Feature extraction and dimensionality reduction in pattern recognition with applications in speech recognition

Jiang, Hai

2006

Jiang, H. (2006). Feature extraction and dimensionality reduction in pattern recognition with applications in speech recognition. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3441>

<https://doi.org/10.32657/10356/3441>

Nanyang Technological University

Downloaded on 26 Apr 2023 17:53:17 SGT

**Feature Extraction and Dimensionality Reduction
in Pattern Recognition with Applications in Speech
Recognition**

JIANG HAI

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

Statement of Originality

I hereby certify that the content of this thesis is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

Date

Jiang Hai

Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision of Associate Professor Er Meng Joo during this work. Without his encouragement, excellent guidance and support throughout the duration of this research and the development of this thesis, the accomplishment of this research would not be possible.

I am grateful to all staff and colleagues in the Instrumentation and System Engineering Laboratory at the School of Electrical and Electronic Engineering for their help and encouragement during the course of my study. I would like to acknowledge the support of the School of Electrical and Electronic Engineering, Nanyang Technological University and the University itself for providing me a university scholarship and logistic support.

Finally, I am forever indebted to my parents and wife for their understanding, endless patience and encouragement when it was most required.

Summary

Speech recognition has become a challenging task to create an intelligent recognizer that emulates a human being's ability in speech perception under all environments. The feature extraction of speech is one of the most important issues in the field of speech recognition. In order to achieve high recognition accuracy, the feature extractor is required to discover salient characteristics suited for classification. In this thesis, feature extraction methods and dimensionality reduction methods for feature space are examined. This thesis is divided into three parts. In the first part, speech recognition techniques are reviewed, and several linear and non-linear dimensionality reduction methods are investigated. In the second part, a new linear and a non-linear dimensionality reduction method are proposed in this thesis. In the last part, a new feature extraction technique for speech recognition is presented.

Chapter 2 gives an introduction to speech recognition in general. The fundamentals for building a speech recognition system are discussed. The fundamentals include digitization, preprocessing, feature extraction, vector quantization and classification. In Chapter 3, we introduce the basic underlying concepts, conventional techniques, and subsequent development of both supervised and

unsupervised dimensionality reduction methodologies for feature extraction and feature space projection.

In Chapter 4, after investigating the essence of current linear supervised mapping techniques based on scatter matrices, the concept of step weight is introduced to solve the problem that the optimal criterion of general linear supervised mapping techniques is not directly related to the accuracy in the output-space. Based on this concept, a Step-Weighted Linear Discriminant Analysis (SW-LDA) dimensionality reduction technique is proposed and tested on a set of speech recognition experiments. General linear mapping techniques are not sophisticated enough to deal with real-word applications especially such as speech recognition or image processing. Moreover, it is much more important to include non-linear properties of the feature space in linear mapping techniques for dimensionality reduction problems. Therefore, in Chapter 5, we extend the linear weighted discriminant analysis method to the Non-linear Weighted Discriminant Analysis (NWDA) method by using a kernel function operator. We derive the formulation of the proposed method using dot product in matrix form, and successfully develop the particular eigenvalue resolution for maximizing the optimized criterion. The proposed NWDA is applied on experiments based on the Fisher's iris data, the Leptograpsus Crabs data and the speech database. Some limitations and drawbacks of the proposed method are also discussed in the last part of this chapter.

Chapter 6 considers multiresolution feature extraction for speech recognition. In the proposed multiresolution feature extraction method, the new Mel-scale-like wavelet packets decomposition strategy is employed to obtain different resolution levels of the original signal. In the different subbands of the signal, the Linear Predictive Cepstral Coefficients (LPCCs) are calculated to capture the characteristics of each subband in our feature extractor. The conventional full-band

Mel-scale Frequency Cepstral Coefficients (MFCCs) are also combined with the subband features for maximizing the recognition accuracy. To reduce the high dimensionality of the new feature space, the linear dimensionality reduction strategy, namely the SW-LDA is employed to reduce the dimension of the feature space. The design and development of the multiresolution feature extractor will be discussed in details, and the performance of the proposed method will be evaluated by speech data set under static noisy condition. Comparative studies with other state-of-the-art methods show that our method is superior.

Contents

Acknowledgements	i
Summary.....	ii
List of Figures.....	vii
List of Tables	x
List of Abbreviations	xi
1. Introduction.....	1
1.1 Motivation.....	1
1.2 Objectives	6
1.3 Major Contributions of the Thesis	7
1.4 Outline of Thesis.....	8
2. Review of Speech Recognition System	11
2.1 Fundamentals of Speech Recognition.....	11
2.1.1 Digitization of Speech Signals.....	12
2.1.2 Preprocessing	13
2.1.3 Feature Extraction.....	17
2.1.4 Vector Quantization	22
2.2 Pattern Classification	24
2.2.1 Template-Based Approach	24
2.2.2 Statistical-Based Approach.....	26
2.2.3 Neural-Networks-Based Approach.....	37
3. Dimensionality Reduction in Pattern Recognition	42
3.1 Introduction.....	42

3.2 Dimensionality Reduction Techniques	47
3.2.1 Supervised Mapping Techniques	49
3.2.2 Unsupervised Mapping Techniques	51
4. Step-Weighted Linear Discriminant Analysis.....	56
4.1 Introduction.....	56
4.2 Review of Conventional LDA	58
4.3 Step-Weighted Linear Discriminant Analysis (SW-LDA)	60
4.4 Simulation Studies	64
4.5 Conclusions.....	68
5. Non-linear Weighted Discriminant Analysis.....	70
5.1 Introduction.....	70
5.2 Non-linear Weighted Discriminant Analysis.....	72
5.3 Simulation Studies	81
5.3.1 Fisher's Iris Data.....	81
5.3.2 Leptograpsus Crabs Data Set	87
5.4 Speech Recognition Experiments	93
5.4.1 Sampling Method.....	95
5.4.2 Experiment Result.....	101
5.5 Discussions and Conclusions.....	103
5.5.1 Discussions	103
5.5.2 Conclusions.....	104
6. Multiple Resolution Analysis in Speech Feature Extraction	106
6.1 Introduction.....	106
6.2 MFCCs and WP Transform-Based Features	109
6.2.1 Review of WP Transform	109
6.2.2 The Difference Between MFCCs and WP Transform-Based Parameters.....	111
6.3 Multiresolution Features Based on Mel-scale-like Wavelet Packets (WP) Decomposition	113
6.4 Emphasis of the Multiresolution Feature Extraction Method.....	116
6.5 Application to Speech Database	122
6.5.1 Database Description and Experimental Results	122
6.5.2 Discussions	126
6.6 Conclusions.....	128
7. Conclusions and Recommendations.....	130
7.1 Conclusions.....	130
7.2 Recommendations for Future Research.....	132
Author's Publications	134

Bibliography	136
---------------------------	------------

List of Figures

Figure 2.1: Block diagram of a standard word recognition system	12
Figure 2.2: The digital pre-emphasis filter	14
Figure 2.3: Segmentation of speech data	15
Figure 2.4: MFCCs feature extraction procedure	20
Figure 2.5: Dynamic time warping (a) alignment path; (b) local path constraints	25
Figure 2.6: A simple HMM with two states and two output symbols A and B.....	26
Figure 2.7: A five-state left-to-right HMM	27
Figure 2.8: Time-delay neural networks.....	39
Figure 4.1: Illustration of the pairwise problem by an example	61
Figure 4.2: Testing classification accuracy in reducing dimensionality from 32 to 4. The weighting function for the WLDA is $\omega_{kl} = d_{kl}^{-4}$ and for the SW-LDA is $\omega_{kl} = d_{kl}^{-12}$ respectively.....	67
Figure 5.1: Projection of iris data on the first two axes. The three classes are shown by various symbols.	82
Figure 5.2: 2D representations of iris data reduced by the LDA method.....	83
Figure 5.3: 2D representations of iris data reduced by the WLDA method with a weighting function (power=1).	83

Figure 5.4: 2D representations of iris data reduced by the WLDA method with a weighting function (power=2).	84
Figure 5.5: 2D representations of iris data reduced by the WLDA method with a weighting function (power=4).	84
Figure 5.6: 2D representations of iris data reduced by the GDA method with a Gaussian kernel ($\sigma = 0.7$).	85
Figure 5.7: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=1).	85
Figure 5.8: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=2).	86
Figure 5.9: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=4).	86
Figure 5.10: Projection of Leptograpsus Crabs data on the first two axes. The four classes are shown by various symbols.	88
Figure 5.11: 2D representations of Leptograpsus Crabs data reduced by the LDA method.....	88
Figure 5.12: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=1).	89
Figure 5.13: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=2).	89
Figure 5.14: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=4).	90
Figure 5.15: 2D representations of Leptograpsus Crabs data reduced by the GDA method with a Gaussian kernel ($\sigma = 0.7$).	90
Figure 5.16: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=1).....	91

Figure 5.17: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=2).....	91
Figure 5.18: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=4).....	92
Figure 5.19: 2D representations before sampling the points of class. Where h represents the width of zone for deleting points and H represents the width of zone for keeping points.	96
Figure 5.20: 2D representations after sampling the points of class	96
Figure 5.21: Projection of Leptograpsus Crabs data on the first two axes. The four classes are shown by various symbols and each contains 50 points.....	98
Figure 5.22: Projection of sampled Leptograpsus Crabs data on the first two axes when set $delt_n=3$ and $delt_d=3$. The number of points in Class 1 is 32, in Class 2 is 29, in Class 3 is 32 and in Class 4 is 32 after sampling.....	98
Figure 5.23: Projection of sampled Leptograpsus Crabs data on the first two axes when set $delt_n=5$ and $delt_d=5$. The number of points in Class 1 is 14, in Class 2 is 14, in Class 3 is 27 and in Class 4 is 14 after sampling.....	99
Figure 5.24: Projection of iris data on the first two axes. The three classes are shown by various symbols and each contains 50 instances.....	99
Figure 5.25: Projection of sampled iris data on the first two axes when set $delt_n=3$ and $delt_d=3$. The number of points in Class 1 is 26, in Class 2 is 27 and in Class 3 is 32 after sampling.....	100
Figure 5.26: Projection of sampled iris data on the first two axes when set $delt_n=5$ and $delt_d=5$. The number of points in Class 1 is 25, in Class 2 is 22 and in Class 3 is 19 after sampling.....	100

Figure 6.1: Wavelets packets decomposition tree. Here, $A(i, j)$ represents the i^{th} second level in depth and the j^{th} subspace in this level in the tree.....	110
Figure 6.2: Comparison of DFT-based and WPT-based frequency analysis for parameterization.....	112
Figure 6.3: Block diagram of the multiresolution feature extraction method	120
Figure 6.4: Detailed building blocks of Section II of Figure 6.3.....	121

List of Tables

Table 4.1: Testing classification accuracy on different weighting functions	66
Table 5.1: Comparison of error rates based on various methods under different SNR for the EH set	102
Table 6.1: Frequency bands for 4-subband Mel-scale-like WP Decomposition ..	115
Table 6.2: Frequency bands for 7-subband Mel-scale-like WP Decomposition ..	116
Table 6.3: Comparison of error rates based on various methods for English digits test.....	124

List of Abbreviations

A/D	Analog to digital
BP	Backpropagation networks
CMN	Cepstral Mean Normalization
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
FDA	Flexible Discriminant Analysis
FFT	Fast Fourier Transform
FLT	Frame Level Training
FT	Fourier Transform
GDA	Generalized Discriminant Analysis
GLA	Generalized Lloyd Algorithm
HDA	Heteroscedastic Discriminant Analysis
HMM	Hidden Markov Model
KFD	Kernel Fisher Discriminant
KFM	Koontz and Fukunaga Mapping
LDA	Linear Discriminant Analysis
LPCCs	Linear Predictive Cepstral Coefficients
LPC	Linear Predictive Coding
MDA	Mixture Discriminant Analysis

MFCCs	Mel-scale Frequency Cepstral Coefficients
MLPNN	Multi-Layer Perceptron Neural Networks
NLM	Non-Linear-Mapping
NWDA	Non-linear Weighted Discriminant Analysis
PCA	Principal component Analysis
PLP	Perceptual Linear Prediction
PNN	Probabilistic Neural Networks
RDA	Regularized Discriminant Analysis
SBS	Simple Backward Selection
SFS	Simple Forward Selection
SLT	Segment Level Training
SW-LDA	Step-Weighted Linear Discriminant Analysis
SOM	Self-Organizing Map
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SNR	Signal-to-Noise Ratio
TDNN	Time-Delay Neural Networks
TI	Texas Instruments
VQ	Vector Quantization
WLT	Word Level Training
WLDA	Weighted LDA
WP	Wavelet Packets
WPT	Wavelet Packet Transform

Chapter 1

Introduction

1.1 Motivation

Speech is the most natural form of human communication and speech processing has been one of the most exciting areas in signal processing. In the last several decades, speech research has resulted in much technological impact on society. Speech recognition i.e. the ability to recognize spoken words provides an effective and powerful communication method for human beings. At present, we are using it so excessively in our everyday life that we do not notice the complex and intricate mechanism behind it. Speech recognition technology has made it possible for computers to follow human voice commands and even understand human languages. As computers become faster and more ubiquitous, these and other areas in speech processing are expected to flourish further and bring about an era of true human-computer interactions. Speech production is a highly non-linear process involving the cooperation of our vocal apparatus including mouth, tongue, nostril and trachea and so on. Speech production is also strongly influenced by factors

such as age, gender, fatigue level, health conditions and so on. Speech perception is even more complex since it involves a high degree of variability due to additional background noise, different room acoustics and transmission characteristics in the case of telephone lines or voice-over-IP. Despite this immense variability, human beings are able to use this form of communication very well even in adverse environments such as noisy parties.

Research in automatic speech recognition by machine has been carried out for almost four decades. The earliest attempts to devise systems for automatic speech recognition by machines were made in the 1950s, when various researchers tried to exploit fundamental ideas of acoustic phonetics. The research areas cover every aspect from signal processing, physics (acoustics), pattern recognition, communication and information theory to linguistics, physiology, computer science and psychology.

In 1952, at Bell Laboratories, Davis, Biddulph and Balashek built a system for isolated digit recognition for a single speaker [1]. In the 1970s, speech-recognition research achieved a number of significant milestones. First, the area of isolated-word or discrete-utterance recognition became a viable and usable technology based on fundamental studies by Velichko and Zagoruyko in the Soviet Union [2], Sakoe and Chiba in Japan [3], and Itakura in the United States [4]. Another milestone was the beginning of longstanding and highly successful group efforts in large vocabulary automatic speech dictation at IBM. Finally, at AT&T Bell Laboratory, researchers began a series of experiments aimed at making speech recognition systems that were truly speaker-independent for telecommunication applications [5].

Speech research in the 1980s was characterized by a shift in technology from template-based approaches to statistical modeling methods, especially the HMM approach [6], [7]. The statistical approach to speech recognition relies heavily on the training data that are available for creating reference models. What is often observed in speech-recognition applications is that a recognizer designed on a data set in the laboratory does not perform as well in the field. In other words, a mismatch between the modeling (training) and the operating (testing) conditions usually exists and causes degradation in the recognizer's performance.

Besides the mismatch, several adverse conditions are also often present during operation, such as ambient and transmission noise, distortions due to room acoustics and transducers, and even changes in speech characteristics due to psychological awareness of talking to a machine. These conditions need to be dealt with in order for the recognizer to be able to deliver reliable results. This is the so-called "robustness" problem in automatic speech recognition.

One method that achieves robust results is to collect an extremely large amount of data that reflect the actual operating conditions of the recognizer; another important method is to extract the most representative feature from the speech utterance in bad environment conditions. The feature extraction of speech is one of the most important issues in the field of speech recognition. In order to achieve high recognition accuracy, the feature extractor is required to discover salient characteristics suited for classification.

There are two dominant feature extraction methods in speech recognition. One occurs in the temporal domain and it is a parametric modeling approach. This kind of method is developed to match closely the resonant structure of the human vocal tract that produces the corresponding speech sound. It is mainly derived from

linear predictive analysis and the Linear Predictive Coding (LPC) method is a representative. The other occurs in the frequency domain and it is the nonparametric modeling method originated from the human auditory perception system. Mel-scale Frequency Cepstral Coefficients (MFCCs), which are based on Fast Fourier Transform (FFT) are utilized for this purpose. The MFCCs feature extraction method was widely adopted in many popular speech recognition systems by many researchers, such as M.N. Stuttle and M.J.F. Gales in Cambridge University of United Kingdom [8], I. Potamitis and N. Fakotakis in University of Patras of Greece [9], Wei-Wen Hung and Hsiao-Chuan Wang in Taiwan [10], Shang-Ming Lee and Shi-Hau Fang [11], Alain Biem and Shigeru Katagiri [12] and Jia Lei [13], etc.

Some researchers have designed new optimal feature extraction methods, for example, M. Chetouani uses neural predictive coding method which is an extension of linear predictive coding method by modeling non-linear speech signals [14] and Satya Dharanipragada uses minimum variance distortionless response spectrum based modeling of speech in [15].

In feature extraction procedure, it is also very important to keep the dimensionality of the feature vector as small as possible for the reason of measurement cost. Moreover, a small number of features can alleviate the curse of dimensionality when the number of training samples is limited. In order to achieve high recognition accuracy, the feature extractor is required to discover salient characteristics and include more useful information. However, including more information will result in increasing the dimensionality of the feature spaces, the number of classes, the size of the training sets and even the number of unimportant information. The higher the dimensionality of the feature space, the more data may be needed to find out what is important and what is not. Therefore, the

performance of the resulting classifiers may degrade with the increase of the dimensionality of the feature space. To effectively reduce the dimensionality of the feature vector and keep more discriminant information, some modern speech recognizers employed transform methods such as Discrete Cosine Transform (DCT)[16],[17], Principal Component Analysis (PCA) [18]-[20] method and Linear Discriminant Analysis (LDA) [21].

In spite of these developments, effective feature extraction is still far from being a solved problem. The LPC method is not optimal because the underlying speech production model is non-linear. LPC starts with the assumption that the speech signal is produced by a buzzer at the end of a tube. For ordinary vowels, the vocal tract is well represented by a single tube. However, for nasal sounds, the nose cavity forms a side branch. In practice, this difference is partly ignored and partly dealt with during the encoding of the residue.

Most of the speech recognition systems use MFCCs for phoneme recognition. Essentially, in all of these computations, Fourier Transform (FT) is used. It is a well-known fact that the windowed FT or the Short-Time Fourier Transform (STFT) has uniform resolution over the time-frequency plane. Because of this, it is difficult to detect sudden burst in a slowly-varying signal by using STFT. This phenomenon is observed in phoneme recognition when “stops” are encountered.

Of the dimensionality reduction techniques, since PCA does not use class information, linear supervised mapping technique such as LDA usually outperforms PCA for pattern classification and dimension reduction. However, there are still some significant difficulties in the linear supervised mapping method. One of the drawbacks is that the optimal criterion based on scatter matrices in general is not directly related to classification accuracy. Moreover, it is still a

linear analysis method and it fails for a non-linear problem especially in complicated speech and image recognition applications.

Since these problems are faced by the feature extraction method and the dimension reduction approaches, we have decided to focus our research in this area.

1.2 Objectives

The main objectives of this research project are to propose improved dimensionality reduction methods and feature extraction strategy for speech recognition systems. We focus on solving the problems in the following three main areas.

1. In the linear supervised mapping techniques based on scatter matrices, the optimal criterion is not directly related to classification accuracy. The scatter matrix is calculated in the input-space and the true value of the scatter matrix in the output-space is far from correct when the dimensionality is reduced.
2. General linear mapping techniques are not sophisticated enough to deal with real-world applications. Meanwhile, we could have a lot more classes than just two and there are very different numbers of points in each class. Therefore, it is much more important to include non-linear properties of the feature space in linear mapping techniques for classification and dimensionality reduction problems.
3. Speech signals have very complex waveforms because of the composition of various frequency components. How to define a representation that is well

adapted for extracting the information content of speech signals is an important problem in speech recognition. Classical feature extraction methods are carried out by computing acoustic feature vectors over the full frequency band of the spectral representation of speech. The major drawback of this approach is that even partial band-limited noise corruption affects all feature vector components.

Concepts referring to pattern recognition and speech recognition in Chapter 1 will be explained more fully in the later chapters.

1.3 Major Contributions of the Thesis

There are three major contributions in the thesis. First, a concept of step weight is introduced to solve the problem that optimal criterion of general linear supervised mapping techniques is not directly related to the accuracy in the output-space after the dimensionality is reduced. Based on this concept, a Step-Weighted Linear Discriminant Analysis (SW-LDA) dimensionality reduction technique is presented, and it is tested on a set of speech recognition experiments.

Second, we extend the linear weighted discriminant analysis method to Non-linear Weighted Discriminant Analysis (NWDA) method by using a kernel function operator. The formulation of the proposed NWDA using dot product and matrix formulation is described in details and the particular eigenvalue resolution for maximizing the optimal criterion is successfully developed. The NWDA is applied on dimensionality reduction experiments based on the Fisher's iris data and the Leptograpsus Crabs data, and we attempt to utilize it in a simple speech recognition system.

Finally, a multiresolution analysis method for speech feature extraction is proposed. Mel-scale-like Wavelet Packets (WP) decomposition is employed to obtain different resolution levels of the original signal. In the different subbands of the signal, Linear Predictive Cepstral Coefficients (LPCCs) are calculated to capture the characteristic of each band in our extraction algorithm. The conventional MFCCs are also combined with the subband features for maximizing the recognition accuracy. In order to reduce the high dimensionality of the new feature space, SW-LDA, the linear dimensionality reduction strategy, is employed in the proposed method.

1.4 Outline of Thesis

This thesis is organized into seven chapters.

- Chapter 1 gives motivations and contributions of the thesis and gives a brief outline of each chapter in the thesis.
- Chapter 2 introduces the foundation of implementing a general speech recognition system. The fundamentals of speech recognition and different approaches adopted by other researches are reviewed. The main blocks of a standard recognition system including digitization of speech signal, preprocessing, feature extraction and pattern classification.
- Chapter 3 gives an overview of the basic underlying concepts, some conventional techniques and the subsequent development of dimensionality reduction techniques, which are commonly used for feature extraction and feature space projection. There are several ways to reduce the dimension of

the feature space. Feature selection could be the easiest and feature extraction is another common way. We mainly discuss two families of dimensionality reduction techniques, namely supervised mapping techniques and unsupervised mapping techniques.

- Chapter 4 investigates deficiencies of linear dimension reduction methods based on the classical LDA and the weighted LDA. Subsequently, we present the development of the proposed SW-LDA to deal with the deficiencies. The SW-LDA algorithm for linear dimension reduction is described in details. Finally, experiments performed on the speech recognition to verify the efficiency of the proposed SW-LDA. Compared with the general weighted LDA, the proposed SW-LDA provides a better lower dimensional representation.
- Chapter 5 presents the NWDA method for non-linear discrimination analysis and dimensionality reduction. The NWDA concentrates on two drawbacks of classical LDA. The first one is that between-class scatter matrix in LDA assumes that each class is equally confusable with all other classes. The second drawback is that the LDA cannot deal with non-linear discriminant information. In the simulation studies, the Fisher's iris data and the Leptograpsus Crabs data set are used to perform dimensionality reduction. The proposed method is employed in confusable English letter recognition task and some issues are discussed.
- Chapter 6 presents a new and robust multiresolution speech feature extraction method based on time-frequency multiresolution analysis. The Mel-scale-like WP decomposition is applied to obtain various uncorrelated frequency channels. By using the SW-LDA linear discriminant

dimensionality reduction technique, more discriminant information is retained and lower dimensionality is attained. Subsequently, simulation studies of the proposed method on speech recognition experiments demonstrate the efficiency of the method for feature extraction.

- Chapter 7 concludes the thesis and suggests several promising directions for future research based on the results presented in this thesis.

Chapter 2

Review of Speech Recognition System

In this chapter, we present a brief review of speech recognition system including the fundamentals to speech recognition and different approaches adopted by other researches.

2.1 Fundamentals of Speech Recognition

Speech recognition can be considered as a multi-level pattern recognition task where acoustical signals are examined into a hierarchy of sub-word units (e.g. phonemes, words, phrases, and sentences). The block diagram of a standard word recognition system is shown in Figure 2.1.

Figure 2.1: Block diagram of a standard word recognition system

2.1.1 Digitization of Speech Signals

Speech digitization is simply recording audible input speech of a user, converting it into electrical pulses and storing it in digital format. This is accomplished by using a microphone and a sound card, which acts as an analog to digital (A/D) converter.

Although digital storing of speech is much more accurate and has high fidelity, speech storing was as in the analog format used in early sound recording. Digitally stored speech has a primary advantage over analog speech because digitally stored speech is the only kind that can be stored and processed by a computer.

When recording, audio data is characterized by the following parameters, which correspond to settings of the A/D converter when the data was recorded. Naturally, the same settings must be used to play the data.

- Sampling rate (in samples per second), e.g. 8000 or 44100
- Number of bits per sample, e.g. 8 or 16
- Number of channels (1 for mono, 2 for stereo, etc.)

Approximate sampling rates are often quoted in hertz or kilohertz. However, the technically correct term is samples per second (samples/sec). Sampling rates are always measured per channel, so for stereo data recorded at 8000 samples/sec, there are actually 16000 samples from both channels in a second.

Multi-channel samples are generally interleaved on a frame-by-frame basis. If there are N channels, the data is a sequence of frames, where each frame contains N samples, one from each channel. For stereo, the left channel usually comes first.

2.1.2 Preprocessing

Raw digitized speech should be initially preprocessed to simplify subsequent processing. Typically, preprocessing includes three sub-modules, namely pre-emphasis, windowing and end-point detection. The output of this module is a sequence of frames each of which containing the same amount of preprocessed raw data.

(a) Pre-emphasis

According to the source-filter model of [22], the speech produced by a human being has a characteristic of 6 dB of energy loss as the frequency is doubled. Therefore, in order to obtain a good dynamic range along the spectrum, digitized speech signal needs to compensate for this loss. One of the widely used method is to let the digitized speech signal pass through a low-order digital system (for example, a first-order FIR filter) to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing stage. The most widely used pre-emphasis network is the first-order filter which is given by

$$H(z) = 1 - az^{-1}, \quad 0.9 \leq a \leq 1.0 \quad (2.1.1)$$

where $H(z)$ is the transfer function of the filter and z^{-1} is the delay operator. We choose $a = 0.95$ in our system.

The pre-emphasis filter can be achieved digitally as shown in Figure 2.2. In this case, the output of the pre-emphasis network, $\tilde{s}(n)$ is related to the input of the network, $s(n)$ by the difference equation

$$\tilde{s}(n) = s(n) - 0.95 \times s(n-1) \quad (2.1.2)$$

Figure 2.2: The digital pre-emphasis filter

(b) Windowing

Firstly, the pre-emphasized speech signal $\tilde{s}(n)$ is organized into frames of N samples, with adjacent frames being separated by M samples. Figure 2.3 illustrates the blocking into frames for the case where $M = \frac{1}{2} \times N$. The first illustrated frame consists of first N speech samples. The second frame begins M samples after the first frame, and overlaps it by $(N - M)$ samples. Similarly, the third frame begins $2M$ samples after the first frame (or M samples after the second frame) and overlaps the second frame by $(N - M)$ samples. This process continues until each of the speech data is organized into one or more frames.

Figure 2.3: Segmentation of speech data

It is easy to see that if $M \leq N$, then adjacent frames overlap as shown in the figure, and the resulting feature parameters obtained from each frame will be correlated from frame to frame. On the other hand, if $M > N$, there will be no overlap donor between adjacent frames. In fact, some of the speech signals will be totally lost (i.e., never appear in any analysis frame), and the correlation between the resulting feature parameters of adjacent frames will contain an incorrect component whose magnitude increases as M increases (i.e. as more speech is omitted from the analysis). This situation is intolerable in any practical feature extraction process for speech recognition.

If we denote the k^{th} frame of the speech signal by $x_k(n)$ and the total number of samples in the speech signal by L and assume that there are K frames within the entire speech signal, we have

$$x_k(n) = \tilde{s}(Mk + n), \quad n = 0, 1, \dots, N-1, \quad k = 0, 1, \dots, K-1, \quad \text{if } Mk + n \leq L$$

or

$$x_k(n) = 0, \quad \text{if } Mk + n > L$$

That is, the first frame of speech, $x_0(n)$, encompasses speech samples $\tilde{s}(0)$, $\tilde{s}(1)$, $\dots$, $\tilde{s}(N-1)$, the second frame of speech, $x_1(n)$, encompasses speech samples $\tilde{s}(M)$, $\tilde{s}(M+1)$, $\dots$, $\tilde{s}(M+N-1)$, the K^{th} frame of speech, $x_{K-1}(n)$, encompasses speech samples $\tilde{s}(M(K-1))$, $\tilde{s}(M(K-1)+1)$, $\dots$, $\tilde{s}(M(K-1)+N-1)$. If the samples are not sufficient to fill all the K frames, zeros are padded to the signal.

The general values for N and M are 300 and 100 respectively when the sampling rate of speech is 6.67 kHz. These correspond to 45 msec frames, separated by 15 msec.

The second step in the windowing sub-module is to window each individual frame so as to minimize signal discontinuities at the beginning and the end of each frame. The concept here is to use the window to taper the signal to zero at the beginning and the end of each frame. If we define the window as $w(n)$, $0 \leq n \leq N-1$, then the result of windowing is the signal given by

$$\tilde{x}_k(n) = x_k(n)w(n), \quad 0 \leq n \leq N-1 \quad (2.1.3)$$

A typical window used most widely for speech recognition systems is the Hamming window, which can be described by the following equation

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N, \quad N > 1 \quad (2.1.4)$$

where n is the sample number and N is the total number of samples per window.

(c) End-point Detection

If recording is done using a tape recorder, there exist undesired silence frames during the recording process. These silent frames not only occupy space but also

provide no information about the speech. Hence, end-point detection is applied to capture the starting and ending points of the recorded speech though it is not an easy task to determine the points since sound artifacts that appear as false start or false end of the speech exist.

There are some typical end-point detection algorithms:

- Energy-based algorithms [23]. They are intuitive approaches based on energy levels and durations of silence and speech. Sometimes, several pairs of boundaries are produced in order for their rank to be correct.
- Pitch-information-based algorithms [24], [25]. Pitch information is extracted directly from the speech waveform to detect boundaries of spoken words.
- Adaptive noise cancellation algorithms. The signal energy, the zero-crossing rate, and duration information are used to detect word boundaries. The thresholds used for energy and zero-crossings are adapted automatically from a few frames provided by the signal environment.
- Frequency-features-based algorithms [26], [27]. Some frequency analysis such as Fast Fourier Transform (FFT) is applied to raw speech data. Features obtained from the frequency domain are used to detect boundaries of spoken words.

2.1.3 Feature Extraction

Preprocessed data frames are fed into this module so that they are transformed and compressed. Many signal analysis techniques, which can extract useful features and compress the data by a factor of ten without losing any important information, are available. The most popular techniques are Linear Predictive Coding (LPC)

and Mel-scale Frequency Cepstral Coefficients (MFCCs). This module produces a sequence of feature vectors extracted from the preprocessed data.

(a) Linear Predictive Coding (LPC) Analysis

The theory of LPC, as applied to speech, has been well understood for many years [28]. The LPC method has been widely used in many successful speech recognition systems [29], [30]. It has several well-known properties:

- LPC provides a good model of the speech signal. This is especially true for quasi steady-state voiced regions of speech where the all-pole model of LPC provides a good approximation to the vocal tract spectral envelope. During unvoiced and transient regions of speech, the LPC model is less effective than in voiced regions, but it still provides an acceptably useful model for speech-recognition purposes.
- The way in which LPC is applied to analysis of speech signals leads to a reasonable source-vocal tract separation. As a result, a parsimonious representation of the vocal tract characteristics (which we know are directly related to the speech sound being produced) becomes possible.
- LPC is an analytically tractable model. The method of LPC is mathematically precise, simple and is straightforward to implement in either software or hardware. The computation involved in LPC processing is considerably less than other methods such as bank-of-filters model and wavelet transformation.
- The LPC model works well in recognition applications. Experience has shown that the performance of speech recognizers, based on LPC front ends, is comparable to or better than that of recognizers based on filter-bank front ends [31].

The idea behind the LPC is that a given speech signal $s(n)$ can be approximated by a linear combination of past samples. This can be described by the following equation:

$$s(n) \approx \sum_{k=1}^p a(k)s(n-k) \quad (2.1.5)$$

where $s(n)$ is the actual speech signal, p is the order of the predictor and $a(k)$ are the predictor coefficient.

The total squared prediction error is:

$$E = \sum_n \left(s(n) - \sum_{k=1}^p a(k)s(n-k) \right)^2 \quad (2.1.6)$$

The main objective of LPC is to determine the coefficients $a(k)$ for each speech frame so that minimize (2.1.6). The coefficients $a(k)$ can be obtained by setting the partial derivatives of (2.1.5) with respect to $a(k)$ to zero. This leads to so called Yule-Walker equations that can be efficiently solved using Levinson-Durbin recursion [32].

Linear Predictive Cepstral Coefficients (LPCCs) [33] which are deriving from LPC are one of feature sets that are widely used for speech processing up to date [34]-[37]. Given the LPC coefficients $a(k)$, $k=1, \dots, p$, the LPCC coefficients are computed using the recursion [38]:

$$c(n) = \begin{cases} a(n) + \sum_{k=1}^{n-1} \frac{k}{n} c(k)a(n-k), & 1 \leq n \leq p \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c(k)a(n-k), & n > p \end{cases} \quad (2.1.7)$$

(b) Mel-scale Frequency Cepstral Coefficients (MFCCs)

MFCCs is well known as a feature extraction method for speech recognition systems, it was introduced in 1980 by Davis and Mermelstein [39]. It uses a filterbank based on the human auditory system: spacing filters in frequency based on the Mel-frequency scale to reshape and resample the frequency axis. A logarithm of each channel models loudness compression. Then a low-dimensional representation is computed using the DCT. The DCT not only removes high-frequency ripples in the spectrum, but serves to decorrelate the coefficients. MFCCs have proven to be one of the most effective set of features for speech recognition. For example, it has been introduced as part of the ETSI standard front-end [40].

The stages involved in creating a stream of MFCCs vectors are illustrated in Figure 2.4.

Figure 2.4: MFCCs feature extraction procedure

Assuming that $y(n)$ is the input speech signal, the complete calculation process of the coefficients can be described step by step as follows:

Step 1: Transform the input speech signal from time domain to frequency domain by applying short-time Fast Fourier Transform (FFT) method. The transformed input speech signal $Y(m)$ is given by:

$$Y(m) = \sum_{n=0}^{F-1} y(n)W(n)e^{-j2n\pi\frac{m}{F}} \quad (2.1.8)$$

where $m = 0, 1, 2, \dots, F-1$, F is frame size, which is generally equal to the power of 2, $W(n)$ is the Hamming window function, which is based on the fact that the signal can be regarded as stationary and uninfluenced by the others within a short period of time, i.e. the frame size.

Step 2: Find the energy spectrum $X(m)$ according to

$$X(m) = |Y(m)|^2 \quad (2.1.9)$$

Step 3: Calculate the energy $S[k]$ in each Mel window according to

$$S[k] = \sum_{j=0}^{k/2-1} W_k(j)X(j) \quad (2.1.10)$$

where $1 \leq k \leq M$, M is the number of the Mel windows in Mel-scale, which ranges from 20 to 24 generally and $W_k(j)$ is the triangular weighted function associated with the k^{th} Mel window in Mel-scale.

Step 4: Proceeding with logarithm and cosine transforms, we can compute the Mel-scale frequency cepstral coefficients $C[n]$ as follows:

$$C[n] = \sum_{k=1}^M \text{Log}(S[k]) \cos \left[n(k-0.5) \frac{\pi}{M} \right] \quad (2.1.11)$$

where $1 \leq n \leq L$ and L is the desired order of MFCCs.

2.1.4 Vector Quantization

The results of MFCCs are a series of vectors which capture the time-varying characteristics of the detected spoken-word speech signal. The n -dimensional speech feature vectors may be considered as representing the points within an n -dimensional Euclidean space. One of the most obvious ways whereby we may establish a measure of similarity between or among such feature vectors is by means of their proximities to one another.

Vector Quantization (VQ) is a mapping of input data to one of the finite number of prestored codevectors (or codewords) in a codebook [41], [42]. It is a powerful data compression technique used in a number of signal processing applications including speech coding, image coding, and speech recognition [43]-[46]. The goal in VQ design is to obtain an optimal codebook that minimizes the average distortion introduced by the approximation of each input data to one of the codevectors. Such an optimal quantizer must satisfy two necessary conditions, the nearest neighbor condition for the partition and centroid condition for the codevectors.

To date, the most widely used algorithm to design a VQ codebook is the K-means algorithm. It is also known as the Generalized Lloyd Algorithm (GLA), or sometimes called the Linde-Buzo-Gray (LBG) algorithm. It updates an initial codebook and satisfies the two necessary conditions iteratively.

Some nomenclatures are established before presenting the K-means algorithm. Let $X^{(p)}$ represent the p -th input feature vector. The complete set of feature vectors will then be $\{X^{(1)}, X^{(2)}, \dots, X^{(p)}\}$. The vector Z represents the cluster center for each of the K clusters. That is, it points to the position in the Euclidean space at

which the cluster center is located. Since there are K clusters, there are also K cluster centers, namely $Z_1, Z_2, \dots, Z_k$. Finally, the notation $S_j = \{X | X \text{ is close to cluster } j\}$ will be used to represent the set of samples that belongs to the j -th cluster center.

The algorithm is described as follows:

Step 1: Choose the number of clusters, K , and choose first the sample vectors as initial cluster centers: $\{Z_1(d), Z_2(d), \dots, Z_k(d)\}$, where $Z_j(d)$ is the value of the cluster center at the d -th iteration.

Step 2: Distribute all sample vectors. Each sample vector $X^{(p)}$ is attached to one of the K clusters according to the following criterion $X^{(p)} \in S_j(d)$, if $\|X^{(p)} - Z_j(d)\| < \|X^{(p)} - Z_i(d)\|$, for all $i=1, 2, \dots, K, i \neq j$, where $S_j(d)$ represents the population of cluster j at iteration d .

Step 3: Using the new cluster membership sets established in the previous step, recalculate the position of each cluster center such that the sum of the distances from each member vector to the new cluster center is minimized. Specifically, we wish to minimize J_j :

$$J_j = \sum_{X^{(p)} \in S_j(d)} \|X^{(p)} - Z_j(d+1)\|^2 \quad j=1, 2, \dots, K \quad (2.1.12)$$

The value of $Z_j(d+1)$ which minimize equation (2.1.12) is simply the mean taken over the samples of $S_j(d)$. Therefore, the new cluster center is calculated as follows:

$$Z_j(d+1) = \frac{1}{N_j} \sum_{X^{(p)} \in S_j(d)} X^{(p)} \quad (2.1.13)$$

where N_j is the number of sample vectors attached to S_j .

Step 4: If there are any position changes for K cluster centers, return to the Step 2. Otherwise, the algorithm ends.

2.2 Pattern Classification

This is the heart of any speech recognition system. Speech features are classified in this module. There are basically three kinds of approaches to pattern classification, namely template-based approach, statistical-based approach and neural-networks-based approach. The following section will introduce all these three techniques.

2.2.1 Template-Based Approach

The simplest way to recognize an isolated word sample is to compare it with a number of stored word templates and determine which is the best match. This goal is complicated by a number of factors. First, different samples of a given word will have somewhat different durations. This problem can be eliminated by simply normalizing the templates and the unknown speech so that they all have an equal duration. Another problem is that the rate of speech may not be constant throughout the word. In other words, the optimal alignment between a template and the speech sample may be non-linear. Dynamic Time Warping (DTW), one of the oldest and the most important algorithms in speech recognition, is an efficient method for finding this optimal non-linear alignment [3].

DTW is an instance of a general class of algorithms known as dynamic programming. Its time and space complexity are merely linear in the duration of speech sample and vocabulary size. The algorithm makes a single pass through a

matrix of frame scores while computing locally optimized segments of the global alignment path. Figure 2.5 shows the basic idea behind DTW.

If $D(x, y)$ is the Euclidean distance between frame x of the speech sample and frame y of the reference template, and if $C(x, y)$ is the cumulative score along an optimal alignment path that leads to (x, y) , then

$$C(x, y) = \text{Min}(C(x-1, y), C(x-1, y-1)) + D(x, y) \quad (2.2.1)$$

The resulting alignment path may be visualized as a low valley of Euclidean distance scores, meandering through the hilly landscape of the matrix, beginning at $(0,0)$ and ending at the final point (X,Y) . By keeping track of back pointers, the full alignment path can be recovered by tracing backward from (X,Y) . An optimal alignment path is computed for each reference word template, and the one with the lowest cumulative score is considered to be the best match for the unknown speech sample.

Figure 2.5: Dynamic time warping (a) alignment path; (b) local path constraints

2.2.2 Statistical-Based Approach

In this approach, variations in speech are modeled statistically using automatic learning process. This approach represents the current state-of-art methodology. The most flexible and successful technique being used widely is the Hidden Markov Model (HMM) [7].

An HMM is a collection of states connected by transitions, as depicted in Figure 2.6. It begins in a designated initial state. In each discrete time step, a transition is taken into a new state, and then one output symbol is generated in that state. The choice of transitions and output symbols are both random, governed by probability distributions. The HMM can be thought of as a black box, where the sequence of output symbols generated over time is observable. But, the sequence of states visited over time is hidden. This is why it is called a Hidden Markov Model.

Figure 2.6: A simple HMM with two states and two output symbols A and B

HMM has a variety of applications. When an HMM is applied to speech recognition, the states are interpreted as acoustic models, indicating what sounds are likely to be heard during their corresponding segments of speech while the transitions provide temporal constraints, indicating how the states may follow each other in sequence. Because speech always goes forward in time, transitions in

speech applications always go forward (or make a self-loop, allowing a state to have arbitrary duration). An HMM can be structured or deployed in a hierarchical way at different levels (e.g. phonemes, words, and sentences) within the recognizer providing a coherent framework for a complete recognition system.

To be able to apply the HMM as the commonly used speech recognition system, there are three major components to be considered:

- A number of finite states used to describe an utterance, i.e. an ergodic or fully connected HMM to allow any state of the HMM to be reached by any other states. This could be simply a left-to-right HMM;
- The transition probability matrix that represents the probability function of the transition among states in the HMM;
- The observation symbol probability matrix that represents the probability for all possible symbols in each transition state.

Figure 2.7: A five-state left-to-right HMM

Figure 2.7 shows a five-state left-to-right HMM. The HMM states and transitions are represented by nodes and arrows, respectively. There are five states in the HMM model. Thus, a state matrix of 5×5 elements can be identified. Since this model is not an ergodic model, only 9 transition cases (arrows) exist. Therefore, a 5×5 states matrix with 9 non-zero elements is identified. This state matrix is

usually called the A matrix. In each time interval, for any transitions to exist, an observation symbol will be emitted. These symbols are used for the representation of a speech signal. Generally speaking, these are the symbols for the acoustic feature vectors of speech utterances.

There is a probability for each observation symbol associated with each state transition. For example, in a transition that has 128 possible observations, the state matrix will be size of 9×128 . In HMM, this matrix is called the B matrix. The B matrix, can be in the form of a discrete distribution or a continuous distribution and is also known as the discrete HMM and continuous HMM, respectively.

In the training mode, the task is to find an HMM model that can best describe an utterance (observation sequence) with parameters associated with the state model A and B matrix. A maximum likelihood is computed among the test utterance and the HMMs, and the one with the highest probability value is considered as the candidate of the matched pattern.

(a) Discrete Hidden Markov Model

A discrete HMM λ is characterized by the following parameters and these are defined as follow:

- N — The number of states in the model. This parameter is pre-defined at the beginning of the problem. For example, for a phonetic recognition system, the HMM model will have at least three states corresponding to the initial state, the steady state and the ending state.
- M — The number of distinct observation symbols per state is considered as the feature vector sequence of the model. Naturally, this number can be

large because the speech utterances are continuous. As for the discrete HMM, a limited number of distinct symbols is reached via VQ.

- $A = \{a_{ij}\}$ — The state transition probability distribution, where

$$a_{ij} = P[q_{t+1} = j | q_t = i] \quad 1 \leq i, j \leq N \quad (2.2.2)$$

is the probability of state for transition i to j .

- $B = \{b_j(k)\}$ — The observation symbol output probability distribution, where

$$b_j(k) = P[o_t = v_k | q_t = j] \quad 1 \leq j \leq M \quad (2.2.3)$$

is the output probability of symbol v_k emitted by state j .

- $\pi = \{\pi_i\}$ — The initial state distribution, where

$$\pi_i = P[q_1 = i] \quad (2.2.4)$$

After defining the necessary five parameters that are required for the HMM, we now require computing the probability distribution A , B and π , i.e.

$$\lambda = (A, B, \pi) \quad (2.2.5)$$

The number of states and the number of distinct symbols are hidden but implicitly used within the model.

Considering the following simple isolated word speech recognizer, for each word of a W word vocabulary, we want to design a separate N -state HMM. At first, we represent the speech signal of a given word as a time sequence of coded spectral vectors. We assume that the coding is done using a spectral codebook with M unique spectral vectors; hence each observation is the index of the spectral vector closest to the original speech signal.

For each vocabulary word, we have a training sequence consisting of a number of repetitions of sequences of codebook indices of the word. Now, the first task is to build individual word models. In other word, it is to optimally estimate model parameters for each word model. The problem 1 is: how do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

To develop an understanding of the physical meaning of the model states, we want to segment each of the word training sequences into states, and then study the properties of the spectral vectors that lead to the observations occurring in each state. Now, the problem 2 is: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and the model λ , how do we choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal in some meaningful sense?

Finally, once the set of W HMMs has been designed and optimized and thoroughly studied, recognition of an unknown word is performed to score each word model based on the given test observation sequence, and select the word whose model score is highest. Therefore, the problem 3 is: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and the model $\lambda = (A, B, \pi)$, how do we efficient compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

To solve the problem 3, the forward-backward procedure is utilized.

The Forward-Backward Procedure [47] [48]: Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda) \quad (2.2.6)$$

We can solve for $\alpha_t(u)$ inductively, as follows:

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N. \quad (2.2.7)$$

2) Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1$$

$$1 \leq j \leq N \quad (2.2.8)$$

3) Termination:

$$P(O|\lambda) = \sum_{j=1}^N \alpha_T(j). \quad (2.2.9)$$

We only need the forward part of the forward-backward procedure to solve problem 3.

In a similar manner, we can consider a backward variable $\beta_t(i)$ defined as

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \lambda) \quad (2.2.10)$$

We can solve for $\beta_t(i)$ inductively, as follows:

1) Initialization:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \quad (2.2.11)$$

2) Induction:

$$\beta_t(i) = \sum_{j=1}^N \alpha_{ij} b_j(O_{t+1}) \beta_{t+1}(j),$$

$$t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \quad (2.2.12)$$

The backward procedure will be used in the solution to problem 1.

Unlike problem 3, there are several possible ways of solving problem 2, namely finding the “optimal” state sequence associated with the given observation sequence.

The most widely used criterion is to find the single best state sequence (path), i.e., to maximize $P(Q|O, \lambda)$ which is equivalent to maximizing $P(Q, O|\lambda)$. A formal technique for finding this single best state sequence exists, based on dynamic programming methods, and is called the Viterbi algorithm.

Viterbi Algorithm [49][50]: To find the single best state sequence, $Q = \{q_1 q_2 \cdots q_T\}$, for the given observation sequence $O = \{O_1 O_2 \cdots O_T\}$, we need to define the quantity

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda] \quad (2.2.13)$$

The complete procedure for finding the best state sequence can now be state as follows:

1) Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2.2.14)$$

$$\psi_1(i) = 0. \quad (2.2.15)$$

2) Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T$$

$$1 \leq j \leq N \quad (2.2.16)$$

$$\psi_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T$$

$$1 \leq j \leq N \quad (2.2.17)$$

3) Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.2.18)$$

$$q_T^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.2.19)$$

4) Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1. \quad (2.2.20)$$

There is no known way to analytically solve for the problem1. We can, however, choose $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximized using an iterative procedure, i.e., the Baum-Welch method. In order to describe the procedure for reestimation of HMM parameters, we first define

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (2.2.21)$$

From definitions of forward and backward variables, we can write $\xi_t(i, j)$ in the form

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \end{aligned} \quad (2.2.22)$$

We define $\gamma_t(i)$ as the probability of being in state S_i at time t , given the observation sequence and the model; hence we can relate $\gamma_t(i)$ and $\xi_t(i, j)$ by summing over j , giving

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (2.2.23)$$

A set of reasonable reestimation formulas for π , A , and B are

$$\bar{\pi}_i = \gamma_1(i) \quad (2.2.24)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (2.2.25)$$

$$\bar{b}_j = \frac{\sum_{t=1}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \quad (2.2.26)$$

If we define the current model as $\lambda = (A, B, \pi)$, and use that to compute the right-hand sides of (2.2.24)-(2.2.26), and we define the reestimated model as $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, as determined from the left-hand sides of (2.2.24)-(2.2.26), then it has been proven by Baum and his colleagues [48][51] that either 1) the initial model λ defines a critical point of the likelihood function, in which case $\bar{\lambda} = \lambda$; or 2) model $\bar{\lambda}$ is more likely than model λ in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$, i.e., we have found a new model $\bar{\lambda}$ from which the observation sequence is more likely to have been produced. Based on the above procedure, if we iteratively use $\bar{\lambda}$ in place of λ and repeat the reestimation calculation, we then can improve the probability of O being observed from the model until some limiting point is reached. The final result of this reestimation procedure is called a maximum likelihood estimate of the HMM.

(b) Continuous Hidden Markov Model

The formal specifications of an HMM with continuous observation densities are characterized by the following model parameters and the same notation as in Rabiner [7] is utilized. This is very much similar to the discrete HMM except that the finite mixture of the Gaussian distribution function is used as the output observation symbol probability function. To facilitate the following development, we define the following terms:

- N — The number of states in the model.
- M — The number of mixtures in the random function.
- A — The transition probability distribution matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability of the Markov chain transiting to state j , given the current state i , that is

$$a_{ij} = P[q_{t+1} = j | q_t = i] \quad (2.2.27)$$

where q_t is the state at time t and a_{ij} has the following properties:

$$a_{ij} \geq 0 \quad 1 \leq i, j \leq N \quad (2.2.28)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad 1 \leq i \leq N \quad (2.2.29)$$

The observation symbol probability distribution matrix $B = \{b_j(o)\}$ is defined below where $b_j(o)$ is the random function associated with state j . The most general representation of the random function is a finite mixture of Gaussian distributions which have the following form:

$$b_j(o) = \sum_{k=1}^M c_{jk} G(o, \mu_{jk}, U_{jk}) \quad 1 \leq j \leq N \quad (2.2.30)$$

where o is the observation vector, c_{jk} is the mixture coefficient for the k -th mixture in state j , and G is the Gaussian distribution with mean vector μ_{jk} and covariance matrix U_{jk} for the k -th mixture component in state j .

The mixture coefficient c_{jk} has the properties that satisfy the following stochastic constraints:

$$c_{jk} \geq 0 \quad 1 \leq j \leq N, 1 \leq k \leq M \quad (2.2.31)$$

$$\sum_{k=1}^M c_{jk} = 1 \quad 1 \leq j \leq N \quad (2.2.32)$$

π is the initial-state distribution matrix:

$$\pi = \{\pi_i\} \quad (2.2.33)$$

where $\pi_i = P[q_1 = i]$, $1 \leq i \leq N$.

It can be seen that the elements of an HMM include the model parameters of N , M , A , B and π . However, the values of N and M exist implicitly as indicated by the dimension of the matrices A and B respectively. Similar to the discrete HMM, the continuous HMM can be represented as follows:

$$\lambda = (A, B, \pi)$$

It was shown in [52]-[54] that the reestimation formulas for the coefficients of the mixture density, i.e., c_{jk} , μ_{jk} , and U_{jk} are of the following forms:

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)} \quad (2.2.34)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot O_t}{\sum_{t=1}^T \gamma_t(j, k)} \quad (2.2.35)$$

$$\bar{U}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot (O_t - \mu_{jk})(O_t - \mu_{jk})^T}{\sum_{t=1}^T \gamma_t(j, k)} \quad (2.2.36)$$

where $\gamma_t(j, k)$ is the probability of being in state j at time t with the k -th mixture component accounting for O_t , i.e.,

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \right] \left[\frac{c_{jk} G(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^M c_{jm} G(O_t, \mu_{jm}, U_{jm})} \right] \quad (2.2.37)$$

The reestimation formula for a_{ij} is identical to the one used for discrete observation densities (i.e., (2.2.25)).

2.2.3 Neural-Networks-Based Approach

Because speech recognition is basically a pattern recognition problem, and because neural networks are good in dealing with this sort of problem, many early researchers naturally tried applying neural networks to speech recognition. The earliest attempts involved highly simplified tasks, e.g., classifying speech segments as voiced/unvoiced, or nasal/fricative/plosive. Successes in these experiments motivated researchers to move on to phoneme classification. This task became a proving ground for neural networks as they quickly achieved excellent results. The same technique also achieved some successes at the level of word recognition. Here, three successful neural-networks-based approaches, namely multi-layer perceptron neural networks, time-delay neural networks and neural networks hidden Markov model hybrid system are reviewed.

(a) Multi-Layer Perceptron Neural Networks (MLPNN)

In [55], Huang, Lippmann and Gold performed a simple but elegant experiment using MLPNN. They demonstrated that perceptron neural networks could form complex decision surfaces from speech data. They applied a multi-layer perceptron with only 2 inputs, 100 hidden units, and 10 outputs to the vowels formant data. After 50,000 iterations of training, the network produced the decision regions which are nearly optimal, resembling the decision regions that would be drawn by hand, and they yield classification accuracy comparable to that of more conventional algorithms, such as the k-nearest neighbor and Gaussian

classification. In [56], a two-layer perceptron neural network trained by the back-propagation algorithm was used to recognize five English digits “one” to “five”. They achieved an overall accuracy of 89% for speaker-dependent test.

Sequential Multi-Layer Perceptron (SMLP) network was introduced in [57]. Each MLP in the SMLP is regarded as a state recognizer for distinguishing an acoustic event of the input speech signal. They achieved the best recognition rate of 98.3% for speaker-dependent ten Chinese digits recognition.

(b) Time-Delay Neural Networks (TDNN)

TDNN shown in Figure 2.8 for phoneme recognition was introduced in [58]. This architecture has only 3 and 5 delays in the input and hidden layer, respectively, and the final output is computed by integrating over 9 frames of phoneme activations in the second hidden layer. The TDNN’s design is attractive for several reasons:

- Its compact structure economizes on weights and forces the network to develop general feature detectors;
- Its hierarchy of delays optimizes these feature detectors by increasing their scope at each layer;
- Its temporal integration at the output layer makes the network shift invariant (i.e., insensitive to exact positioning of the speech signal).

The TDNN was trained and tested on 2000 samples of /b,d,g/ phonemes manually excised from a database of 5260 Japanese words. The TDNN achieved an error rate of 1.5%, compared to 6.5% achieved by a simple HMM-based recognizer.

Figure 2.8: Time-delay neural networks

(c) Hybrid Neural-Networks-HMM-Based System

Neural networks are excellent at acoustic modeling and parallel implementations, but weak at temporal and compositional modeling. HMMs are good models overall, but they are weak in their discrimination ability. Some researchers have tried to combine these two approaches into various hybrid systems, capitalizing on the strengths of each approach.

Perhaps the simplest way to integrate neural networks and HMM is to simply implement various pieces of HMM systems using neural networks. Although this does not improve the accuracy of an HMM, it does permit it to be parallelized in a natural way, and incidentally showcases the flexibility of neural networks. Lippmann introduced the Viterbi net in [59], which is a neural network that

implements the Viterbi algorithm. It was tested on 4000 word tokens from the nine-speaker thirty-five-word Lincoln stress-style speech database, and obtained results essentially identical to a standard HMM (0.56% error).

Rather than simply reimplementing an HMM using neural networks, most researchers have been exploring ways to enhance HMM by designing hybrid systems that capitalize on the respective strength of each technology: temporal modeling in the HMM and acoustic modeling in neural networks. In particular, neural networks are often trained to compute emission probabilities for HMM. Neural networks are well suited to this mapping task, and they also have a theoretical advantage over HMM. Because unlike discrete density HMM, they can accept continuous-valued inputs and hence do not suffer from quantization errors. Furthermore, unlike continuous density HMM, they do not make any unrealistic assumptions about the parametric shape of the density function. There are many ways to design and train a neural network for this purpose. The simplest approach is to map frame inputs directly to emission symbol outputs, and to train such a network on a frame-by-frame basis e.g. [60]. This approach is called Frame Level Training (FLT).

An alternative to FLT is Segment Level Training (SLT) e.g. [61], in which a neural network receives inputs from an entire segment of speech (e.g., the whole duration of a phoneme), rather than from a single frame or a fixed window of frames. This allows the network to take better advantage of the correlation that exists among all the frames of the segment, and also makes it easier to incorporate segmental information, such as duration. The drawback of this approach is that the speech must first be segmented before the neural network can evaluate the segments.

A natural extension to SLT is Word Level Training (WLT) e.g. [62], in which a neural network receives inputs from an entire word, and is directly trained to optimize word classification accuracy. WLT is appealing because it brings the training criterion still closer to the ultimate testing criterion of sentence recognition accuracy. Unfortunately, the extension is nontrivial, because in contrast to a simple phoneme, a word cannot be adequately modeled by a single state, but requires a sequence of states. Activations of these states cannot be simply summed over time as in a TDNN, but must first be segmented by a DTW, identifying which states to be applied to which frames. Thus, WLT requires that DTW be embedded into a neural network.

Chapter 3

Dimensionality Reduction in Pattern Recognition

This chapter provides an introduction to the basic underlying concepts, some conventional techniques and the subsequent development of dimensionality reduction methodologies, which are commonly used for feature extraction and feature space projection.

3.1 Introduction

Pattern recognition has a long history in engineering, especially for military applications. However, it was not a popular subject due to the cost of hardware for acquiring the data and computation. In recent years, pattern recognition has gained more attention since the advances made in hardware.

The discipline of pattern recognition or pattern recognition by machine essentially deals with the problem of developing algorithms, methodologies and devices that can enable implementation of many of the recognition tasks that human beings normally perform on computers. According to Jain et al. [63], pattern recognition is a general term used to describe a wide range of problems like recognition, description, classification, and grouping of patterns. These problems are important in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, artificial intelligence, speech recognition, computer vision and remote sensing.

From [64], we can say that pattern recognition is about guessing or predicting the unknown nature of an observation, a discrete quantity such as black or white, one or zero, sick or healthy, real or fake. And a pattern is defined in [65] as “opposite of a chaos; it is an entity, vaguely defined, that could be given a name.” For example, a pattern could be a speech signal, a fingerprint image, a handwritten word or a human face. The goal of pattern recognition research is to devise ways and means of automating certain decision-making processes that lead to classification and recognition.

The recognition problem here is considered as a classification or categorization task, where the classes are defined by the system designer in supervised classification or are learnt based on the similarity of patterns in unsupervised classification. The pattern is described by quantitative values and the object is represented by a set of D features or attributes. We can say that the object is a d -dimensional feature vector. To make a classification, the space spanned by these feature vectors is subdivided using decision boundaries. Each part of the space then represents a class and concepts from statistical decision theory are utilized to establish these decision boundaries.

The decision making process in pattern recognition can be summarized as follows: A given pattern is to be assigned to one of L categories $C_1, C_2, \dots, C_L$ based on a vector of n feature values $x = (x_1, \dots, x_n)$. Statistically, the features are assumed to have a probability density or mass function conditioned on the pattern class. Thus, a pattern vector x belonging to class C_i is viewed as an observation drawn randomly from the class-conditional probability function $f(x|C_i)$. To define the “optimal” decision rule, the Bayes decision rule is used.

(a) Bayes Decision Theory [64]

The following approach builds upon probabilistic arguments based on the statistical nature of the generated features. If the probability density functions (pdf) of all classes are known, the discrimination of classification is based on the Bayes decision theory. Given a classification task of L classes $C_1, C_2, \dots, C_L$, and an unknown pattern represented by a feature vector x , we form the L conditional probabilities $P(C_i|x)$, $i = 1, \dots, L$. These are called a-posteriori probabilities. Now the task is to compute these conditional probabilities. For this, the Bayes rule proves useful. Let $P(C_i)$ be the a-priori probability and these values can be known or easily estimated from the training data. Let $p(x|C_i)$ be the class-conditional probabilities describing the distribution of the feature vectors in each of the classes. These values also have to be estimated. Using Bayes rule, a-posterior probability can be computed using a-prior and the likelihood as follows:

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)} \quad (3.1.1)$$

where $p(x) = \sum_{i=1}^L p(C_i)p(x|C_i)$ is the pdf of x .

The Bayes decision rule is based on finding the class that gives the maximal posterior probability taking into account the loss. It has been shown that this rule has the smallest misclassification probability if one assumes a simple zero-one loss function, and the smallest Bayes risk (that is, the total expected loss assuming both the feature vector and the class label to be random) if we assume a more general loss-function [66]-[68].

(b) Generalization and the Curse of Dimensionality

The objective of classification is to build a set of decision boundaries, using only a finite set of example objects. This limited set puts constraints on the learning process, in the sense that it is important to limit the number of features. Generalization and the curse of dimensionality are two important concepts in the learning theory.

No matter what classifier or decision rule is used, it must be trained using the available training samples. The performance of the classifier depends on the interrelationship between sample sizes, number of features, and classifier complexity. To precisely classify future samples is the goal of designing a recognition system and likely to be different from the training samples. Therefore, optimizing a classifier to maximize its performance on the training set may not always result in the desired performance on a test set. Generalization is the ability of a classifier to deliver good performance on a test set of samples, which are not used during the training stage. Poor generalization can be attributed to one of the following factors:

- The number of features is too large relative to the number of training samples;

- The number of unknown parameters associated with the classifier is large;
- A classifier is too intensively optimized on the training set.

When a classification of high-dimensional vectors is implemented, the curse of dimensionality [69] becomes the main factor affecting the classification performance. The curse of dimensionality is due to the inherent sparsity of high-dimensional spaces, implying that, in the absence of simplifying assumptions, the amount of training data is really high in order to get reasonably low variance estimators. If the class densities are completely known, an increase in the number of features will not result in an increase in the probability of misclassification. In fact, it has often been observed that the added features may actually degrade the performance of a classifier when the number of training samples is small relative to the number of features. We give a simple explanation in the subsequent section. In most classifiers, parametric estimation is employed to find the probability density of the different classes. The unknown parameters of the classifier are estimated and then are applied for searching the true parameters. If the sample size is fixed, the reliability of the parameter estimates will decrease as the number of features is increased. Therefore, the performance of the resulting classifiers may degrade with the increase of the number of features when the sample size is fixed. To deal with this problem, we can assume that an important structure in the data actually lies in a much smaller dimensional space, and try to reduce the dimensionality before classification. The important premise to utilize the dimensionality reduction approach is that it can keep as useful information as possible in the transformation from high-dimensional space to the low-dimensional one.

There are several ways to reduce the dimension of the feature space. From [70], a limited set of features is selected out of the total set. Feature selection could be the

easiest way to reduce the dimensionality. Feature extraction is another common way. In feature extraction, features are extracted as linear or non-linear functions of the original features. Supervised linear feature extraction techniques usually rely on the discriminant analysis technique [71] which uses the within and between-class scatter matrices. Unsupervised linear feature extraction techniques are usually based on Principal Component Analysis (PCA), which rotates the original feature space before projecting the feature vectors onto a limited amount of axes.

Several non-linear mapping techniques were already developed in pattern recognition. Some methods attempt to map the original data space into a lower dimensional space and preserve as much as possible useful information according to some predefined error criterion, for example, multidimensional scaling [72], [73] and Sammon's mapping [74]. With the development of neural networks, new possibilities for non-linear mapping were created. The most well-known method is self-organizing maps of [75]. Other ways include auto-associative feedforward networks of [76] and a neural network version of Sammon's mapping etc. [77],[78]. Recently a supervised neural network approach was presented for feature extraction for classification purposes, and the results were compared with PCA in [79].

We will give a detailed review of a variety of dimensionality reduction strategies at the following section.

3.2 Dimensionality Reduction Techniques

There are two main reasons to keep the dimensionality of the pattern representation as small as possible: measurement cost and classification accuracy. A limited yet salient feature set simplifies both the pattern representation and the

classifiers that are built on the selected representation. Consequently, the resulting classifier will be faster and will use less memory. Moreover, as stated earlier, a small number of features can alleviate the curse of dimensionality when the number of training samples is limited. On the other hand, a reduction in the number of features may lead to a loss in the discrimination power and thereby lower the accuracy of the resulting recognition system.

It is important to make a distinction between feature selection and feature extraction. In pattern recognition, for feature vectors $V = [v_1, v_2, \dots, v_n]^T$, feature extraction and feature selection are defined as a transformation $J(A) = \max_A J(A(v))$ and $J(X) = \max_X J(X(v))$ respectively. In feature selection, the best features are retained and the remaining ones are discarded according to a chosen criterion J . In feature extraction, all features are retained and are subject to transformation A . In both cases, a mapping $W: \Phi^n \rightarrow \Psi^m$ maximizing a criterion J is determined, where $m < n$ and $y = [y_1, y_2, \dots, y_m]^T$. Here, $y = A(v)$ can be a linear or non-linear transformation and can employ unsupervised or supervised information. Feature computation and extraction or selection can be cascaded for classification, which carries out partitioning of the final feature space. This feature space should be optimum with regard to both constraints compactness and discriminance. Feature visualization requires a dimensionality reduction to a two or three-dimensional representation (e.g., [80]-[82]).

The main issue in dimensionality reduction is the choice of a criterion function. A commonly used criterion is the classification error of a feature subset. One has to be careful, since the classification error itself cannot be reliably estimated when the ratio of sample size to the number of features is small. In addition to the choice of

a criterion function, we also need to determine an appropriate dimensionality of the reduced feature space.

3.2.1 Supervised Mapping Techniques

Dimensionality reduction can be achieved in essentially two different ways: feature selection and feature extraction, given a set of example feature vectors with their corresponding class labels. Feature selection is to seek m features out of the available n measurements and the classification error is a natural choice for the criterion function. This makes the feature selection procedure dependent on the specific classifier that is used. The only “optimal” feature selection method which avoids the exhaustive search is based on the branch and bound algorithm [83]. Other strategies are suboptimal due to the fact that the best pair of features need not contain the best single feature [84]. As a result, the naive method of selecting just the best individual features may fail dramatically. There are more advanced methods which depend on the feature, for example, the Simple Forward Selection (SFS) and the Simple Backward Selection (SBS). Like SFS and SBS, simple sequential methods add or delete one feature at a time. In large feature selection problem, more sophisticated techniques perform better than SFS and SBS, such as the “plus 1 - take away r ” strategy and the sequential floating search methods, SFFS and SBFS [85]. These methods backtrack as long as they find improvements compared to previous feature sets of the same size. The SFFS and SBFS methods find nested sets of features which remain hidden, but the number of features set evaluations, however, may easily increase by a factor of 2 to 10.

Feature extraction is usually performed by discriminant analysis in supervised mapping techniques. The class information associated with each pattern is used for linearly extracting the most discriminatory feature. The interclass separation is

emphasized by finding the eigenvectors of a general separability measure, like the Fisher criterion $S_w^{-1}S_b$, where S_w is the within-class scatter matrix and S_b is the between-class scatter matrix [86]. The Patrick-Fisher distance using Parzen density estimates is another supervised criterion for non-Gaussian class-conditional densities [64].

(a) Linear Mapping Techniques

The LDA method was introduced in [86]. The linear mapping tries to increase interclass and decrease intraclass distance while reduces the dimension of feature vectors. One possible quality measure for separability is given by $J = \text{Tr}(S_w^{-1}S_b)$. This measure attains large values if interclass scatter becomes large and if intraclass scatter becomes small. As variance for PCA, most discriminance is related to those m eigenvectors of $(S_w^{-1}S_b)$ associated with the largest m eigenvalues. This approach works well if the Gaussian assumption is met; however for nonparametric or even multimodal data, results may be unsatisfying [86]. This method will be described in the following chapter in details.

(b) Non-linear Mapping Techniques

Koontz and Fukunaga introduced a mapping (KFM) [87], which is related to Kruskal's work and Sammon's Non-Linear-Mapping (NLM) [88]. They introduced an additional term in the cost function, e.g. Sammon's stress, dedicated to separability which implies supervised information, i.e. class affiliation, for each pattern vector. During mapping, both the constraints of distance preservation and separability of class regions are pursued.

An interesting alternative is already introduced where the KFM mapping is used to train a neural network. Although this approach is not straightforward to achieve network convergence and good approximation of the KFM, it allows us to project to an arbitrary dimension and to map single points or test sets. By reducing the second hidden layer size to the desired target dimension and training the network with the training set, a comparable mapping is achieved. However, the KFM's property of structure preservation is not explicitly available in the backpropagation network.

These kinds of mappings are very powerful to separate and shape clusters due to the use of discriminant analysis, in the case of significant class regions' overlap for test-data-set-related errors remain. And the step of discriminant analysis should not be necessary if a backpropagation network is used as a classifier, because any mapping can be represented by the network. However, it will be tedious to find the appropriate configuration, and a hierarchical network approach can be beneficial.

3.2.2 Unsupervised Mapping Techniques

The PCA is the best known unsupervised feature extractor. Since PCA uses the most expressive features (eigenvectors with largest eigenvalues), it effectively approximates the data by a linear subspace using the mean squared error criterion. Projection pursuit [89] and Independent Component Analysis (ICA) [90]-[94] are more appropriate for non-Gaussian distributions since they do not rely on second-order properties of the data.

(a) Linear Mapping Techniques

A mapping can most simply be achieved by explicit selection of two or three salient components or factors and context knowledge is required for the selection. Supervised automatic feature selection methods can be employed to find salient features for visualization [69]. In some techniques, simultaneous display of multiple linked pairwise plots of the data set is proposed as a selection. However, the combinatorial explosion limits practical applications of this approach, where there are $\frac{M(M-1)}{2}$ plots for M-dimensional data. The PCA can also implement the mapping by the first two or three principal components [86]. If the PCA assumption of Gaussian distribution is met and the most variance is presented by the first two principal components, suitable plots can be achieved by this linear method [81]. The criterion for PCA mapping is based on signal preservation, but not structure preservation or discriminance gain. Namely, the PCA mapping tries to obtain the least error distortion when the signal is reconstructed from the selected principal components. Finally, it might result in accuracy and numerical problems when the data set size is not adequate with regard to data dimension.

(b) Neural-Networks-Based Mapping Techniques

The Backpropagation networks (BP) mapping method has been applied in autoassociative mode and a bottleneck topology to achieve mappings comparable to PCA [95]. However, it is hard to interpret and it strongly depends on other network parameters, e.g. the number of hidden layer neurons.

In terms of structure reservation, the most promising mapping methods use the criteria of either topology or distance preservation for the non-linear mapping process. The most popular method for data visualization is Kohonen's Self-Organizing Map (SOM) [96]. During training, the SOM unfolds in pattern space

and creates a topology preserving mapping of the multivariate data on the fixed neuron grid. In this mapping, only neighborhood relation information is concerned, and there is no information on intra or inter-cluster distance. The unified-distance-matrix complemented the SOM with this distance information by working on exploratory data analysis [98]. Kohonen proposed the NLM in [88] to include distance information in SOM visualization by mapping SOM weight vectors. There are several other practical problems in using SOM display [97]. Firstly, the SOM is not suited for identifying the position of individual data vectors in the map visualization, due to the quantization carried out by SOM along with the topology-preserving mapping. Secondly, SOM interpolation properties cause the placement of weight vectors in pattern space regions to be void of data vectors. Thirdly, if the training data possesses not only a high absolute dimension, but also a high intrinsic dimension [86] larger than two or three dimensions, the SOM starts to fold and twist in an attempt to establish a mapping to the plane of the neuron grid. This can lead to the scattered representation of an intrinsically high-dimensional cluster all over the map and cause misinterpretations. The Growing-Cells method employed dynamically allocated neurons only in regions of significant pattern density to solve the second SOM problem [99]. However, there are no improvements for the other two issues.

(c) Non-linear Mapping Techniques

The NLM is a distance preserving non-linear mapping [88]. According to the cost function $C(m)$, the interpoint distances d_{xij} implicating the data structure shall be preserved in the mapping i.e.

$$C(m) = \frac{1}{c} \sum_{j=1}^N \sum_{i=1}^j \frac{(d_{xij} - d_{yij}(m))^2}{d_{xij}} \quad (3.2.1)$$

where $d_{yij}(m) = \sqrt{\sum_{q=1}^d (y_{iq}(m) - y_{jq}(m))^2}$ are the distance of the respective data points in the visualization plane, $d_{xij}(m) = \sqrt{\sum_{q=1}^M (v_{iq} - v_{jq})^2}$ in the original data space and $c = \sum_{j=1}^N \sum_{i=1}^j d_{xij}$. Based on a gradient descent approach, the new coordinates of the N pivot vectors in the visualization plane y_i are determined be:

$$y_{iq}(m+1) = y_{iq}(m) - MF * \Delta y_{iq}(m) \quad (3.2.2)$$

with

$$\Delta y_{iq}(m) = \frac{\partial C(m)}{\partial y_{iq}(m)} \bigg/ \left| \frac{\partial^2 C(m)}{\partial y_{iq}(m)^2} \right| \text{ and } 0 < MF \leq 1 \quad (3.2.3)$$

It is well-known that the NLM achieves a fair projection of the data. However, if it is used for feature extraction in a pattern recognition application, it will need recomputation of the NLM to map individual points or a complete test set.

According to many research on NLM, it has become well known to researchers that in general, all distances could not be perfectly preserved by the mapping anyway. Lee, Slaggle and Blum developed a fast distance preserving mapping that focuses on exact preservation of only a limited number of $2N - 3$ distances, neglecting all remaining ones[80]. Minimal-Spanning-Tree (MST) of the data distance graph is computed in this mapping. However, MST computation and traversal itself has $O(N^2)$ complexity. Viso mapping presents a faster mapping algorithm, and has $O(N)$ complexity and thus provides data projections with a very short response time [100].

As a conclusion, distance-preserving mappings are regarded as the most convenient, powerful and practical option for fast mapping and visualization compared with other mapping techniques [80], [100].

Chapter 4

Step-Weighted Linear Discriminant Analysis

In this chapter, we first present a brief introduction to Linear Discriminant Analysis (LDA), a classical linear dimensional reduction technique, and fundamentals and some developments proposed by other researchers. After that, a Step-Weighted Linear Discriminant Analysis (SW-LDA) technique is presented and applied to dimensionality reduction applications.

4.1 Introduction

Dimensionality reduction is the process of mapping high dimensional patterns to a lower dimensional subspace and is typically used as a preprocessing step in classification applications. Generally speaking, dimensionality reduction can be represented by $z = W(x)$, where $x \in \Phi^n$, $z \in \Psi^m$ and $m < n$. We can obtain linear or non-linear W with or without the use of a class label. The most important issues in dimensionality reduction are finding a W and a suitable optimality criterion.

The best known technique for linear dimension reduction is LDA [101], [102]. The idea of the LDA is as follows: A transform is determined such that the Fisher optimality criterion of the ratio of the between-class scatter to the within-class scatter is maximized [86]. The LDA usually gives a good discrimination performance, but there exist some deficiencies. To circumvent the deficiencies of the LDA, a number of extensions of LDA have been proposed. One of the deficiencies is that the LDA is too flexible when there are many highly correlated variables; in this case, the LDA can overfit the data. Friedman [103] proposed Regularized Discriminant Analysis (RDA) using a regularization technique to estimate the sample covariance more reliably to solve the overfitting problem. If the class boundaries are complex or non-linear, the LDA becomes too rigid and can underfit the data. To solve the underfitting problem, Hastie et al. proposed two alternative methods, namely, Mixture Discriminant Analysis (MDA) [104] and Flexible Discriminant Analysis (FDA) [105]. The MDA fits each class using the Gaussian mixture model that works well for clustered data. The FDA is a nonparametric version of the LDA using multivariate nonparametric regression. The Heteroscedastic Discriminant Analysis (HDA) generalized the LDA by removing the equal within-class covariance constraint because the LDA is known to be inappropriate for the case where classes are of unequal sample covariances [106].

In this chapter, we exploit another implicit assumption of LDA. The between-class covariance matrix in the LDA assumes that each class is equally confusable with all other classes. It becomes obvious from the rewritten between-class scatter expression [107] that the large class distances are overemphasized and the small class distances are underemphasized in the Fisher criterion. In order to alleviate this assumption, the authors of [107] proposed the weighted pairwise scatter

constructed in which smaller distances are more heavily weighted than larger distances. A similar approach by introducing a weighted variant to the linear dimension reduction criterion is also presented in [108]. The method of fractional-step dimensionality reduction is proposed to weight the between-class scatter more precisely [109]. However, the weighted scatter matrix is still based on the assumption that the between-class distances are not very different along each coordinate direction of the n dimensional feature space. The weighted scatter matrix is calculated in the input-space and the true value of the weighted scatter matrix in the output-space is often far from correct when some dimensionality is reduced. This will result in loss of useful discriminant information in the projection process. Consequently, we introduce the concept of step weight in the procedure of dimensionality reduction. The key idea is the following: Choose different weighting functions in terms of distances between different classes first, and then recompute the ratio of between-class scatter matrix to the within-class scatter matrix in the output-space after each dimension of the feature space is reduced. The proposed method will be illustrated using some simulation results.

4.2 Review of Conventional LDA

Linear dimension reduction is concerned with the search for a linear transformation that reduces the dimension of a given n -dimensional feature space to m ($m < n$) dimension, while preserving a maximum amount of discrimination information in the lower-dimensional subspace. As we know, the Bayes error is the optimality criterion for classification, which is the minimum achievable classification error. However, since the Bayes error cannot always be easily used as a criterion, one resorts to criteria that are suboptimal but that are easier to optimize. The LDA is a suboptimal approach and this method maximizes the ratio

of the between-class variance to the within-class variance in any particular data set, thereby guaranteeing maximal separability.

The objective of the LDA is to find the subspace Ψ of dimensionality m so that the optimality criterion is maximized. A common form of an optimality criterion to be maximized is the function $J = \text{tr}(\hat{S}_b \hat{S}_w^{-1})$, where $\hat{S}_b$ is the between-class scatter matrix and $\hat{S}_w$ is the within-class scatter matrix in the output space. The LDA problem is formulated as follows [110]: Let $x \in \Phi^n$ be a feature vector. We seek to find a transformation $z = W(x)$, $W : \Phi^n \rightarrow \Psi^m$ with $m < n$, such that in the transformed space, there is minimum loss of discrimination information. In practice, m is much smaller than n .

In classical LDA, the corresponding within-class and between-class scatter matrices in the input space are defined by

$$S_b = \sum_{k=1}^N n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T \quad (4.2.1)$$

$$S_w = \sum_{k=1}^N \sum_{j=1}^{n_k} (x_{kj} - \bar{x}_k)(x_{kj} - \bar{x}_k)^T \quad (4.2.2)$$

where $\bar{x}_k = \frac{1}{n_k} \sum_{p=1}^{n_k} x_{kp}$, $\bar{x} = \frac{1}{M} \sum_{k=1}^N n_k \bar{x}_k$, n_k is the number of points in class k , N is the number of classes and x_{kj} is the j -th vector of class k of the input set X with M elements. Furthermore, M and n_k are related by $M = \sum_{k=1}^N n_k$.

To be specific, the optimality criterion of LDA is maximized as the following objective function:

$$\max_{\Theta} \frac{|\Theta^T S_b \Theta|}{|\Theta^T S_w \Theta|} \quad (4.2.3)$$

where the columns of the optimum Θ are the relative generalized eigenvectors corresponding to the first m maximal values of the eigenvalues corresponding to the following equation:

$$S_b \Theta = \lambda S_w \Theta \quad (4.2.4)$$

4.3 Step-Weighted Linear Discriminant Analysis (SW-LDA)

In the proposed SW-LDA method, we first consider constructing a weighted scatter matrix in S_b . According to the definition (4.2.1), we know the between-class covariance matrix S_b is a measure of how distributed the means of each class are from the center. Intuitively, it is better to have a “bigger” value of S_b since it shows that the classes are more spread out in the transformed space and it is easier to discriminate them. However, it is not clear how the classes are discriminated from each other pairwise. To clearly illustrate this problem, an example from [107] is used.

Suppose there are four classes in Φ^2 and each of them has the same number of feature vectors and has equal variance. The problem is illustrated in Figure 4.1. Their means are

$$\begin{aligned} \bar{x}_1 &= (1, \varepsilon) & \bar{x}_2 &= (-1, \varepsilon) \\ \bar{x}_3 &= (-1, -\varepsilon) & \bar{x}_4 &= (1, -\varepsilon) \end{aligned}$$

The between-class scatter matrix S_b can be written as follows:

$$S_b = 4 \times \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon^2 \end{pmatrix} \rightarrow 4 \times \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{when } \varepsilon \rightarrow 0.$$

When $\varepsilon \rightarrow 0$, the between-class scatter matrix does not contain any discrimination in the vertical direction. There is only discrimination in the horizontal direction. We can say that the between-class scatter matrix is dominated by the covariance of the class pairs other than (1, 4) and (2, 3). The covariance of class pairs (1, 4) and (2, 3) is more important than others since they are more confusable when considering the classification problem. However, the conventional between-class scatter obviously does not capture this information.

Figure 4.1: Illustration of the pairwise problem by an example

From the above discussion, because the definition of between-class scatter matrix is not directly related to classification accuracy, a weighted scatter matrix is constructed in which smaller distances are more heavily weighted than larger distances as follows:

$$S_b = \frac{1}{2M} \sum_{k=1}^N \sum_{l=1}^N \omega_{kl} n_k n_l (\bar{x}_k - \bar{x}_l)(\bar{x}_k - \bar{x}_l)^T \quad (4.3.1)$$

where $\bar{x}_l = \frac{1}{n_l} \sum_{p=1}^{n_l} x_{lp}$, ω_{kl} is a non-negative weight assigned to class pair (k, l) and it represents how important it is to discriminate class k from class l .

In the LDA analysis, if we wish to reduce the dimensionality from n to $(n-1)$, we need to compute S_b , S_w and eigenvectors $\Theta_1, \Theta_2, \dots, \Theta_n$. Fortunately, we can find a set of discriminant eigenvectors that are orthogonal one to another suggested by J. Duchene and S. Leclercq [110]. We would project patterns onto the subspace Ψ^{n-1} spanned by orthogonal eigenvectors $\Theta_1, \Theta_2, \dots, \Theta_{n-1}$. If the between-class distances are not very different along each direction of the n -dimensional feature space spanned by eigenvectors, the differences of between-class distances will have minor changes after projecting one dimension, and the infection is minor to the selected weighting function in advance. However, when there are sufficient classes, it is very possible that a pair of classes (say class p and class q) such that $(\bar{x}_p - \bar{x}_q)$ has the same orientation as Θ_n . Because the eigenvectors $\Theta_1, \Theta_2, \dots, \Theta_n$ are orthogonal respectively, the two classes would heavily overlap in the $(n-1)$ dimensional space. In the original n dimensional space, the distance between class p and class q is big and a small weighting value is enough to keep discriminant information and calculate S_b . After projecting the original space to the $(n-1)$ dimensional subspace, the distance between class p and class q

compared with other between-class distances become very small. Under this circumstance, the predefined weighting function is not proper when we continue projecting the $(n-1)$ dimensional subspace to $(n-2)$. Therefore, we reason that the $(\bar{x}_p - \bar{x}_q)$ was insufficiently weighted in computing S_b using (4.3.1). This will result in loss of some important information, when we reduce the dimensionality from n to m (m is much smaller than n).

In the SW-LDA, instead of reducing the dimensionality from n to m directly, we gradually compress the feature space one dimension per step. At each step in the dimension reduction process, we recompute the between-class and within-class scatter matrices according to the changed interclass distances and intraclass distances and rebuild the weighting function. As a consequence, those class centers that come closer together can be increasingly weighted to avoid severe overlap of the classes. The corresponding output-space within-class and between-class scatter matrices are defined by

$$\hat{S}_b = \frac{1}{2M} \sum_{k=1}^N \sum_{l=1}^N \omega_{kl} n_k n_l (\hat{\bar{x}}_k - \hat{\bar{x}}_l)(\hat{\bar{x}}_k - \hat{\bar{x}}_l)^T \quad (4.3.2)$$

$$\hat{S}_w = \sum_{k=1}^N \sum_{j=1}^{n_k} (\hat{x}_{kj} - \hat{\bar{x}}_k)(\hat{x}_{kj} - \hat{\bar{x}}_k)^T \quad (4.3.3)$$

$$\hat{\bar{x}}_k = \frac{1}{n_k} \sum_{p=1}^{n_k} \hat{x}_{kp} \quad (4.3.4)$$

where $\hat{\bar{x}}_k$ denotes the mean of class k in the output space and $\hat{x}_{kp}$ is the p -th vector of class k in the output space.

The entire SW-LDA algorithm in reducing the dimension from n to m can be summarized as follows:

Step 1: Calculate S_b and S_w according to (4.3.1) and (4.2.2);

-
- Step 2: Calculate the transformation matrix $T_{n-1} = [\Theta_1, \Theta_2, \dots, \Theta_{n-1}]$ and reduce the dimension of the feature space from n to $n-1$;
- Step 3: Calculate new weighting functions ω_{kl} according to the between-class distances in the output space. If we have N classes, the weighting function is chosen as $\omega_{kl} = \frac{1}{d_{kl}^n}$, where $d_{kl} = \|\bar{x}_k - \bar{x}_l\|^2$ and $k, l = 1, \dots, N$;
- Step 4: Calculate $\hat{S}_b$ and $\hat{S}_w$ according to (4.3.2) and (4.3.3);
- Step 5: Calculate the transformation matrix $T_{n-2} = [\hat{\Theta}_1, \hat{\Theta}_2, \dots, \hat{\Theta}_{n-2}]$ and reduce the dimension from $n-1$ to $n-2$;
- Step 6: Repeat Step 3, 4 and Step 5 until the feature space reaches m dimension;
- Step 7: Calculate the transformation matrix $T = T_m * T_{m+1} * \dots * T_{n-2} * T_{n-1}$;
- Step 8: After training, use the transformation matrix T to project the observed feature vectors or testing features from dimension n to m .

4.4 Simulation Studies

Coping with inter-speaker variability (i.e., differences in the vocal tract characteristics of speakers) is still a major challenge for automatic speech recognizers in isolated word recognition or continuous speech recognition [111]. In order to show the effectiveness of the proposed feature extraction method, isolated-word recognizers were built. The proposed algorithm is applied to a data set from TI46 corpus which were collected from 16 different adults. TI46 corpus were designed and collected at Texas Instruments (TI) [112]. This database is a set of English speech including ten digits and it contains the speech of 16 speakers

including 8 males and 8 females. There are 10 training tokens and 5 testing tokens for each speaker.

We present results obtained using the general Weighted LDA (WLDA) of [107] and the proposed SW-LDA. Both the WLDA and the SW-LDA algorithms were run with all the choices of the weighting function.

Training accuracies are measured by the method based on a five-state left-to-right HMM. To classify a given test data into one of k classes, train up k HMMs, one per class and then compute the likelihood that each model gives to the test sequence; if the i -th model is the most likely, then declare the class of the sequence to be class i . The software of HMM recognizer for measuring different methods was built using MATLAB 6.5. The algorithms of recognizer were derived from Rabiner's work [7].

The data set consists of ten English digits from 0 to 9. The feature vector was built by 32 MFCCs. To clearly demonstrate the classification results based on the proposed method, 10 class labels were obtained by the word. After manually labeling, the classes are: ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT and NINE. There are a total of 1600 training patterns with 160 for each of the ten classes. The testing patterns are 800 and each class has 80 testing patterns. We ran the simulation for ω_{kl} taken from the set $\{d_{kl}^{-2}, d_{kl}^{-4}, d_{kl}^{-6}, d_{kl}^{-8}, d_{kl}^{-10}, d_{kl}^{-12}, d_{kl}^{-14}, d_{kl}^{-16}\}$ when the dimensionality is reduced from 32 to 28.

	d_{kl}^{-2}	d_{kl}^{-4}	d_{kl}^{-6}	d_{kl}^{-8}	d_{kl}^{-10}	d_{kl}^{-12}	d_{kl}^{-14}	d_{kl}^{-16}
WLDA	92%	94%	91%	93%	93%	93%	92%	91%
SW-LDA	93%	94%	93%	92%	93%	95%	93%	93%

Table 4.1: Testing classification accuracy on different weighting functions

Table 4.1 shows the testing accuracies obtained with the WLDA and the SW-LDA for different weighting functions. The accuracy shows roughly invariant result over the range of powers of d_{kl} for the WLDA and the SW-LDA. We can also see that the WLDA and SW-LDA have no more performance difference after reducing only 4 dimensions for the feature space. In the following projecting procedure, the dimensionality is gradually reduced from 32 to 4, and the best weighting function for the WLDA turns out to be $\omega_{kl} = d_{kl}^{-4}$ and for the SW-LDA, it is $\omega_{kl} = d_{kl}^{-12}$.

Figure 4.2: Testing classification accuracy in reducing dimensionality from 32 to 4.

The weighting function for the WLDA is $\omega_{kl} = d_{kl}^{-4}$ and for the SW-LDA is

$$\omega_{kl} = d_{kl}^{-12} \text{ respectively.}$$

Figure 4.2 shows the testing classification accuracies when the feature space is gradually projected from dimension 32 to 4. For the SW-LDA method, we can observe slight improvement over the WLDA method when the dimensionality is reduced from 28 to 12. The reason is that the between-class distances are not very different along each coordinate direction of the dimensional feature space when the feature space is projected from dimension 32 to 12. Therefore, most of the useful information is retained using WLDA and SW-LDA when reducing the

dimensionality of feature space. According to the simulation results, more unimportant information is included in the high dimensionality input spaces. After reducing the dimensionality and cleaning the unimportant information, the performance of the resulting classifiers is good and the computation load is lower. However, the classification accuracy of the WLDA method shows an obvious drop than that of the SW-LDA after the output space is projected to dimension 8. The most important reason is that some useful information is lost during the dimensionality projecting procedure. After projecting the original space to the 12-dimensional subspace, the distances between some classes compared with other between-class distances become very small. Under this circumstance, the predefined weighting functions are not proper when we continue projecting the space. Therefore, the classification accuracy of the WLDA is worse than that of the SW-LDA.

According to the simulation results, the classification accuracy of the SW-LDA is better than that of the WLDA. However, extra computation load was imposed by the SW-LDA at the same time. This extra computation load was applied on reducing the dimensionality of the feature space. Therefore, after compressing the feature space, the computation load will be reduced in building speech models and recognizing test data. Considering the classification accuracy is increased, the SW-LDA is superior to the WLDA in reducing the dimensionality of feature space.

4.5 Conclusions

In this chapter, we consider the effectiveness of step-weighted idea for LDA in linear discriminant reduction, and propose the SW-LDA algorithm for improving the LDA method. The optimal criterion of the LDA based on scatter matrices in general is not directly related to classification accuracy. A weighted scatter matrix

can keep more discriminant information due to different weights. However, the between-class scatter matrix is calculated in the input-space and the true value of the scatter matrix in the output-space is far from correct when the feature space is projected. Our investigations show that the general WLDA will result in loss of useful discriminant information in the process of projection.

In order to precisely keep more discriminant information during the procedure of feature space dimensionality reduction, we studied the process of feature space projection using the WLDA and proposed SW-LDA dimensionality reduction method. We gradually compress the feature space one dimension per step. At each step of dimension reduction, we recompute the between-class and within-class scatter matrices. As a consequence, those centers of classes that come closer together can be increasingly weighted. Therefore, using the SW-LDA dimensionality reduction method, one can obtain a projection subspace with good classification performance without having to resort to non-linear optimization techniques.

A data set from the TI46 corpus is employed to test the new algorithm. The classification accuracy based on the proposed SW-LDA method is compared to that of the WLDA method. When the dimensionality is reduced more, the SW-LDA method shows better classification performance than the general WLDA as illustrated by the simulation results. Therefore, we can say that within the context of linear dimensionality reduction, the proposed method provides a better lower-dimensional representation.

Chapter 5

Non-linear Weighted Discriminant Analysis

In this chapter, we present a new non-linear dimensionality reduction technique. Detailed working principles of the proposed method will be introduced in the follow section. Finally, the capability and properties of this method will be demonstrated via some simulation studies.

5.1 Introduction

Classical LDA has been proven to be successful on classification and dimensionality reduction problems. The procedure of LDA is based on eigenvalue resolution and it gives an exact solution to the maximum value of the inertia. Therefore, the LDA attempts to reduce dimensionality with minimal loss in discrimination information. There are many developments in classical LDA and we have demonstrated some improvement in Chapter 4. It is well-known that one of the major drawbacks of LDA is that the optimality criterion based on scatter

matrices is not directly related to classification accuracy. To solve this problem, a weighted scatter matrix was constructed in [107] where smaller distances are more heavily weighted than larger distances. Unfortunately, the weighted LDA (WLDA) is still a linear analysis method and it fails for more complicated non-linear problems. In recent years, Scholkopf et al. explored the basic idea behind Support Vector Machine (SVM) [113] and extended the PCA to non-linear kernel PCA for extracting structure from high-dimensional data set [114]. The authors also proposed a non-linear variant of other algorithms such as Independent Component Analysis (ICA) or kernel-k-means. They pointed out that it would be desirable to develop a non-linear form of discriminant analysis based on the kernel method. The main idea is to map the input space into a convenient feature space in which variables are non-linearly related to the input space. Suykens et al. improved the kernel PCA by using a simple and straightforward primal-dual support vector machine formulation [115]. Subsequently, Kernel Fisher Discriminant (KFD) [116] and Generalized Discriminant Analysis (GDA) [117] were proposed using a kernel function operator, which enabled applications of classical LDA in non-linear discriminant analysis in the transformed space. However, it is still a problem to implement the WLDA in non-linear discriminant analysis problems using a kernel idea.

In this chapter, we concentrate on two drawbacks of classical LDA. The first one is that the between-class scatter matrix in LDA assumes that each class is equally confusable with all other classes. We have discussed this problem in Chapter 4 in details. The second drawback is that the LDA cannot deal with non-linear discriminant information. To overcome these drawbacks, we propose the Non-linear Weighted Discriminant Analysis (NWDA) classification and dimensionality reduction algorithm by using a kernel function operator on the WLDA. The WLDA is generalized to NWDA by mapping the input space into a feature space

in which variables are non-linearly related to the input space. The formulation of the NWDA using dot product and matrix form is described and the eigenvalue resolution is successfully developed. In the simulation study, to make results more sense, the Fisher's iris data and the Leptograpsus Crabs data set, which are used in simulation of GDA [117], are used to perform dimensionality reduction. The proposed method is employed in confusable English letter recognition task and some issues concerning sampling data, selecting kernel functions and adjusting weighting functions are discussed. Conclusions are drawn in the final section.

5.2 Non-linear Weighted Discriminant Analysis

In LDA, the corresponding between-class and within-class scatter matrices are defined by (4.2.1) and (4.2.2). The optimality criterion of LDA is maximized as the (4.2.3).

To overcome the drawback that the optimality criterion based on scatter matrices is not directly related to classification accuracy, the between-class scatter matrix is rewritten as follows [107]:

$$S_b = \frac{1}{2M} \sum_{k=1}^N \sum_{l=1}^N \omega_{kl} n_k n_l (\bar{x}_k - \bar{x}_l)(\bar{x}_k - \bar{x}_l)^T \quad (5.2.1)$$

where $\bar{x}_l = \frac{1}{n_l} \sum_{p=1}^{n_l} x_{lp}$, ω_{kl} is a non-negative weighting function assigned to class pair (k, l) and it represents how important it is to discriminate class k from class l .

General LDA is not sophisticated enough to deal with real-world applications such as speech and image recognition. Meanwhile, we could have a lot more classes than just two and there are very different numbers of points in each class.

Therefore, it is much more important to include non-linear properties of the feature space in LDA for dimensionality reduction and classification problems.

In the NWDA method, we map the data non-linearly into some feature space and perform LDA there in order to find non-linear directions and increase the discriminant power of the analysis. The kernel operator allows the construction of non-linear separating function in the input space which is equivalent to a linear separating function in the feature space Ψ^m . Considering that the space Φ^n is mapped into a Hilbert space Ψ^m through a non-linear mapping function W , we have $z = W(x)$, $W : \Phi^n \rightarrow \Psi^m$.

The NWDA is similar to the general LDA in that it also maximizes the between-class inertia and minimizes the within-class inertia. This maximization is equivalent to the following eigenvalue resolution problem, i.e. to find eigenvalues λ and eigenvectors η from the following equation:

$$\lambda S_w^f \eta = S_b^f \eta \quad (5.2.2)$$

where $\eta \in \Psi^m$, S_w^f and S_b^f represent the within-class scatter matrix and the between-class scatter matrix in the space Ψ^m respectively.

The largest eigenvalue of (5.2.2) gives the maximum value of the following quotient of the inertia:

$$\lambda = \frac{\eta^T S_b^f \eta}{\eta^T S_w^f \eta} \quad (5.2.3)$$

As the eigenvectors are linear combinations of Ψ^m elements, there exist coefficients γ_{pq} ($p = 1, \dots, N; q = 1, \dots, n_p$) such that all solutions, η can be expressed as

$$\eta = \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} W(x_{pq}) \quad (5.2.4)$$

i.e. η lie in the span of $W(x_{pq})$.

Assuming that the observations are centered in Ψ^m [117], the terms S_b^f and S_w^f can be written as follows:

$$\begin{aligned} S_b^f &= \frac{1}{2M} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \left(\bar{W}_k \bar{W}_k^T - \bar{W}_k \bar{W}_l^T - \bar{W}_l \bar{W}_k^T + \bar{W}_l \bar{W}_l^T \right) \\ &= \frac{1}{M} \left(\frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_k^T - \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_l^T \right. \\ &\quad \left. - \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_k^T + \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_l^T \right) \end{aligned} \quad (5.2.5)$$

$$S_w^f = \frac{1}{M} \sum_{l=1}^N \sum_{k=1}^{n_l} W(x_{lk}) W^T(x_{lk}) \quad (5.2.6)$$

where $\bar{W}_k = \frac{1}{n_k} \sum_{p=1}^{n_k} W(x_{kp})$ and $\bar{W}_l = \frac{1}{n_l} \sum_{p=1}^{n_l} W(x_{lp})$.

Equation (5.2.2) has the same eigenvector as the following equation:

$$\lambda W^T(x_{ij}) S_w^f \eta = W^T(x_{ij}) S_b^f \eta \quad (5.2.7)$$

Using this formula for all class i and for all its element j , the equation (5.2.7) can be rewritten as:

$$\lambda W_{str} S_w^f \eta = W_{str} S_b^f \eta \quad (5.2.8)$$

To be convenient, $(W^T(x_{11}), \dots, W^T(x_{1n_1}), \dots, W^T(x_{ij}), \dots, W^T(x_{N1}), \dots, W^T(x_{Nn_N}))$ is represented by W_{str} in equation (5.2.8).

The right hand side of equation (5.2.8) can be written as:

$$\begin{aligned}
 W_{str} S_b^f \eta &= W_{str} \frac{1}{M} \left(\frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_k^T - \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_l^T \right. \\
 &\quad \left. - \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_k^T + \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_l^T \right) \eta \\
 &= \frac{1}{M} \left(\frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_k^T \eta - \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_l^T \eta \right. \\
 &\quad \left. - \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_k^T \eta + \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_l^T \eta \right) \\
 &= \frac{1}{M} (T_1 - T_2 - T_3 + T_4)
 \end{aligned} \tag{5.2.9}$$

By substituting (5.2.4) into (5.2.9), we obtain T_1 , T_2 , T_3 and T_4 as follows:

$$\begin{aligned}
 T_1 &= \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_k^T \eta \\
 &= \frac{W_{str}}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \left(\sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_k^T \right) W(x_{pq}) \\
 &= \frac{1}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l W_{str} \left[\frac{1}{n_k} \sum_{m=1}^{n_k} W(x_{km}) \right] \left[\frac{1}{n_k} \sum_{n=1}^{n_k} W^T(x_{kn}) \right] W(x_{pq}) \\
 &= A \alpha
 \end{aligned} \tag{5.2.10}$$

where A is a $M \times M$ matrix defined by

$$\begin{aligned}
 A_{ij} &= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \frac{n_l}{n_k} \left[\sum_{s=1}^M W^T(x_{is}) \sum_{p=1}^{n_k} W(x_{kp}) \right] \left[\sum_{q=1}^{n_k} W^T(x_{kq}) \sum_{s=1}^M W(x_{sj}) \right] \\
 &= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \frac{n_l}{n_k} \left(\sum_{p=1}^{n_k} k_{ip} \right) \left(\sum_{q=1}^{n_k} k_{qj} \right)
 \end{aligned} \tag{5.2.11}$$

and k_{ij} is defined according to (5.2.19).

$$\begin{aligned}
T_2 &= \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_l^T \eta \\
&= \frac{W_{str}}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \left(\sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_k \bar{W}_l^T \right) W(x_{pq}) \\
&= \frac{1}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l W_{str} \left[\frac{1}{n_k} \sum_{m=1}^{n_k} W(x_{km}) \right] \left[\frac{1}{n_l} \sum_{n=1}^{n_l} W^T(x_{ln}) \right] W(x_{pq}) \\
&= B\alpha
\end{aligned} \tag{5.2.12}$$

where B is a $M \times M$ matrix defined by

$$\begin{aligned}
B_{ij} &= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \left[\sum_{s=1}^M W^T(x_{is}) \sum_{p=1}^{n_k} W(x_{kp}) \right] \left[\sum_{q=1}^{n_l} W^T(x_{lq}) \sum_{s=1}^M W(x_{sj}) \right] \\
&= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \left(\sum_{p=1}^{n_k} k_{ip} \right) \left(\sum_{q=1}^{n_l} k_{qj} \right);
\end{aligned} \tag{5.2.13}$$

$$\begin{aligned}
T_3 &= \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_k^T \eta \\
&= \frac{W_{str}}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \left(\sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l \bar{W}_l \bar{W}_k^T \right) W(x_{pq}) \\
&= \frac{1}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l W_{str} \left[\frac{1}{n_l} \sum_{m=1}^{n_l} W(x_{lm}) \right] \left[\frac{1}{n_k} \sum_{n=1}^{n_k} W^T(x_{kn}) \right] W(x_{pq}) \\
&= C\alpha
\end{aligned} \tag{5.2.14}$$

where C is a $M \times M$ matrix defined by

$$\begin{aligned}
C_{ij} &= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \left[\sum_{s=1}^M W^T(x_{is}) \sum_{p=1}^{n_l} W(x_{lp}) \right] \left[\sum_{q=1}^{n_k} W^T(x_{kq}) \sum_{s=1}^M W(x_{sj}) \right]; \\
&= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \left(\sum_{p=1}^{n_l} k_{ip} \right) \left(\sum_{q=1}^{n_k} k_{qj} \right);
\end{aligned} \tag{5.2.15}$$

$$\begin{aligned}
T_4 &= \frac{W_{str}}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \overline{W}_l \overline{W}_l^T \eta \\
&= \frac{W_{str}}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \left(\sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \overline{W}_l \overline{W}_l^T \right) W(x_{pq}) \\
&= \frac{1}{2} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} n_k n_l W_{str} \left[\frac{1}{n_l} \sum_{m=1}^{n_l} W(x_{lm}) \right] \left[\frac{1}{n_l} \sum_{n=1}^{n_l} W^T(x_{ln}) \right] W(x_{pq}) \\
&= D\alpha
\end{aligned} \tag{5.2.16}$$

where D is a $M \times M$ matrix defined by

$$\begin{aligned}
D_{ij} &= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \frac{n_k}{n_l} \left[\sum_{s=1}^M W^T(x_{is}) \sum_{p=1}^{n_l} W(x_{lp}) \right] \left[\sum_{q=1}^{n_k} W^T(x_{lq}) \sum_{s=1}^M W(x_{sj}) \right]; \\
&= \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^N \omega^f_{kl} \frac{n_k}{n_l} \left(\sum_{p=1}^{n_l} k_{ip} \right) \left(\sum_{q=1}^{n_k} k_{qj} \right);
\end{aligned} \tag{5.2.17}$$

The left side of (5.2.8) can be written as:

$$\begin{aligned}
\lambda W_{str} S_w^f \eta &= \frac{\lambda}{M} W_{str} \sum_{l=1}^N \sum_{k=1}^{n_l} W(x_{lk}) W^T(x_{lk}) \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} W(x_{pq}) \\
&= \frac{\lambda}{M} \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} \sum_{l=1}^N \sum_{k=1}^{n_l} [W_{str} W(x_{lk})] [W^T(x_{lk}) W(x_{pq})] \\
&= \frac{\lambda}{M} K K \gamma
\end{aligned} \tag{5.2.18}$$

Let K be a $M \times M$ matrix defined on the class elements given by $(K_{ij})_{\substack{i=1,\dots,N \\ j=1,\dots,N}}$,

where (K_{ij}) is a matrix composed of dot product $(k_{pq})_{\substack{p=1,\dots,n_p \\ q=1,\dots,n_q}}$ in the feature space

Ψ^m . The kernel function is expressed as follows:

$$(k_{pq})_{ij} = k(x_{ip}, x_{jq}) = W^T(x_{ip}) W(x_{jq}) \tag{5.2.19}$$

In order to keep sufficient discriminant information, we need to select a weighting function. We select a natural candidate equal to the square of the inverse of the Euclidean distance between class means as a weighting function as follows:

$$\begin{aligned}\omega_{kl} &= \frac{1}{\|\bar{x}_k - \bar{x}_l\|^2} \\ &= \frac{1}{(\bar{x}_k - \bar{x}_l)^T (\bar{x}_k - \bar{x}_l)} \quad k, l=1, \dots, N\end{aligned}\quad (5.2.20)$$

The classes which have their means closer to each other are weighted more than the ones which have means farther away. In other words, more confusable classes are weighted more and vice versa.

After converting ω_{kl} into the Ψ^m domain, we obtain the transformed ω_{kl}^f as follows:

$$\begin{aligned}\omega_{kl}^f &= \frac{1}{\|\bar{W}_k - \bar{W}_l\|^2} \\ &= \frac{1}{(\bar{W}_k - \bar{W}_l)^T (\bar{W}_k - \bar{W}_l)} \\ &= \frac{1}{\bar{W}_k^T \bar{W}_k - \bar{W}_k^T \bar{W}_l - \bar{W}_l^T \bar{W}_k + \bar{W}_l^T \bar{W}_l}\end{aligned}\quad (5.2.21)$$

According to (5.2.19), we obtain

$$\begin{aligned}\bar{W}_k^T \bar{W}_k &= \frac{1}{n_k^2} \sum_{p=1}^{n_k} \sum_{q=1}^{n_k} W^T(x_{kp}) W(x_{kq}) \\ &= \frac{1}{n_k^2} \sum_{p=1}^{n_k} \sum_{q=1}^{n_k} (k_{pq})_{kk}\end{aligned}$$

$$\begin{aligned}
\overline{W}_k^T \overline{W}_l &= \frac{1}{n_k n_l} \sum_{p=1}^{n_k} \sum_{q=1}^{n_l} W^T(x_{kp}) W(x_{lq}) \\
&= \frac{1}{n_k n_l} \sum_{p=1}^{n_k} \sum_{q=1}^{n_l} (k_{pq})_{kl} \\
\overline{W}_l^T \overline{W}_k &= \frac{1}{n_k n_l} \sum_{p=1}^{n_l} \sum_{q=1}^{n_k} W^T(x_{lp}) W(x_{kq}) \\
&= \frac{1}{n_k n_l} \sum_{p=1}^{n_l} \sum_{q=1}^{n_k} (k_{pq})_{lk} \\
\overline{W}_l^T \overline{W}_l &= \frac{1}{n_l^2} \sum_{p=1}^{n_l} \sum_{q=1}^{n_l} W^T(x_{lp}) W(x_{lq}) \\
&= \frac{1}{n_l^2} \sum_{p=1}^{n_l} \sum_{q=1}^{n_l} (k_{pq})_{ll}
\end{aligned}$$

Therefore,

$$\omega_{kl}^f = \frac{1}{\frac{1}{n_k^2} \sum_{p=1}^{n_k} \sum_{q=1}^{n_k} (k_{pq})_{kk} - \frac{1}{n_k n_l} \sum_{p=1}^{n_k} \sum_{q=1}^{n_l} (k_{pq})_{kl} - \frac{1}{n_k n_l} \sum_{p=1}^{n_l} \sum_{q=1}^{n_k} (k_{pq})_{lk} + \frac{1}{n_l^2} \sum_{p=1}^{n_l} \sum_{q=1}^{n_l} (k_{pq})_{ll}}$$

(5.2.22)

Combining all the equations above, we obtain:

$$\lambda KK\gamma = (A - B - C + D)\gamma$$

(5.2.23)

Multiplying both sides of equation (5.2.23) by γ^T , we obtain the largest eigenvalue

$$\lambda = \frac{\gamma^T (A - B - C + D) \gamma}{\gamma^T KK \gamma}$$

(5.2.24)

We can find the discriminant information in the transformed space Ψ by finding the leading eigenvector of $(KK)^{-1}(A - B - C + D)$. If we want to have m dimensional space Ψ^m , we can find the m leading eigenvectors of $(KK)^{-1}(A - B - C + D)$ and put them in a matrix, R . Where each γ is a row in R .

The coefficients γ are normalized by requiring that η be normalized in Ψ i.e. $\eta^T \eta = 1$. Using (5.2.4), we have

$$\begin{aligned} \eta^T \eta &= \sum_{p=1}^N \sum_{q=1}^{n_p} \sum_{l=1}^N \sum_{h=1}^{n_l} \gamma_{pq} \gamma_{lh} W^T(x_{pq}) W(x_{lh}) \\ &= \sum_{p=1}^N \sum_{l=1}^N \gamma_p^T K_{pl} \gamma_l \\ &= \gamma^T K \gamma = 1 \end{aligned} \quad (5.2.25)$$

The normalized vectors η are calculated as follows:

$$\eta = \frac{\gamma}{\sqrt{\gamma^T K \gamma}} \quad (5.2.26)$$

Therefore, we can compute projections of a test point t as follows:

$$(\eta^T \cdot W(t)) = \sum_{p=1}^N \sum_{q=1}^{n_p} \gamma_{pq} k(x_{pq}, t) \quad (5.2.27)$$

The procedure of the proposed NWDA is summarized here. We first compute the matrices K , A , B , C and D according to (5.2.19), (5.2.11), (5.2.13), (5.2.15) and (5.2.17) respectively. After decomposing the $(KK)^{-1}(A - B - C + D)$ using eigenvector decomposition, we obtain m -dimensional space Ψ^m . Then, compute eigenvectors η in term of (5.2.26) and compute projections of test points onto the eigenvectors η according to (5.2.27).

5.3 Simulation Studies

In this section, the Fisher's iris data and the Leptograpsus Crabs data set are first studied using the proposed NWDA method. To evaluate the performance of our new approach, we performed an extensive comparison with other state-of-the-art dimensional reduction and classification methods. Classical LDA, WLDA and GDA were employed as a baseline. To apply NWDA and GDA to the data, we need to decide on two important parameters.

Firstly, a good kernel function should be decided in the NWDA and the GDA method. Some kernel functions e.g. Gaussian kernel $k(x, y) = \exp(-\frac{\|x - y\|^2}{\sigma})$ or polynomial kernel $k(x, y) = (a + b \times x \cdot y)^d$ for some constants a , b , c and d have been proven useful. Usually, the best performance is determined by choosing some appropriate constants. We utilize a Gaussian kernel function in all experiments.

Secondly, it is very important to determine suitable weighting functions in the WLDA and the NWDA. In our experiments, the weighting function is selected in terms of ω_{kl}^p according to equation (5.2.20). We adjust the value of power p to be 1, 2 and 4 in all simulation studies.

5.3.1 Fisher's Iris Data

Fisher published the iris flower data for discriminant analysis and cluster analysis in 1936 [118]. There are three classes, namely iris setosa, iris versicolor and iris

virginica. Sepal length, sepal width, petal length, and petal width are four parameters that were measured on fifty iris specimens from each of the three species. Two classes are not linearly separable from each other, but the other class is linearly separable from the other two. In the experiments, all iris examples are centered at first. We choose a two-dimensionality output space reduced by the LDA, the WLDA, the GDA and the proposed NWDA method.

Figure 5.1: Projection of iris data on the first two axes. The three classes are shown by various symbols.

Figure 5.2: 2D representations of iris data reduced by the LDA method.

Figure 5.3: 2D representations of iris data reduced by the WLDA method with a weighting function (power=1).

Figure 5.4: 2D representations of iris data reduced by the WLDA method with a weighting function (power=2).

Figure 5.5: 2D representations of iris data reduced by the WLDA method with a weighting function (power=4).

Figure 5.6: 2D representations of iris data reduced by the GDA method with a Gaussian kernel ($\sigma = 0.7$).

Figure 5.7: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=1).

Figure 5.8: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=2).

Figure 5.9: 2D representations of iris data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=4).

The original Fisher's iris data space is shown on the first two axes in Figure 5.1. Figure 5.2 shows 2D representations of the examples using the LDA. It is insufficient for the LDA to separate the three classes of Fisher's iris data when reduced to a 2D subspace. Figure 5.3 - Figure 5.5 show 2D representations of the examples using the WLDA method and the power value of the weighting function is chosen as 1, 2 and 4 respectively. The WLDA is also not good at classifying the Fisher's iris data according to the results. For the WLDA method, the separation results are not obviously influenced by different power values of weighting functions according to the simulation results. The result using the GDA method is shown in Figure 5.6. We can see that the GDA can produce well-separated classes on the 2D subspace. As can be seen from Figure 5.7, Figure 5.8 and Figure 5.9, the three classes are very well separated under the NWDA method than those under the LDA, the WLDA and the GDA methods. With two axes, each class is nearly projected on one point and has great distances to others. We can see that the classes are very well separated because the NWDA can produce a different representation that is dependent on the distances of the between-class inertia. The separation results are not obviously influenced by different power values of weighting functions according to Figure 5.7-Figure 5.9.

5.3.2 Leptograpsus Crabs Data Set

For this simulation, the Leptograpsus Crabs data set available from the FTP archived at the Department of Statistics, University of Oxford, United Kingdom was used [121]. The data set consists of four classes each containing 50 instances. Each instance has five features and all the data are used for training. To clearly show the performance of different dimensionality reduction methods, we project training patterns into two dimension and compare classification accuracies directly through 2D representations.

Figure 5.10: Projection of Leptograpsus Crabs data on the first two axes. The four classes are shown by various symbols.

Figure 5.11: 2D representations of Leptograpsus Crabs data reduced by the LDA method.

Figure 5.12: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=1).

Figure 5.13: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=2).

Figure 5.14: 2D representations of Leptograpsus Crabs data reduced by the WLDA method with a weighting function (power=4).

Figure 5.15: 2D representations of Leptograpsus Crabs data reduced by the GDA method with a Gaussian kernel ($\sigma = 0.7$).

Figure 5.16: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=1).

Figure 5.17: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=2).

Figure 5.18: 2D representations of Leptograpsus Crabs data reduced by the NWDA method with a Gaussian kernel ($\sigma = 0.7$) and a weighting function (power=4).

The original Leptograpsus Crabs data is shown on the first two axes in Figure 5.10. The four classes are shown by various symbols. Figure 5.11 shows 2D representations of the examples projected by the LDA method. From the result, we can see that the LDA method is incapable of separating the four classes of Leptograpsus Crabs data when reduced to a 2D subspace. In order to show the effectiveness of different weighting functions of the WLDA method, the simulation results based on different power values of weighting functions are displayed in Figure 5.12 - Figure 5.14. From the simulation results by using the WLDA when the power value of the weighting function is set as 1, 2 and 4 respectively, we can see that the WLDA cannot separate the four classes of Leptograpsus Crabs data when reduced to a 2D subspace. We can also observe that the performance of classification become worse when the power value of the weighting function is increased. From Figure 5.15, the result of classification

based on the GDA method is much better than those of the LDA method and the WLDA method because the GDA includes non-linear discriminant information of the feature space. However, it is insufficient for the GDA to completely separate all the four classes of Leptograpsus Crabs data when reduced to a 2D subspace. Figure 5.16- Figure 5.18 show the classification results of applying the NWDA when the power value of the weighting function is also set to be 1, 2 and 4 respectively. Classification accuracies projected by the NWDA algorithm show the best separation of different classes because of the use of the non-linear discriminant analysis and the weighting function. We can also see that the classification results based on the NWDA method are robust to different power values of weighting functions.

5.4 Speech Recognition Experiments

In this section, we try to apply the proposed NWDA dimensionality reduction method in a simple speech recognition system. We give a particular description of solutions for the problems and difficulties, which arose in implementing the speech recognition system.

One obstacle not overcome by discrete or isolated letter speech recognition is the high confusability of individual letters. Although the perplexity of the letters is only 26, many letters are so similar as to be indistinguishable. The most obvious groupings are those consonant letters sharing a vowel in their individual pronunciation, such as the EH set: M, N, F and S [119][120]. (An “A set” and “E set” exist as well.) The vowel sound dominates the pronunciation with only minimal influence from the consonants; thus the letters in the EH set are difficult to distinguish.

The proposed method is performed on a confusable English letters group, i.e. the EH set. The speech data are based on the TI46 corpus, which is designed and collected at the Texas Instruments (TI). The TI46 corpus contains 16 speakers: 8 males and 8 females. There are 15 utterances of each English letter from each speaker: ten is designated as training tokens and five are designed as testing tokens in the proposed system.

The kernel function and the weighting function are chosen the same as those in the experiments of Section 5.3. After calculating 32-dimensional MFCCs on the speech data set, thirty-two continuous valued features are extracted. Therefore, the original dimensionality of the speech feature space is 32. To demonstrate the effect of the proposed NWDA, we apply the LDA, the WLDA, the GDA and the NWDA on our speech recognition system to reduce the feature space from 32 to 16 respectively. Training accuracies were measured by the method of HMM. A five-state left-to-right HMM is utilized to perform training and classification. Before applying dimensionality reduction methods, we need to label the training data first. The class label of each word is the HMM state sequence obtained by performing Viterbi alignment over the training data. Therefore, after labeling each training data, there are altogether 20 classes: M1, M2, M3, M4, M5, N1, N2, N3, N4, N5, F1, F2, F3, F4, F5, S1, S2, S3, S4 and S5.

The procedure of applying NWDA to our speech recognition system is as follows:

- Step 1: Estimate HMM parameters in the original data space using Baum-Welch algorithm and get a baseline result;
- Step 2: Perform Viterbi alignment over the training data to get class labels, i.e., the HMM state sequence with the corresponding word;
- Step 3: Compute the eigenvectors η in terms of the steps of NWDA;
- Step 4: Transform the original data into the space Ψ^m , where $m=16$;

Step 5: Re-estimate HMM parameters in Ψ^m and obtain the decoding results. Note that we will repeat Step 3-5 for every kernel function.

5.4.1 Sampling Method

When we follow the above procedure, there are some implementation issues with our speech recognition system. We have almost up to 29,000 data samples in the training set, which means the size of matrices S_b^f and S_w^f of the GDA and the NWDA will become gigantic in the training procedure. To invert such a gigantic matrix and to perform multiplications is impractical. Hence, in order to demonstrate the performance of our new method more directly and effectively, we sample the training data so as to reduce the size of matrices. Our sampling method is to approximately maximize the within-class variances while approximately keeps the within-class means the same as those before sampling.

Figure 5.19: 2D representations before sampling the points of class. Where h represents the width of zone for deleting points and H represents the width of zone for keeping points.

Figure 5.20: 2D representations after sampling the points of class

Before presenting the sampling algorithm, some nomenclatures are established. Let $X_p = \{x_{p1}, x_{p2}, \dots, x_{pn_p}\}$ represent the p -th class of the input feature vectors where n_p is the number of points in class p . The complete set of feature vectors will then be $X = \{X_1, X_2, \dots, X_p\}$. The center of the p -th class of the input feature vector is $\bar{x}_p$. The procedure of sampling the points of classes is as follows:

Step 1: Calculate the center of each class $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p$;

Step 2: In class p , find the distance d_p from the class center to the point that is the farthest away from the center;

Step 3: In class p , delete point x_{pm} that satisfies the following conditions:

$$\begin{aligned} \|x_{pm} - \bar{x}_p\| &\leq \left\| \bar{x}_p + n * \frac{d_p}{delt_n} + \frac{d_p}{delt_d * delt_n} \right\|, \text{ and} \\ \|x_{pm} - \bar{x}_p\| &\geq \left\| \bar{x}_p + n * \frac{d_p}{delt_n} - \frac{d_p}{delt_d * delt_n} \right\|, n = 1, 2, \dots, delt_n, \end{aligned} \quad (5.4.1)$$

where $delt_d, delt_n > 0$.

Step 4: Repeat Step 2 and Step 3 for every class.

Through adjusting the parameters $delt_n$ and $delt_d$, we can change the range and quantity of sampling data easily and quickly.

To demonstrate the effect of our sampling method, we apply it on the Leptograpsus Crabs data set and the Fisher's iris data set. Figure 5.21 and Figure 5.24 show the projection of original points in the data set on the first two axes. Figure 5.22 and Figure 5.25 show the results of applying the sampling method with $delt_n = 3$ and $delt_d = 3$. From Figure 5.23 and Figure 5.26, the number of points in the Leptograpsus Crabs data set is successfully reduced from 200 to 69 and the number of points in the iris data set is reduced from 150 to 66.

Figure 5.21: Projection of Leptograpsus Crabs data on the first two axes. The four classes are shown by various symbols and each contains 50 points.

Figure 5.22: Projection of sampled Leptograpsus Crabs data on the first two axes when set $delt_n=3$ and $delt_d=3$. The number of points in Class 1 is 32, in Class 2 is 29, in Class 3 is 32 and in Class 4 is 32 after sampling.

Figure 5.23: Projection of sampled *Leptograpsus* Crabs data on the first two axes when set $\text{delt_n}=5$ and $\text{delt_d}=5$. The number of points in Class 1 is 14, in Class 2 is 14, in Class 3 is 27 and in Class 4 is 14 after sampling.

Figure 5.24: Projection of iris data on the first two axes. The three classes are shown by various symbols and each contains 50 instances.

Figure 5.25: Projection of sampled iris data on the first two axes when set $delt_n=3$ and $delt_d=3$. The number of points in Class 1 is 26, in Class 2 is 27 and in Class 3 is 32 after sampling.

Figure 5.26: Projection of sampled iris data on the first two axes when set $delt_n=5$ and $delt_d=5$. The number of points in Class 1 is 25, in Class 2 is 22 and in Class 3 is 19 after sampling.

5.4.2 Experiment Result

In order to evaluate the performance of the NWDA, we employ classical LDA, the WLDA and the GDA on the same speech recognition task and reduce the dimensionality of the feature space from 32 to 16. To test the robust performance of the proposed speech recognition system under noisy environments, we add white Gaussian noises to the original clear speech data set according to different Signal-to-Noise Ratio (SNR) values.

The SNR is defined as

$$10\log_{10}\left(\sum_n x^2(n)/\sum_n w^2(n)\right) \quad (5.4.2)$$

where the summation is over the entire length of the original signal.

In order to really implement the GDA and the NWDA, we reduce the number of training data samples from 29049 to 1122 in the EH set by using the proposed sampling method, and the parameters $delt_n$ and $delt_d$ of the sampling method are both equal to 50. The number of training data can be adjusted by changing the $delt_n$ and $delt_d$. It is important to determine suitable weighting functions in the WLDA and the NWDA. In our experiments, the weighting function is selected in terms of ω_{kl}^p according to equation (5.2.20). We adjust the value of power p to be 1, 2 and 4 in all simulation studies.

The recognition results of speech-word error rate were obtained under quiet and noisy environments. The SNR values of noisy environments are equal to 10 and 20 respectively. The test results are displayed in Table 5.1.

Method \ SNR (dB)	Quiet	20DB SNR	10DB SNR
LDA	30%	35%	41%
WLDA (power=1)	27%	33%	42%
WLDA (power=2)	29%	35%	45%
WLDA (power=4)	30%	40%	48%
GDA	25%	30%	35%
NWDA (power=1)	20%	26%	30%
NWDA (power=2)	23%	30%	32%
NWDA (power=4)	30%	38%	40%

Table 5.1: Comparison of error rates based on various methods under different SNR for the EH set

Table 5.1 shows the error rates that are based on the LDA, the WLDA, the GDA and the proposed NWDA method for recognizing the EH set. Here the columns represent different methods used. For example, the LDA represents the error rates in all ranges of testing data set including clear and lower SNRs level when the LDA is utilized to reduce the dimensionality of the feature space from 32 to 16.

From the error rates for recognizing the EH set, we can see that the performance of the WLDA and the LDA is almost the same when the power value is set 1. The

error rates of the WLDA are worse than that of the LDA when the power value is set at 2 and 4. It shows that the choice of weighting function is an important step in the WLDA method. The GDA method obtains clearly lower error rates than those of the LDA and the WLDA methods because it includes non-linear properties of the feature space. The best error rates come from the NWDA when the power value is set at 1. It shows that the choice of weighting function is also an important step in the NWDA method. From the error rates for recognizing the EH set, the NWDA obtain 7%, 7% and 12% more performance over those of the best WLDA when the power value is set at 1. Comparing with the GDA method, the NWDA also obtain 5%, 4% and 5% performance promotion. Moreover, although all these results are only based on a very limited training set, they still show that the NWDA is better than the LDA and the WLDA on this simple speech recognition tasks and can be better than GDA on such tasks. Considering the complexity of the NWDA, extra computation load was imposed by this method and it will reduce the speed of classification process. However, the NWDA keeps more useful non-linear information and removes more unimportant information when reducing the dimensionality of the feature space. Therefore, the performance of the NWDA is better than those of the LDA and the GDA. Without considering the computation load, we expect to obtain better speech recognition results if we increase the number of training patterns considerably.

5.5 Discussions and Conclusions

5.5.1 Discussions

In speech recognition and image processing, the dimensionality of the feature space is huge and depends on the size of the input space. It is very likely that a

function can show good results in separating the training data but may not generalize well. Therefore, we need to find a compromise between the training and generalization performances for the training method. The test error of SVM depends only on the expectation of the number of support vectors and the number of training examples but not on the dimensionality of the feature space [113]. Our research is to establish the connection between general WLDA and SVM resolution on the basis of the GDA [117]. Using the SVM techniques [122], [114], we can improve the performance of NWDA in generalization, accuracy and speed. Furthermore, we can construct an optimal separating hyperplane in the feature space using a kernel approach. We only need to compute the dot product without considering the details about this space. It is still not easy to select a proper kernel function and weighting function in the proposed NWDA. From the essence of the methods based on a kernel function, we can find some similarities between Probabilistic Neural Networks (PNN) and NWDA (and GDA) with Gaussian kernel. PNN can also be regarded as a mapping function based on a set of input-output observations very like the NWDA or GDA method. Like more general neural networks, PNN always finds a local minimum and it finds separation by trial-and-error measurement on the training set. Trial and error is a method for obtaining knowledge, both propositional knowledge and know-how. In trial and error, one tries an option to see if it works. If it works, then we have a solution. If it doesn't work - there is an error - then one tries another option. In NWDA and GDA, a hyperplane separation is defined by a set of vectors obtained by the eigenvalue resolution, and a global minimum is given according to the optimum criterion.

5.5.2 Conclusions

In this chapter, a NWDA strategy for non-linear discrimination analysis is proposed. The motivation for exploring the algebraic approach is to develop an

exact solution and not an approximate optimization. The NWDA method gives an exact solution even if some points require further investigation, such as the choice of the weighting function and the kernel function. Simulation studies based on the Fisher's iris data, the Leptograpsus Crabs data set and the confusable English letters speech recognition experiments were carried out successfully. To apply the NWDA on the speech recognition and reduce the computation load, a sampling method is proposed and the working principle of the algorithm are discussed in this chapter. In terms of classification performance and recognition results, for the small databases studied here, the NWDA method is better than the LDA, the WLDA and the GDA method. Making the proposed system computationally feasible for the complicated problem, such as large vocabulary speech recognition task, is also one of the important objectives in future.

Chapter 6

Multiple Resolution Analysis in Speech Feature Extraction

6.1 Introduction

Several adverse conditions are often present during operation, such as ambient and transmission noise, distortions due to room acoustics and transducers, and even changes in speech characteristics due to psychological awareness of talking to a machine. These conditions need to be dealt with in order for the recognizer to be able to deliver reliable results. Thus, the issue of robustness is one of the important problems in automatic speech recognition. One method that achieves robust results is to extract the most representative feature from the speech utterance in bad environment conditions. Feature extraction of speech is one of the most important issues in the field of speech recognition. In order to achieve high recognition accuracy, the feature extractor is required to discover salient characteristics suited for classification.

There are two dominant acoustic measurements of speech signals. One, which occurs in the temporal domain, is the parametric modeling approach, which is developed to match closely the resonant structure of the human vocal tract that produces the corresponding speech sound. It is mainly derived from linear predictive analysis, such as the Linear Predictive Coding (LPC) method. The other method, which occurs in the frequency domain, is the nonparametric modeling method originated from the human auditory perception system. FFT-based Mel-scale Frequency Cepstral Coefficients (MFCCs) are typically utilized for this purpose [123]. Although it is widely acknowledged that MFCCs do not cope with noisy speech well, the MFCCs representation of speech is being standardized for cellular telephone systems nowadays [124]. Many alternate feature extraction methods have also been proposed for speech recognition over the years. A set of Perceptual Linear Prediction (PLP) coefficients [125], which attempt to incorporate some of the features of the human auditory mechanism, has been suggested for robust automatic speech recognition. Combined with RASTA filtering, these features have been shown to perform better under noisy conditions [126]. In [127], an auditory model-based front-end that relies on short-term adaptation proved to be more robust than conventional MFCCs features of name dialing applications. Some kind of speech enhancement techniques, such as non-linear spectral subtraction [128], have also been applied to ASR without any greater success. Different cepstral-domain parameter normalization techniques, e.g., Cepstral Mean Normalization (CMN) [129] and feature vector normalization [130], have also turned out to be efficient for compensating the effects of the communication channel. Wavelets are functions with compact support capable of representing signals with good time and frequency resolution. Much research has been done on wavelet analysis for speech recognition [131]-[133]. The wavelet

based sub-band features has shown better performance than that of MFCCs [134][135].

In spite of these developments, effective feature extraction is certainly far from being a solved problem. The LPC method is not optimal because the underlying speech production model is non-linear. LPC starts with the assumption that the speech signal is produced by a buzzer at the end of a tube. For ordinary vowels, the vocal tract is well represented by a single tube. However, for nasal sounds, the nose cavity forms a side branch. In practice, this difference is partly ignored and partly dealt with during the encoding of the residue. In particular, the MFCCs, which are based on human auditory perception, have been used extensively for speech recognition. Fundamentally, it is not clear whether all of the information necessary to distinguish between words is retained in the Mel-cepstrum. Most wavelet analysis for speech recognition suffer from one main drawbacks which filter banks implementing wavelet transforms are typically dyadic, splitting the spectrum in half.

In this thesis, we propose a new and robust speech feature extraction method based on time-frequency multiresolution analysis. First, Mel-scale-like Wavelet Packets (WP) decomposition is applied to decompose the input signal into various uncorrelated frequency channels. The decomposition of the input signal into some “approximation” and “detailed” spaces is called the multiresolution approximation. It is obvious that the multiresolution features can be easily extracted from these frequency channels only by using their related WP coefficients. For capturing the key information in the speech signal, the Linear Predictive Cepstral Coefficients (LPCCs) of the lower-frequency channels and the log energy value of the highest-frequency channel are calculated respectively. To grasp more information in the signals, we combine the feature from the all-band MFCCs (or LPCCs) with the

subband feature into a single feature vector before decoding. By using the step-weighted linear discriminant dimensionality reduction technique which is proposed in Chapter 4, more discriminant information is retained and lower dimensionality is attained.

6.2 MFCCs and WP Transform-Based Features

6.2.1 Review of WP Transform

Basically, wavelet transform can decompose a signal space A_i into a lower resolution (approximation) space A_{i+1} and a detailed space D_{i+1} by dividing the orthogonal basis $(\phi_i(t - 2^i n))_{n \in Z}$ of A_i into two new orthogonal bases, namely $(\phi_{i+1}(t - 2^{i+1} n))_{n \in Z}$ of A_{i+1} and $(\psi_{i+1}(t - 2^{i+1} n))_{n \in Z}$ of D_{i+1} where Z is a set of integers, and $\phi(t)$ and $\psi(t)$ are scaling and wavelet functions respectively [136].

The WP decomposes the approximate space as well as the detailed space. Each subspace in the tree is indexed by its depth i and the number of subspaces p below it. The two WP orthogonal bases at a parent node (i, p) are defined by:

$$\psi_{i+1}^{2p}(k) = \sum_{n=-\infty}^{\infty} h[n] \psi_i^p(k - 2^i n) \quad (6.2.1)$$

$$\text{and } \psi_{i+1}^{2p+1}(k) = \sum_{n=-\infty}^{\infty} g[n] \psi_i^p(k - 2^i n) \quad (6.2.2)$$

where $h[n]$ is a low-pass filter and $g[n]$ is a high-pass filter given by

$$h[n] = \langle \psi_{i+1}^{2p}(u), \psi_i^p(u - 2^i n) \rangle \quad (6.2.3)$$

$$\text{and } g[n] = \langle \psi_{i+1}^{2^{p+1}}(u), \psi_i^p(u - 2^i n) \rangle \quad (6.2.4)$$

respectively. Thus, a balanced binary tree structure is formed having more than $2^{2^{i-1}}$ orthogonal bases by using full i level WP decomposition. WP decomposition can partition the frequency axis at the higher frequency side into smaller bands. From Figure 6.1, we can see that WP decomposition gives an over-complete set of basis like a tree structure. In our method, the problem is to select the best set of basis for a given signal. In other words, our task is to select the best partitioning of the frequency axis.

Figure 6.1: Wavelets packets decomposition tree. Here, $A(i, j)$ represents the i^{th} second level in depth and the j^{th} subspace in this level in the tree.

6.2.2 The Difference Between MFCCs and WP Transform-Based Parameters

It is well-known that a Discrete Fourier Transform (DFT) is computed in MFCCs. In WP transform-based method, a Wavelet Packet Transform (WPT) is applied. In Figure 6.2, a comparison of time-frequency tiling is shown for both the DFT and the WPT [137]. The DFT samples the Fourier Transform (FT) at equally spaced points in the frequency domain, whereas WPT is computed through time-domain filtering with a subsignal representation obtained from frequency components within each subband. If we consider band i , depending on the local changes in the FT of the frame in that band, the DFT coefficients will be different due to changes in environmental conditions and various speech characteristics for speaker's psychological state. However, the subsignals are relatively robust to local variations within a band since they reflect the overall spectral shape for a band. We can see that the WPT reflects global changes in the FT while being relatively immune to local changes within a particular band. It gives its best performance when it is applied to detection of short-time phenomena, discontinuities, or abrupt changes in a signal. Therefore, WPT can be considered to be a time-scale method embedded with the characteristic of frequency. WPT can also be considered a multiresolution approximation, and the multiresolution features can be easily extracted from these frequency channels only by using their related WP coefficients.

Figure 6.2: Comparison of DFT-based and WPT-based frequency analysis for parameterization

6.3 Multiresolution Features Based on Mel-scale-like Wavelet Packets (WP) Decomposition

Speech signals have very complex waveforms due to the composition of various frequency components. Therefore, how to define a representation that is well adapted for extracting the information content of a speech signal is an important issue in speech recognition systems. It is well-known that two available sources of information involved with time and frequency domains are inherent in speech signals. The sharp variations of signal amplitude are generally the most meaningful features in the time space. It means that complicated signals can be determined by their detailed contours. When the signal includes important structures that belong to different scales, to represent the signal more precisely, it is often very helpful to decompose the signal into a set of subsignals in different scales. As for the frequency domain, the middle frequency region is the dominant frequency channel, but different speakers may have different responses in all frequency regions. It may lose some useful information in the feature extraction process if the methods only consider fixed frequency channels. Therefore, using the multiresolution decomposition technique can decompose the speech signal into different resolution levels. The characteristics of multiple frequency channels and any change in the smoothness of the signal can then be detected to perfectly represent the signals.

The WP decomposition is able to segment the frequency axis and make uniform translation in time. We consider using this property to obtain different resolution frequency channels. In order to obtain more reasonable multiresolution decomposition and emulate human's ability in speech perception, the frequency axis can be divided similarly to the Mel scale by applying WP decomposition on the input signals. However, due to the dyadic nature of the WP, the exact

bandwidth as in the Mel scale cannot be obtained. The problem that arises is therefore how to divide the frequency axis by using the WP transform so that it matches closely with the Mel scale.

The Mel scale was previously derived from purely psycho-acoustic experiments [138]. For the Mel scale, the edge bandwidths were calculated based on

$$y = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (6.3.1)$$

where y is the Mel scale value and f is the bandwidth frequency in Hz.

We consider two different kinds of strategy to obtain multiresolution decomposition by applying Mel-scale-like WP transform. The Daubechies 8 tap wavelet is used in our WP decomposition.

(a) Strategy 1

Firstly, a 4-subband Mel-scale-like WP decomposition is implemented. Our database TI46 is sampled at 11.025 kHz and the bandwidth is 5.512 kHz. At the beginning, a full one-level WP decomposition is carried out. This partitions the frequency axis into two subbands, each of which has a bandwidth of 2756 Hz. The lower band of 0~2756 Hz is further decomposed by applying again a full one-level WP decomposition. This divides the 0~2756 Hz band into two subbands, each of which has a bandwidth of 1378 Hz. Finally, the frequency band of 0~1378 Hz is decomposed again by applying one-level WP decomposition, thereby giving two subbands of 689 Hz. The results are shown in Table 6.1.

Filter Number	Bandwidth Frequency (Hz)	Mel Scale	Mel Bandwidth
1	689	772	346
2	1378	1226	454
3	2756	1800	251
4	5512	2460	282

Table 6.1: Frequency bands for 4-subband Mel-scale-like WP Decomposition

(b) Strategy 2

We consider decomposing the signal space into 7 Mel-scale-like frequency bands using WP transform. It is the same as Strategy 1 whereby the speech data are sampled at 11.025 kHz and the bandwidth is 5.512 kHz. First, a full two-level WP decomposition is carried out. This partitions the frequency axis into four bands, and each of which has a bandwidth of 1378 Hz. The lowest band of 0~1378 Hz is further decomposed by applying a full one-level WP decomposition. This divides the 0~1378 Hz band into two subbands each of which has a bandwidth of 689 Hz. The frequency band of 0~689 Hz is further decomposed by applying one-level WP decomposition again, then giving two subbands of 344 Hz each. Next, the 1378~2756 Hz is selected to complete one-level decomposition. This gives two bands, each of bandwidth of 689 Hz. Finally, we obtain 7 subbands and the results are summarized in Table 6.2.

Filter Number	Bandwidth Frequency (Hz)	Mel Scale	Mel Bandwidth
1	344	426	426
2	689	772	346
3	1378	1226	454
4	2067	1549	323
5	2756	1800	251
6	4134	2178	378
7	5512	2460	282

Table 6.2: Frequency bands for 7-subband Mel-scale-like WP Decomposition

After performing decomposition by Mel-scale-like WP in Strategy 1 and Strategy 2, the division of the frequency axis is performed in a manner such that it matches closely with the Mel scale and results in independent multiresolution subband signals.

6.4 Emphasis of the Multiresolution Feature Extraction Method

According to Allen's paper [139], the human auditory system processes the features from different subbands independently and merging is carried out at some higher point of processing to produce the final decision. Therefore, the multiband feature extraction approach has become a new and exploratory area of speech recognition. The advantages of using multiband processing are multifold and have been described in earlier publications [140]-[142].

After performing decomposition by Mel-scale-like WP, the information within the signals will be partitioned into different resolution levels or different bands. Furthermore, the LPCCs within each band except for the highest frequency subband are calculated to capture the special information. The highest frequency subband includes most detailed information about the feature. Therefore, the LPCCs are not calculated about this subband and the log energy value of this subband is utilized in the proposed methods. The main reasons for using LPCCs are their good representation on the envelope of the speech spectrum of vowels and their simplicity. Consequently, an effective and robust speech feature for representing detailed characteristics of a speech can be constructed by using LPCCs of multiple channels.

Although some promising results can be achieved by the subband-based ASR systems, we still believe that performing full-band processing is very important when computing the feature vectors. The information loss on the correlation between various subbands is probably the major drawback of a pure subband-based approach. Therefore, the full-band features are not to be ignored, but should be combined with the subband features for maximizing the recognition accuracy [143]. In the proposed multiresolution feature extraction method, we try to combine the full-band MFCCs and the LPCCs in the multiresolution feature respectively.

It is also very important to decide at which temporal level the combination of subband features should be carried out. From Bourlard and Hermansky [140], [144], different classifiers for each band were used and the likelihood recombination was carried out at Hidden Markov Model (HMM) state, phone, or word level in the multi-band approach. Another approach which is the combination of individual features of each subband into a single feature vector prior to decoding

is also proposed in [145]. Our approach tries to combine features from the subband and the full-band MFCCs (or LPCCs) into a single feature vector before decoding. The advantage of this approach over likelihood recombination is that all feature components can be jointly represented by statistical models.

The details of the proposed multiresolution feature extraction are illustrated in Figure 6.3 and Figure 6.4. Figure 6.3 shows the different blocks of the proposed method. In addition to the full-band processing of MFCCs, the LPCCs are calculated in separate parallel blocks except for the highest frequency subband. In this way, different subbands with different LPCCs could independently be produced in each block and added to the full-band features. Further processing in each of the block to produce the subband feature vectors is shown in Figure 6.4. According to the idea of our method, to avert too high dimension of the final feature space, the number of orders of LPCCs determined is obviously dependent on the number of subbands. Therefore, consideration is needed for a trade-off between the recognition rate and computation cost. For this reason, we employ two WP strategies i.e. the 4-subband Mel-scale-like WP (Strategy 1) and the 7-subband Mel-scale-like WP (Strategy 2). There are more orders of LPCCs (tenth order) in each subband but fewer subbands (4 subbands) in Strategy 1. On the contrary, in 7-subband Mel-scale-like WP strategy, we calculate only fifth order LPCCs in each of 7 subbands.

Figure 6.4 shows the multiresolution feature extraction method based on the 4-subband WP. The method based on 7-subband WP is almost the same as the 4-subband except it has more subbands. Besides, in order to fortify the feature vectors and reduce the computation load, the log energy value of WP coefficients in the highest frequency channel was calculated in the proposed method. The log energy value applied here is given as follows:

$$E_a(C_a) = \log_{10} \left(\sum_{i=1}^{N_a} c_a^2(i) \right) \quad (6.4.1)$$

$$E_d(C_d) = \log_{10} \left(\sum_{j=1}^{N_d} c_d^2(j) \right) \quad (6.4.2)$$

$$C_a = \{ c_a(1), c_a(2), \dots, c_a(N_a) \} \quad (6.4.3)$$

$$C_d = \{ c_d(1), c_d(2), \dots, c_d(N_d) \} \quad (6.4.4)$$

where C_a and C_d are the WP coefficients in the approximation and detailed channels respectively, N_a and N_d are the number of coefficients, $E_a(C_a)$ and $E_d(C_d)$ are the log energy value within the approximation channel and detailed channel.

In the Intermediate Feature stage of Section II in Figure 6.3, the combination of these LPCCs and the log energy value is implemented according to the following equation:

$$IntermediateFeature = \sum_{i=1}^L LPCCs(i) + w \times E_d(C_d) \quad (6.4.5)$$

where L is the number of blocks and it is set to be 3 or 6 according to different Mel-scale-like WP decomposition strategies, w represents a weighting value and it is set to be 1 in our proposed methods.

In the Final Feature stage, a combination of outputs of Section I and Section II is implemented as follows:

$$FinalFeature = \{F1, F2\} \quad (6.4.6)$$

In the Dimensionality Reduction modules in Figure 6.3, we use the newly proposed method, the Step-Weighted Linear Discriminant Analysis (SW-LDA), to reduce the dimension of feature space without losing more useful discriminant

information. By using SW-LDA, better decorrelation of the subband features and lower dimension feature space is achieved resulting in more salient feature vectors. The SW-LDA has been described in Chapter 4 in details.

Figure 6.3: Block diagram of the multiresolution feature extraction method

Figure 6.4: Detailed building blocks of Section II of Figure 6.3

6.5 Application to Speech Database

6.5.1 Database Description and Experimental Results

To evaluate the performance of our new feature extraction approach, we implemented LPC, LPCCs, MFCCs and the proposed multiresolution feature extraction method on a set of speech recognition data. Like Chapter 4, the proposed methods were applied on TI46 corpus to show the effectiveness coping with inter-speaker variability. TI46 corpus was collected from 16 different adults. Our speech recognition experiments utilized HMM-based speech recognizer for speaker-independent isolated English digits test. A five-state left-to-right HMM is adopted in the experiment. The TI46 corpus is used in the proposed system and the TI46 corpus contains 16 speakers namely 8 males and 8 females labeled. There are 15 utterances of each English digit (0~9) from each speaker: 10 designated as training tokens and 5 designed as testing tokens in the proposed system. The recognition system was trained by 10 clear training tokens per person. To further test the performance of the proposed method, the clear testing utterances were corrupted by white noise according to different Signal-to-Noise Ratio (SNR).

We need to label the training data in order to apply LDA and SW-LDA in the experiments. In the beginning, we estimate HMM parameters in the original data space using Baum-Welch algorithm and get a baseline result. Next, we employ Viterbi alignment over the training data to get class labels, i.e., the HMM state sequence. To successfully apply SW-LDA, we also need to select a proper weighting function. In our experiments, the weighting function is the square of the

inverse of the squared Euclidean distance between class means, i.e., $\omega_{kl} = d_{kl}^{-2}$, where $d_{kl} = \|v_k - v_l\|^2$ and $k, l = 1, \dots, N$, if we have N classes.

Both the 4-subband and the 7-subband multiresolution feature extraction methods are tested in the experiments. To compare the performance of different strategies, the multiresolution feature vectors combined with the full-band MFCCs and the feature vectors combined with the full-band LPCCs are all evaluated in the experiments.

In the 4-subband feature extraction strategy, the full-band stream is built by 16 MFCCs (or LPCCs) and plus 16 deltas. After projecting the feature space by the SW-LDA method, the output of F1 in Section I is a set of 16-dimensional coefficients. In the feature extraction stage of Section II, tenth order LPCCs in each block are calculated. The dimension of the final output of F2 is reduced by SW-LDA from 31 to 16. The dimension of the final combined feature vector is 32.

The 7-subband feature extraction strategy is also implemented in the speech recognition experiments. The full-band stream is also built by 16 MFCCs (or 16th LPCCs) plus 16 deltas. The dimension of feature vectors of F1 in Section I is projected by the SW-LDA method from 32 to 16. In Section II, fifth order LPCCs in each block are calculated. The dimension of the final output of F2 is also reduced by the SW-LDA from 31 to 16. After combining two output feature vectors, the final feature space is 32 in dimension.

The baseline front-end features are MFCCs, LPC and LPCCs. The feature vectors are built by 16 MFCCs, 16 LPC and 16 LPCCs plus 16 deltas respectively and class labels are also obtained by the HMM state sequence. After using LDA on the

refereed feature space obtained by the LPC, the LPCCs and the MFCCs, we obtain 32-dimensional feature space.

The results are tabulated in Table 6.3. Here, the columns represent the LPC, the LPCCs, the MFCCs and the proposed method. For example, the 4-subband-LPCCs represents the proposed 4-subband Mel-scale-like WP-based multiresolution feature extraction method using LPCCs as the full-band feature, and the 7-subband-MFCCs represents the method based on 7-subband Mel-scale-like WP decomposition and using the full-band MFCCs.

Method \ SNR (dB)	Clean	20dB SNR	15dB SNR	10dB SNR	5dB SNR	0dB SNR	Average
LPC (%)	5	29	45	60	76	86	50.2
LPCCs (%)	1	19	31	41	59	77	38
4-subband-LPCCs (%)	0	5	15	32	53	67	28.7
7-subband-LPCCs (%)	1	9	20	36	50	65	30.2
MFCCs (%)	8	15	18	22	30	44	22.8
4-subband-MFCCs (%)	1	4	9	16	26	45	16.8
7-subband-MFCCs (%)	1	6	12	22	38	50	21.5

Table 6.3: Comparison of error rates based on various methods for English digits test

(a) LPC-based Method

Table 6.3 shows the speech recognition results when different feature extraction techniques are employed. We can see that the LPC gets an acceptable result i.e. 5% error rate when it is applied on a clear testing data set. Applying the LPCCs

method on the testing data set, it obtains better performance than that of the LPC at different SNRs. It is seen that the LPCCs outperforms the LPC with approximately an average of 11.8% more reduction in word error rate. However, when the testing patterns are corrupted with a Gaussian white noise, the performance of all methods is extremely affected by the added noise. In particular, the performances of LPC-based methods drop quickly with decreasing SNR. When the Mel-scale-like WP-based method is combined with the full-band LPCCs, and projected by the SW-LDA without increasing the dimension of feature space, about 10% more performance improvement was achieved not only at clear but also at different lower SNRs testing data. This shows that the improvement is successfully achieved by applying multiresolution feature extraction strategy. However, it can also be seen from the table that the performances of LPC-based methods including LPCCs method are not very good at lower SNRs.

(b) MFCCs-based Method

We have known that the multiresolution front-end combined with full-band LPCCs considerably improves the performance of the traditional LPC and the LPCCs, but the LPC-based methods all are not very good for those test data degraded by Gaussian noise. From the table, the average error rate of the best 4-subband-LPCCs is still 5.9% higher than that of the traditional MFCCs. However, from the results, we can see that the best performance acquired from existing MFCCs has only about 8% error rate on clear speech testing data, and it is obviously not better than the LPC-based method. On the other hand, the MFCCs obtains about 20% more performance improvement at below 10 dB noise level even though it was only trained by clear data. The MFCCs shows better recognition performance under noisy data set, but it fails to obtain better results than those of 4-subband-LPCCs at 20, 15 SNR and clear test. Therefore, combining full-band MFCCs with

the proposed multiresolution feature vector is more promising. In fact, the 4-subband-MFCCs achieves an average of 16.8%, i.e. the lowest error rate in all feature extraction methods, and it increases the recognition performance in all ranges of testing data set including clear and lower SNRs level. When we apply 7-subband-MFCCs in experiments, we can also observe that significant performance improvements in clean as well as in noisy conditions in comparison to other baseline results.

(c) 4-subband and 7-subband WP-Based Method

In this section, we compare the results based on the 4-subband and 7-subband methods in experiments. From Table 6.3, both 4-subband and 7-subband methods obtain performance improvement over other baseline results. The 4-subband-based method performs better than the 7-subband-based method according to the recognition results. We can see that the 4-subband-LPCCs surpasses 7-subband-LPCCs, and it achieves approximately 1.5% slight reduction in error rate. For the 4-subband-MFCCs method, we can also observe slight improvement over the 7-subband-MFCCs. In the case of multiresolution front-end, we notice that combining the full-band MFCCs with the multiresolution subband feature is better than combining the full-band LPCCs with them. The 4-subband-MFCCs and the 7-subband-MFCCs have achieved about 10% more performance than those of the 4-subband-LPCCs and the 7-subband-LPCCs respectively.

6.5.2 Discussions

In this chapter, a multiresolution-based feature extraction technique is presented for robust speech recognition. The input speech signals were partitioned into different frequency subbands similar to the Mel scale by using Mel-scale-like

wavelet packets decomposition. Furthermore, the LPCCs are calculated within each subband to capture special information about the envelope of the speech spectrum of vowels. In addition to multiresolution processing, the proposed front-end includes other novel features as well. The use of full-band MFCCs (or LPCCs) has been shown to improve the recognition rate. According to the works of [142],[145], the use of conventional full-band coefficients is still very important because general subband-based feature extraction algorithms are incapable of obtaining high recognition accuracy under noisy conditions. From our investigation, using full-band MFCCs is much more effective than full-band LPCCs. Considering the computation load, the higher feature vector dimension is an apparent drawback of the proposed multiresolution feature extraction method. To effectively reduce the dimension of the feature space, the SW-LDA, which is proposed and discussed in Chapter 4, is utilized to reduce high dimensionality successfully without losing more useful information. The proposed multiresolution front-end, especially the 4-subband-MFCCs-based feature extraction method produces significant performance improvement over a variety of different stationary background noises.

Although we utilized the SW-LDA method to reduce the high dimensional feature space, it still results in losing some useful information. The high computational complexity of the SW-LDA and the Mel-scale-like WP decomposition are also introduced in the multiresolution feature extraction method. Therefore, the higher computation load is still an apparent drawback of the proposed multiresolution front-end. There appears to a trade-off between the number of subbands and the order of the LPCCs accompanying the recognition rate. When the number of the subbands was increased from 4 to 7, the recognition accuracy did not increase as we expected. The reason is probably that the order of LPCCs is reduced to realize the lower dimension of the feature vectors. Relative fewer subbands and higher

orders of the LPCCs in each subband will obtain better recognition rate according to the speech recognition experiments. In this thesis, only speaker-independent isolated English digits recognition experiments are implemented. In future, if condition permit, we may like to employ the proposed method on more speech data sets, which are based on large vocabulary and continuous speech recognition.

6.6 Conclusions

We present a multiresolution-based feature extraction technique for robust speech recognition in this chapter. Mel-scale-like WP decomposition is employed in the method to obtain different resolution levels of the original signal. In the different subbands of the signal, LPCCs are calculated to capture the characteristic of each subband in the proposed algorithm. The conventional full-band MFCCs are also combined with the subband features for maximizing the recognition accuracy. To optimize the proposed method, we employ the projection strategy to reduce the computational load of the complicated multiresolution-based front-end. The new step-weighted linear discriminant dimensionality reduction technique is used instead of Discrete Cosine Transform (DCT) or LDA to produce final feature parameters.

The proposed algorithm was experimentally evaluated in speaker-independent English-digit recognition tasks. The testing data were corrupted by Gaussian noise at different SNRs level. The training and testing speech utterances are from the TI46 database. Experimental results show that using this mechanism not only improves the recognition rate but also effectively reduces the influence of noise interference. Even in very low SNR environments, the error rate of the proposed method is satisfactory. Compared with other existing popular feature extraction

methods, the proposed multiresolution front-end, especially 4-subband-MFCCs method is highly effective and robust.

Chapter 7

Conclusions and Recommendations

In this chapter, the main results are summarized and recommendations for future work are presented.

7.1 Conclusions

The emphasis of this thesis is on current interests of pattern recognition. Methods are investigated and developed specifically for feature extraction and feature dimensionality reduction, and applied to problems in speech recognition.

In the first part, a study on the supervised linear feature dimensionality reduction methodologies is carried out. A concept of step weight is introduced to solve the problem that the optimal criterion of the general linear supervised mapping techniques is not directly related to the accuracy in the output space. Based on this concept, a Step-Weighted Linear Discriminant Analysis (SW-LDA) dimensionality reduction method is proposed. During the projection procedure using the SW-LDA, the feature space is gradually compressed one dimension per

step. At each step of dimension reduction, the between-class and the within-class scatter matrices are recomputed. As a consequence, those centers of classes that come closer together can be increasingly weighted. Experiments are conducted for speaker-independent English digits recognition. The proposed method is demonstrated to outperform other LDA-based dimensionality reduction techniques when there is a need to reduce more feature dimensions.

In the second part, the emphasis is on the supervised non-linear feature dimensionality reduction techniques. We extend the linear weighted discriminant analysis to Non-linear Weighted Discriminant Analysis (NWDA) by using a kernel function operator. The formulation of the NWDA using dot product and matrix formulation is derived in details and a particular eigenvalue resolution method is developed. From dimensionality reduction experiments based on the Fisher's iris data and the Leptograpsus Crabs data, we can conclude that the use of NWDA improves classification and projection performance compared to other linear and non-linear dimensionality reduction methods. This method is not very computationally feasible for speech recognition based on large vocabulary due to the enormous size of the matrices involved. We attempt to apply this method on a simple confusable English letters recognition task by sampling the training data, and we observe that the NWDA performs favorable than the baseline methods.

Finally, a multiresolution analysis method for speech feature extraction is proposed. We utilize Mel-scale-like Wavelet Packets (WP) decomposition strategy to obtain different resolution levels of the original signal. In the different frequency subbands of the signal, Linear Predictive Cepstral Coefficients (LPCCs) are calculated to capture the characteristic of each subband in the feature extraction algorithm. The full-band Mel-scale Frequency Cepstral Coefficients (MFCCs) are also calculated and combined with the subband features. The SW-LDA, which we

have introduced in Chapter 4, is employed to reduce the high dimensionality of the feature space derived from the multiresolution front-end. From the speaker-independent speech recognition experiments, we can see that the new multiresolution feature extraction strategy is very promising and much better than other existing feature extraction methods.

7.2 Recommendations for Future Research

Two possible areas of further works emerge from this thesis. The first area is to improve the NWDA algorithm. The second area is to further investigate and improve the proposed multiresolution speech feature extraction method.

To the proposed NWDA, how to clearly select the weighting function and the kernel function is still an open problem in future. In the training procedure, the size of the matrix S_b^f and S_w^f of the NWDA method will become rather large. To invert such a gigantic matrix and to perform multiplications is impractical. Although we utilized a sampling method to reduce the computation load, the difficulty of computation is still a problem to the NWDA. In future, we may like to make the NWDA more feasible by improving approaches to estimating NWDA. On the other hand, to reduce computation complexity, it is also a very promising strategy that makes use of a more reasonable sampling method to reduce the training samples without losing more useful information. All in all, to apply this method to practical applications in speech recognition, we need to further reduce the computation load.

The proposed multiresolution feature extraction method which satisfies recognition results is compared with other methods. However, it is rather difficult to decide the appropriate number of subbands through applying Mel-scale-like WP

decomposition, and the order of LPCCs computed in each subband. In future, we may like to improve the multiresolution feature extraction method so that the number of subbands and the order of LPCCs are not chosen intuitively and can be obtained by an optimum criterion. To reduce the high dimension of the feature space, the multiresolution feature extraction method relies on the use of the proposed SW-LDA, which permits efficient dimensionality reduction in a high dimensional feature space. The SW-LDA algorithm is a simple and direct extension of the general weighted linear discriminant analysis. It might be possible to improve the performance of multiresolution front-end if the output dimension of the feature space reduced by the SW-LDA is more reasonably determined. The computation complexity of the Mel-scale-like WP decomposition and the SW-LDA is another apparent drawback of the proposed method. Future work will be dedicated to optimizing and improving the multiresolution feature extraction algorithm for reducing the computation load.

Due to the source of speech database and the limitation of computation ability, our speech recognition experiments are only based on an isolated English words speech database. The speech database is not large enough to satisfy more practical applications. We hope we can apply the proposed methods on large vocabulary speech database and continuous speech recognition in future.

Author's Publications

- Jiang Hai, Er Meng Joo, and Gao Yang, “Feature Extraction Using Wavelet Packets Strategy,” *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 5, pp. 4517-4520, Dec., USA, 2003.
- Jiang Hai, and Er Meng Joo, “Improved Linear Predictive Coding Method for Speech Recognition,” *Proceedings of the Fourth International Conference on Information, Communications & Signal Processing and Fourth IEEE Pacific-Rim Conference On Multimedia*, vol. 3, pp. 1614–1618, Dec., Singapore 2003.
- Li Zheng Rong, Er Meng Joo, and Jiang Hai, “Adaptive Nonlinear Filter by Ellipsoidal-Basis-Function Networks,” *Proceedings of the Fourth International Conference on Information, Communications & Signal Processing and Fourth IEEE Pacific-Rim Conference On Multimedia*, vol. 2, pp. 1056–1060, Dec., Singapore 2003.
- Jiang Hai, and Er Meng Joo, “Using Sub-Band Wavelet Packets Strategy for Feature Extraction,” *Proceedings 2nd WSEAS ICECS and 2nd WSEAS E-Activities*, Dec., Singapore 2003.

- Jiang Hai, and Er Meng Joo, "Speech Feature Analysis Using Step-Weighted Linear Discriminant Analysis," *Proceedings of 2nd WSEAS ICECS and 2nd WSEAS E-Activities*, Dec. 7-9, Singapore 2003.
- Jiang Hai, and Er Meng Joo, "Nonlinear Weighted Discriminant Analysis, " *Accepted for presentation at the 12th European Signal Processing Conference (EUSIPCO)*, September 6-10, 2004, Vienna, Austria.
- Jiang Hai, and Er Meng Joo, "Nonlinear Weighted Discriminant Analysis Method Using a Kernel Function Operator," *Submitted to IEEE Transactions on Signal Processing*, Nov. 2004.
- Jiang Hai, and Er Meng Joo, "Multiresolution Analysis in Speech Feature Extraction," *Submitted to IEEE Transactions on Signal Processing*, Nov. 2004.
- Jiang Hai, and Er Meng Joo, "Step-Weighted Linear Discriminant Analysis," *Submitted to IEEE Signal Processing Letters*. Nov. 2004.

Bibliography

- [1] K.H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *Journal of the Acoustics Soc. of America*, vol. 24, pp. 637-642, 1952. Cited in S.R. Hyde, *Automatic Speech Recognition: A Critical Survey and Discussion of the Literature*, McGraw-Hill, 1972.
- [2] V.M. Velichko, and N.G. Zagoruyko, "Automatic recognition of 200 words," *International Journal of Man-Machine Studies*, 2:223, Jun. 1970.
- [3] H. Sakoe, and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 26, issue 1, pp. 43-49, Feb. 1978.
- [4] F. Itakura, "Minimum prediction residual applied to speech recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 23, issue 1, pp. 67-72, Feb. 1975.
- [5] L.R. Rabiner, S.E. Levinson, A.E. Rosenberg, and J.G. Wilpon, "Speaker independent recognition of isolated words using clustering techniques," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 27, issue 4, pp. 336-349, Aug. 1979.

[6] J. Ferguson, *Hidden Markov Models for Speech*, IDA-CRD, Princeton, New Jersey, 1980.

[7] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, issue 2, pp. 257-286, Feb. 1989.

[8] M.N. Stuttle, and M.J.F. Gales, "A Mixture of Gaussians Front End for Speech Recognition," *Proceedings of Eurospeech-2001*, Denmark, Sep. 2001.

[9] I. Potamitis, N. Fakotakis, and G. Kokkinakis, "Improving the robustness of noisy MFCC features using minimal recurrent neural networks," *Proceedings of IJCNN 2000*, vol. 5, pp. 271-276, Jul. 2000.

[10] Wei-Wen Hung, and Hsiao-Chuan Wang, "On the use of weighted filter bank analysis for the derivation of robust MFCCs" *IEEE Signal Processing Letters*, vol. 8, issue 3, pp. 70-73, Mar. 2001.

[11] Shang-Ming Lee, Shi-Hau Fang, Jieh-weih Hung, and Lin-Shan Lee, "Improved MFCC feature extraction by PCA-optimized filter-bank for speech recognition," *IEEE Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy, pp. 49-52, 2001.

[12] Alain Biem, Shigeru Katagiri, Erik McDermott, and Biing-Hwang Juang, "An application of discriminative feature extraction to filter-bank-based speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, No.2, Feb. 2001.

[13] Jia Lei, and Xu Bo, "Including detailed information feature in MFCC for large vocabulary continuous speech recognition," *Proceedings of ICASSP'02*, vol. 1, pp. 13-17, May 2002.

[14] M. Chetouani, B. Gas, J.L. Zarader and C. Chavy, "Discriminative training for neural predictive coding applied to speech features extraction," *Proceedings of IJCNN'02*, vol. 1, pp. 12-17, May 2002.

[15] Satya Dharanipragada, "Feature extraction for robust speech recognition," *Proceedings of ISCAS2002*, vol. 2, pp. II-855 – II-858, 2002.

[16] Jong-Hwan Lee, Ho-Young Jung, Te-Won Lee, and Soo-Young Lee, "Speech feature extraction using independent component analysis," *Proceedings of ICASSP'00*, vol. 3, pp. 1631-1634, 2000.

[17] Oh-Wook Kwon, Te-Won Lee, and Kwokleung Chan, "Application of variational Bayesian PCA for speech feature extraction," *Proceedings of ICASSP'02*, vol. 1, pp. 825-1 – 825-8, 2002.

[18] M.E. Tipping, and C.M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society*, series B, 61, part 3, pp.611-622, 1999.

[19] M.E. Tipping, and C.M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, pp. 443-482, Feb. 1999.

[20] Ralf Schlüter, and Hermann Ney, "Using phase spectrum information for improved speech recognition performance," *Proceedings of ICASSP'01*, vol. 1, pp. 133-136, 2001.

-
- [21] C.M. Bishop, "Variational principal components," *Proceedings of ICANN*, pp. 509-514, 1999.
- [22] F. J. Owen, *Signal Processing of Speech*, McGraw-Hill, Inc., 1st Edition, Feb. 1993.
- [23] M.H. Savoji, "A robust algorithm for accurate endpointing of speech," *Speech Communication*, vol. 8, pp. 45-60, Mar. 1989.
- [24] M. Hamada, Y. Takizawa, and T. Norimatsu, "A noise robust speech recognition system," *Proceedings of ICSLP-90*, pp. 893-896, 1990.
- [25] G.V.Ramana Rao, and J. Srichand, "Word boundary detection using pitch variations," *Proceedings of ICSLP-96*, pp. 813-816, Oct. 1996.
- [26] J.H.L. Hansen, and O. Bria, "Lombard effect compensation for robust automatic speech recognition in noise," *Proceedings of ICSLP-90*, pp. 1125-1128, Nov. 1990.
- [27] J.C. Junqua, B. Mak, and B. Reaves, "A robust algorithm for word boundary detection in the presence of noise," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 406-412, Jul. 1994.
- [28] J.E. Markel, and A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag New York, Inc., 1976.
- [29] P.B. Patil, "Multilayered network for LPC based speech recognition," *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 435-438, May 1998.

-
- [30] L. Rabiner, and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
- [31] G. White, and R. Neely, "Speech recognition experiments with linear predication, bandpass filtering, and dynamic programming," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 24, issue 2, pp. 183-188, Apr. 1976.
- [32] J. Harrington, and S. Cassidy, *Techniques in Speech Acoustics*, Kluwer Academic Publishers, Dordrecht, 1999.
- [33] X. Huang, A. Acero, and H. -W. Hon, *Spoken Language Processing : a Guide to Theory, Algorithm, and System Development*, Prentice-Hall, New Jersey, 2001
- [34] Andrew Teoh, S.A. Samad, and A. Hussain, "Nearest neighbourhood classifiers in a bimodal biometric verification system fusion decision scheme," *Journal of Research and Practice in Information Technology*, vol. 36, no. 1, pp. 47-62, Feb. 2004.
- [35] Tomi Kinnunen, Ville Hautamäki, and Pasi Fränti, "Fusion of spectral feature sets for accurate speaker identification," *Proceedings of the 9th International Conf. Speech and Computer*, pp. 361-365, St. Petersburg, Russia, Sep. 2004
- [36] Benjamin J. Shannon, and Kuldip K. Paliwal, "Influence of autocorrelation lag ranges on robust speech recognition," *Proceedings of the IEEE International Conf. on Acoustics, Speech and Signal Processing*, vol. I, pp. 545-548, Philadelphia, Mar. 2005
- [37] Jhing-Fa Wang, Shun-Chieh Lin, Hsueh-Wei Yang, and Fan-Min Li, "Multiple-translation spotting for Mandarin-Taiwanese speech-to-speech

translation,” *Computational Linguistics and Chinese Language Processing*, vol. 9, no. 2, pp. 13-28, Aug. 2004.

[38] B. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” *Journal of the Acoustic Society of America*, vol. 56, no. 6, pp. 1304-1312, 1974.

[39] S.B. Davis, and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357-366, Aug. 1980.

[40] “ETSI ES 201 108 v1.1.2 Distributed speech recognition; front-end feature extraction algorithm; compression Algorithm,” Apr. 2000.

[41] Y. Linde, A. Buzo, and R.M. Gray, “An algorithm for vector quantizer design,” *IEEE Transactions on Communication*, vol. 28, pp. 84-95, Jan. 1980.

[42] A. Gersho, and R.M Gray, *Vector Quantization and Signal Compression*, Boston, Kluwer Academic Publishers, Jan. 1992.

[43] H. Abut, editor, *Vector Quantization*, IEEE Reprint Collection, IEEE Press, Piscataway, NJ, USA, May 1990.

[44] W.B. Kleijn, and K.K. Paliwal, *Speech Coding Synthesis*, Netherlands, Elsevier Science Publisher, 1995.

[45] N.M. Nasrabadi, and R.A. King, “Image coding using vector quantization: a review,” *IEEE Transactions on Communication*, vol. 36, pp. 957-971, Aug. 1988.

-
- [46] K.K Paliwal, and B.S. Atal, "Efficient vector quantization of LPC parameters are 24 bits/frame," *IEEE Transactions on Speech and Audio Processing*, vol. 1, issue 1, pp. 3-14, Jan. 1993.
- [47] L.E. Baum, and J.A. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360-363, 1967.
- [48] L.E. Baum, and G.R.Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, vol. 27, no. 2, pp. 211-227, 1968.
- [49] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informat. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.
- [50] G.D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp 268-278, Mar. 1973.
- [51] L.K.Baker, "The dragon system-An overview," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-23, no. 1, pp. 24-29, Feb. 1975.
- [52] L. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Transactions on Information Theory*, vol. 28, issue 5, pp. 729-734, Sep. 1982.
- [53] B.H. Juang, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *AT&T Tech. Journal*, vol. 64, no. 6, pp. 1235-1249, Jul. 1985.

-
- [54] B.H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains," *IEEE Transactions on Information Theory*, vol. 32, issue 2, pp. 307-309, Mar. 1986.
- [55] W. Huang, R. Lippmann, and B. Gold, "A neural net approach to speech recognition," *Proceedings of IEEE International Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 99-102, 1988.
- [56] M. H. Kostepen, and G. Kumar, "Speech recognition using back-propagation neural networks," *Proceedings of IEEE Conf. Energy, Computer, Communication and Control Systems*, vol. 2, pp. 144-148, Aug. 1991.
- [57] W.Y. Chen, S.H. Chen, and C.J. Lin, "A speech recognition method based on the sequential multi-layer perceptrons," *Neural Networks*, vol. 9, No. 4, pp. 655-669, 1996.
- [58] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328-339, 1989.
- [59] R.P. Lippmann, and B. Gold, "Neural classifiers useful for speech recognition," *Proceedings of the First International Conf. on Neural Networks*, vol. 4, pp. 417-422, Jun. 1987.
- [60] H. Bourlard, and C.J. Wellekens, "Links between Markov models and multilayer perceptrons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, issue 12, pp. 1167-1178, Dec. 1990.

-
- [61] S. Austin, G. Zavaliagkos, J. Makhoul, and R. Schwartz, "Speech recognition using segmental neural nets," *Proceedings of IEEE Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 625-628, Mar. 1992.
- [62] H. Sakoe, R. Isotani, K. Yoshida, K. Iso, and T. Watanabe, "Speaker-independent word recognition using dynamic programming neural networks," *Proceedings of IEEE Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 29-32, May 1989.
- [63] A.K. Jain, R.P.W. Duin, and Jianchang Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, issue 1, pp. 4-37, Jan. 2000.
- [64] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, New York, Springer Verlag, 1996.
- [65] S. Watanabe, *Pattern Recognition: Human and Mechanical*, New York, John Wiley and Sons, 1985.
- [66] T.W. Anderson, *An Introduction to Multivariate Statistical Analysis*, New York, John Wiley and Sons, third edition, Jul. 2003.
- [67] R.O. Duda, and P.E. Hart, *Pattern Classification and Scene Analysis*, New York, John Wiley and Sons, 1973.
- [68] R.O. Duda, D.G. Stork, and P.E. Hart, *Pattern Classification and Scene Analysis*, New York, John Wiley and Sons, Second Edition, 2000.
- [69] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, New Jersey, 1961.

-
- [70] P.J. Devijver, and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, New Jersey, 1982.
- [71] K. Fukunaga, editor, *Introduction to Statistical Pattern Recognition*, Academic Press, London, Second Edition, 1990.
- [72] R.N. Shepard, "The analysis of proximities: multidimensional scaling with an unknown distance function," *Psychometrika*, vol. 27, pp. 125-140, 1962.
- [73] J.B. Kruskal, "Nonmetric multidimensional scaling: a numerical method," *Psychometrika*, vol. 29, pp. 115-129, 1964.
- [74] J.W. Sammon, "A non-linear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, pp. 401-409, 1969.
- [75] T. Kohonen, *Self-organizing maps*, Springer-Verlag, 1995.
- [76] P. Baldi, and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53-58, 1989.
- [77] J. Mao, and A.K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, issue 2, pp. 296-317, Mar. 1995.
- [78] M.A. Kraaijveld, J. Mao, and A.K. Jain, "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Transactions on Neural Networks*, vol. 6, issue 3, pp. 548-559, May 1995.

-
- [79] C. Lee, and D.A. Landgrebe. "Decision boundary feature extraction for neural networks," *IEEE Transactions on Neural Networks*, vol. 8, issue 1, pp. 75-83, Jan. 1997.
- [80] R.C.T. Lee, J.R. Slagle, and H. Blum, "A triangulation method for the sequential mapping of points from N-Space to two-Space," *IEEE Transactions on Computers*, vol. 26, pp. 288-292, 1977.
- [81] Wojciech Siedlecki, Kinga Siedlecka, and Jack Sklansky, "An overview of mapping techniques for exploratory pattern analysis," *Pattern Recognition*, vol. 21, issue 5, pp. 411-429, 1988.
- [82] Wojciech Siedlecki, Kinga Siedlecka, and Jack Sklansky, "Experiments on mapping techniques for exploratory pattern analysis," *Pattern Recognition*, vol. 21, issue 5, pp.431-438, 1988.
- [83] P.M. Narendra, and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. 26, issue 9, pp. 917-922, Sep. 1977.
- [84] T.M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, pp. 326-334, 1965.
- [85] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 11, pp. 1119-1125, 1994.
- [86] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, Second Edition, 1990.

-
- [87] Warren L.G. Koontz, and Keinosuke Fukunaga, "A non-linear feature extraction algorithm using distance transformation," *IEEE Transactions on Computers*, vol. 21, pp. 56-63, 1972.
- [88] J.W. Sammon, "A non-linear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, pp. 401-409, 1969.
- [89] J.H. Friedman, "Exploratory projection pursuit," *Journal of the American Statistical Association*, vol. 82, no. 39, pp. 249-266, 1987.
- [90] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, issue 3, pp. 287-314, 1994.
- [91] A.J. Bell, and T.J. Sejnowski. "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, issue 6, pp. 1129-1159, 1995.
- [92] J.-F. Cardoso. "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 86, issue. 10, pp. 2009-2025, Oct. 1998.
- [93] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley and Sons, 2001.
- [94] T.W. Lee, *Independent Component Analysis: Theory and Applications*, Kluwer Academic Publishers, Dec. 1998.
- [95] Erkki Oja, "Data compression, feature extraction, and autoassociation in feed forward neural networks," *Proceedings of the International Conf. on Artificial Neural Networks ICANN'91*, vol. 1, pp. 737-742, 1991.

-
- [96] Teuvo Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, New York, Third Edition, 1989.
- [97] Nikhil R. Pal, *Pattern Recognition in Soft Computing Paradigm*, World Scientific Publishing Co. Pte. Ltd., Apr. 2001.
- [98] Alfred Ultsch, and H.P. Siemon, "Exploratory data analysis: using Kohonen networks on transputers," *Research Report No.329*, University of Dortmund, 1989.
- [99] Bernd Fritzke, "Growing cell Structures – a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, issue 9, pp. 1441-1460, 1994.
- [100] Andreas König, Olaf Bulmahn, and Manfred Glesner, "Systematic methods for multivariate data visualization and numerical assessment of class separability and overlap in automated visual industrial quality control," *Proceedings of the 5th British Machine Vision Conf. BMVC'94*, pp. 195-204, 1994.
- [101] R.A. Fisher, "The statistical utilization of multiple measurements," *Annals of Eugenics*, vol. 8, pp. 376-386, 1938.
- [102] C.R. Rao, "The utilization of multiple measurements in problems of biological classification," *Journal of the Royal Statistical Society*, series B, vol. 10, pp. 159-203, 1948.
- [103] J.H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, 405, pp. 165-175, 1989.
- [104] T. Hastie, and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society*, Series B, pp.155-176, 1996.

-
- [105] T. Hastie, R. Tibshirani, and A. Buja, "Flexible discriminant analysis by optimal scoring," *Journal of American Statistical Association*, pp. 1255-1270, 1994.
- [106] G. Saon, M. Padmanabhan, R.A. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," *Proceedings of IEEE Conf. Acoustics Speech Signal Processing*, pp. 1747-1750, Jun. 2000.
- [107] Yongxin Li, Yuqing Gao, and H. Erdogan, "Weighted pairwise scatter to improve linear discriminant analysis," *Proceedings of ICSLP*, vol. 4, pp. 608-611, 2000.
- [108] M. Loog, R.P.W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise Fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, No. 7, pp. 762-766, Jul. 2001.
- [109] Rohit Lotlikar, and Ravi Kothari, "Fractional-step dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, issue 6, pp. 623-627, Jun. 2000.
- [110] Duchene, and S. Leclercq, "An optimal transformation for discriminant principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, issue 6, pp. 978-983, Nov. 1988.
- [111] Om Deshmukh, Carol Y. Espy-Wilson, and Amit Juneja, "Acoustic-phonetic speech parameters for speaker-independent speech recognition," *Proceedings of ICASSP'02*, pp. 593-596, Orlando, May 2002.
- [112] TI46-Word, LDC1993S9, ISBN: 1-58563-017-9, collected by Texas Instruments, Inc.

-
- [113] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, N.Y., 1995.
 - [114] B. Schölkopf, A. Smola, and K.R. Müller, "Non-linear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.
 - [115] Suykens J.A.K., Van Gestel T., Vandewalle J., and De Moor B, "A support vector machine formulation to PCA analysis and its kernel version," *IEEE Transactions on Neural Networks*, vol. 14, issue 2, pp. 447-450, Mar. 2003.
 - [116] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller, "Fisher discriminant analysis with Kernels," *Neural Networks for Signal Processing IX*, pp. 41-48, 1999.
 - [117] G. Baudat, and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, issue 10, pp. 2385-2404, Oct. 2000.
 - [118] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. Part II, pp. 179-188, 1936.
 - [119] P. Loizou and A. Spanias, "Improved speech recognition using a subspace projection approach," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 343-345, 1999.
 - [120] Deshmukh, O., Espy-Wilson, C., and Juneja, A., "Acoustic-phonetic speech parameters for speaker-independent speech recognition," *Proceedings of IEEE ICASSP 2002*, pp. 593-596, Orlando, May 13-17, 2002.

-
- [121] B.D. Ripley, "Ftp archive," Department of Statistics, Univ. of Oxford, <http://www.stats.ox.ac.uk/pub/PRNN/>.
- [122] C.J.C. Burges, and B. Schölkopf, "Improving the accuracy and speed of support vector machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 375-381, MIT Press, Cambridge, MA, 1997.
- [123] S.B. Davis, and P. Mermelstein, "Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Audio Processing*, vol. 28, no. 4, pp. 357-366, Aug. 1980.
- [124] ETSI Draft Standard, *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithm*, ETSI ES 201 108 v0.0.8, 1999.
- [125] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of Acoustics Society of America*, vol. 87, no. 4, pp. 1738-1752, Apr. 1990.
- [126] H. Hermansky, and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, issue 4, pp. 578-589, Oct. 1994.
- [127] J. Tian, R. Hariharan, and K. Laurila, "Noise robust two-stream auditory feature extraction method for speech recognition," *Proceedings of International Conf. Spoken Language Processing*, vol. 3, pp. 991-994, 1998.
- [128] P. Lockwood, and J. Boudy, "Experiments with a non-linear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars," *Speech Communication*, vol. 11, no. 2-3, pp. 215-228, 1992.

-
- [129] A.E. Rosenberg, C.-H. Lee, and F.K. Soong, "Cepstral channel normalization techniques for HMM-based speaker verification, " *Proceedings of International Conf. on Spoken Language Processing*, pp. 1835-1838, 1994.
 - [130] O. Viikki, D. Bye, and K. Laurila, "A recursive feature vector normalization approach for robust speech recognition in noise," *Proceedings of International Conf. Acoustics, Speech, Signal Processing*, pp. 733-736, 1998.
 - [131] M.Gupta and A.Gilbert, "Robust speech recognition using wavelet coefficient features, " *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 445-448, Dec. 2001.
 - [132] K.Kim, D.H.Youn, and C.Lee, "Evaluation of wavelet filters for speech recognition," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 2891-2894, 2000.
 - [133] B.T.Tan, R.Lang, H.Schroder, Minyue, A.Spray, and P.Dermody, "Applying wavelet analysis to speech segmentation and classification," *Wavelet Application, Proc. SPIE 2242*, pp. 750-761, 1994
 - [134] O. Farooq, and S.Datta, "Mel Filter-like admiddible wavelet packet structure for speech recognition" *IEEE Signal Processing Letters*, vol. 8, no. 7, July 2001.
 - [135] O.Farooq, and S.Datta, "Wavelet based robust sub-band features for phoneme recognition," *IEE Proceedings of Vision, Image and Signal Processing*, vol. 155, issue 3, pp. 187-193, June 2004.
 - [136] S. Mallat, *A wavelet tour of signal processing*, Academic Press, Second Edition, 1999.

[137] Ruhi Sarikaya, and John H.L.Hansen, "High resolution speech feature parametrization for monophone-based stressed speech recognition," *IEEE Signal Processing Letters*, vol. 7, issue 7, pp. 182-186, Jul. 2000.

[138] E. Zwicker, and E. Terhardt, "Analytical expressions for critical band rate and critical bandwidth as a function of frequency," *The Journal of Acoustics Society America*, vol. 68, pp. 1523-1525, Dec. 1980.

[139] J.B. Allen, "How do humans process and recognize speech?" *IEEE Transactions on Speech and Audio Processing*, vol. 2, issue 4, pp. 567-577, Oct. 1994.

[140] H. Bourlard, and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," *Proceedings of International Conf. On Spoken Language Processing*, pp. 426-429. Oct. 1996.

[141] H. Hermansky, "Should recognizers have ears?" *Speech Communication*, vol. 25, no. 1-3, pp. 3-27, 1998.

[142] S. Tibrewala, and H. Hermansky, "Sub-band based recognition of noisy speech," *Proceedings of International Conf. Acoustics, Speech, and Signal Processing*, pp. 1255-1258, 1997.

[143] Nikki Mirghafori, and Nelson Morgan, "Combining connectionist multi-band and full-band probability streams for speech recognition of natural numbers," *Proceedings of International Conf. On Spoken Language Processing*, Sydney, pp. 743-746, Dec. 1998.

[144] H. Hermansky, S. Tibrewala, and M. Pavel, "Towards ASR on partially corrupted speech," *Proceedings of International Conf. on Spoken Language Processing*, pp. 462-465, Oct. 1996.

[145] S. Okawa, E. Bocchieri, and A. Potamianos, "Multi-band speech recognition in noisy environments," *Proceedings of International Conf. on Acoustics, Speech and Signal Processing*, pp. 641-644, May 1998.