on Computational Science and Applications*, pages 489–495, Perugia, Italy, 2008. 87

[31] D. Huang, C. Xiang, and S. S. Ge. Feature extraction for face recognition using recursive bayesian linear discriminant. In *2007 5th International Symposium on Image and Signal Processing and Analysis*, pages 299–304, Istanbul, Turkey, Sept. 2007. 2007 5th International Symposium on Image and Signal Processing and Analysis, IEEE. 64

[32] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(2):153–158, 1997. 5

[33] J. M. Jerez, L. Franco, and I. Molina. CBA generated receptive fields implemented in a facial expression recognition task. In *IWANN 2003 : international work-conference on artificial and natural neural networks*, volume 2686, pages 734–741, 2003. 87

[34] Y. Jiang and Z. hua Zhou. Editing training data for knn classifiers with neural network ensemble. In *Lecture Notes in Computer Science, Vol.3173*, pages 356–361. Springer, 2004. 59, 60, 61

[35] X. Jing, D. Zhang, and X. Yao. Improvements on the linear discrimination technique with application to face recognition. *Pattern Recognition Letters*, 24:2695–2701, 2003. 15, 17

[36] T. Kanade. *Computer Recognition of Human Faces*, 47, 1977. 63

[37] M. D. Kelly. Visual identification of people by computer. Stanford AI Project 130, Stanford, Stanford, CA, 1970. 63

[38] M. Kirby and L. Sirovich. Application of the karhumen-loève procedure for the characterization of human faces. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 12: 103–108, 1990. 66

[39] T. Kohonen. Som toolbox online documentation. URL `http://www.cis.hut.fi/projects/somtoolbox/package/docs2/som_supervised.html`. 36

[40] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, third extended edition edition, 2001. ISBN 3-540-67921-9. 35, 36

[41] T. Kohonen, K. Mäkivasara, and T. Saramäki. Phonetic maps - insightful representation of phonological features for speech recognition. In *In proceedings of International Conference on Pattern Recognition (ICPR)*, pages 182–185, Montreal, Canada, 1984. 36

[42] K.-C. Kwak and W. Pedrycz. Face recognition using an enhanced independent component analysis approach. *IEEE Transactions on Neural Networks*, 18(2): 530–541, Mar 2007. 64

[43] T. Lal, T. Hinterberger, G. Widman, M. Schroder, J. Hill, W. Rosenstial, C. Elger, B. Scholkopf, and N. Birhaumer. *Methods towards invasive human brain computer interfaces*, volume 17, pages 734–744. MIT Press, Cambridge, MA, USA, 2005. 92

[44] E. Leuthardt, G. Schalk, J. Wolpaw, J. Ojemann, and D. Moran. A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering*, 1:63–71, 2004. 89

[45] Z. Li, W. Liu, D. Lin, and X. Tang. Nonparametric subspace analysis for face recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 961–966, 2005. 19, 20, 76

[46] C. Liu. Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28 (5):725 – 737, May 2006. 17

[47] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Trans. Image Processing*, 11:467–476, Apr. 2002. 12, 17, 64, 75

[48] M. Loog and R. Duin. Linear dimensionality reduction via a heteroscedastic extension of LDA: The Chernoff criterion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):732 – 739, 2004. 18

[49] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *Proceedings, Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, Nara Japan, April 1998. Third IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society. 36, 67, 87

[50] M. J. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(12):1357–1362, Dec. 1999. 87

[51] A. M. Martínez and A. C. Kak. PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:228–233, Feb 2001. 12, 17

[52] G. J. Mclachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, New York, 1992. 12

[53] A. Miller. *Subset Selection in Regression.* Chapman & Hall, CRC, Los Angeles, CA, second edition, 2002. 5

[54] H. Moon and P. J. Phillips. Computational and performance aspects of pca-based face recognition algorithms. *Perception*, 30(3):301–321, 2001. 67

[55] C. N. S. G. Murthy and Y. V. Venkatesh. Encoded pattern classification using constructive learning algorithms based on learning vector quantization. *Neural Networks*, 11:315–322, 1998. 85

[56] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998. URL `http://archive.ics.uci.edu/ml/index.html`. 54, 56, 57

[57] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559C572, 1901. 10

[58] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 84–91, 1994. 66

[59] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalogra. Clin. Neurophysiol.*, 8(4):441–446, 1997. 91

[60] P. Phillips, R. Mccabe, and R. Chellappa. Biometric image processing and recognition. In *European Signal Processing Conference*, 1998. 64

[61] S. Raudys and A. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practioners. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991. 5

[62] S. Raudys and V. Pikelis. On dimensionality, sample size, classification error, and complexity of classification algorithms in pattern recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2:243–251, 1980. 5

[63] L. Rueda and M. Herrera. A new approach to multi-class linear dimensionality reduction. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 634–643, 2006. 19

[64] J. Ruiz-del Solar and P. Navarrete. Eigenspace-based face recognition: A comparative study of different approaches. *IEEE Trans. Syst., Man, Cybern. C, Cybern.*, 35(3):315–325, 2005. 65

[65] T. Sabisch, A. Ferguson, and H. Bolouri. Identification of complex shapes using a self organizing neural system. *IEEE Transactions on Neural Networks*, 11(4): 921–934, Jul 2000. 35

[66] P. Sajda, A. Gerson, K.-R. Müller, B. Blankertz, and L. Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *IEEE Trans. Neural Sys. Rehab. Eng.*, 11(2):184–185, 2003. 89

[67] J. C. Sanchez, N. Alba, T. Nishida, C. Batich, and P. R. Carney. Structural modifications in chronic microwire electrodes for cortical neuroprosthetics: A case study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14 (2):217C221, 2006. 90

[68] G. Santhannam, S. Ryu, B. Yu, A. Afshar, and K. Shenoy. A high-performance brain-computer interface. *Nature*, 442(7099):195–198, 2006. 89

[69] S. Scott. Converting thoughts into action. *Nature*, 442(7099):141C142, 2006. 90

[70] F. Y. Shih, C.-F. Chuang, and P. S. P. Wang. Performance comparisons of facial expression recognition in JAFFE database. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(3):445–459, May 2008. 87

[71] Y. Shinohara and N. Otsu. Facial expression recognition using Fisher weight maps. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 499–504, Korea, 2004. 87

[72] E. Suter. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Application*, 15:31–45, 1992. 90, 91

[73] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–836, Aug 1996. 12, 17

[74] E. K. Tang, P. N. Suganthan, X. Yao, and A. K. Qin. Linear dimensionality reduction using relevance weighted lda. *Pattern Recognition*, 38:485, 2005. 60, 61

[75] M. Thangavelu and R. Raich. Multiclass linear dimension reduction via a generalized Chernoff bound. In *IEEE Workshop on Machine Learning for Signal Processing*, 2008. 19

[76] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cogn. Neurosci.*, 3:72C86, 1991. 66

[77] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, 2004. 12, 17

[78] Q. Wei, F. Meng, Y. Wang, and S. Gao. URL `http://ida.first.fraunhofer.de/projects/bci/competition$_$iii/results/index.html`. 92, 108

[79] J. Wilson, E. Felton, P. Garell, G. Schalk, and J. Williams. ECoG factors underlying multimodal control of a brain-computer interface. *IEEE transations on Neural Systems and Rehabilitation Engineering*, 14(2):246–250, 2006. 89

[80] C. Xiang and D. Huang. Feature extraction using recursive cluster-based linear discriminant with application to face recognition. *IEEE Transactions on Image Processing*, 15(12):3824–3832, Dec 2006. 29

[81] C. Xiang, X. A. Fan, and T. H. Lee. Face recognition using recursive Fisher linear discriminant. *IEEE Transactions on Image Processing*, 15(8):2097–2105, Aug 2006. 14, 17, 64

[82] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005. 34

[83] J. Yang and C. Liu. Color image discriminant models and algorithms for face recognition. *IEEE Transactions on Neural Networks*, 19(12):2088–2098, Dec 2008. 64

[84] J. T. yau Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 10:1018–1031, 1999. 62

[85] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001. 12, 16, 17

[86] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 454–9, 1998. 87

[87] W. Zheng, X. Zhou, C. Zou, and L. Zhao. Facial expression recognition using kernel canonical correlation analysis. *IEEE Transactions on Neural Networks*, 17 (1):233–8, Jan. 2006. 87

[88] J. Zou, Q. Ji, and G. Nagy. A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing*, 16(10):2617–2628, Oct 2007. 65

Author's Publications

Journal Papers

1. Xiang, C. and Huang, D. , "Feature Extraction Using Recursive Cluster-based Linear Discriminant with Application to Face Recognition," *IEEE Trans. on Image Processing*, v15, pages 3824-3832, 2006.

2. Huang, D., and Xiang, C., and Gu, W.F. and Ge, S.S.,"Recursive Cluster-based Bayesian Linear Discriminant for Pattern Recognition," submitted to *IEEE Trans. on Neural Networks*, 2009.

3. Gu Wenfei, and Xiang Cheng, and Venkatesh Yedatore, and Huang Dong, and Lin Hai, "Biologically Inspired Facial Expression Recognition using Radial Encoded Local Gabor Features and Classifier Synthesis," submitted to *IEEE trans. on Image Processing*, 2009.

Conference Papers

1. Huang, D. and Xiang, C. and Ge,S.S., "Recursive Fisher Linear Discriminant for BCI applications," in *2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing.* Melbourne, Australia: IEEE, Dec 2007, pp. 383-388.

2. Huang, D. and Xiang, C. and Ge,S.S., "Feature extraction for face recognition using recursive Bayesian linear discriminant," in *2007 5th International Symposium on Image and Signal Processing and Analysis*, 2007, pp. 299-304.

3. Huang, D. and Xiang, C., "Recursive Bayesian Linear Discriminant for Classification," in *Lecture Notes in Computer Science*, vol. 4492, *4th International Symposium on Neural Networks, ISNN 2007.* Nanjing, China, Jun 2007, pp. 1002-1011.

4. Huang, D. and Xiang, C., "A Novel LDA algorithm based on approximate error probability with application to face recognition," Atlanta, U.S.A.: *2006 IEEE International Conference on Image Processing*, 2006, pp. 653-656.

5. Xiang, C. and Huang, D., "Face Recognition Using Recursive Cluster-based Linear Discriminant," Proceedings of *2005 IEEE Seventh Workshop on Multimedia Signal Processing.* Shanghai, 30 Oct - 2 Nov 2005, Shanghai, China. pp. 401-405.

6. Xiang, C. and Huang, D., "Feature Extraction Using Recursive Cluster-Based Linear Discriminant with Application to Face Recognition," Proceedings of the *2005 IEEE Signal Processing Society Workshop on Machine Learning for Singal Processing.* Hilton Mystic, Connecticut, United States, 28 - 30 Sep 2005, pp. 123-128.