

# STARS

University of Central Florida  
**STARS**

---

Electronic Theses and Dissertations, 2004-2019

---

2005

## Speaker Identification Based On Discriminative Vector Quantization And Data Fusion

Guangyu Zhou  
*University of Central Florida*



Part of the [Electrical and Electronics Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Zhou, Guangyu, "Speaker Identification Based On Discriminative Vector Quantization And Data Fusion" (2005). *Electronic Theses and Dissertations, 2004-2019*. 521.  
<https://stars.library.ucf.edu/etd/521>



**SPEAKER IDENTIFICATION BASED ON DISCRIMINATIVE VECTOR  
QUANTIZATION AND DATA FUSION**

by

GUANGYU ZHOU

B.S. EE, Zhejiang University, P.R. China, 1993

M.S. EE, Zhejiang University, P.R. China, 1996

A dissertation submitted in partial fulfillment of the requirements for the degree  
of Doctor of Philosophy  
in the Department of Electrical and Computer Engineering  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida.

Summer  
2005

Major Professor: Dr. Wasfy B. Mikhael

© 2005 Guangyu Zhou

## **ABSTRACT**

Speaker Identification (SI) approaches based on discriminative Vector Quantization (VQ) and data fusion techniques are presented in this dissertation.

The SI approaches based on Discriminative VQ (DVQ) proposed in this dissertation are the DVQ for SI (DVQSI), the DVQSI with Unique speech feature vector space segmentation for each speaker pair (DVQSI-U), and the Adaptive DVQSI (ADVQSI) methods. The difference of the probability distributions of the speech feature vector sets from various speakers (or speaker groups) is called the interspeaker variation between speakers (or speaker groups). The interspeaker variation is the measure of template differences between speakers (or speaker groups). All DVQ based techniques presented in this contribution take advantage of the interspeaker variation, which are not exploited in the previous proposed techniques by others that employ traditional VQ for SI (VQSI).

All DVQ based techniques have two modes, the training mode and the testing mode. In the training mode, the speech feature vector space is first divided into a number of subspaces based on the interspeaker variations. Then, a discriminative weight is calculated for each subspace of each speaker or speaker pair in the SI group based on the interspeaker variation. The subspaces with higher interspeaker variations play more important roles in SI than the ones with lower interspeaker variations by assigning larger discriminative weights. In the testing mode, discriminative weighted average VQ distortions instead of equally weighted average VQ distortions are used to make the SI decision. The DVQ based techniques lead to higher SI accuracies than VQSI.

DVQSI and DVQSI-U techniques consider the interspeaker variation for each speaker pair in the SI group. In DVQSI, speech feature vector space segmentations for all the speaker pairs are exactly the same. However, each speaker pair of DVQSI-U is treated individually in the speech feature vector space segmentation. In both DVQSI and DVQSI-U, the discriminative weights for each speaker pair are calculated by trial and error. The SI accuracies of DVQSI-U are higher than those of DVQSI at the price of much higher computational burden.

ADVQSI explores the interspeaker variation between each speaker and all speakers in the SI group. In contrast with DVQSI and DVQSI-U, in ADVQSI, the feature vector space segmentation is for each speaker instead of each speaker pair based on the interspeaker variation between each speaker and all the speakers in the SI group. Also, adaptive techniques are used in the discriminative weights computation for each speaker in ADVQSI. The SI accuracies employing ADVQSI and DVQSI-U are comparable. However, the computational complexity of ADVQSI is much less than that of DVQSI-U.

Also, a novel algorithm to convert the raw distortion outputs of template-based SI classifiers into compatible probability measures is proposed in this dissertation. After this conversion, data fusion techniques at the measurement level can be applied to SI. In the proposed technique, stochastic models of the distortion outputs are estimated. Then, the posteriori probabilities of the unknown utterance belonging to each speaker are calculated. Compatible probability measures are assigned based on the posteriori probabilities. The proposed technique leads to better SI performance at the measurement level than existing approaches.

## **ACKNOWLEDGMENTS**

First, I would like to express my sincere respect and appreciation to my advisor, Dr. Wasfy B. Mikhael. His excellent guidance, great wisdom, patience and understanding helped me overcome all problems and difficulties that I encountered throughout my Ph.D. study, both academically and personally.

I am grateful to acknowledge the support and technical guidance of Dr. Brent Myers and Conxant, Inc.

I appreciate the help and cooperation from the members of my dissertation committee.

Last but not least, I wish to express my deepest thanks to my wife and my parents for their continuous love, patience and support.

## TABLE OF CONTENTS

TABLE OF CONTENTS.....	vi
LIST OF FIGURES .....	ix
LIST OF TABLES.....	xi
CHAPTER ONE: INTRODUCTION.....	1
1.1. Classification of Speaker Recognition.....	1
1.2. Applications of Speaker Recognition .....	3
1.3. Main Results and Organization of the Dissertation .....	4
CHAPTER TWO: INTRODUCCION OF SPEAKER RECOGNITION .....	7
2.1. Introduction.....	7
2.2. Speech Feature Extraction .....	8
2.2.1. Digital Filter Bank .....	8
2.2.2. Mel_Frequency Cepstral Coefficients (MFCC).....	11
2.2.3. Linear Prediction Analysis.....	16
2.2.5. Dynamic Feature.....	21
2.3. Patten Matching Methods for Speaker Recognition .....	22
2.3.1. Vector Quantization.....	23
2.3.2. Dynamic Time Warping (DTW).....	31
2.3.3. Hidden Markov Model.....	34
2.4. Data Fusion Techniques.....	39
CHAPTER THREE: DISCRIMINATIVE VECTOR QUANTIZATION APPROACH FOR SPEAKER IDENTIFIICATION .....	44

3.1. Introduction.....	44
3.2. Speaker Identification Based on Discriminative Vector Quantization .....	47
3.2.1. The Training Mode .....	47
3.2.2. The Testing Mode .....	52
3.3. Experimental Results .....	55
3.4. Conclusions.....	62
CHAPTER FOUR: AN ENHANCED PERFORMANCE DISCRIMINATIVE VECTOR QUANTIZATION TECHNIQUE FOR SPEAKER IDENTIFICATION .....	64
4.1. Introduction.....	64
4.2. The DVQSI Approach with Unique Feature Vector Space Segmentation for Each Speaker Pair (DVQSI-U) .....	65
4.2.1. Speech Feature Vector Space Segmentation Based on Linear Discriminant Functions.....	66
4.2.2 A Novel Algorithm for Discriminative Weighted Average Distortions Calculation .....	71
4.3. Experimental Results .....	74
4.4. Conclusions.....	80
CHAPTER FIVE: SPEAKER IDENTIFICATION BASED ON ADAPTIVE DISCRIMINATIVE VECTOR QUANTIZATION .....	81
5.1. Introduction.....	81
5.2. Adaptive Discriminative Vector Quantization for Speaker Identification (ADVQSI). 82	
5.2. 1. The Training Mode .....	83
5.2.2. The Testing Mode .....	90



5.3. Experimental Results .....	91
5.4. Conclusions.....	96
CHAPTER SIX: COMPATIBLE PROBABILITY MEASURES for THE OUTPUTS OF THE TEMPLATE-BASED SPEAKER IDENTIFICATION CLASSIFIER FOR DATA FUSION....	97
6.1. Introduction.....	97
6.2. Compatible Probability Measures for the Outputs of the Template-based SI Classifier for Data Fusion .....	99
6.3. Experimental Results .....	102
6.4. Conclusions.....	105
CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK.....	106
7.1. Main Contributions .....	106
7.2. Areas of Future Research.....	108
LIST OF REFERENCES .....	109

## LIST OF FIGURES

Figure 1: Speech processing and speaker recognition .....	2
Figure 2. The diagram of the speaker recognition system .....	7
Figure 3. Block diagram of filter bank analysis.....	9
Figure 4. The variation of bandwidths versus the frequency based on the critical band scale .....	11
Figure 5. The block diagram of MFCCs.....	12
Figure 6. Filter allocation in the frequency domain before normalization .....	15
Figure 7. The block diagram of LPC processor for speaker recognition.....	17
Figure 8. The vector quantization system .....	24
Figure 9. An example of time normalization of two sequential patterns to a common time index. 32	
Figure 10. A Markov chain with three states.....	36
Figure 11. The hidden Markov chain.....	37
Figure 12. The general flow chart of the training mode of DVQSI.....	47
Figure 13. The flow chart of the testing mode of DVQSI .....	54
Figure 14. SI accuracy rates versus the number of subspaces $m$ and the parameter $h$ of (3.5a) in text-independent experiments .....	57
Figure 15. SI accuracy rates versus the number of subspaces $m$ and the parameter $c$ of (3.5b) in text-independent experiments .....	58
Figure 16. SI accuracy rates versus the number of subspaces $m$ and the parameter $h$ of (3.5a) in text-dependent experiments .....	60
Figure 17. SI accuracy rates versus the number of subspaces $m$ and the parameter $c$ of (3.5b) in text-dependent experiments .....	61

Figure 18. The speech feature vector space segmentation procedure for the number of subspaces $m=4$ .....	68
Figure 19. Average distortion $d1_j^1(1, 2)$ , $d2_j^1(1, 2)$ , $d_j^t(R, 1)$ and $d_j^t(R, 2)$ versus subspace index $j$ for DVQSI-U with $R \in 1$ .....	77
Figure 20. Average distortion $d1_j^{15}(15, 27)$ , $d2_j^{15}(15, 27)$ , $d_j^t(R, 15)$ and $d_j^t(R, 27)$ versus subspace index $j$ for DVQSI-U with $R \in 15$ .....	77
Figure 21. The numbers of training feature vectors of speaker 11 and 19 in the subspaces of their speaker pair .....	78
Figure 22. The discriminative weight $w_j(3,19)$ for speaker pair 11 and 19, where $h$ in (3.5a) equals 1 .....	79
Figure 23. The diagram of the training mode of ADVQSI .....	89
Figure 24. The average discriminative weights for different subspaces versus the number of adaptive iterations .....	94
Figure 25. The value of the cost function $J$ in (5.2) versus adaptive the number of adaptive iterations .....	95
Figure 26. The average value of $h_{dis}(k1,k2)$ for all speaker pairs versus the number of adaptive iterations .....	95
Figure 27. The flow chart of the estimation of $m(j, k)$ .....	101
Figure 28. The SI accuracies versus $\alpha$ of the second classifier by employing Eq. (6.1) in the data fusion, where $\alpha$ of the first classifier is set to 1 .....	105

## LIST OF TABLES

Table 1. SI accuracy rates of VQSI and DVQSI versus the number of subspaces $m$ and the parameter $h$ of (3.5a) in text-independent experiments .....	57
Table 2. SI accuracy rates of VQSI and DVQSI versus the number of subspaces $m$ and the parameter $c$ of (3.5b) in text-independent experiments .....	59
Table 3. SI accuracy rates of VQSI and DVQSI versus the number of subspaces $m$ and the parameter $h$ of (3.5a) in text-dependent experiments .....	60
Table 4. SI accuracy rates of VQSI and DVQSI versus the number of subspaces $m$ and the parameter $c$ of (3.5b) in text-dependent experiments .....	61
Table 5. The SI accuracy rates employing VQSI, DVQSI and DVQSI-U .....	75
Table 6. The SI accuracy rates employing DVQSI and DVQSI-U, with $h=1$ in (3.5a) .....	75
Table 7. The SI accuracy rates of DVQSI-U versus $T_2$ in (4.4), with $h=1$ in (3.5a) .....	75
Table 8. The SI accuracy rates employing VQSI, DVQSI, and ADVQSI .....	92
Table 9. The average $d(\mathbf{v})$ for the first speaker in the speech feature vector space segmentation	93
Table 10. The SI accuracies rate by employing individual classifiers and data fusion techniques .....	103

# CHAPTER ONE: INTRODUCTION

## 1.1. Classification of Speaker Recognition

Language is the engine of civilization, and speech is its most powerful and natural form. Research in the area of speech processing has attained remarkable progress in past decades. One of the major challenges in the field of speech research is speaker recognition. Speaker recognition is a process of automatically recognizing who is speaking on the basis of the individual information included in the speech waveforms.

Speaker recognition is one of the speech processing fields. Figure 1 shows a few areas of speech processing and how speaker recognition relates to the rest of the fields [4].

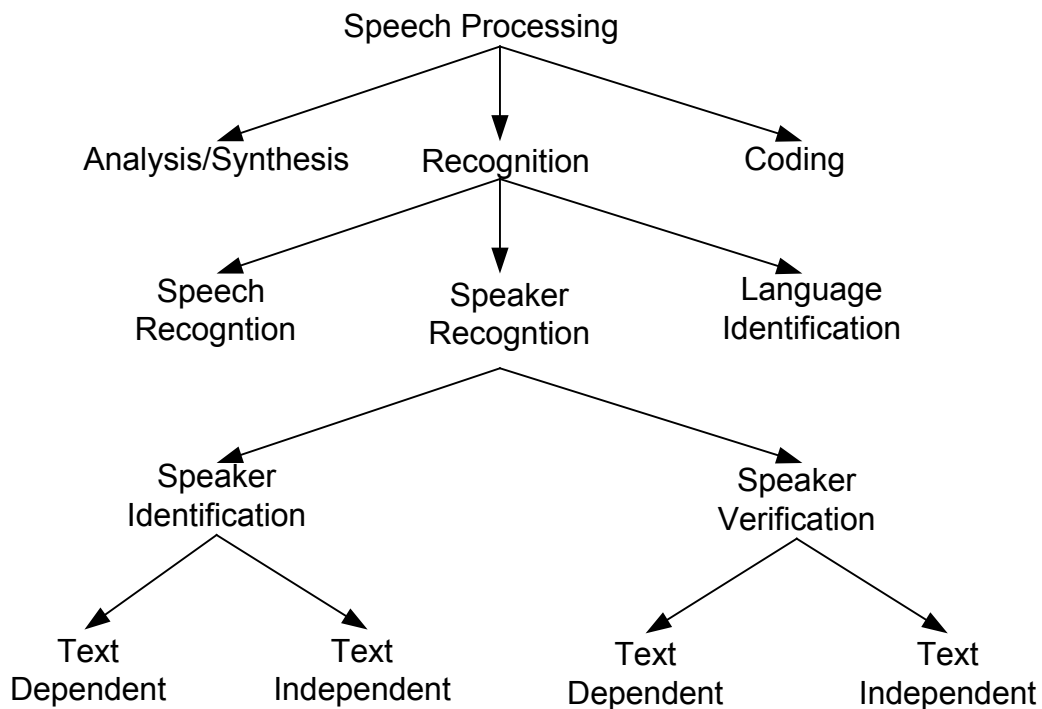


Figure 1: Speech processing and speaker recognition

Speaker recognition can be divided into two categories, namely, speaker identification and speaker verification. In the speaker verification process, by setting a threshold, a decision is made about whether the speaker is who he/she claimed to be. Most applications in which a voice is used as the key to confirm the identity of a speaker are classified as speaker verification. In the literature, speaker verification is also called voice verification, speaker authentication, voice authentication, talker authentication or talker verification [4].

Unlike speaker verification, where a claim of an identity is accepted or rejected based on the speaker's voice, the research work on speaker identification lets the computer identify who is talking, from a large number of enrolled speakers, based on a small sample of his or her voice. Most speaker identification problems are closed-set problems, where the unknown speech waveform is from one of the enrolled speakers. There is also the case called open-set identification, where the unknown speech waveform may be from a speaker without enrollment. In this situation, an additional decision alternative, the unknown speech does not match any enrolled speakers' models, is required.

Speaker recognition can also be categorized into text-dependent recognition and text-independent recognition. The former requires the speaker to say key words or sentences having the same text for both training and recognition trials, whereas the latter do not rely on a specific text being spoken.

## **1.2. Applications of Speaker Recognition**

The applications of the speaker recognition technology are quite varied and continually growing. Speaker recognition has lots of commercial applications, as well as military applications. In 2000, the President of the United States established an organization with the Department of Defense to develop and promulgate biometrics technologies to achieve information system security. Speaker recognition is one of the most important topics in the voice biometrics technologies' research.

Below is an outline of some board areas where speaker recognition technology has been or is currently used [52].

- Access Control: Speaker Recognition is one of the most natural and economic methods to help solving unauthorized uses of computer, communications systems and multilevel access control. Besides the password and/or token, speech biometric factors can be added for extra security.
- Transaction Authentication: For telephone banking, remote electronic and mobile purchase, in addition to access control, speaker recognition can be used for transaction authentication
- Law Enforcement: Some applications are home-parole monitoring (call parolees at random times to verification they are at home) and prison call monitoring (validate inmate prior to outbound call). There has also been discussion of using automatic systems to corroborate aural/spectral inspections of voice sample for forensic analysis [52].

- **Speech Data Management:** In the phone mail system, it is very helpful to automatically label the incoming voice mails with the speakers' names. Also, in a telephone meeting, it is preferable to identify the speakers who are speaking automatically.

### **1.3. Main Results and Organization of the Dissertation**

In this work, Speaker Identification (SI) approaches based on discriminative Vector Quantization (VQ) techniques and data fusion techniques are investigated.

In SI, all the speakers share the same speech feature vector space, since they use the same type of speech feature. The probability distributions of the speech feature vectors of different speakers (or speaker groups) in the speech feature vector space are different. In this work, this difference of the probability distributions is called the interspeaker variation between speakers (or speaker groups). When the interspeaker variation in a subspace of the speech feature vector space is large, the speech templates between speakers (or speaker groups) have a large difference in this subspace, and vice versa.

The Discriminative VQ (DVQ) based techniques presented in this work, the DVQ approach for SI (DVQSI), the DVQSI approach with Unique speech feature vector space segmentation for each speaker pair (DVQSI-U) and the Adaptive DVQSI (ADVQSI) approaches consider the interspeaker variation inside the speech feature vector space. All DVQ based approaches have two modes, namely, the training mode and the testing mode. In the training mode, the speech feature vector space is firstly segmented into a number of subspaces. Then, for each subspace of each speaker pair or each speaker, a discriminative weight is assigned based on interspeaker variations. In the testing mode, weighted average VQ distortions instead of average



VQ distortions are used for the SI decision. DVQ based techniques lead to higher SI accuracies than the traditional VQ technique for SI (VQSI) [4, 49].

In DVQSI and DVQSI-U, discriminative weights are assigned to each speaker pair in the SI group. The discriminative weights are obtained by trial and error and based on the interspeaker variation of each speaker pair. The speech feature vector space segmentation for DVQSI-U considers each speaker pair individually, while the space segmentation for DVQSI is the same for all speaker pairs. The SI accuracy of DVQSI-U is higher than that of DVQSI at the price of the increased computational complexity.

ADVQSI assigns discriminative weight to each speaker instead of each speaker pair based on the interspeaker variation between each speaker and all the speakers in the SI group. Adaptive techniques are used to compute the optimal discriminative weights. The SI accuracy by employing ADVQSI is comparable with that of DVQSI-U. However, the computational burden of ADVQSI is increased approximately proportional to the number of speakers in the SI group, whereas, in DVQSI-U, the computational burden increases with the square of the number of speakers in the SI group.

A novel approach, which transfers the raw distortion outputs of template-based SI classifiers into compatible probability measures, is also presented in this work. In the proposed approach, the statistic models of the raw distortion outputs of template-based SI classifiers are estimated. Then, a posteriori probability of the unknown utterance belonging to each speaker is calculated for each given distortion output. Compatible probability measures of the distortion outputs are obtained based on the posteriori probabilities. After raw outputs of SI classifiers are converted into compatible probability measures, data fusion techniques at the measurement level

can be applied to SI. Experimental results confirm the effectiveness of the proposed approach for SI data fusion at the measurement level.

This dissertation is organized as follows. Chapter two gives a brief overview of speaker recognition techniques. The DVQSI approach is proposed in Chapter three. In Chapter four, techniques associated with DVQSI-U are proposed and analyzed. The ADVQS method is formulated in Chapter five. The technique to transfer the raw distortion outputs of template-based SI classifiers into compatible probability measures is given in Chapter six. Finally, the conclusions are summarized in Chapter seven.

## CHAPTER TWO: INTRODUCTION OF SPEAKER RECOGNITION

### 2.1. Introduction

The speaker recognition problem is popularly considered a pattern recognition problem. The general approach of speaker recognition consists of two stages, namely, the feature extraction stage and the pattern matching stage. The diagram of the speaker recognition system is expressed in Figure 2.

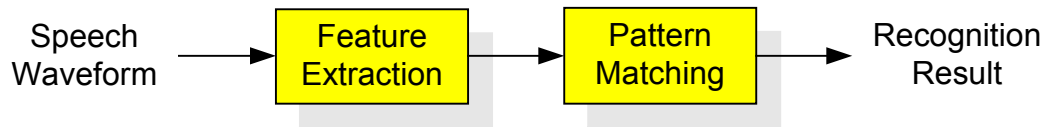


Figure 2. The diagram of the speaker recognition system

This chapter is organized as follows. Section 2.2 explains the feature extraction methods. The filter bank method, the linear predictor analysis, the Mel\_Frequency Cepstral Coefficients (MFCC) approach, and the dynamic features analysis are presented in this section. Pattern matching techniques are introduced in Section 2.3. In this section, Vector Quantization (VQ), Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs) based techniques are illustrated. Finally, data fusion techniques for pattern recognition and speaker recognition are given in Section 2.4.

## **2.2. Speech Feature Extraction**

Most speech or speaker recognition systems contain a signal processing front end. This front end converts the speech waveform into some type of parametric representation (feature vectors) for further analysis and processing. The major task of this step is the data reduction by converting the speech waveform into feature vectors while preserving the useful information for applications.

Many feature extraction methods for speech waveforms have been developed over the past several decades [12, 15, 33, 44, 47, 49, 56, 60]. This section is devoted to the discussion of the most commonly used speech feature extraction techniques for speaker recognition. This section is organized as follows. The digital filter bank technique is presented in Subsection 2.2.1. Subsection 2.2.2 introduces MFCCs. The linear predictor analysis approach is given in Subsection 2.2.3. Finally, Subsection 2.2.4 explains the dynamic features.

### **2.2.1. Digital Filter Bank**

The filter bank approach separates the signal frequency bandwidth into a number of frequency bands and measures the signal energy in each band. This approach estimates the spectral envelope of the speech waveform.

The main advantages of a filter-bank over a Discrete Fourier Transform (DFT) lie in the small number of parameters used to represent the spectrum envelope and the possibility to have a different frequency resolution for each filter. This variable resolution is often used in spectral analyses, which attempt to simulate auditory processes. When a constant frequency resolution is

needed, a filter-bank is typically implemented on the basis of a Fast Fourier Transform (FFT) [25].

A block diagram of the canonic structure of a complete filter-bank front-end analyzer is given in Figure 3 [49].

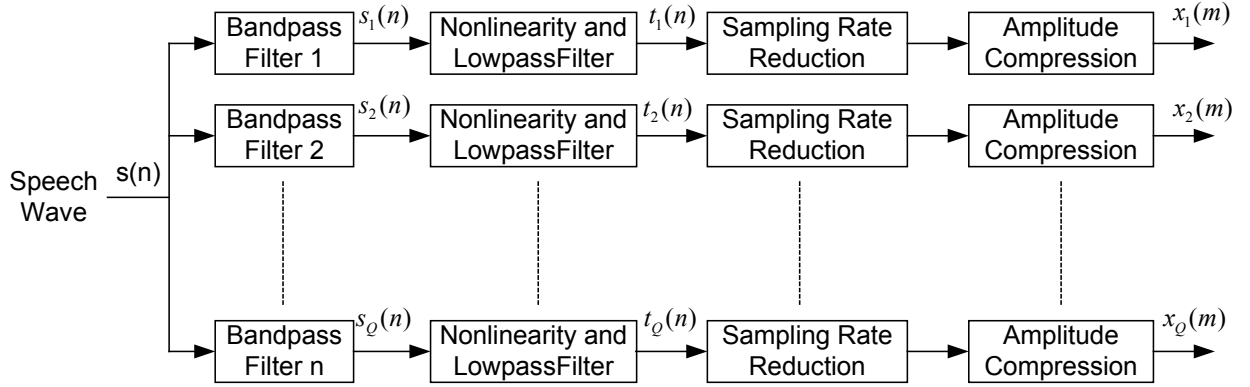


Figure 3. Block diagram of filter bank analysis

In the filter bank analysis, the sampled speech signal,  $s(n)$ , is firstly passed through a bank of  $Q$  bandpass filters. Thus, the  $i$ th bandpass-filtered signal  $s_i(n)$  is given by

$$s_i(n) = s(n) * h_i(n) = \sum_{m=0}^{M_i-1} h_i(m) s(n-m) \quad 1 \leq i \leq Q$$

where  $h_i(m)$  is the impulse response of the  $i$ th bandpass filter with a duration of  $M_i$  samples. Since the purpose of the filter-bank analyzer is to give a measurement of the energy of the speech signal in a given frequency band, each of the bandpass-filtered signal  $s_i(n)$  is passed through a nonlinearity block. The nonlinearity block shifts the bandpass-filtered signal's spectrum to the low frequency band as well as creates high-frequency images. A low pass filter is used to eliminate high frequency images. Each lowpass-filtered signal  $t_i(n)$  is resampled at a

rate on the order of 40-60 Hz. Finally, the signal's dynamic range is compressed by using an amplitude compression scheme to obtain output  $x_i(m)$  ( $1 \leq i \leq Q$ ) [49].

How to select the suitable filter bank is the key problem in the filter bank analysis. The typical filter bank used for speaker recognition is a uniform filter bank. The center frequency,  $f_i$ , of the  $i$ th bandpass filter is defined as

$$f_i = \frac{F_s}{N} i, 1 \leq i \leq Q$$

where  $F_s$  is the sampling rate of the speech signal, and  $N$  is the number of uniformly spaced filters required to span the frequency range of the speech. The actual number of filters used in the filter bank,  $Q$ , satisfies the relation  $Q \leq N/2$  [49].

The alternative to the uniform filter bank is the nonuniform filter bank. Its design is based on certain criterion for how the individual filter should be spaced in the frequency domain. The critical band is the most popularly used criterion in the filter bank design. Experiments suggest the existence of an auditory filter in the vicinity of the tone that effectively blocks extraneous information from interfering with the detection of the tone. This vicinity is called a critical band and can be viewed as the bandwidth of each auditory filter. The experimental results show that the width of a critical band increases with the higher frequency if the tone is masked. Thus, these results yield important information about the bandwidth of the auditory filter [17]. The scale is close to linear for frequencies below 1000 Hz (i.e. the bandwidth is essentially constant as a function of the frequency), and is close to logarithmic for frequencies above 1000 Hz (i.e. the bandwidth is essentially exponential as a function of the frequency). Several variants on the critical band scale have been used, including Mel scale and bark scale. The differences between these variants are small and are, for the most part, insignificant with regard to the design of filter

banks for the speaker recognition purpose. In Figure 4, the variation of bandwidths versus the frequency based on the critical band scale is given.

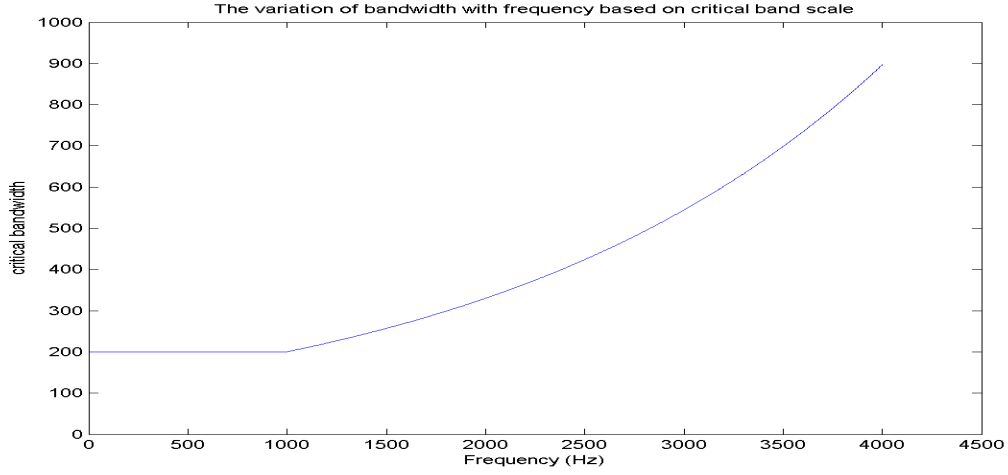


Figure 4. The variation of bandwidths versus the frequency based on the critical band scale

### 2.2.2. Mel\_Frequency Cepstral Coefficients (MFCC)

In the speech analysis, speech is most commonly viewed as the output of a linear, time-varying system (the vocal tract) excited by either quasi-periodic pulse or random noise. Since the easily observable speech signal is the result of convolving the excitation with the vocal tract sample response, it would be useful to separate or deconvolute these two components.

Cepstral deconvolution transforms a product of two spectras into a sum of two signals. If the summed signals are sufficiently different in the spectrum, they may be separated by a linear filter. The desired transformation is logarithmic, which is given by

$$\log(X)=\log(EV)=\log(E)+\log(V)$$

where  $E$  is the Fourier transform of the excitation waveform,  $V$  is the Fourier transform of the vocal tract response and  $X$  is the Fourier transform of the speech signal. Since the formant structure of  $V$  varies slowly in frequency compared to the harmonics or noise in  $E$ , contribution due to  $E$  and  $V$  can be linearly separated after an inverse Fourier transform [56].

The real cepstrum is computed by taking the inverse  $z$  transform on the unit circle. It is given by

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log|x(w)|e^{jwn} dw$$

It is also possible to define a complex cepstrum that gives a useful insight into properties of actual systems [17].

Mel\_Frequency Cepstral Coefficients (MFCCs), or shortly called mel\_cepstrum, uses the cepstrum with a nonlinear frequency axis following the Bark or mel scale. It provides an alternative representation for speech spectra [33, 34, 60].

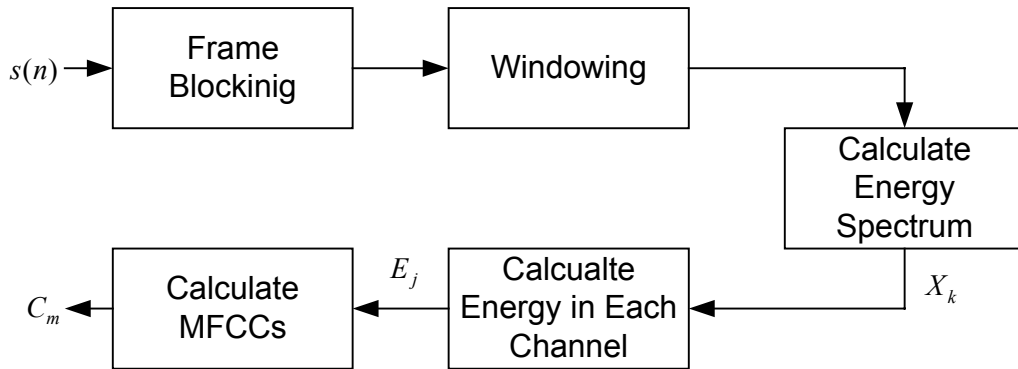


Figure 5. The block diagram of MFCCs



The evaluation techniques of MFCCs are given in Figure 5 and can be summarized as follows [60].

#### 1) Block and window frames

One of the key measures used in speech processing is the short-term spectrum. In all of its many forms, this measure consists of some kind of the local spectral estimate, which is typically measured over a relatively short region of speech. This measure is trying to capture the time-varying spectral envelope for the speech and to reduce the effect of pitch.

To extract the short-time features of a speech signal, the speech waveform is blocked into short segments called frames. The duration of each frame varies from 15 to 30 ms. The speech belonging to each frame is assumed to be stationary.

To reduce the edge effect of each segment, a smoothing window is applied to each frame. Each successive frame is allowed to overlap each other, so that a smoother feature set over time can be generated.

The popularly used window functions are Hamming window and Hanning window. Hamming window is given by

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N_w - 1}\right), 0 \leq n < N_w$$

and Hanning window is presented by

$$W(n) = 0.5 - 0.5 \cos\left(\frac{2\pi n}{N_w - 1}\right), 0 \leq n < N_w$$

#### 2) Calculate the energy spectrum

Assume there is a speech waveform inputted. After segmentation and windowing, Fast Fourier Transform (FFT) is used. It converts each frame of  $N$  samples from the time domain into the frequency domain. The transform is given as follows

$$x(k) = \sum_{n=0}^{N_w-1} s(n)W(n)e^{-j2\pi nk/N_w}, 0 \leq k < N_w$$

where  $s(n)$  is the input speech waveform,  $N_w$  corresponds to the size of each frame and  $W(n)$  is the window function.

The energy spectrum is expressed as

$$X_k = |x(k)|^2, 0 \leq k < K$$

where  $K$  is taken equal to  $N_w/2$ , since only half of the spectrum needs to be considered.

3) Calculate the energy in each channel

The energy in each channel is given by

$$E_j = \sum_{k=0}^{K-1} \phi_j(k) X_k, 0 \leq j < J$$

Where  $\phi_j$  is triangular filters and  $J$  is the number of triangle filters. Triangle filter  $\phi_j$  has the following constraint:

$$\sum_{k=0}^{K-1} \phi_j(k) = 1, \forall j$$

The distribution of these filters before normalization is shown in Figure 6.

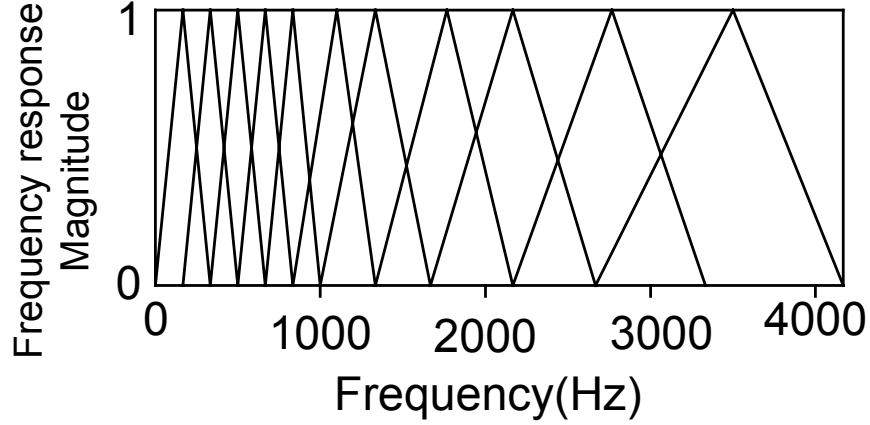


Figure 6. Filter allocation in the frequency domain before normalization

### 3) Calculate the MFCC

In the final step, we convert the log mel spectrum back into time. The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the MFCCs (and so their logarithm) are real numbers, we can convert them into the time domain using Discrete Cosine Transform (DCT).

$$c_m = \sum_{j=0}^{J-1} \cos\left(m \frac{\pi}{J}(j+0.5)\right) \log_{10}(E_j) \quad (2.1)$$

Let weighting factors be

$$V_m = \left\{ \cos\left(m \frac{\pi}{J}(j+0.5)\right) \middle| 0 \leq j < J \right\}$$

(2.1) can be rewritten as

$$c_m = \sum_{j=0}^{J-1} V_{m,j} \log_{10}(E_j)$$

where  $c_m$  is MFCC. Generally, only the first 15 values of  $c_m$  are retained.

### 2.2.3. Linear Prediction Analysis

Linear prediction has been widely used in the model-based representation of signals [30, 34, 44, 49, 61]. The premise of such representation is that a broadband and spectrally flat excitation,  $r(n)$ , is processed by an all pole filter to generate the signal. The coefficients of all poles autoregressive system are derived by the LP analysis, a process that derives a set of moving average coefficients,  $A_i = [a_{i0}, -a_{i1}, \dots, -a_{im}]^T$  with  $a_{i0}=1$ . The LP predicts the present signal sample,  $x_i(n)$ , from  $m$  previous values by minimizing the energy in the system output. The system output is referred to as the prediction residual error  $R_i = [r_i(0), r_i(1), \dots, r_i(N-1)]^T$ . The frame size  $N$  is chosen such that the signal is relatively stationary.

The LP analysis process in the time domain is expressed by

$$r_i(n) = x_i(n) - \sum_{k=1}^m a_{ik} x_i(n-k), \quad n=0,1, \dots, N-1$$

Equivalently, in z domain, the response of the LP analysis filter is given by

$$A_i(z) = 1 - \sum_{k=1}^m a_{ik} z^{-k}$$

The LP analysis filter decorrelates the excitation and the impulse response of the all pole synthesis filter to generate the prediction residual,  $R_i$ , that is an estimate of the excitation signal  $r(n)$ .

While decoding, the signal  $x_i(n)$  is synthesized by filtering the excitation,  $r_i(n)$ , by the autoregressive synthesis filter whose pole locations correspond to zeros of the LP analysis filter. The response of the synthesis filter is given by

$$H_i(z) = \frac{1}{1 - \sum_{k=1}^m a_{ik} z^{-k}}$$

The sinusoid frequency response of the synthesis filter,  $H_i(f)$ , is obtained by evaluating over the unit circle in the  $z$  plane. Thus

$$H_i(f) = \frac{1}{1 - \sum_{k=1}^m a_{ik} \exp(-j2\pi kf)}$$

where  $z = \exp(j2\pi f)$  and frequency  $f$  is normalized with respect to the sampling frequency.

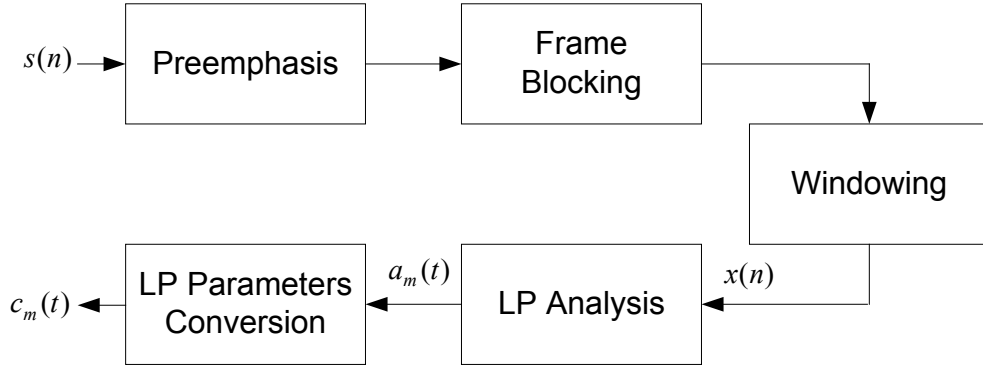


Figure 7. The block diagram of LPC processor for speaker recognition

Figure 7 shows the block diagram of the LPC processor for speaker recognition. The basic steps in the processing include the following [49]:

#### 1) Preemphasis

In many feature extraction approaches, the speech is first pre-emphasized with a pre-emphasis filter, which is initially motivated by the speech production model. From the speech production model of the voiced speech, there is an overall of -6 dB/octave due decay (-12

dB/octave due to excitation source and +6 dB/octave due to the radiation compensation) as frequency increases. The spectrum of the speech is flattening by a pre-emphasis filter of the form

$$H(z) = 1 - \tilde{a}z^{-1}$$

Typically, parameter  $\tilde{a}$  is selected around 0.95. While the pre-emphasis filter does its job for voiced speech, it causes a +12 dB/octave rise in unvoiced speech.

## 2) Frame blocking and windowing

The frame blocking and windowing steps for LPC are exactly the same as those of the MFCCs process, which is given in the last subsection.

## 3) LPC analysis

The original LP coefficients can be calculated by the autocorrelation method or the covariance method. The least-squares autocorrelation method chooses LP coefficients  $a_k$  to minimize the mean energy in the error signal over a frame of speech data.

Let  $E$  be the error energy:

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left[ x(n) - \sum_{k=1}^p a_k x(n-k) \right]^2$$

where  $e(n)$  is the residual corresponding to the windowed signal  $x(n)$ .

Minimum prediction error  $E$  is obtained, when

$$\sum_{k=1}^p a_k R(i-k) = R(i)$$

where autocorrelation  $R(i)$  of  $x(n)$  is  $R(i) = \sum_{n=i}^{N-1} x(n)x(n-i)$ ,  $i=1, 2, \dots, p$ .

The most popularly used method for converting autocorrelation coefficients to an LP parameter set is known as Durbin's method and can be formally given as the following algorithm

$$E^{(0)} = r(0) \quad (2.2a)$$

$$k_i = \frac{\left\{ r(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} r(|i-j|) \right\}}{E^{(i-1)}} \quad (2.2b)$$

$$\alpha_i^{(i)} = k_i \quad (2.2c)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad (2.2d)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (2.2e)$$

where the summation in (2.2b) is omitted for  $i=1$ . The set of Equations (2.2a-2.2e) are solved recursively for  $i=1, 2, \dots, p$ , and the final solution is given as

$$\alpha_m = \text{LPC coefficients} = \alpha_m^{(p)}, 1 \leq m \leq p$$

$$k_m = \text{the reflection (or PARCOR) coefficients}$$

$$g_m = \text{log area ratio (LAR) coefficients} = \log[(1 - k_m)/(1 + k_m)]$$

The reflection coefficients obey the condition  $|k_m| < 1$ , for  $m=1, 2, \dots, p$ . If they are coded within the limits of  $-1$  and  $+1$ , the stability of the synthesis filter can be ensured. Alternatively, a quantization error in encoding the LAR parameters, maintains the condition  $|k_m| < 1$ , and thus ensures the poles of the reconstructed synthesis filter lying within the unit circle.

#### 4) LPC parameters conversion

In order to assure the stability of the synthesis filter, LP coefficients are not directly encoded. Other equivalent representations of the LP coefficients, such as Linear Spectral Frequency (LSF) [57], Log Area Ratio (LAR) [61] or LPC cepstrum [49] are used. Among them, Log Area Ratio (LAR) has been introduced in Durbin's algorithm.

LSP is introduced as follows. The recursive relationship of  $A_{n+1}(z)$  in term of  $A_n(z)$  is

$$A_{n+1}(z) = A_n(z) - k_{n+1} z^{-(n+1)} A_n(z^{-1})$$

Let  $P_{n+1}(z)$  be  $A_{n+1}(z)$  with  $k_{n+1}=1$ . The difference filter is obtained

$$P_{n+1}(z) = A_n(z) - z^{-(n+1)}A_n(z^{-1})$$

Likewise, let  $Q_{n+1}(z)$  be  $A_{n+1}(z)$  with  $k_{n+1}=-1$ . The sum filter is achieved

$$Q_{n+1}(z) = A_n(z) + z^{-(n+1)}A_n(z^{-1})$$

Decomposing the difference filter, we have

$$P_{n+1}(z) = (1 - z^{-1}) \prod_{k=1}^{N/2} (1 + d_k z^{-1} + z^{-2})$$

where  $d_k = -2\cos(2\pi f_k t_s)$ ,  $f_k$  is the  $k$ th LSF associated with  $P_{n+1}(z)$ , and  $t_s$  is the speech sampling time interval.

Similarly, decomposing the sum filter gives

$$Q_{n+1}(z) = (1 + z^{-1}) \prod_{k=1}^{N/2} (1 + d'_k z^{-1} + z^{-2})$$

where  $d'_k = -2\cos(2\pi f'_k t_s)$  and  $f'_k$  is the  $k$ th LSF associated with  $Q_{n+1}(z)$ .

A very important LPC parameter set, which can be derived directly from the LPC coefficient set, is the LPC cepstral coefficients,  $c(m)$  [49]. They are given by

$$c_0 = \ln \sigma^2$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left( \frac{k}{m} \right) c_k a_{m-k}, \quad 1 \leq m \leq p$$

$$c_m = \sum_{k=m-p}^{m-1} \left( \frac{k}{m} \right) c_k a_{m-1}, \quad m > p$$

where  $\sigma^2$  is the gain term in the LPC model. The cepstral coefficients, which are the coefficients of the Fourier transform representation of the log magnitude spectrum, have been shown to be a more robust, reliable feature set for speech recognition than LP coefficients, PARCOR coefficients, or LAR coefficients. Generally,  $Q$  coefficients are used, where  $Q \approx 3/2p$ .



### 2.2.5. Dynamic Feature

Feature vectors from the LPC analysis or the MFCCs analysis provide good smooth estimates of the local spectra. They are considered the static measures of the speech. However, it could be argued that a key characteristic of speech is its dynamic behavior [17].

The dynamic feature of speech is often represented by a time differential log spectrum. The time differential log spectrum is typically implemented as a least square approximate to the local slope, which is a smoother estimate of the local derivative than a simple difference between cepstrals for neighboring frames. Thus, the dynamic feature vector  $\Delta C_i$  is given by [49]

$$\Delta C_i(n) = \frac{\sum_{k=-N}^N k C_i(n+k)}{\sum_{k=-N}^N k^2}$$

where  $C_i$  is the corresponding static feature vector. The second order derivative can be retrieved by a similar method.

It has been reported that the correlation between the differential spectral distance and the spectral distance was found to be 0.6, which is quite small relative to correlations between spectral representations observed in speech. [49]

Many speech recognition systems have the incorporate dynamic feature. They tend to emphasize the dynamic aspects of the speech spectrum over time. The dynamic feature is relatively insensitive to constant spectral characteristics that might be unrelated to the linguistic content in speech, such as the long-term average spectral slope. However, dynamic feature vectors miss some of the gross characteristics that are salient in the static spectral representation, and dynamic feature vectors are not often sufficient for good recognition performance. In

practice, most systems that incorporate the dynamic features use the dynamic feature as an addition to static measures [17].

### **2.3. Pattern Matching Methods for Speaker Recognition**

The speech waveform can be directly represented by the time sequence of feature vectors, which are obtained from the front-end feature extraction analysis as we have discussed in the previous section. A key question in speaker recognition is how speech patterns are compared to determine their similarity (or equivalently, the distance between patterns). The most popular pattern matching methods for speaker recognition include the VQ based approach [4, 58], the Dynamic Time-Warping (DTW) based approach [54] and the Hidden Markov Models (HMMs) based approach [49, 59]. The Gaussian Mixture Models (GMMs) based approach [51] is a special case of the HMMs based approach. DTW is used exclusively for text-dependent applications, while VQ and HMMs deal with both text-dependent and text-independent speaker recognition.

In the training mode of the DTW approach, the speaker templates, which are the sequences of feature vectors obtained from the text-dependent speech waveforms, are created. In the testing mode, matching scores are produced by using DTW to align and measure the similarities between the test waveform and the speaker templates [4, 54].

In the VQ based approach, a codebook for each speaker is obtained as a reference template for the speaker in the training mode. In the testing mode, the average VQ distortions of testing speech feature vector quantized by speakers' codebooks are calculated. The average VQ distortions here show the similarities between the unknown speaker's speech and the reference

templates. The smaller the average VQ distortion, the better the matching between the testing speech and the reference template. The lack of time warping in the VQ approach greatly simplifies the system. However, some speaker-dependent temporal information, which is present in the waveforms, is neglected in the VQ technique [4].

In the HMMs approach, the sequences of feature vectors, which are extracted from the speech waveforms, are assumed to be a Markov Process and can be modeled with an HMM. During the training mode, HMMs' parameters are estimated from the speech waveforms. In the testing mode, the likelihood of the test feature sequence is computed against the speaker's HMMs [59].

It is reported that the performance of the continuous ergodic HMMs is about the same as that of the VQ method and is much higher than that of the discrete ergodic HMMs. From the viewpoint of the number of model parameters, the continuous ergodic HMMs outperformed the VQ method [38, 62, 64]. However, the computational complexity of the VQ approach is much less than that of the HMMs approach [49].

In this section, VQ and DTW approaches are introduced in Subsection 2.3.1 and 2.3.2. Then, the HMMs method is presented in Subsection 2.3.3.

### **2.3.1. Vector Quantization**

Vector quantization is a generalization of Scalar Quantization (SQ). It is the quantization of a vector, an order set of real numbers. The jump from one dimension to multiple dimensions is a major step. It allows a wealth of new ideas, concepts, techniques, and applications to arise that often have no counterpart in the simple case of SQ. VQ considers the correlation between the

items in the vector, and also gives an enhanced flexibility in the quantizer's structure. This makes VQ inherently better than SQ [13, 18, 27, 32, 40-42, 57].

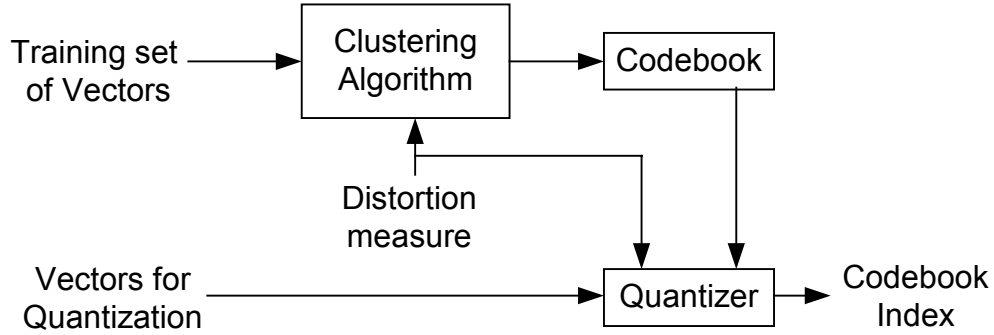


Figure 8. The vector quantization system

A vector quantizer  $Q$  of dimension  $k$  and size  $N$  is a process that maps the a  $k$ -dimension vector into a set of  $N$   $k$ -dimension vectors

$$Q: R^k \rightarrow C$$

where codebook  $C=(y_1, y_2, \dots, y_N)$  is the set of  $N$   $k$ -dimension vectors and  $y_i \in R^k$  ( $i=1, 2, \dots, N$ ) is denoted by code vector or codeword. The codebook size  $N$  is a critical parameter. It determines the accuracy (or the average distortion) of the quantization, the encoding complexity needed for searching through the codebook and the memory required to store the codebook.

The resolution or code rate of VQ is defined as

$$r=(\log_2 N)/k$$

It measures the number of bits per vector component to represent input vectors.

The Signal to Noise ratio (SNR) for VQ in the high rate region is given by the relation

$$SNR=6(\log_2 N)/k+h_k \quad (2.3)$$

where  $h_k$  is a constant depending on the dimension  $k$ . SNR and  $h_k$  are expressed in dB units in (2.3). SNR for the VQ increases approximately at the rate of  $6/k$  dB for each doubling of the codebook size (for each additional bit used to code the entire vector).

VQ produces less distortion than SQ for the same number of bits in the quantization. Firstly, VQ exploits linear and non-linear dependence among the vectors to be quantized. For many data sources, such as images and audios, the datum are often highly correlated. VQ considers their inherent relationship that SQ ignores. Secondly, in multi-dimension SQ, the quantization cells are always rectangles, but in VQ, the quantization cell is much more flexible in shape. The flexibility of VQ over SQ leads to a higher compression rate. In fact, even when the components in a vector are statistically independent of each other, VQ has better performance than SQ.

The key part of VQ is the codebook construction. In the codebook construction, the overall performance of VQ is evaluated by the statistical average of a suitable distortion measure.

The optimality design of the VQ codebook must meet the following three conditions [18].

#### 1) The Nearest Neighbor Condition

For a given codebook  $C=(y_1, y_2, \dots, y_N)$ , the optimal partition cell must satisfy

$$Q(x)=y_j, \text{ only if } d(x, y_j) \leq d(x, y_i) \text{ for all } i,$$

where  $x$  is the input vector for quantization,  $Q$  is the vector quantizer, and  $d(x, y_i)$  is the distortion measure between vector  $x$  and  $y_i$ .

#### 2) The Centroid Condition

For a given partition  $R_i (i=1, 2, \dots, N)$ , the optimal code vectors must satisfy

$$y_i = \text{cent}(R_i)$$

It means that code vectors are the centroids of partition  $R_i (i=1, 2, \dots, N)$ .

### 3) The Zero Probability Boundary Condition

The necessary condition for a codebook to be optimal for a given source distribution is

$$P\left(\bigcup_{j=1}^N B_j\right) = 0$$

Thus, the boundary point occurs with zero probability.

The most popularly used VQ codebook construction method is the General Lloyd (GL) algorithm (also known as the LBG algorithm or the k-means algorithm). This algorithm can be described by following steps [18].

Step1.

Begin with an initial codebook  $C_1$ , set  $m=1$ . (The selection of the initial codebook will be explained later.)

Step 2.

Given the codebook  $C_m$ , perform the Lloyd Iteration to generate the improved codebook  $C_{m+1}$ . (Lloyd Iteration will be given later).

Step3.

Compute the average distortion for  $C_{m+1}$ . If the average distortion has changed by a small enough amount since the last iteration, stop. Otherwise set  $m+1 \rightarrow m$  and go to Step 2.

The Lloyd Iteration for empirical data is given by

(a) Given a codebook,  $C_m = \{y_i\}$ , by using the Nearest Neighbor condition, partition the training vector set  $T$  into clustering sets  $R_i$  ( $i=1, 2, \dots, N$ ),

$$R_i = \{x \mid x \in T : d(x, y_i) \leq d(x, y_j); \forall j \neq i\}$$

- (b) Using the Centroid Condition, compute the centroid for the cluster sets to obtain the new codebook,  $C_{m+1} = \{\text{cent}(R_i)\}$ . If an empty cell was generated in step (a), an alternate codeword needs to assign for this cell.

The GL algorithm is a decent algorithm. It always decreases the average distortion for each iteration. Thus, it can be viewed as an additive algorithm for any other codebook construction approach. The GL algorithm can only lead to a local optimal, and cannot guarantee the global optimal. The final results of the GL algorithm depend on the initial codebooks for the codebook construction.

The simplest way to select the initial codebook is the random initialization. The initial codebook may be randomly chosen among the set of vectors that are used for clustering.

Another popularly used algorithm to get the initial codebook is the centroid split [32]. It is formally implemented by the following recursive procedure.

- 1). Let the codebook size  $N=1$ , calculate the centroid of the entire set of training vectors (Hence, no iteration is required here).
- 2). Double the size of the codebook by splitting each current codeword, for  $i=1, 2, \dots, N$

$$C_{new}^i = (1+\varepsilon) C_{old}^i$$

$$C_{new}^{N+i} = (1-\varepsilon) C_{old}^i$$

where  $\varepsilon$  is a small splitting parameter (such as  $\varepsilon=0.01$ ),  $C_{old}^i$  and  $C_{new}^i$  are the  $i$ th codeword in the old codebook and the  $i$ th codeword in the new codebook respectively. Then set the codebook size  $N = 2N$ .

- 3), Use a codebook construction algorithm such as the GL algorithm to construct the renewed codebook.

4), Repeat Steps 3 and 4 until the desired number of codewords is obtained.

The computational complexity of VQ is often larger than that of SQ. One approach to circumvent the complexity problem is to impose certain structural barriers on the codebook construction. This means that codewords in the codebook cannot have arbitrary locations in the  $k$ -dimensional space, but are distributed in a restricted manner. The new structure of the codebook construction in the restricted manner should bring a much easier search strategy for VQ. However, any constraints imposed on the codebook construction will certainly lead to an inferior codebook for a given rate and dimension [18].

One of the most effective and widely used techniques for reducing the search complexity in VQ is the tree-structured codebook search [18]. Tree-Structured VQ (TSVQ) greatly reduces the encoding complexity at the expense of a need for more memory and a lower SNR. The encoding process in a TSVQ search is completed in stages. In an  $m$ -ary balance tree search, the vector for quantization is compared with  $m$  pre-designed test vectors at each stage of the searching tree. Then, one out of  $m$  paths through the tree is selected for the next stage by using the nearest neighborhood criterion. At each stage, the number of candidate codewords in the codebook is reduced to  $1/m$  of the previous set of candidates.

If the codebook size is  $N=m^d$ , then  $d$   $m$ -ary search stages are needed to locate the chosen codeword. An  $m$ -ary tree with  $d$  stages is said to have breadth  $m$  and depth  $d$ .

The TSVQ design procedure is given as follows [18]

Step1.

Use the training set  $T$  to generate a codebook  $C^*$  of size  $m$  test vectors for the root node (level 1) of the tree. Partition the training set into  $m$  new subsets  $T_0, T_1, \dots, T_{m-1}$ .

Step 2.



For each  $i$  ( $i=1, 2, \dots, m-1$ ), design a test codebook  $C_i$  of size  $m$  using the GL algorithm applied to  $T_i$ . Then, the test codebooks for the  $m$  nodes at level 2 of the tree are obtained.

Step 3.

Continue this process until level  $d-1$  is reached.

The TSVQ encoder is expressed as follows [18].

0).

Give depth  $d$ , breath  $m$  and vector  $x$  for quantization.

1).

Root node: Find the codeword  $y \in C^*$  minimizing  $d(x, y)$ , and let  $u_0 \in \{0, 1, \dots, m-1\}$  be the index of this minimum distortion word. Set the one-dimensional channel  $m$ -ary codeword to  $u^1 = u_0$  and advance to node  $(u^1)$ . Set the current tree depth  $k=1$ .

2).

Given the  $k$ -dimension channel codeword  $u^k = (u_0, u_1, \dots, u_{k-1})$  and the current node  $(u^k)$ , find the codeword  $y \in C_{uk}$  to minimize the distortion  $d(x, y)$ . Let  $u_k$  denote the index of the minimum distortion codeword. Set the  $(k+1)$ -dimension channel  $m$ -ary codeword  $u^{k+1}$  equal to the concatenation of  $u^k$  and  $u_k$

$$u^{k+1} = (u^k, u_k) = (u_0, u_1, \dots, u_k)$$

3).

If  $k+1=d$ , halt with the final channel codeword  $u^d$  (corresponding to a reproduced vector in  $C_u^{d-1}$ ). Otherwise set  $k+1 \rightarrow k$  and go to 2).

The total search complexity of tree-structured VQ is proportional to  $md$  rather than  $m^d$ . On the other hand, the storage requirement of the tree-structured VQ is increased compared to the unstructured VQ.

A vector quantizer can be used to describe almost any type of patterns, such as templates of speech and image, by constructing codebooks for them. The VQ encoding process can be considered a pattern matching process. Each vector is encoded by comparison with a set of stored reference vectors, known as codewords or patterns. Each pattern will be used to represent input vectors that are somehow identified as “similar” to this pattern. The best matching pattern in the codebook, the set of stored reference patterns, is selected by the encoding process according to a suitable fidelity measure.

In speaker recognition, the VQ based approach is one of the most important template-based pattern matching methods. In some restricted cases, a good recognition performance can be obtained with straightforward use of VQ as a recognizer. Comparing with other pattern matching methods such as the DTW based approach and the HMMs based approach, the VQ based approach has much lower computational complexity.

Like other pattern matching methods in speaker recognition, the VQ based approach contains two modes, namely, the training mode and the testing mode. In the training mode, a VQ codebook for each enrolled speaker is constructed as the reference pattern. In the testing mode for speaker verification, the feature vector set of the testing speech waveform is VQ encoded by the claimed speaker’s codebook and the average VQ distortion is calculated. If the average distortion is smaller than a given threshold, the waveform is accepted. Otherwise, it is rejected. For speaker identification, the feature vector set of the testing speech waveform is VQ encoded by every enrolled speaker’s codebook. The testing speech waveform is identified to the speaker whose codebook gives the least average VQ distortion.

### 2.3.2. Dynamic Time Warping (DTW)

The different acoustic tokens of a same speech utterance are rarely realized at the same speaking rate across the entire utterance. This fact makes that when comparing different tokens of the same utterance, the speaking rate and the duration of the utterance should not contribute to the similarity measurement. Thus, there is a need to normalize out speaking rate fluctuation in order for the utterance comparison to be meaningful, before a recognition decision can be made. A solution to this problem can be achieved using dynamic programming techniques for time alignment. In DTW, the problem is presented as finding the minimum distance between a set of template speech streams and the input speech streams [4, 17, 49].

Consider two speech patterns,  $X$  and  $Y$ , represented by the spectral sequence  $(x_1, x_2, \dots, x_{T_x})$  and  $(y_1, y_2, \dots, y_{T_y})$ , respectively, where  $x_i$  and  $y_i$  are feature vectors. The time indices of  $X$  and  $Y$  are denoted by  $i_x$  and  $i_y$  respectively, where  $i_x=1, 2, \dots, T_x$  and  $i_y=1, 2, \dots, T_y$ . The duration,  $T_x$  and  $T_y$  need not be identical. The dissimilarity between  $X$  and  $Y$  is defined by considering the distortion  $d(x_{i_x}, y_{i_y})$ . For simplicity,  $d(x_{i_x}, y_{i_y})$  is denoted by  $d(i_x, i_y)$ .

By using two warping functions,

$$i_x=\phi_x(k) \text{ and } i_y=\phi_y(k), \quad k=1, 2, \dots, T$$

the global pattern dissimilarity measure  $d_\phi(X, Y)$  is given by

$$d_\phi(X, Y) = \sum_{k=1}^T d(\phi_x(k), \phi_y(k)) \left( \frac{m(k)}{M_\phi} \right)$$

where  $d(\phi_x(k), \phi_y(k))$  is a defined distortion between  $x_{\phi_x(k)}$  and  $y_{\phi_y(k)}$ ,  $\phi = (\phi_x, \phi_y)$  is the warping function pair,  $m(k)$  is a nonnegative weighting coefficient and  $M_\phi$  is a normalization factor.

The DTW problem can be considered an optimal path problem. The DTW technique is used to find  $d(X, Y)$  as the minimum of  $d_\phi(X, Y)$ , over all possible paths, such that

$$d(X, Y) = \min_{\phi} d_{\phi}(X, Y)$$

where the warping function pair  $\phi$  must satisfy a set of requirements, which is to be discussed later [49]. An example of time normalization of two sequential patterns to a common time index is shown in Figure 9. In Figure 9, the time warping function  $\phi_x$  and  $\phi_y$  map the individual time index  $i_x$  and  $i_y$  respectively, to the common time index  $k$ .

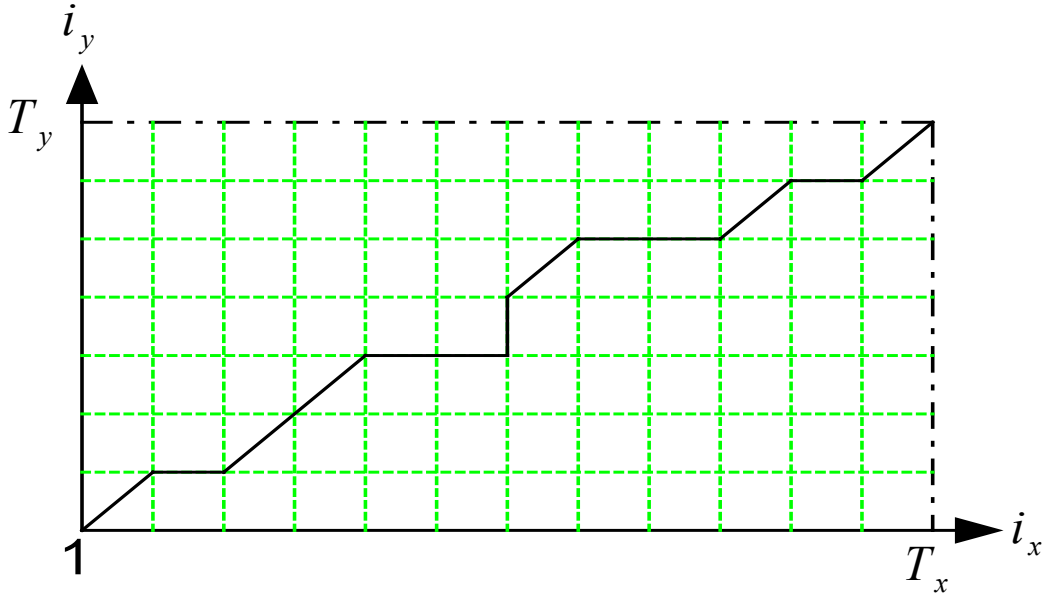


Figure 9. An example of time normalization of two sequential patterns to a common time index

Slope weighting  $m(k)$  along the path adds another dimension of control in the search for the optimal warping path for speech waveform matching. Weighting function  $m(k)$  controls the contribution of each  $d(i_x, i_y)$ . Some of the popularly used slope weightings are given as follows.

$$\text{Type (a): } m(k) = \min[\phi_x(k) - \phi_x(k-1), \phi_y(k) - \phi_y(k-1)]$$

Type (b):  $m(k)=\max[\phi_x(k)-\phi_x(k-1), \phi_y(k)-\phi_y(k-1)]$

Type (c):  $m(k)=\phi_x(k)-\phi_x(k-1)$

Type (d):  $m(k)=\phi_x(k)-\phi_x(k-1), \phi_y(k)-\phi_y(k-1)$

Denote the minimum cost from step 1 to  $j$  by  $\varphi(1, j)$  and one step cost from  $j$  to  $i$  by  $\xi(j, i)$ .

$i$ ). The algorithm used to solve the optimal path problem can be summarized as follows [49].

1) Initialization

$$\varphi_1(1, n) = \xi(i, n) \text{ and } \xi_1 = i, \text{ for } n=1, 2, \dots, N.$$

2) Recursion

$$\varphi_{m+1}(i, n) = \min_{1 \leq l \leq N} [\varphi_m(i, l) + \xi(l, n)] \text{ and}$$

$$\xi_{m+1}(n) = \arg \min_{1 \leq l \leq N} [\varphi_m(i, l) + \xi(l, n)] \text{ for } n=1, 2, \dots, N \text{ and } m=1, 2, \dots, M-2$$

3) Termination

$$\varphi_M(i, n) = \min_{1 \leq l \leq N} [\varphi_{M-1}(i, l) + \xi(l, n)]$$

$$\xi_M(n) = \arg \min_{1 \leq l \leq N} [\varphi_{M-1}(i, l) + \xi(l, n)]$$

4) Path Backtracking

$$\text{Optimal path} = (i, i_1, i_2, \dots, i_{M-1}, j)$$

$$\text{where } i_m = \xi_{m+1}(i_{m+1}), m = M-1, M-2, \dots, 1, \text{ with } i_M = j$$

For the alignment process to be meaningful in terms of time normalization for different renditions of an utterance, some constraints on warping function are necessary. Popularly used warping constraints that are considered necessary and reasonable for time alignment between utterances include the endpoint condition, the monotonic condition, the continuity condition, the adjustment window condition and the slope constraint condition [49].

The DTW based approach can be categorized into the training mode and the testing mode. In the training mode of the DTW approach, enrolled speakers' templates, which are the sequences of feature vectors obtained from the text-dependent speech waveforms, are created. In the testing mode, for speaker identification, matching scores are produced by using DTW to align and measure the similarities between the test waveform and enrolled speakers' templates. The test waveform is classified to the speaker that leads to the highest similarity. However, for speaker verification, the similarity between testing waveform and claimed speaker's template is measured and compared with a threshold to make the decision [4, 54].

### **2.3.3. Hidden Markov Model**

The pattern matching methods can be approximately divided into template-based methods and stochastic-based methods. VQ and DTW are template-based approaches. One key idea in the template-based method is to derive typical sequences of speech frames for a pattern via some average procedure, and relies on the use of the local distance measure to compare patterns. In the template-based approach, the reference pattern can be viewed as the mean of some assumed distribution. While another important statistic information, covariance, is not considered in template-based approaches. In the stochastic-based approaches, both mean and covariance of the training vectors are taken into consideration. Stochastic-based approaches have inherent advantages over template-based approaches.

In stochastic-based approaches, stochastic speaker models are used. The pattern-matching problem can be formulated as measuring the likelihood of an observation for a given speaker model. The observation is a random vector with a conditional probability distribution function

that depends upon the speaker. The conditional probability distribution function for a speaker can be estimated from a set of training vectors. Then, given the estimated density, the probability that the observation was generated by the speaker can be determined [4].

The most popularly researched stochastic method is the Hidden Markov Models (HMMs) based approach [4, 28, 37, 49, 51, 62, 73]. HMMs have a formal probabilistic basis, which has been studied since the 1960s. HMMs have successfully been used in biology as well as speech and speaker recognition. The general problem addressed by the HMM is to build a probabilistic model of a sequence of observations. Gaussian Mixture Model (GMM) can be viewed as a special case of the continuous HMM, where only the probabilities of one observation instead of the probabilities of a sequence of observations are taken into consideration.

In the HMMs based approach, speech is assumed to be a piecewise stationary process. This means that every acoustic utterance is modeled as a series of discrete stationary states, with instantaneous transitions between them.

HMM is the extension of Markov model. Before we discuss HMM, Markov model is introduced first.

Consider a discrete system that can be described at any time as being in one of a set of  $N$  distinct states indexed by  $\{1, 2, \dots, N\}$ . We denote the state  $q_t$  changing with time  $t=1, 2, \dots, m$ . A full probabilistic description of the system would in general, require specification of the current state, as well as all the predecessor states. For the special case, the probability of the current state only depends on the previous state. That is first order Markov chain. It can be described as

$$P[q_t=j | q_{t-1}=i, q_{t-2}=k, \dots] = P[q_t=j | q_{t-1}=i]$$

The state transition matrix  $A$  is given as following

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix}$$

where  $a_{ij} = P[q_t=j | q_{t-1}=i]$ ,  $a_{ij} \geq 0$  for  $\forall j, i$ ,  $1 \leq i, j \leq N$ , and  $\sum_{j=1}^N a_{ij} = 1$  for  $\forall i$ . A Markov chain with

three states is given in Figure 10.

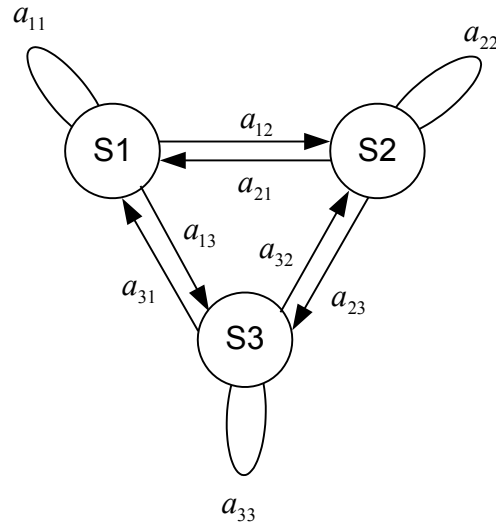


Figure 10. A Markov chain with three states

A Hidden Markov Model (HMM) is simply a Markov model in which the states of Markov model are hidden. Figure 11 shows a hidden Markov Chain. Each output of a Markov model corresponds to a deterministic event, whereas, each output of HMM corresponds to a probabilistic density function of the Markov states. HMM can be classified into discrete models and continuous models according to whether observable events assigned to each state are discrete or continuous.



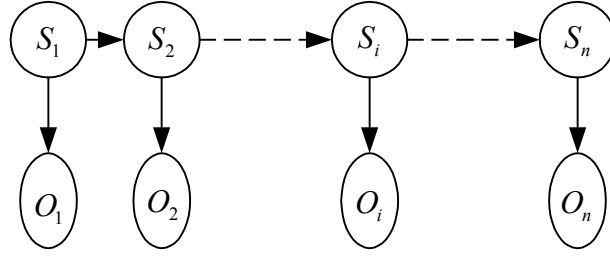


Figure 11. The hidden Markov chain

The elements of HMM are characterized by the following:

- 1),  $Q = \{q_1, q_2, \dots, q_N\}$ , hidden states in the model, where  $N$  is the number of states.
- 2), The state transition probability distribution matrix  $A$ , which is defined before.
- 3), The initial state distribution  $\pi = \{\pi_i\}$ , in which  $\pi_i = P[q_1 = i]$ ,  $1 \leq i \leq N$
- 4), Number of the distinct observation symbols per state,  $M$  (only for discrete HMM). The individual symbol set is denoted by  $V = \{v_1, v_2, \dots, v_M\}$ .
- 5), The observation symbol probability distribution  $B = \{b_1, b_2, \dots, b_M\}$ , in which

For discrete case

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

For continuous case

$$b_j(o) = \sum_{k=1}^M c_{jk} N(o, u_{jk}, U_{jk})$$

where  $o$  is the observation vector being modeled,  $c_{jk}$  is the mixture coefficient for the  $k$ th mixture in state  $j$  and  $N(0, u_{jk}, U_{jk})$  is any log-concave or elliptically symmetric density. Without loss generality, we assume  $N(0, u_{jk}, U_{jk})$  is Gaussian distribution with mean  $u_{jk}$  and covariance matrix  $U_{jk}$  for the  $k$ th mixture component in state  $j$ . The mixture gain  $c_{jk}$  meets the constraint

$$\sum_{k=1}^M c_{jk} = 1, 1 \leq j \leq N \text{ and } c_{jk} \leq 0, 1 \leq j \leq N, 1 \leq k \leq M$$

and the pdf is properly normalized, i.e.

$$\int_{-\infty}^{\infty} b_j(o) do = 1, 1 \leq j \leq N$$

There are three basic problems in the research of HMM. They are given as follows

- P1), **The Evaluation or Scoring Problem:** Given the observation sequence  $O=\{o_1, o_2, \dots, o_T\}$  and model  $\lambda=(A, B, \pi)$ , how to solve the probability  $P(O | \lambda)$ ? The Forward and Backward Algorithm can solve this problem.
- P2), **The Decoding or Alignment Problem:** Given the observation sequence  $O$  and mode  $\lambda$ , how do we choose a corresponding state sequence  $Q$  that is optimal in some sense? The solution of this problem is the Viterbi algorithm
- P3), **The Estimate or Training Problem:** How to estimate the model parameter  $\lambda=(A, B, \pi)$ , to maximize  $P(O | \lambda)$ ? The Baum-Welch algorithm is used to solve this problem

The process of using HMM for speaker recognition contains two modes: the training mode and the testing mode. Firstly, in the training mode, for each enrolled speaker, HMM is estimated. This is to estimate the model parameters  $\lambda=(A, B, \pi)$  to optimize the likelihood of the training observation vector set for each speaker. Then, in the testing mode of speaker identification, for each input waveform, likelihood is measured for estimated HMM of each speaker. The speaker whose model likelihood is the highest is selected as the identification result. For speaker verification, if the likelihood of the input observation for claimed speaker's HMM is larger than a given threshold, the input is accepted. Otherwise the speech waveform is rejected.

## **2.4. Data Fusion Techniques**

Usually, in pattern recognition problems, numerous classifiers with different types of features and/or various pattern matching methods are available. It has been observed that different classifiers for pattern recognition potentially offer complementary information about the patterns to be classified, which could be used to improve the performance of the pattern recognition systems [1, 21, 55, 63]. The idea of data fusion is not to rely on a single classifier to make a decision. Instead, all the classifiers are used for decision making by combining individual opinions of multiple classifiers to obtain a consensus decision. Ideally, the combination should take advantage of the strengths of the individual classifiers, avoid their weakness, and improve the classification accuracy [21, 26]. Data fusion has different names in the literature. They include combination of multiple classifiers, classifier fusion, mixture of experts, consensus aggregation, composite classifier systems and classifier ensembles [29].

Data fusion has proved to be one of the most promising approaches in a variety of pattern recognition fields. These include speaker recognition [9, 11], face identification [3], handwritten character recognition [63], and machine printed word/character recognition [21], etc.

Xu and his colleagues categorize data fusion systems with respect to the type of the raw output information of each classifier into three levels [63]. The first level is the abstract level, where the output of each classifier is a unique class label. The second level is the rank level, where the classifiers rank the candidate classes from the highest to lowest likelihood. The third level is the measurement level, where a similarity score is assigned for each candidate class by each classifier.

At the abstract level, only the identity of the top class is given. Data fusion approaches are based on voting procedures that adopted from the group decision-making theory. The most popularly used approaches, majority voting and plurality decision rule, are introduced here.

Majority voting is a process that chooses the classification decision made by more than half of the classifiers. When no such class is found, the result is considered to be an error [1].

In plurality decision rule, the combined decision is the class, which gets more voted than any other class. This rule is a relaxation of the majority voting rule. The winning class is no longer required to have more than half of the votes. It is shown theoretically and experimentally that the recognition performance of the plurality rule is better than that of the majority voting rule [1].

In the rank level combination, the classifier modules provide us with rank information instead of just top class choices. Each classifier provides a sorted list of classes for every input pattern, arranged in order of the preference. One useful group consensus approach is referred as Borda count. Borda count for a class is the sum of the number of classes ranked below it by each classifier.

Borda count rule can be described as follows. For any class  $\omega_j$ , let  $B_j^i$  be the number of classes, which are ranked below  $\omega_j$  by classifier  $i$  ( $i=1, 2, \dots, M$ ). The Borda count for class  $\omega_j$  is

$$B_j = \sum_{i=1}^M B_j^i$$

Borda count rule picks the class with the highest  $B_j$ .

Many statistic based approaches are presented to solve the data fusion on the rank level. Logistic regression is one of them [21].

In the logistic regression approach, the true class is denoted by  $Y=1$  and other classes are denoted by  $Y=0$ . The probability  $P(Y=1 | x)$  is represented by  $\pi(x)$ , where  $x=(x_1, x_2, \dots, x_m)$  represents the rank scores assigned to that class by classifiers  $C_1, C_2, \dots, C_m$ . The logistic response function is given as

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m)}$$

and

$$\log \frac{\pi(x)}{1 - \pi(x)} = (\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m)$$

where  $\alpha$  and  $\beta=(\beta_1, \beta_2, \dots, \beta_m)$  are constant parameters. The transformation  $L(x) = \log \frac{\pi(x)}{1 - \pi(x)}$  is referred to as the logit, which is linearly related to  $x$  [21].

Methods based on the maximum likelihood or the weighted least squares can be used to estimate the model parameters  $\alpha$  and  $\beta$ . In the testing mode, for each test pattern, the logit for each class is predicted by the estimated model. For speaker identification, the class with the largest logit is considered most likely to be the true class. For speaker verification, the value of  $\pi(x)$  or the logit can be used as a confidence measure. A threshold on these values can be determined experimentally, so that classes with confidence lower than the threshold are rejected [21].

The combination algorithm at the measurement level has accessed to a set of numerical scores provided by the classifiers. For a Bayes classifier  $e_k$  and an input  $x$ , the classification of  $x$  is based on a set of postprobabilities

$$P_k(x \in C_i | x), \text{ for } i=1, 2, \dots, M; k=1, 2, \dots, N$$

where input  $x$  from class  $C_i$  is denoted by  $x \in C_i$ ,  $M$  is the number of classes, and  $N$  is the number of classifiers used for data fusion [63]

For any classifier  $e_k$ , a definitive decision is made as

$$e_k(x)=j \text{ with } P_k(x \in C_i | x) = \max_{i \in \Lambda} P_k(x \in C_i | x)$$

where  $\Lambda = \{1, 2, \dots, M\}$  represents the set of classes.

In the data fusion approach at the measurement level, the most commonly used consensus rule is the linear opinion pool, which is simply a linear weighted average of the experts' estimated probabilities [63].

$$P_E(x \in C_i | x) = \sum_{k=1}^N \alpha_k P_k(x \in C_i | x)$$

where  $\sum_{k=1}^N \alpha_k = 1$  and  $0 < \alpha_k < 1$  ( $k=1, 2, \dots, N$ ) is the weight for class  $k$ .

The data fusion decision is given by

$$E(x)=j, \text{ with } P_k(x \in C_i | x) = \max_{i \in \Lambda} P_E(x \in C_i | x)$$

An alternative to the linear opinion pool is the log opinion pool. The log opinion pool consists of a weighted product of the model output [50]. It is given by

$$P_E(x \in C_i | x) = \prod_{k=1}^N [P_k(x \in C_i | x)]^{\alpha_i}$$

The data fusion decision for the log opinion pool is the same as that of the linear opinion pool.

In the log opinion pool method, if one of  $P_k(x \in C_i | x)$  is zero, the combined probability is also zero. However, in the linear opinion pool approach, the zero probability would be averaged with the other probabilities.

The speaker recognition problem is popularly treated as a pattern recognition problem. The data fusion techniques presented for pattern recognition are applicable to speaker recognition. As introduced before, speaker recognition classifiers are categorized into template-based approaches and stochastic-based approaches. The raw outputs of the template-based approaches are distortions between input speeches and speakers' templates. The raw outputs of stochastic-based approaches are the measures of the likelihood of the speech observation. Since the raw outputs for speaker recognition classifier are similarity scores, it is preferable to consider the data fusion problem of speaker recognition at the measurement level. Then, an incompatible problem is raised in the data fusion for speaker recognition. The raw outputs of classifiers need to be converted into some compatible probability measures, so that the presented data fusion techniques at the measurement level can be applicable to speaker recognition.

## CHAPTER THREE: DISCRIMINATIVE VECTOR QUANTIZATION

### APPROACH FOR SPEAKER IDENTIFICATION

#### 3.1. Introduction

Vector Quantization (VQ) is an important pattern-matching method for automatic speaker recognition. This is due to its simplicity, robustness and efficiency [4, 49]. In the existing VQ techniques for SI (VQSI), a codebook for each speaker is obtained as a reference template in the training mode. Then, in the testing mode, SI is performed by finding the codebook, and its corresponding speaker that gives the smallest average VQ quantization distortion, to represent the unknown speaker waveform [58]. In this chapter, a novel Discriminative Vector Quantization method for Speaker Identification (DVQSI) is proposed, and its parameters selection is discussed. The proposed DVQSI technique takes advantage of the interspeaker variation between two speakers of each speaker pair in the SI group. DVQSI employs discriminative weighted average VQ distortions instead of equally weighted average VQ distortions to make SI decisions in the testing mode.

In SI, all the speakers in the SI group share the same speech feature vector space, since they use the same type of speech feature. The probability distribution of the speech feature vectors of speaker  $a$  (or speaker group  $a$ ) in subspace  $c$  (or region  $c$ ) of the speech feature vector space and the probability distribution of the speech feature vectors of speaker  $b$  (or speaker group  $b$ ) in the same subspace are different. In this dissertation, this difference of the probability distributions is called the interspeaker variation between speaker  $a$  (or speaker group  $a$ ) and speaker  $b$  (or speaker group  $b$ ) in subspace  $c$  (or region  $c$ ). When this interspeaker variation is



large, in subspace  $c$  (or region  $c$ ), the speech templates between speaker  $a$  (or speaker group  $a$ ) and speaker  $b$  (or speaker group  $b$ ) have a large difference, and vice versa. If the subspace  $c$  (or region  $c$ ) equals the whole speech feature vector space, this interspeaker variation is called the interspeaker variation between speaker  $a$  (or speaker group  $a$ ) and speaker  $b$  (or speaker group  $b$ ).

The average distortion measure in the testing mode for VQSI does not consider interspeaker variations inside the speech feature vector space. To increase the SI accuracy, it is expected that the regions of the feature space with higher interspeaker variations should play more important roles than the ones with lower interspeaker variations.

The proposed DVQSI approach can be divided into two modes: the training mode and the testing mode. In the training mode, the training speech waveforms for each speaker are available. Also, a training speech feature vector set is created for each speaker from the speaker's training waveforms. In this mode, the vector space of speech features is firstly divided into a number of subspaces for all speakers and speaker pairs in the SI group. Then, a VQ codebook for each speaker in each subspace is constructed. For every possible combination of speaker pairs, a discriminative weight is assigned for each subspace of the speaker pair, based on the subspace's ability to discriminate between speakers in the speaker pair. Consequently, the subspace, which contains a larger interspeaker variation for the speaker pair, plays a more important role by assigning it a larger discriminative weight. In the testing mode, unknown speaker waveforms are presented for identification. In this mode, discriminative weighted average VQ distortions for speaker pairs are computed for the unknown speaker input waveform. Then, a technique is described that find the best match between the unknown waveform and speakers' templates.

The proposed DVQSI approach can be considered a generalization of the existing VQ technique for Speaker Identification (VQSI). As will be shown later, when suitable parameters of

DVQSI are selected, DVQSI yields better SI accuracies than VQSI. This is confirmed experimentally. In addition, a computationally efficient implementation of the DVQSI technique is given which uses a tree-structured-like approach to obtain the codebooks.

In this dissertation, for VQ and its codebook construction, the definition of the average VQ distortion is given as follows: the average VQ distortion of the vector set  $V=\{\mathbf{v}_i | i=1, 2, \dots, M\}$  quantized by the codebook  $\mathbf{C}$ , is defined by

$$d = \frac{1}{M} \sum_{i=1}^M \min_{1 \leq j \leq n} D(\mathbf{v}_i, \mathbf{y}_j)$$

where  $\mathbf{y}_j(j=1, 2, \dots, n)$  is the codeword of the codebook  $\mathbf{C}$  and  $D(\mathbf{v}_i, \mathbf{y}_j)$  is the distortion (distance) between the vector  $\mathbf{v}_i$  and  $\mathbf{y}_j$  [49]. In this dissertation, the squared error distortion measure is defined by the square Euclidean distance between two vectors. It is given as

$$D(\mathbf{P}, \mathbf{Q}) = \|\mathbf{P} - \mathbf{Q}\| = \sum_{i=1}^k (p_i - q_i)^2$$

where  $\mathbf{P}=[p_1, p_2, \dots, p_k]$  and  $\mathbf{Q}=[q_1, q_2, \dots, q_k]$ .

This chapter is organized as follows: The proposed DVQSI approach and its parameters selection is presented in Section 3.2. Experimental results to evaluate the DVQSI technique are given in Section 3.3. Section 3.4 contains the conclusions.

## **3.2. Speaker Identification Based on Discriminative Vector Quantization**

### **3.2.1. The Training Mode**

In this mode, training speech waveforms for each speaker in SI are available. Also, the training speech feature vector set  $T(k)$  is obtained from training waveforms of each speaker  $k$  by feature extraction techniques, where speaker  $k \in \Lambda$  and  $\Lambda = \{\text{speaker 1, speaker 2, ..., speaker } N\}$  is the closed set of speakers for SI.

The general flow chart of the training mode of the proposed DVQSI technique is given in Figure 12. The first step of DVQSI is to divide the speech feature vector space into a number of subspaces for all speakers and speaker pairs. Next, in each segmented subspace, the codebook for each speaker is constructed by the speaker's training feature vector set in this subspace to represent the speaker's template in the subspace. Finally, a discriminative weight is appropriately calculated for each subspace based on the subspace's interspeaker variation.

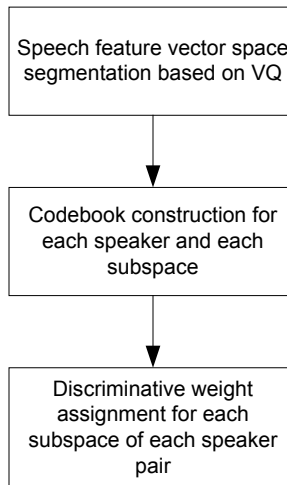


Figure 12. The general flow chart of the training mode of DVQSI

A tree-structured-like codebooks' construction is used in the DVQSI approach. Similar to the tree-structured method, by introducing subspaces, the computational complex of VQ is decreased at the expense of degrading of the speakers' templates represented by codebooks [2]. More subspaces lead to more degradation, which decreases SI accuracy. Also, increasing the number of subspaces results in fewer training feature vectors for the codebooks' construction. This increases the probability that the obtained codebooks for some speakers in certain subspaces do not yield accurate speaker models. In contrast, more subspaces describe the interspeaker variation in more detail. Detailed information of the interspeaker variation enables measuring more accurately the different roles of the various parts of the feature vector space. This is of particular importance in the work presented here. The main difference between the DVQSI approach and the existing VQ technique is due to taking into account the roles of different parts of the feature vector space.

In the first step for DVQSI, the speech feature vector space  $S$  is divided into a desired number of subspaces. Space Segmentation is based on VQ (SSVQ). In SSVQ, for all speakers and speaker pairs, a codebook is constructed by training set  $T^s = \{T(1), T(2), \dots, T(N)\}$ . This codebook is only used for the speech feature vector space segmentation. The codebook size of this codebook equals  $m$ , the desired number of subspaces [18, 27, 32]. The feature space is divided into  $m$  subspaces by using the nearest neighborhood algorithm with codewords of the codebook as centroids of subspaces [20]. In this technique, the space segmentation for all the speakers and speaker pairs is the same and it is only processed once.

After the speech feature space segmentation, subspace codebooks and corresponding average VQ distortions are calculated. For each speaker pair  $k1$  and  $k2$ , and each subspace  $j$  of the speaker pair, a subspace codebook denoted by  $C_j^{kl}(k1, k2)$  is obtained by using training set

$Tr_j^{k1}(\mathbf{k1}, \mathbf{k2})$ , the set for all speech feature vectors of  $T(\mathbf{k1})$  located in subspace  $j$  of the speaker  $k1$  and  $k2$  pair (speaker  $k1 \in \Lambda$ , speaker  $k2 \in \Lambda$ , and  $k1 \neq k2$ ). Similarly,  $C_j^{k2}(\mathbf{k1}, \mathbf{k2})$  is constructed by  $Tr_j^{k2}(\mathbf{k1}, \mathbf{k2})$ , the set for all speech feature vectors of  $T(\mathbf{k2})$  located in the same subspace. Then, average VQ distortions represented by  $d1_j^{k1}(k1, k2)$  and  $d2_j^{k1}(k1, k2)$  for  $Tr_j^{k1}(\mathbf{k1}, \mathbf{k2})$  quantized by subspace codebooks  $C_j^{k1}(\mathbf{k1}, \mathbf{k2})$  and  $C_j^{k2}(\mathbf{k1}, \mathbf{k2})$  are calculated. Meanwhile, average VQ distortions  $d1_j^{k2}(k1, k2)$  and  $d2_j^{k2}(k1, k2)$  are obtained for  $Tr_j^{k2}(\mathbf{k1}, \mathbf{k2})$  quantized by subspace codebooks  $C_j^{k1}(\mathbf{k1}, \mathbf{k2})$  and  $C_j^{k2}(\mathbf{k1}, \mathbf{k2})$ . In this work, the size of the subspace codebook for each speaker and each subspace is the same.

In DVQSI, the SSVQ technique cannot guarantee that the number of the training feature vectors have a small difference for each subspace of each speaker. It is possible that for some speakers and some subspaces, only a few training feature vectors are available. In the construction of subspace codebooks, for each speaker pair  $k1$  and  $k2$ , and each subspace  $j$  of the speaker pair, if the training feature vector set for speaker  $k1$  in the subspace is so small that it cannot guarantee to represent the model of speaker  $k1$  in subspace  $j$  correctly, an empty codebook  $C_j^{k1}(\mathbf{k1}, \mathbf{k2})$  is built and flagged. When the codebook  $C_j^{k1}(\mathbf{k1}, \mathbf{k2})$  is empty, average distortions  $d1_j^{k1}(k1, k2)$  and  $d1_j^{k2}(k1, k2)$  are set to zero and flagged.

The discriminative weight denoted by  $w_j(k1, k2)$  for the speaker pair  $k1$  and  $k2$  (speaker  $k1 \in \Lambda$ , speaker  $k2 \in \Lambda$ , and  $k1 \neq k2$ ) in subspace  $j$  is assigned based on the interspeaker variation of the speaker pair  $k1$  and  $k2$  in subspace  $j$ .

If none of  $d1_j^{k1}(k1, k2)$ ,  $d2_j^{k1}(k1, k2)$ ,  $d1_j^{k2}(k1, k2)$  or  $d2_j^{k2}(k1, k2)$  is zero,  $w_j(k1, k2)$  is obtained by defining  $e_j(k1, k2)$  as one of the following:

$$e_j(k1, k2) = \Gamma[d_{dis(j)}(k1, k2) + d_{dis(j)}(k2, k1)] \quad (3.1a)$$

where

$$d_{dis(j)}(k1, k2) = \left[ \frac{d2_j^{k1}(k1, k2) - d1_j^{k1}(k1, k2)}{d1_j^{k1}(k1, k2)} \right],$$

$$d_{dis(j)}(k2, k1) = \left[ \frac{d1_j^{k2}(k1, k2) - d2_j^{k2}(k1, k2)}{d2_j^{k2}(k1, k2)} \right]$$

are the measurements of the interspeaker variation of speaker pair  $k1$  and  $k2$  in subspace  $j$ , and

$$\Gamma(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

If any of  $d1_j^{k1}(k1, k2)$ ,  $d2_j^{k1}(k1, k2)$ ,  $d1_j^{k2}(k1, k2)$  or  $d2_j^{k2}(k1, k2)$  is zero,  $e_j(k1, k2)$  is defined by

$$e_j(k1, k2) = 0 \quad (3.1b)$$

The normalized  $e_j(k1, k2)$ ,  $\hat{e}_j(k1, k2)$ , is defined as

$$\hat{e}_j(k1, k2) = \frac{me_j(k1, k2)}{\sum_{i=1}^m e_i(k1, k2)} \quad (3.2)$$

Then a threshold  $T1$ , real and greater than zero, is used to limit the maximum value of  $\hat{e}_j(k1, k2)$ . This leads to

$$\bar{e}_j(k1, k2) = \begin{cases} \hat{e}(k1, k2) & \hat{e}(k1, k2) < T1 \\ T1 & \hat{e}(k1, k2) \geq T1 \end{cases} \quad (3.3)$$

Finally, the discriminative weight  $w_j(k1, k2)$  for the speaker pair  $k1$  and  $k2$  in subspace  $j$  is given by

$$w_j(k1, k2) = q(\bar{e}_j(k1, k2)) \quad (3.4)$$

where  $q(x)$  is a non-decreasing function.  $q(x)$  can be selected as

$$q(x)=x^h \quad (3.5a)$$

or

$$q(x)=\Gamma(x-c) \quad (3.5b)$$

where  $h \geq 0$  and  $0 \leq c \leq 1$  are constant scalars.

When one of  $d1_j^{k1}(k1, k2)$ ,  $d2_j^{k1}(k1, k2)$ ,  $d1_j^{k2}(k1, k2)$  or  $d2_j^{k2}(k1, k2)$  is zero, it means that the training feature vector set is not large enough to decide the distribution difference between the feature vector sets of speaker  $k1$  and speaker  $k2$  inside subspace  $j$ . This leads to  $e_j(k1, k2)=0$ , and consequently,  $w_j(k1, k2)=0$ . In the testing mode, when  $w_j(k1, k2)=0$ , the testing vectors in the subspace  $j$  are ignored in the identification of the speaker pair  $k1$  and  $k2$ .

If none of  $d1_j^{k1}(k1, k2)$ ,  $d2_j^{k1}(k1, k2)$ ,  $d1_j^{k2}(k1, k2)$  or  $d2_j^{k2}(k1, k2)$  is zero,  $d_{dis(j)}(k1, k2)$  and  $d_{dis(j)}(k2, k1)$  are the normalized average distortion differences in subspace  $j$ , for the codebooks of speaker  $k1$  and speaker  $k2$ , when the input waveform is from speaker  $k1$  or speaker  $k2$ .  $d_{dis(j)}(k1, k2)$  and  $d_{dis(j)}(k2, k1)$  are the measurements of the distribution difference of the training feature vector sets from speaker  $k1$  and speaker  $k2$ , and the estimators of the distribution difference of the testing feature vector sets from speaker  $k1$  and speaker  $k2$ . Since the discriminative weights are used to identify the speaker pair  $k1$  and  $k2$ , the input from speaker  $k1$  and from speaker  $k2$  should be considered at the same time by adding  $d_{dis(j)}(k1, k2)$  and  $d_{dis(j)}(k2, k1)$ . When  $d_{dis(j)}(k1, k2)+d_{dis(j)}(k2, k1)$  is less than or equal to zero, the distribution difference of the feature vector sets from speaker  $k1$  and speaker  $k2$  cannot be identified in subspace  $j$ , i.e., this subspace is useless or even harmful to identify the speaker pair  $k1$  and  $k2$  in the testing mode. This leads to  $\bar{e}_j(k1, k2) = 0$  and  $w_j(k1, k2)=0$ . The testing feature vectors in subspace  $j$  are ignored in the identification of the speaker pair  $k1$  and  $k2$ . A higher value of  $\bar{e}_j(k1, k2)$  means

that the interspeaker variation between speaker  $k1$  and speaker  $k2$  is larger in subspace  $j$ . A higher discriminative weight  $w_j(k1, k2)$  should be given to subspace  $j$ . Similarly, a smaller value of  $\bar{e}_j(k1, k2)$  leads to a smaller discriminative weight for subspace  $j$  in the identification of the speaker pair  $k1$  and  $k2$ .

Large  $h$  or  $c$  in the function  $q(x)$  emphasizes the importance of the subspaces that have larger  $\bar{e}_j(k1, k2)$ . A very high value for  $h$  or  $c$  makes the subspace that has the largest  $\bar{e}_j(k1, k2)$  become the dominant one for SI, which may not yield the best results. To prevent a subspace playing a dominant role in SI, the threshold  $T$  is added to  $\hat{e}_j(k1, k2)$  ,.

### 3.2.2. The Testing Mode

In the testing mode, testing waveforms from unknown speakers in SI are presented for speaker identification. For each testing waveform  $R$ , a testing speech feature vector set  $\mathbf{T}(\mathbf{R})$  is created. In this mode, for each testing waveform, the discriminative weighted average distortion pairs for speaker pairs used in SI are calculated. The SI decision is then made based on these weighted distortion pairs.

For each speaker pair  $k1$  and  $k2$  used in SI, and each subspace  $j$  of the speaker pair, the average VQ distortion pair  $d_j'(R, k1)$  and  $d_j'(R, k2)$  for subspace  $j$  is calculated for  $\mathbf{T}e_j^R(\mathbf{k1}, \mathbf{k2})$ , the set for all speech feature vectors of  $\mathbf{T}(\mathbf{R})$  in subspace  $j$  of speaker pair  $k1$  and  $k2$ , quantized by codebooks  $\mathbf{C}_j^{k1}(\mathbf{k1}, \mathbf{k2})$  and  $\mathbf{C}_j^{k2}(\mathbf{k1}, \mathbf{k2})$ .

Then, the discriminative weighted average distortion pair for the input  $R$  and the speaker pair  $k1$  and  $k2$  is given by



$$\bar{d}(R, k1) = \frac{\sum_{j=1}^m d'_j(R, k1) w_j(k1, k2) n_j(R)}{\sum_{j=1}^m w_j(k1, k2) n_j(R)} \quad (3.6a)$$

and

$$\bar{d}(R, k2) = \frac{\sum_{j=1}^m d'_j(R, k2) w_j(k2, k1) n_j(R)}{\sum_{j=1}^m w_j(k2, k1) n_j(R)} \quad (3.6b)$$

where  $n_j(R)$  is the number of the feature vectors of the input waveform  $R$  in subspace  $j$ .  $\bar{d}(R, k1) < \bar{d}(R, k2)$  means the template of speaker  $k1$  matches testing waveform  $R$  better than that of speaker  $k2$ , and vice versa.

The flow chart of the SI decision procedure in the test mode of DVQSI is shown in Figure 13. In the beginning, all the speakers in the speaker set  $\Lambda$  are considered the candidates for each testing waveform  $R$ . A speaker pair  $k1$  and  $k2$  from candidate speaker set is randomly selected for the comparison. If  $\bar{d}(R, k1) > \bar{d}(R, k2)$ , speaker  $k1$  is eliminated from the list of candidates since speaker  $k2$ 's template matches the testing waveform better. Otherwise, speaker  $k2$  is eliminated. If  $R$  is not from either speaker  $k1$  or speaker  $k2$ , the elimination of speaker  $k1$  or speaker  $k2$  does not lead to a wrong SI decision, since neither of them is the correct SI result. When  $R$  belongs to speaker  $k1$ , speaker  $k2$  has higher chance to be eliminated from the candidate list than speaker  $k1$ , and vice versa. This elimination process is repeated  $N-1$  times. Consequently,  $N-1$  speakers are eliminated from the candidate set and only one speaker is left. The remaining speaker's template matches the testing waveform best. Consequently, this speaker is considered the identification result for testing waveform  $R$ .

It is apparent that VQSI approach can be considered a special case of DVQSI, where only one subspace exists.

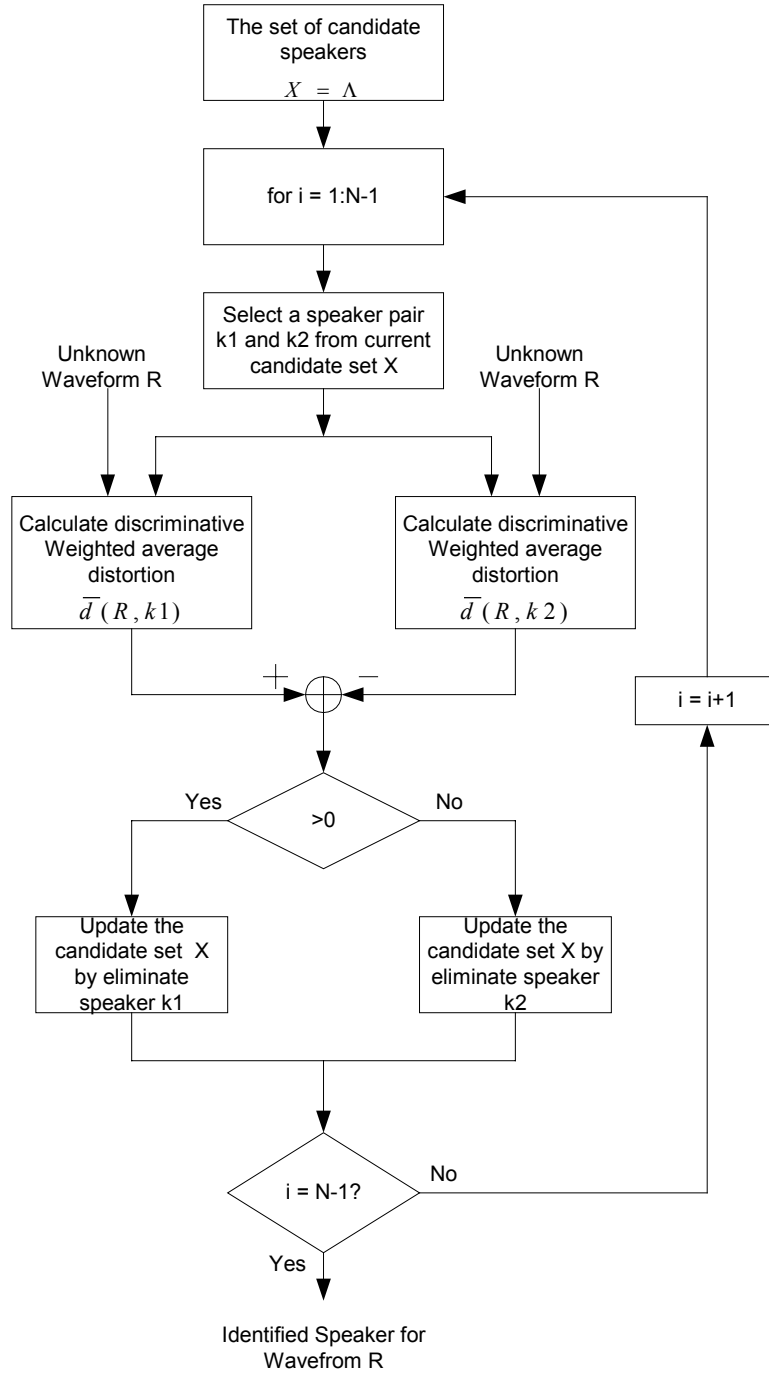


Figure 13. The flow chart of the testing mode of DVQSI

### **3.3. Experimental Results**

In this section, an experiment is given to evaluate the effectiveness of the proposed DVQSI approach. Speech records are obtained from the CSLU (Center for Spoken Language Understanding, Oregon Health & Science University) Speaker Recognition V1.1 corpus. For each speaker, the speech records collected on different collection dates are packaged into different recording sessions. There are mismatches between the speech utterances taken from different speakers or different recording sessions of the same speaker. All the speech files in the corpus were sampled at 8 kHz and 8-bits per sample.

Fifteen speakers are used in the text-independent SI experiments. One spontaneous speech for each speaker is used in the construction of the codebook. Another spontaneous speech taken about one year after the training speech waveform for each speaker is used in the testing mode. Each speech waveform in the training mode lasts about 15 to 20 seconds and the one in the testing mode lasts about 8 seconds.

Twenty speakers are used in the text-dependent SI experiments. The sentence used in the text-dependent experiments is randomly selected. Two text-dependent speech phrases recorded separately about two weeks apart for each speaker are used in the codebooks construction. One text-dependent speech phrase taken about one year after the training speech waveform for each speaker is used in the testing mode. Each phrase in the training mode and the testing mode lasts about 2 to 3 seconds.

Silenced and unvoiced segments are discarded based on an energy threshold. The analysis Hamming window size is 32ms (256samples) with 28ms overlapping. The feature

vector used in the experiment is composed of 15 Mel Frequency Cepstral Coefficients (MFCC's) [33, 60]

The codebook size of VQSI used for comparison is 64. The codebook size in each subspace for DVQSI is  $64/m$ , where  $m$  is the number of the subspaces as mentioned before. The threshold  $T=5m/(m+4)$  for (3.5a), and  $T=5m/(m+4)+c$  for (3.5b). All the codebooks are constructed by the Generalized Lloyd algorithm with the splitting algorithm for the initial values [18, 27, 32].

The experimental results employing DVQSI are shown in Figure 14 and Figure 15, and Table 1 and Table 2 for the text-independent case. The results for the text-dependent case are given in Figure 16 and Figure 17, and Table 3 and Table 4. In Figure 14 to Figure 17, when the number of subspaces  $m$  equals 1, the proposed DVQSI approach corresponds to the VQSI technique.

In text-independent experiments, Figure 14 and Table 1, dividing the feature space into 8 subspaces and selecting  $h$  in (3.5a) equal to 1 or 2 yields the highest SI accuracy. When  $h = 1, 2, 3, 4, 5$ , or 6, and the number of segmented subspaces  $m = 4, 8$ , or 16, the SI accuracies of DVQSI are better than those of VQSI. If  $m = 32$  and  $h = 3, 4, 5$ , or 6, the performance of DVQSI is worse than that of VQSI.

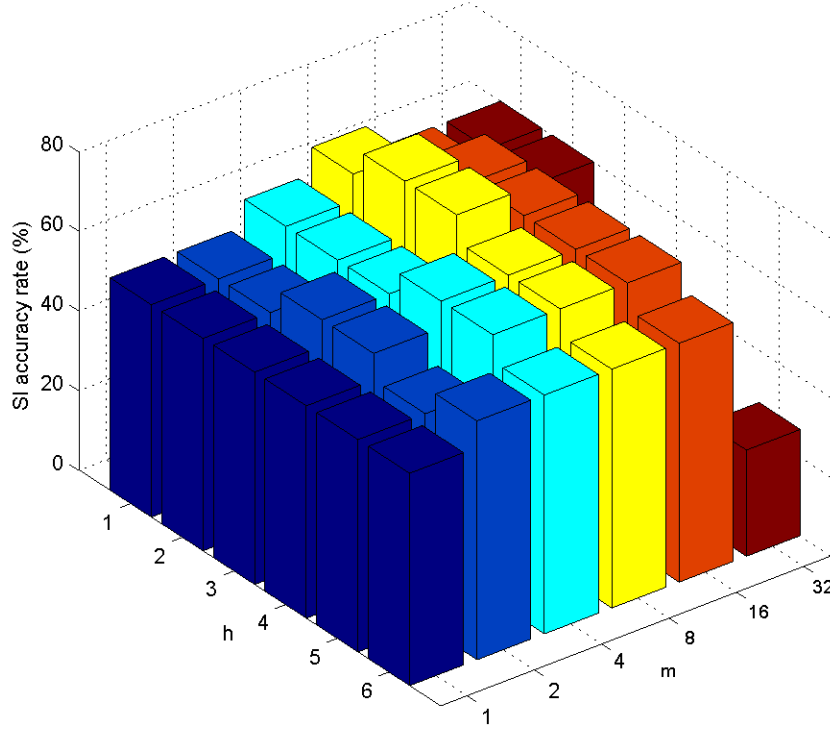


Figure 14. SI accuracy rates versus the number of subspaces  $m$  and the parameter  $h$  of (3.5a) in text-independent experiments

When  $m$  is equal to 1, DVQSI degrades into VQSI.

Table 1. SI accuracy rates of VQSI and DVQSI versus the number of subspaces  $m$  and the parameter  $h$  of (3.5a) in text-independent experiments

		$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$
VQSI	53%						
DVQSI with $m = 2$		53%	53%	60%	60%	53%	60%
DVQSI with $m = 4$		60%	60%	60%	67%	67%	60%
DVQSI with $m = 8$		67%	73%	73%	67%	67%	60%
DVQSI with $m = 16$		60%	67%	67%	67%	67%	60%
DVQSI with $m = 32$		60%	60%	47%	27%	13%	27%

In Figure 15 and Table 2,  $g(x)$  is chosen as given by (3.5b). The best SI performance is obtained when the number of subspaces is 8 and  $c$  is equal to 0.5. The SI accuracies of DVQSI are better than those of VQSI, if  $c=0, 0.25, 0.5$ , or  $0.75$ , and  $m=4, 8$ , or  $16$ . Also, in some cases, such as  $m=32$  and  $c=0.75$  or  $1$ , the performance of DVQSI is not as good as that of VQSI.

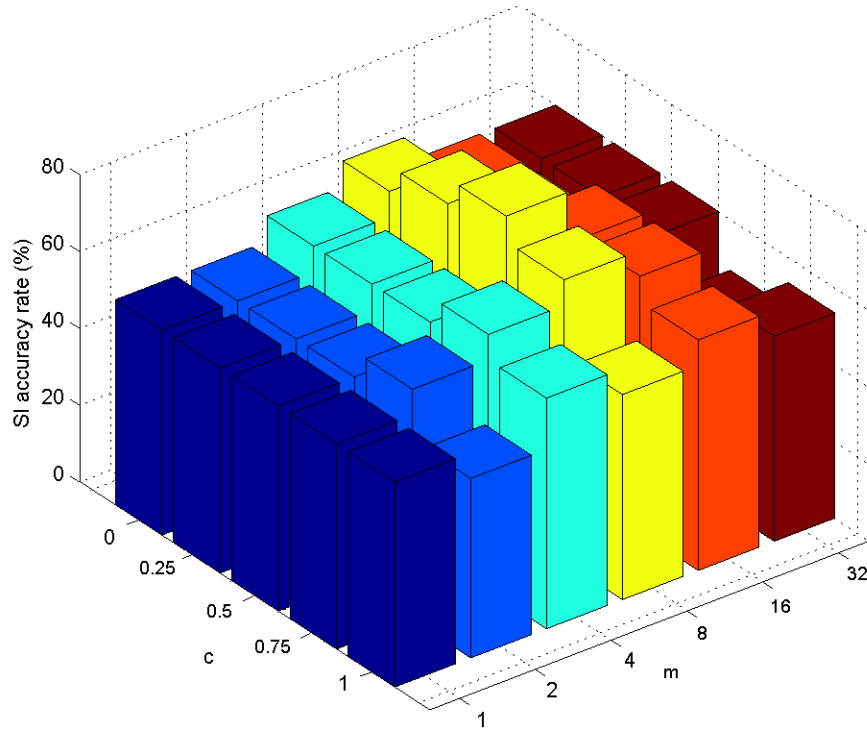


Figure 15. SI accuracy rates versus the number of subspaces  $m$  and the parameter  $c$  of (3.5b) in text-independent experiments

When  $m$  is equal to 1, DVQSI degrades into VQSI.

Table 2. SI accuracy rates of VQSI and DVQSI versus the number of subspaces  $m$  and the parameter  $c$  of (3.5b) in text-independent experiments

		$c=0$	$c=0.25$	$c=0.5$	$c=0.75$	$c=1$
VQSI	53%					
DVQSI with $m=2$		53%	53%	53%	60%	47%
DVQSI with $m=4$		60%	60%	60%	67%	60%
DVQSI with $m=8$		67%	73%	80%	73%	53%
DVQSI with $m=16$		60%	60%	67%	67%	60%
DVQSI with $m=32$		60%	60%	60%	47%	53%

Similarly, in text-dependent experiments, Figure 16 and Figure 17, and Table 3 and Table 4, 8 subspaces with  $h$  equal to 1, 2 or 3, and 8 subspaces with  $c$  equal to 0, 0.25 or 0.5, lead to the highest SI accuracy. When  $m=8$  or 16 and  $h=1$  or 2, or  $m=8$  or 16 and  $c=0, 0.25$  or  $0.5$ , the SI accuracies of DVQSI are better than those of VQSI. However, if the parameters are not properly selected, the performance of DVQSI can be worse than that of VQSI.

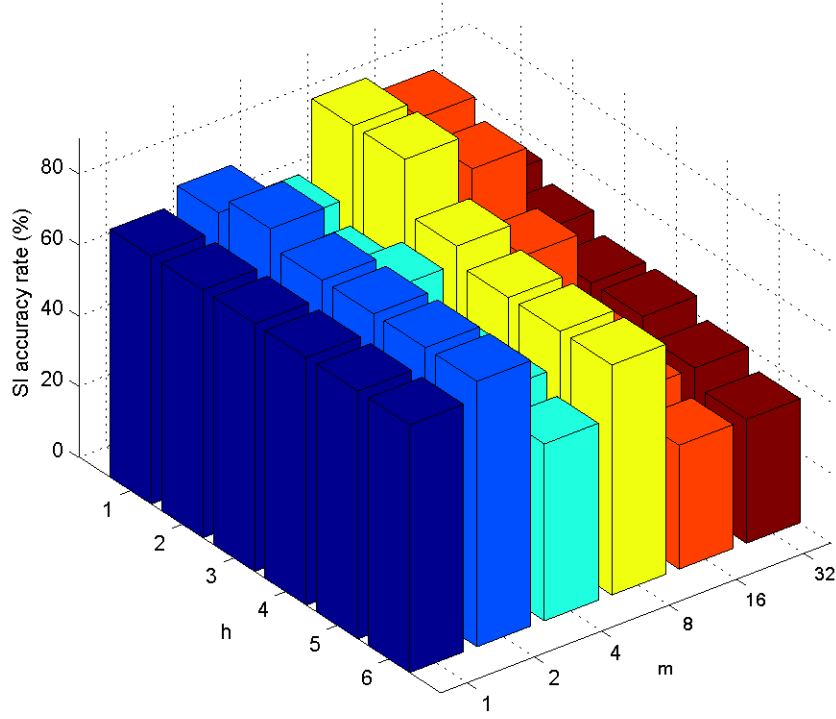


Figure 16. SI accuracy rates versus the number of subspaces  $m$  and the parameter  $h$  of (3.5a) in text-dependent experiments

When  $m$  is equal to 1, DVQSI degrades into VQSI.

Table 3. SI accuracy rates of VQSI and DVQSI versus the number of subspaces  $m$  and the parameter  $h$  of (3.5a) in text-dependent experiments

		$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$
VQSI	70%						
DVQSI with $m = 2$		75%	80%	75%	75%	75%	75%
DVQSI with $m = 4$		65%	60%	65%	55%	55%	50%
DVQSI with $m = 8$		85%	85%	70%	65%	65%	65%
DVQSI with $m = 16$		80%	75%	60%	45%	40%	35%
DVQSI with $m = 32$		55%	50%	45%	45%	40%	35%



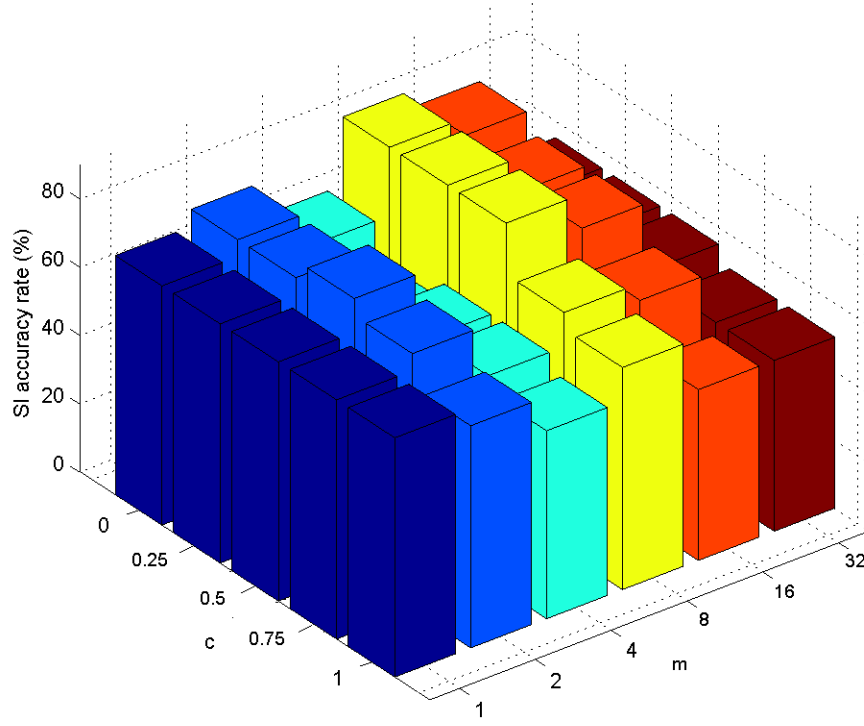


Figure 17. SI accuracy rates versus the number of subspaces  $m$  and the parameter  $c$  of (3.5b) in text-dependent experiments

When  $m$  is equal to 1, DVQSI degrades into VQSI.

Table 4. SI accuracy rates of VQSI and DVQSI versus the number of subspaces  $m$  and the parameter  $c$  of (3.5b) in text-dependent experiments

		$c = 0$	$c = 0.25$	$c = 0.5$	$c = 0.75$	$c = 1$
VQSI	70%					
DVQSI with $m = 2$		75%	75%	80%	75%	65%
DVQSI with $m = 4$		65%	55%	60%	60%	55%

DVQSI with $m = 8$		85%	85%	85%	70%	65%
DVQSI with $m = 16$		80%	75%	75%	65%	50%
DVQSI with $m = 32$		55%	55%	55%	50%	50%

From Figure 14 to Figure 17 and Table 1 to Table 4, it is observed that, for both text-independent and text-dependent experiments, and for either selection of  $q(x)$  in (3.5a) and (3.5b), when the parameters are selected appropriately, in the range of  $m=8$  or 16 and  $h = 1$  or 2 or  $c = 0$ , 0.25 or 0.5, DVQSI achieves better SI accuracy compared with VQSI.

From the experimental results, it is observed that increasing the parameters  $h$  or  $c$  does not always improve SI performance. If the number of segmented subspaces is small, up to certain point, the SI accuracy of DVQSI increases when the number of segmented subspaces increases. After that, the SI accuracy will deteriorate. This is consistent with the discussion of the parameters selection presented in Section 3.2.

### **3.4. Conclusions**

In this chapter, the DVQSI approach is proposed and its parameters selection is discussed. The DVQSI technique takes advantage of the interspeaker variation between speakers inside each speaker pair. The effectiveness of the proposed approach is demonstrated experimentally. It is shown that the proposed DVQSI technique yields better identification accuracies than VQSI approach, when the DVQSI parameters are properly selected. In addition, the tree-structured-like technique is used for codebooks construction and distortion measure computation of DVQSI to improve the computational efficiency. Although the new technique is applied to SI, the proposed

DVQSI approach can be gainfully extended to other pattern identification applications, such as handwritten character identification and face identification.

# **CHAPTER FOUR: AN ENHANCED PERFORMANCE DISCRIMINATIVE VECTOR QUANTIZATION TECHNIQUE FOR SPEAKER IDENTIFICATION**

## **4.1. Introduction**

In this chapter, an enhanced approach, DVQSI with Unique speech feature vector space segmentation for each speaker pair (DVQSI-U), is introduced. In the training mode of DVQSI-U, the speech feature vector space segmentation considers each speaker pair individually based on the interspeaker variation of the speaker pair. Undesired empty subspace codebooks and zero distortions are avoided. In the testing mode of DVQSI-U, an improved approach is presented to calculate the discriminative weighted average distortion pairs. The new approach ignores the subspaces that may lead to wrong SI decisions in the calculation of distortion pairs. The performance of DVQSI-U is analyzed and tested experimentally. The experimental results confirm the SI accuracy improvement employing the proposed DVQSI-U technique in comparison with DVQSI and VQSI.

One of the key factors of the DVQSI approach presented in the last chapter is the speech feature vector space segmentation. In the DVQSI approach, space segmentation is based on VQ and ignores interspeaker variations. The space segmentations for all speaker pairs are exactly the same. Compared with DVQSI, in the DVQSI-U approach presented in this chapter, the linear discriminant functions technique, one of the most popularly used pattern classification techniques, is used in the space segmentation. The space segmentation of DVQSI-U considers

each speaker pair uniquely by exploiting the interspeaker variation of the speaker pair. Moreover, in DVQSI-U, the number of feature vectors for each subspace does not have a large difference for each speaker in each speaker pair. Thus, undesired empty subspace codebooks and zero distortions, which may happen in DVQSI, are avoided.

In the testing mode of DVQSI, all the subspaces are used in the calculation of discriminative weighted average distortion pairs. However, in the testing mode of DVQSI-U, by adding a threshold function, a new algorithm is employed to calculate discriminative weighted average distortion pairs. This algorithm excludes the subspaces that may lead to wrong SI decisions from being counted into the calculation of discriminative weighted average distortion pairs.

This chapter is organized as follows: The proposed DVQSI-U approach is developed in Section 4.2. Experimental results to evaluate the DVQSI-U technique are given in Section 4.3. Section 4.4 contains the conclusions.

#### **4.2. The DVQSI Approach with Unique Feature Vector Space Segmentation for Each Speaker Pair (DVQSI-U)**

In the DVQSI-U approach, a new speech feature vector space segmentation technique and a novel algorithm for the discriminative weighted average distortion pairs calculation are introduced. The discriminative weight calculations for DVQSI and DVQSI-U are the same, except that, in DVQSI-U, empty subspace codebooks and zero distortions need not be considered. This will be explained in detail later.

#### 4.2.1. Speech Feature Vector Space Segmentation Based on Linear Discriminant Functions

In this subsection, Space Segmentation based on Linear Discriminant Functions (SSLDF) for DVQSI-U is presented. The advantages and disadvantages of SSLDF in comparison with SSVQ for DVQSI are analyzed.

Before the presentation of the SSLDF technique, linear discriminant function techniques for the linearly nonseparable pattern classification problem are introduced here [8].

The pattern classification problem in this chapter is to find a suitable linear discriminant function, with which to classify two linearly nonseparable categories  $\omega_1$  and  $\omega_2$  based on the Mean Square Error (MSE) criterion.

A linear discriminant function that is a linear combination of the components of  $\mathbf{x}$  ( $\mathbf{x} \in R^d$ ,  $\mathbf{x}$  is from categories  $\omega_1$  or  $\omega_2$ ) can be written as

$$g(\mathbf{x}) = \mathbf{a}'\mathbf{y} \quad (4.1)$$

where prime means transpose,  $\mathbf{y} = [1, \mathbf{x}']'$ , and  $\mathbf{a} \in R^{d+1}$  is the weight vector to be calculated.

The equation  $g(\mathbf{x}) = 0$  defines a decision surface that divides the  $d$ -dimension vector space into two subspaces. Thus, the two-category linear classifier implements the following decision rule:  $\mathbf{x}$  is from category  $\omega_1$  if  $g(\mathbf{x}) > 0$  and from category  $\omega_2$  if  $g(\mathbf{x}) < 0$ . If  $g(\mathbf{x}) = 0$ ,  $\mathbf{x}$  can ordinarily be assigned to either class [8].

Then, the pattern classification problem is converted into finding a weight vector  $\mathbf{a}$  that minimizes the MSE criterion function

$$J_s(\mathbf{a}) = \|\mathbf{Y}\mathbf{a} - \mathbf{b}\|^2 = \sum_i (\mathbf{a}'\mathbf{y}_i - b_i)^2$$

where  $\mathbf{b} = [b_1, b_2, \dots, b_n]'$  is a column vector, and  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$  [8].

If the matrix  $\mathbf{Y}'\mathbf{Y}$  is nonsingular [8], the solution is given by

$$\alpha = (Y'Y)^{-1}Y'b \quad (4.2)$$

Typically,  $b_i=1$  is selected for the vectors from one category, and  $b_i=-1$  is assigned for the vectors from the other category. It has been shown that, in this case, the MSE solution approximates the Bayes discriminant function as the number of training vectors tends to infinity [8].

The SSLDF approach is based on linear discriminant function techniques. SSLDF considers each speaker pair and the speaker pair's interspeaker variation uniquely. In SSLDF, for each speaker pair, based on the interspeaker variation for the speaker pair, the feature vector space is divided into a desired number of subspaces, which is denoted by  $m$ . The speech feature vector space is firstly segmented into two subspaces. Then, the process is repeated to segment each subspace into two parts until the desired number of the subspaces is obtained. The space segmentation procedure for  $m=4$  is given in Figure 18 as an example. The process, which segments the space or the subspace into two parts, can be divided into two stages. In the first stage, the space segmentation problem is converted into a pattern classification problem by defining two pattern classification training categories. Then, in the second stage, a decision surface is created by linear discriminant function techniques to divide the feature space or subspace into two subspaces [8].

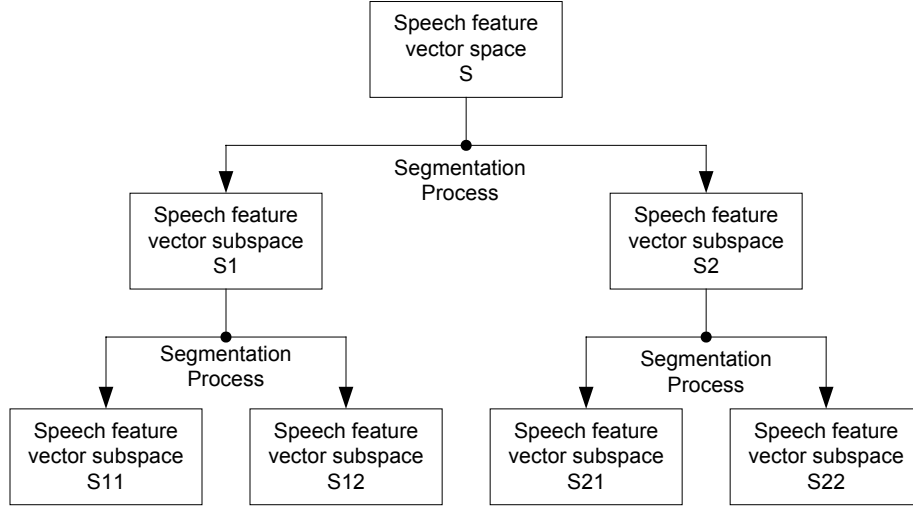


Figure 18. The speech feature vector space segmentation procedure for the number of subspaces

$$m=4$$

In the SSLPF technique, each speaker pair  $k1 \in \Lambda$  and  $k2 \in \Lambda$  is considered, where  $\Lambda$  is the closed set of speakers for the SI group,  $\Lambda = \{\text{speaker 1, speaker 2, ..., speaker } N\}$ , and  $N$  is the number of speakers in the SI group. For each training feature vector  $\mathbf{v}_{1i} \in T(\mathbf{k1})$  of speaker  $k1$ , its nearest training feature vector  $\mathbf{v}_{2i} \in T(\mathbf{k2})$  of speaker  $k2$  is found, where  $T(\mathbf{k1})$  and  $T(\mathbf{k2})$  are the training speech feature vector sets of speaker  $k1$  and  $k2$  respectively. The distance between  $\mathbf{v}_{1i}$  and  $\mathbf{v}_{2i}$  is calculated and denoted by  $d(\mathbf{v}_{1i})$ . Typically,  $\mathbf{v}_{1i}$  located in the part of the feature space with higher interspeaker variation has larger  $d(\mathbf{v}_{1i})$ , and vice versa. After that, the training feature vector set  $T(\mathbf{k1})$  of speaker  $k1$  is divided into two subsets, namely,  $V_{11}$  and  $V_{12}$ , where  $V_{11}$  contains the half set of  $T(\mathbf{k1})$  with smaller  $d(\mathbf{v}_{1i})$ , while  $V_{12}$  includes the remaining half set of  $T(\mathbf{k1})$  with larger  $d(\mathbf{v}_{1i})$ . The numbers of training feature vectors in  $V_{11}$  and  $V_{12}$  are the same. Similarly, the training feature vector set  $T(\mathbf{k2})$  of speaker  $k2$  is divided into two subsets,  $V_{21}$  and  $V_{22}$ . Let  $\mathbf{Q}_1 = (V_{11}, V_{21})$  and  $\mathbf{Q}_2 = (V_{12}, V_{22})$ . The numbers of feature vectors in  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are the



same. Since  $\mathbf{v}_{1i}$  with lower  $d(\mathbf{v}_{1i})$  is typically located in the part of the feature space with lower interspeaker variation, most feature vectors in  $\mathbf{Q}_1$  are located in this part; while feature vectors in  $\mathbf{Q}_2$  are mainly located in the part of the feature space with higher interspeaker variation. The space segmentation problem is converted into a pattern classification problem by letting  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  be the two training categories of the linearly nonseparable pattern classification problem.

After the space segmentation problem has been converted into a pattern classification problem, a discriminant function  $g(\mathbf{x})$  in (4.1) is constructed with its weight vector  $\boldsymbol{\alpha}$  given by (4.2), where  $b_i=1$  for vectors from  $\mathbf{Q}_1$  and  $b_i=-1$  for vectors from  $\mathbf{Q}_2$ . The corresponding decision surface  $g(\mathbf{x})=0$  divides the feature space  $S$  into two subspaces  $S_1$  and  $S_2$ . The subspace for  $\mathbf{Q}_1$  has lower interspeaker variation than the subspace for  $\mathbf{Q}_2$ , since feature vectors in  $\mathbf{Q}_1$  are typically located in part of the feature space with lower interspeaker variation than feature vectors in  $\mathbf{Q}_2$ . The subspace segmentation is based on the interspeaker variation of the speaker pair. Similar procedures are repeated to divide  $S_1$  and  $S_2$ , and their subspaces, until the desired number of subspaces for DVQSI-U is met.

The SSVQ technique in DVQSI is computationally efficient, at the expense of ignoring the interspeaker variations for speaker pairs. Because the space segmentation of SSVQ is not based on interspeaker variations, it is possible that for some speaker pairs, interspeaker variations and then discriminative weights are similar for all subspaces, so that DVQSI does not have an advantage over VQSI for those speaker pairs. Moreover, the space segmentation result of SSVQ depends on the initial values used for VQ codebook training. Different initial values lead to different space segmentations, and consequently different SI results.

In DVQSI, for each speaker in each speaker pair, the numbers of the training feature vectors for different subspaces may have large differences. This may result in the numbers of the

training feature vectors being too small for some subspace codebooks construction, so that empty subspace codebooks and zero distortions are obtained for these subspaces.

In VQ and discriminative VQ based SI, VQ codebooks are used to describe the templates of the speakers in the SI group. The larger the codebook size, the more valuable are the speaker templates that the codebooks represent. These valuable speaker templates result in the high accuracy of SI. In the proposed DVQSI and DVQSI-U approaches, the size of the codebook for each subspace and each speaker is the same. If the codebook size in each subspace is  $n_s$ , and the number of empty subspace codebooks for a speaker is  $m_s$ , thus only  $(m-m_s)n_s$  codewords instead of  $mn_s$  codewords are actually used to represent the speaker's template. This is undesired and harmful to SI. Also, the empty codebook leads to zero distortions. The subspace with a zero distortion has a zero discriminative weight. Thus, the testing feature vectors in this subspace will not take effect in the testing mode and are wasted.

In addition, since the size of the codebook is equal for each subspace of each speaker, it is expected that the numbers of the training feature vectors are also nearly equal for different subspaces of each speaker.

In the SSLDF approach of DVQSI-U, space segmentations for different speaker pairs are different. The number of all possible speaker pairs is equal to the number of different combinations of size 2 from a set of size  $N$ ,  $C_2^N = \binom{N}{2} = \frac{N(N-1)}{2}$ , where  $N$  is the number of speakers in the SI group.  $C_2^N$  feature space segmentations are needed in SSLDF, while for SSVQ, only one feature space segmentation is required. The computational burden of SSLDF is much larger than SSVQ when  $N$  is a large value. At the same time, SSLDF has its advantages over SSVQ. In SSLDF, the space segmentation is determined. SSLDF segments the feature space

based on the differences of the interspeaker variation in different parts of the feature space for the two speakers in the speaker pair. In DVQSI-U, for each speaker pair, different subspaces have the apparently different interspeaker variation, and so different discriminative weights.

SSLDF intends to make the number of feature vectors in each subspace approximately the same for each speaker by letting the two training categories in the space segmentation have the same numbers of training vectors from each speaker. If the categories are perfectly classified, for each speaker, the numbers of training feature vectors are the same in all subspaces.

In the SSLDF approach, for each speaker in the speaker pair, the number of feature vectors in each subspace has no large difference. Therefore, for DVQSI-U, enough training vectors can be obtained for the codebooks construction for each subspace. None of the subspace codebooks need be set to empty. Consequently, none of the corresponding distortions for the discriminative weights assignment need be set to zero. Thus, in DVQSI-U, empty subspace codebooks and zero distortions need not be considered.

#### 4.2.2 A Novel Algorithm for Discriminative Weighted Average Distortions Calculation

In the testing mode of DVQSI-U, a new definition of the discriminative weighted average distortion pair is presented. For testing waveform  $R$  and the speaker pair  $k1$  and  $k2$ , the discriminative weighted average distortion pair  $\bar{d}(R, k1)$  and  $\bar{d}(R, k2)$  is given by

$$\bar{d}(R, k1) = \frac{\sum_{j=1}^m d'_j(R, k1) w_j(k1, k2) n_j(R) L(j)}{\sum_{j=1}^m w_j(k1, k2) n_j(R) L(j)} \quad (4.3.a)$$

and

$$\bar{d}(R, k2) = \frac{\sum_{j=1}^m d_j^t(R, k2) w_j(k2, k1) n_j(R) L(j)}{\sum_{j=1}^m w_j(k2, k1) n_j(R) L(j)} \quad (4.3.b)$$

where

$$L(j) = \begin{cases} 0 & d_j^t(R, k1) > T2, d_j^t(R, k2) > T2 \\ 1 & \text{otherwise} \end{cases} \quad (4.4)$$

and  $T2$  is a positive threshold.

The definition of the discriminative weighted average distortion pair for DVQSI-U is a generalization of the definition in DVQSI by adding a threshold function  $L(j)$ . When the threshold  $T2$  tends to infinity,  $L(j)$  always equals 1. The definitions of the weighted average distortion pair, given by (4.3.a) and (4.3.b), are the same as the definitions for DVQSI.

After the discriminative weighted average distortion pairs used for SI are obtained, the SI decision process of DVQSI-U is the same as that of DVQSI.

For speaker pair  $k1$  and  $k2$ , and  $R \in k1$ ,  $d_j^t(R, k1) - d_j^t(R, k2)$  in subspace  $j$  has a positive effect on SI, if it is a negative value. However, if  $d_j^t(R, k1) - d_j^t(R, k2)$  is positive, larger discriminative weight  $w_j(k1, k2)$  of subspace  $j$  is more likely to lead to  $\bar{d}(R, k1) > \bar{d}(R, k2)$  than the smaller one, when subspace  $j$  is counted in the calculation of  $\bar{d}(R, k1)$  and  $\bar{d}(R, k2)$  in (4.3).  $\bar{d}(R, k1) > \bar{d}(R, k2)$  results in speaker  $k1$  being eliminated from the candidate speaker set. Since  $R \in k1$ , speaker  $k1$  being eliminated from the candidate speaker set leads to a wrong SI decision. In order to make the correct SI decision, subspace  $j$  with positive  $d_j^t(R, k1) - d_j^t(R, k2)$  and larger  $w_j(k1, k2)$  should not be counted in the calculation of the discriminative weighted average distortion pair.

It has been observed that the subspace  $j$  with larger discriminative weight  $w_j(k1, k2)$  always has larger  $d_j'(R, k2)$ , and vice versa. Only when both  $d_j'(R, k1)$  and  $d_j'(R, k2)$  are large, subspace  $j$  may have larger  $w_j(k1, k2)$  and positive  $d_j'(R, k1) - d_j'(R, k2)$ . When  $T2$  is selected to make both  $d_j'(R, k1)$  and  $d_j'(R, k2)$  larger than  $T2$ ,  $L(j)$  equals zero. The subspace with large  $w_j(k1, k2)$  and undesired positive  $d_j'(R, k1) - d_j'(R, k2)$  is excluded in the discriminative weighted average distortion pair calculation in (4.3).

The selection of threshold  $T2$  considerably influences SI results. In order to avoid subspace  $j$  with large  $w_j(k1, k2)$  and undesired positive  $d_j'(R, k1) - d_j'(R, k2)$  to be counted in (4.3), small  $T2$  is preferred. However, the subspaces, which have large  $w_j(k1, k2)$ ,  $d_j'(R, k1) > T2$  and negative  $d_j'(R, k1) - d_j'(R, k2)$ , are also neglected by adding  $L(j)$ . These subspaces can make positive contributions to the SI decision if they are counted. In this case,  $T2$  should be a large value. The optimal  $T2$  for SI is the trade off between these two cases.

Meanwhile, according to the definition of  $L(j)$ , subspaces with small  $w_j(k1, k2)$ , which have small  $d_j'(R, k2) < T2$ , are always counted in the discriminative weighted average distortion pairs calculation, since the corresponding threshold function  $L(j)$  always equals 1 for these subspaces.

It is worthwhile to mention that although the calculation of the training mode of DVQSI-U increases almost proportionally to the square of the number of speakers in the SI group, the calculation of the testing mode is nearly proportional to the number of speakers in the SI group.

### **4.3. Experimental Results**

In this section, an experiment is given to evaluate the effectiveness of the proposed DVQSI-U approach. Speech records are obtained from the CSLU (Center for Spoken Language Understanding, Oregon Health & Science University) Speaker Recognition V1.1 corpus. For each speaker, the speech records collected on different collection dates are packaged into different recording sessions. There are mismatches between the speech utterances taken from different speakers. Also, there are mismatches due to different recording sessions of the same speaker. All the speech files in the corpus were sampled at 8 khz and 8-bits per sample.

Thirty-five speakers are used in the text-independent SI experiments. Four spontaneous speeches for each speaker are used in the training mode. Two other spontaneous speeches, taken about one year after the training speech waveform for each speaker, are used in the testing mode. Each speech waveform lasts about 4 seconds.

Silenced and unvoiced segments are discarded based on an energy threshold. The analysis Hamming window size is 32ms, 256samples, with 24ms overlapping [49]. The feature vector used in the experiment is composed of 15 Mel Frequency Cepstral Coefficients (MFCC's) [60].

The codebook size of the existing VQ technique for SI (VQSI) used for comparison is 64. The codebook size in each subspace for DVQSI and DVQSI-U is  $64/m$ , where  $m$  is the number of the subspaces. The threshold  $T_1$  in (3.5a) is equal to  $5m/(m+4)$ . In this work, the speech feature vector space is divided into 4 subspaces, i.e.,  $m=4$ . All the codebooks are constructed by the Generalized Lloyd algorithm. The initial values of codebooks are obtained by using splitting algorithm [18, 32].

Table 5 shows SI accuracy results employing VQSI, DVQSI and DVQSI-U. It is seen that, the two latter techniques lead to better SI accuracy than the former if the parameters of DVQSI and DVQSI-U are suitably selected. The discussions and simulation results of the parameters selection of DVQSI are given in the last chapter. The parameters selection of DVQSI-U is the same as that of DVQSI, and the selection of  $T_2$  of (4.4) is discussed in Section 4.2.

Table 5. The SI accuracy rates employing VQSI, DVQSI and DVQSI-U

Technique	VQSI	DVQSI with $h=1$	DVQSI-U with $h=1$ and $T_2=300$
SI accuracy	62.9%	68.6%	71.4%

The experimental results employing the DVQSI-U and DVQSI techniques are given in Table 6 and

Table 7. From Table 6, for DVQSI, SI accuracy rates do not change when (4.3) instead of (3.6) is used in the calculation of the weighted average distortion pairs. While using DVQSI-U, the threshold  $T_2 = 300$  leads to better results than  $T_2 = \infty$ . The SI results employing DVQSI-U are better than those achieved using DVQSI, when the threshold  $T_2 = 300$ .

Table 6. The SI accuracy rates employing DVQSI and DVQSI-U, with  $h=1$  in (3.5a)

	DVQSI	DVQSI with Eq. (4.3) and $T_2=300$	DVQSI-U with $T_2=300$	DVQSI-U with $T_2=\infty$
SI accuracy	68.6%	68.6%	71.4%	51.4%

Table 7. The SI accuracy rates of DVQSI-U versus  $T_2$  in (4.4), with  $h=1$  in (3.5a)

$T_2$	200	225	250	275	300	325	350	400	$\infty$
SI accuracy	40.0%	51.4%	68.6%	71.4%	71.4%	71.4%	60.0%	54.3%	51.4%

In Table 7, the SI accuracy rates are given for DVQSI-U versus  $T_2$  in (4.4). The best SI accuracy rate is achieved when  $T_2$  is 275, 300 or 325. Smaller and larger  $T_2$  lead to degraded SI results. The simulation results match the discussion of the selection of  $T_2$  in Section 4.2.

The average distortions  $d1_j^{kl}(k1, k2)$ ,  $d2_j^{kl}(k1, k2)$ ,  $d_j^l(R, k1)$  and  $d_j^l(R, k2)$  versus  $j$  for DVQSI-U are shown in Figure 19 for the speaker pair  $k1=1$  and  $k2=2$  with  $R \in k1$ . This is also represented in Figure 20 for the speaker pair  $k1=15$  and  $k2=27$  with  $R \in k1$ . For simplification, the subspaces are ranked from the lowest discriminative weight to the highest discriminative weight for all the figures in this section. From Figure 19 and Figure 20, it is seen that larger subspace index  $j$ , which has larger discriminative weight  $w_j(k1, k2)$ , leads to larger  $d_j^l(R, k2)$ . Typically,  $d_j^l(R, k1) - d_j^l(R, k2)$  and  $d_j^l(R, k1) - T_2$  for subspace  $j$  are negative values. Then,  $L(j)$  in (4.4) equals 1, subspace  $j$  is counted in the calculation of the discriminative weighted average distortion pair and makes the positive contribution in SI. However, for some cases, such as subspace 4 in Figure 20,  $d_j^l(R, k1) - d_j^l(R, k2)$  for subspace  $j$  is a positive value. In these cases, when subspace  $j$  has a large discriminative weight, both  $d_j^l(R, k1)$  and  $d_j^l(R, k2)$  are larger than  $T_2$ . Thus,  $L(j)$  is set to zero, subspace  $j$  is excluded from the weighted average distortion pair calculation to avoid making wrong SI decisions. The adding of threshold function  $L(j)$  to the calculation of weighted average distortions has the advantage of increasing the SI accuracy.



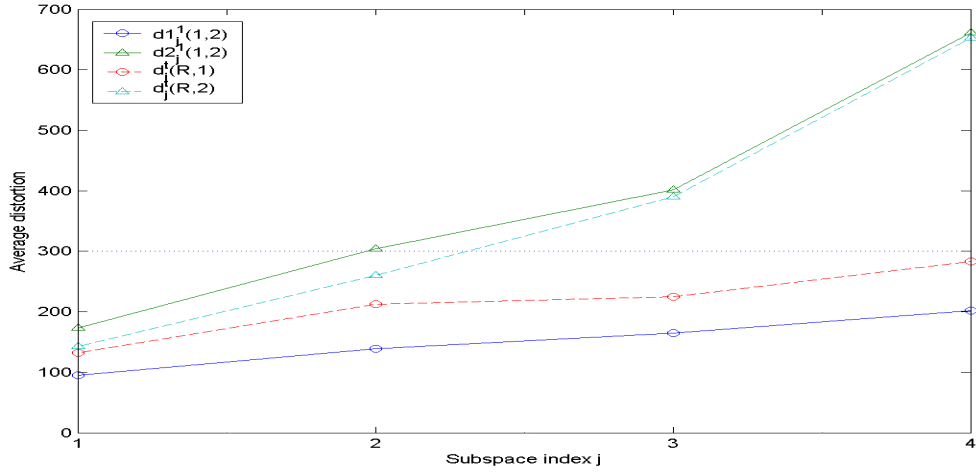


Figure 19. Average distortion  $d1_j^1(1, 2)$ ,  $d2_j^1(1, 2)$ ,  $d_j^t(R, 1)$  and  $d_j^t(R, 2)$  versus subspace index  $j$  for DVQSI-U with  $R \in 1$

The dotted horizontal line in the figure corresponds to the threshold  $T2=300$ .

—○—:  $d1_j^1(1, 2)$ , —△—:  $d2_j^1(1, 2)$ , --○--:  $d_j^t(R, 1)$ , --△--:  $d_j^t(R, 2)$

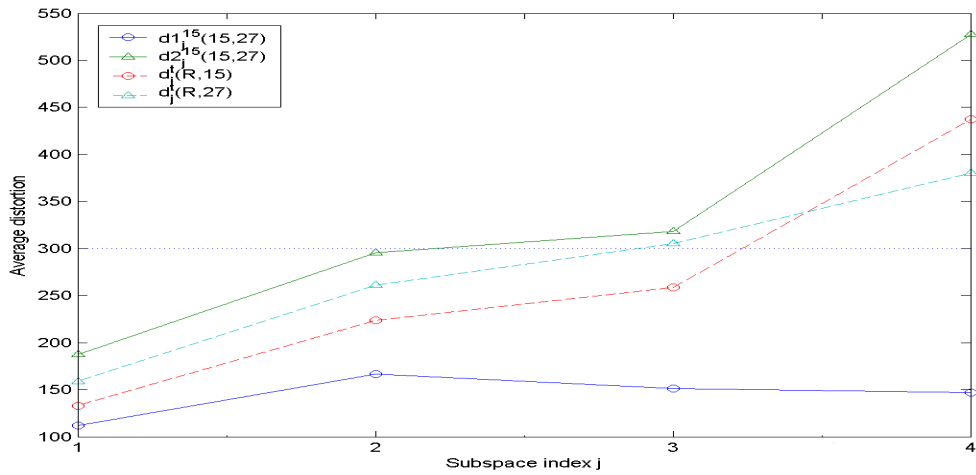


Figure 20. Average distortion  $d1_j^{15}(15, 27)$ ,  $d2_j^{15}(15, 27)$ ,  $d_j^t(R, 15)$  and  $d_j^t(R, 27)$  versus subspace index  $j$  for DVQSI-U with  $R \in 15$

The dotted horizontal line in the figure corresponds to the threshold  $T2=300$ .

$$\text{---}\circ\text{---}: d1_j^{15}(15, 27), \text{---}\Delta\text{---}: d2_j^{15}(15, 27), \text{--}\circ\text{--}: d_j'(R, 15), \text{--}\Delta\text{--}: d_j'(R, 27)$$

The numbers of training feature vectors for speaker  $k1=11$  and  $k2=19$  in the subspaces of their speaker pair are given in Figure 21. For DVQSI-U, the numbers of feature vectors for different subspaces of the same speaker have no large differences. In contrast, as shown in Figure 21, for DVQSI, the numbers of feature vectors for the same speaker in different subspaces have large differences. This may lead to undesired empty subspace codebooks and zero distortions.

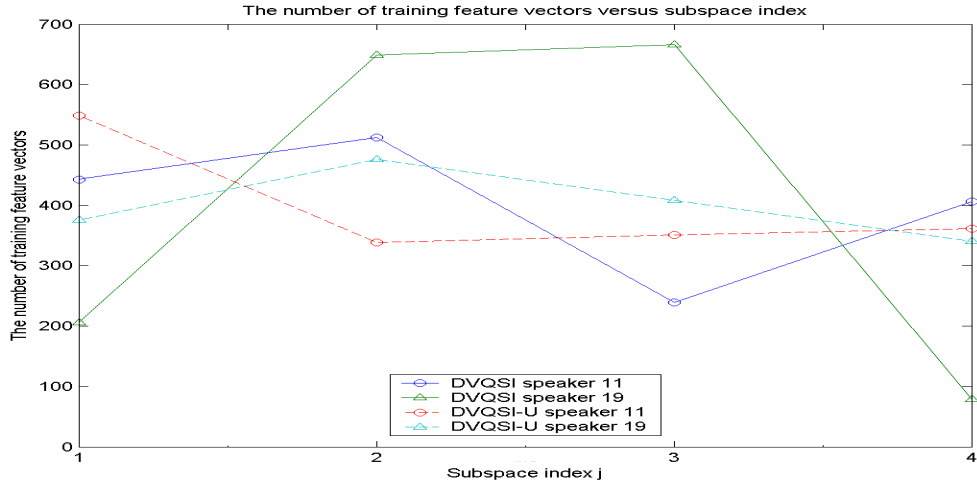


Figure 21. The numbers of training feature vectors of speaker 11 and 19 in the subspaces of their speaker pair

— $\circ$ —: speaker 11 with DVQSI, — $\Delta$ —: speaker 19 with DVQSI, -- $\circ$ --: speaker 11 with DVQSI-U, -- $\Delta$ --: speaker 19 with DVQSI-U

The discriminative weights  $w_j(11, 19)$  for the speaker pair  $k1=11$  and  $k2=19$  with  $h=1$  are illustrated in Figure 22. Discriminative weights for different subspaces have clear differences for DVQSI-U. While employing DVQSI, for some subspaces (for example: subspace 1 and 2), the discriminative weights are approximately equal. This is because the segmentation of SSLDF in DVQSI-U is based on the interspeaker variation of the speaker pair, but SSVQ in DVQSI does not consider the interspeaker variations.

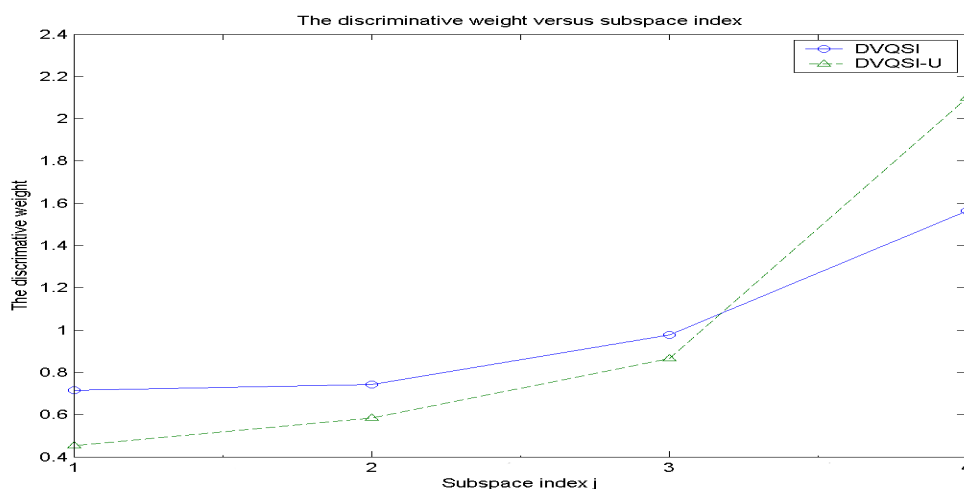


Figure 22. The discriminative weight  $w_j(3,19)$  for speaker pair 11 and 19, where  $h$  in (3.5a) equals 1

The subspaces are ranked from the lowest discriminative weight to the highest discriminative weight.

—○—: DVQSI, --Δ--: DVQSI-U.

#### **4.4. Conclusions**

An enhanced DVQSI technique, DVQSI-U, is presented in this chapter. In the proposed DVQSI-U technique, SSLDF instead of SSVQ in DVQSI is used for the speech feature vector space segmentation. The SSLDF technique considers each speaker pair individually. It divides the feature space of each speaker pair into subspaces based on the interspeaker variation of this speaker pair. Unlike DVQSI, in DVQSI-U, the SSLDF technique guarantees that different subspaces have different discriminative weights for each speaker pair in the SI group. DVQSI-U also avoids empty subspace codebooks and zero distortions which occur in DVQSI. In the testing mode of DVQSI-U, a novel definition of weighted average distortion pairs is presented after a threshold function is introduced. The new definition excludes some subspaces that have large discriminative weights from the calculation of the discriminative weighted average distortion pairs. These subspaces may have negative contributions for SI, if they are included.

From the analysis, supported by the experimental results, DVQSI-U has better performance in the SI accuracy than DVQSI. The performance improvement of DVQSI-U is achieved at the expense of the increased computational burden in the training mode.

## **CHAPTER FIVE: SPEAKER IDENTIFICATION BASED ON ADAPTIVE DISCRIMINATIVE VECTOR QUANTIZATION**

### **5.1. Introduction**

In this chapter, a novel Adaptive Discriminative Vector Quantization technique for Speaker Identification (ADVQSI) is introduced. The proposed ADVQSI technique exploits the interspeaker variation between each speaker and all speakers in the SI group in order to enlarge the speakers' template differences. For each speaker, its speech feature vector space is divided into subspaces. Different discriminative weights are given to different subspaces. Subspaces with larger discriminative weights play more important roles in the SI decision. [4, 58].

The ADVQSI technique has two modes, namely, the training mode and the testing mode. In the training mode, a VQ codebook is constructed for each speaker in the SI group, and a general VQ codebook is constructed for the entire group of speakers. Then, for each speaker, the speech feature vector space is segmented into a number of subspaces based on interspeaker variation between this speaker and all speakers in the SI group. Next, a discriminative weight is determined for each subspace of each speaker by employing adaptive techniques. The adaptively trained discriminative weights are used to represent the optimal roles of subspaces for SI. The VQ codebook for each speaker, together with the feature space segmentation and discriminative weights for each speaker, represent the template of that speaker. In the testing mode, for each input waveform, discriminative weighted average VQ distortions are calculated as matching scores between speakers' templates and the testing waveform. The testing waveform is identified to the speaker that leads to the highest matching score.

DVQSI/DVQSI-U approaches reported in previous chapters also consider the interspeaker variation. Though both DVQSI/DVQSI-U and ADVQSI employ the interspeaker variation, their techniques for the speech feature vector space segmentation, the discriminative weights determination, and the SI decision in the testing mode are different. The DVQSI/DVQSI-U approach is based on each speaker pair in the SI group and discriminative weights are obtained by trial and error, whereas the ADVQSI technique is based on each speaker in the SI group and discriminative weights are calculated by using adaptive techniques. The computational burden of ADVQSI is proportional to the number of speakers in the SI group, while the computational burden of previously reported DVQSI/DVQSI-U increases with the square of the number of speakers in the SI group.

The rest of this chapter is organized as follows: in Section 5.2, the proposed ADVQSI approach is developed. Experimental results to evaluate the ADVQSI technique are given in Section 5.3, while Section 5.4 contains the conclusions.

## **5.2. Adaptive Discriminative Vector Quantization for Speaker Identification (ADVQSI)**

In the training mode of ADVQSI, the training speech waveforms for each speaker in the SI group are available. First, each speaker's training speech feature vector set is created from this speaker's training waveforms by feature extraction techniques. After feature extraction, a VQ codebook for each speaker and a VQ codebook for all speakers are constructed. Then, for each speaker, its feature vector space is segmented into a number of subspaces based on the interspeaker variation between this speaker and all speakers in the SI group. Finally, a discriminative weight for each subspace of each speaker is calculated by employing adaptive

techniques. In the ADVQSI testing mode, speech waveforms of the unknown speakers are presented to identify speakers. A testing feature vector set is created for each testing waveform in this mode. Discriminative weighted average VQ distortions instead of equally weighted average VQ distortions are used as similarity scores between the speakers' templates and the testing waveform for SI decisions.

### 5.2. 1. The Training Mode

In the training mode, training speech waveforms for each speaker in the SI group are available. Through feature extraction, the training speech feature vector set  $\mathbf{T}(k)$  is extracted from the training waveforms of each speaker  $k \in \Lambda$ , where  $\Lambda = \{\text{speaker 1, speaker 2, ..., speaker } N\}$  is the closed set of speakers in the SI group. The training speech feature vector set  $\mathbf{T}(k)$  for each speaker  $k$  shares the same speech feature vector space but has a different probability distribution.

A VQ codebook  $\mathbf{C}(k)$  for speaker  $k$  is constructed by employing  $\mathbf{T}(k)$  of speaker  $k$  for the codebook training [18, 32]. Meanwhile, a general codebook  $\mathbf{C}^g$  is constructed for all the speakers in the SI group by using  $\mathbf{T}^g$  as the training set for the codebook construction [18, 32], where  $\mathbf{T}^g = \{\mathbf{T}(1), \mathbf{T}(2), ..., \mathbf{T}(h)\}$  is the set of all training speech feature vectors for all speakers.

After the codebooks construction, for each speaker, the speech feature vector space is segmented into a number of subspaces based on the interspeaker variation between this speaker and all speakers in the SI group. In the segmentation, the speech feature vector space is firstly segmented into two subspaces. Then, the process is repeated to segment each subspace into two parts until the desired number of the subspaces is obtained. The desired number of subspaces for

the space segmentation is denoted by  $m$ . The process, which segments the space or the subspace into two parts, can be divided into two stages. In the first stage, the space segmentation problem is converted into a pattern classification problem by defining two pattern classification training categories. Then, in the second stage, a decision surface is created by linear discriminant function techniques to divide the feature space or subspace into two subspaces [8].

In the first stage of the space segmentation process, for each speaker  $k$  and each training feature vector  $\mathbf{v} \in \mathbf{T}(\mathbf{k})$  of speaker  $k$ , the distortion  $d(\mathbf{v}, k)$  of  $\mathbf{v}$  quantized by codebook  $\mathbf{C}(\mathbf{k})$  of speaker  $k$  and the distortion  $d(\mathbf{v}, g)$  of  $\mathbf{v}$  quantized by general codebook  $\mathbf{C}^g$  are calculated. Let  $d(\mathbf{v}) = d(\mathbf{v}, k) / d(\mathbf{v}, g)$ . Typically, when  $d(\mathbf{v})$  is lower,  $\mathbf{v}$  is located in the region of the feature space with a higher interspeaker variation between speaker  $k$  and all speakers, and vice versa. Then, the training feature vector set  $\mathbf{T}(\mathbf{k})$  of speaker  $k$  is divided into two subsets, namely  $\mathbf{T}_1$  and  $\mathbf{T}_2$ .  $\mathbf{T}_1$  contains the feature vector with larger  $d(\mathbf{v})$  while  $\mathbf{T}_2$  contains the remaining feature vectors. The numbers of feature vectors in  $\mathbf{T}_1$  and  $\mathbf{T}_2$  are the same. Since  $\mathbf{v}$  with larger  $d(\mathbf{v})$  is typically located in the region of the feature space with a lower interspeaker variation, most feature vectors in  $\mathbf{T}_1$  are located in the regions of the feature space with lower interspeaker variations. In contrast, feature vectors in  $\mathbf{T}_2$  are mainly located in the regions of the feature space with higher interspeaker variations. The space segmentation problem is converted into a pattern classification problem by letting  $\mathbf{T}_1$  and  $\mathbf{T}_2$  be the two training categories of the linear pattern classification problem.

In the second stage of the space segmentation, a linear discriminant function  $g(\mathbf{x})$  in (4.1) is constructed with its weight vector  $\mathbf{a}$  given by (4.2), where  $b_i = 1$  for vectors from  $\mathbf{T}_1$  and  $b_i = -1$  for vectors from  $\mathbf{T}_2$ . The corresponding decision surface  $g(\mathbf{x}) = 0$  divides the speech feature vector space  $S$  into two subspaces  $S_1$  and  $S_2$ . The subspace for  $\mathbf{T}_1$  has a lower interspeaker variation



between speaker  $k_1$  and all speakers than the subspace for  $T_2$ , since feature vectors in  $T_1$  are typically located in regions of the feature space with lower interspeaker variations than feature vectors in  $T_2$ . The feature space segmentation of ADVQSI is based on the interspeaker variation between each speaker and all speakers. Similar procedures are repeated to divide  $S_1$  and  $S_2$ , and their subspaces, until the desired number of subspaces for ADVQSI is met. The feature space segmentation for each speaker is decided by the linear discriminant functions for this speaker.

In ADVQSI, each speaker's template is represented by this particular speaker's codebook, discriminative weights for subspaces, and feature space segmentation. In order to obtain optimal discriminative weights for all speakers by adaptive techniques, an initial positive discriminative weight is assigned to each subspace of each speaker. Then the differences for templates of various speakers are measured based on initial discriminative weights.

The average VQ distortion  $d_{kj}(k_1, k_2)$  of  $T_j(k_1, k_2)$  quantized by  $C(k_2)$  is calculated for each speaker  $k_1$  and each subspace  $j$  of speaker  $k_2$ , where  $T_j(k_1, k_2)$  is the set for all speech feature vectors of  $T(k_1)$  located in subspace  $j$  of speaker  $k_2$ , speaker  $k_1 \in \Lambda$  and speaker  $k_2 \in \Lambda$ , and  $j=1, 2, \dots, m$  is the subspace index. Similarly, the average VQ distortion of  $T_j(k_1, k_2)$  quantized by  $C^g$  is obtained and denoted by  $d_{gj}(k_1, k_2)$ . Let  $d_j(k_1, k_2) = d_{gj}(k_1, k_2) - d_{kj}(k_1, k_2)$ .

The weighted average distortion  $d_{dis}(k_1, k_2)$  is defined as

$$d_{dis}(k_1, k_2) = \frac{W(k_2)' N(k_1, k_2) D(k_1, k_2)}{W(k_2)' n(k_1, k_2)} \quad (5.1)$$

where

$$\begin{aligned} D(k_1, k_2) &= [d_1(k_1, k_2), d_2(k_1, k_2), \dots, d_m(k_1, k_2)]' \\ W(k_2) &= [w_1(k_2), w_2(k_2), \dots, w_m(k_2)]' \\ N(k_1, k_2) &= \text{diag}[n_1(k_1, k_2), n_2(k_1, k_2), \dots, n_m(k_1, k_2)] \end{aligned}$$

$$\mathbf{n}(\mathbf{k1}, \mathbf{k2}) = [n_1(k1, k2), n_2(k1, k2), \dots, n_m(k1, k2)]'$$

$w_j(k2)$  is the discriminative weight for each subspace  $j$  of each speaker  $k2$  and  $n_j(k1, k2)$  is the number of the feature vectors of  $T_j(\mathbf{k1}, \mathbf{k2})$ .

$d_{dis}(k1, k2)$  is the measure of the similarity score between the training set of speaker  $k1$  and the template of speaker  $k2$  under current discriminative weights.  $d_{dis}(k1, k1)$  is always larger than  $d_{dis}(k1, k2)$  ( $k1 \neq k2$ ) for any positive discriminative weights, since the training set always best matches the speaker's template that is created from this training set.

Let  $h_{dis}(k1, k2) = d_{dis}(k1, k1) - d_{dis}(k1, k2)$ .  $h_{dis}(k1, k2)$  is the measure of the template difference between the speaker  $k1$  and  $k2$  under current discriminative weights.  $h_{dis}(k1, k2)$  equals zero when  $k1 = k2$ , and  $h_{dis}(k1, k2)$  is larger than zero for  $k1 \neq k2$ . The larger the  $h_{dis}(k1, k2)$ , the larger the template difference between speaker  $k1$  and  $k2$ .

The cost function to obtain optimal discriminative weights is given by

$$J = \sum_{k1=1}^N \sum_{k2=1, k1 \neq k2}^N f(h_{dis}(k1, k2)) \quad (5.2)$$

where

$$f(x) = e^{-\alpha x + \beta}$$

$\alpha > 0$  and  $\beta$  are scalars.

To increase the SI accuracy,  $h_{dis}(k1, k2)$  and the corresponding template difference between speaker  $k1$  and  $k2$  are required to be as large as possible, thus the cost function (5.2) needs to be minimized. It is desired to find discriminative weights that minimize the cost function  $J$ , so that the template differences between different speakers are maximized. The selection of  $f(x)$  in (5.2) will be explained in detail later.

The gradient vector  $\nabla J(W(k2))$  is given by

$$\begin{aligned}\nabla J(W(k2)) &= \frac{\partial J}{\partial W(k2)} = \sum_{k1=1}^N \frac{d[f(h_{dis}(k1, k2))]}{d[h_{dis}(k1, k2)]} \frac{\partial[h_{dis}(k1, k2)]}{\partial W(k2)} \\ &+ \sum_{k1=1}^N \frac{d[f(h_{dis}(k2, k1))]}{d[h_{dis}(k2, k1)]} \frac{\partial[h_{dis}(k2, k1)]}{\partial W(k2)}\end{aligned}\quad (5.3)$$

where

$$\frac{d[f(x)]}{d[x]} = -\alpha e^{-\alpha x + \beta} \quad (5.4)$$

$$\frac{\partial[h_{dis}(k1, k2)]}{\partial W(k2)} = -\frac{d[d_{dis}(k1, k2)]}{d[W(k2)]}$$

$$\frac{\partial[h_{dis}(k2, k1)]}{\partial W(k2)} = \frac{d[d_{dis}(k2, k2)]}{d[W(k2)]}$$

$$\frac{d[d_{dis}(k1, k2)]}{d[W(k2)]} = \frac{N(k1, k2)D(k1, k2)}{W(k2)'n(k1, k2)} - \frac{n(k1, k2)(W(k2)'N(k1, k2)D(k1, k2))}{(W(k2)'n(k1, k2))^2}$$

$$\frac{d[d_{dis}(k2, k2)]}{d[W(k2)]} = \frac{N(k2, k2)D(k2, k2)}{W(k2)'n(k2, k2)} - \frac{n(k2, k2)(W(k2)'N(k2, k2)D(k2, k2))}{(W(k2)'n(k2, k2))^2}$$

The updating function for discriminative weights is expressed as

$$W = W - \Gamma \times \nabla J(W) \quad (5.5)$$

where

$$W = [W(1), W(2), \dots, W(h)]$$

$$\nabla J(W) = [\nabla J(W(1)), \nabla J(W(2)), \dots, \nabla J(W(h))]$$

and scalar  $\Gamma$  is the convergence factor.

$h_{dis}(k1, k2)$  represents the template difference between the speaker  $k1$  and  $k2$  under current discriminative weights. When two speakers have larger  $h_{dis}(k1, k2)$  and a corresponding larger template difference between them, the testing waveforms from these speakers are less likely to be misidentified to each other. Further increasing large  $h_{dis}(k1, k2)$  has little advantage

for the SI accuracy improvement. On the other hand, increasing smaller  $h_{\text{dis}}(k1, k2)$  is more likely to increase the SI accuracy. In order to increase the SI accuracy, in the discriminative weights updating, it is desirable to give priority to increasing the smaller  $h_{\text{dis}}(k1, k2)$  than larger ones.

In (5.3),  $-\frac{\partial[h_{\text{dis}}(k1, k2)]}{\partial W(k2)}$  is the direction to increase only  $h_{\text{dis}}(k1, k2)$ . The term  $\frac{d[f(h_{\text{dis}}(k1, k2))]}{d[h_{\text{dis}}(k1, k2)]}$  that appears in (5.3) is the multiplier factor of  $-\frac{\partial[h_{\text{dis}}(k1, k2)]}{\partial W(k2)}$ . It is smaller for larger  $h_{\text{dis}}(k1, k2)$  and larger for smaller  $h_{\text{dis}}(k1, k2)$ . Compared with the cost function which is the direct summation of  $h_{\text{dis}}(k1, k2)$ , the effect of smaller  $h_{\text{dis}}(k1, k2)$  for the discriminative weights updating in (5.5) has been enlarged by introducing  $f(x) = e^{-\alpha x + \beta}$  in (5.3). Thus, smaller  $h_{\text{dis}}(k1, k2)$  has higher priority for increasing than larger  $h_{\text{dis}}(k1, k2)$  in the discriminative weights updating.

Similarly,  $h_{\text{dis}}(k2, k1)$  also represents the template difference between the speaker  $k2$  and  $k1$  under current discriminative weights. Again, smaller  $h_{\text{dis}}(k2, k1)$  has higher priority for increasing than larger  $h_{\text{dis}}(k2, k1)$  in the discriminative weights updating.

The diagram of the training mode of ADVQSI is shown in Figure 23. Codebook  $C(k)$ , discriminative weight  $W(k)$  and space segmentation for speaker  $k$  represent the template of speaker  $k$ . All the templates of speakers in the SI group together with general codebook  $C^g$  are used in the testing mode of ADVQSI.

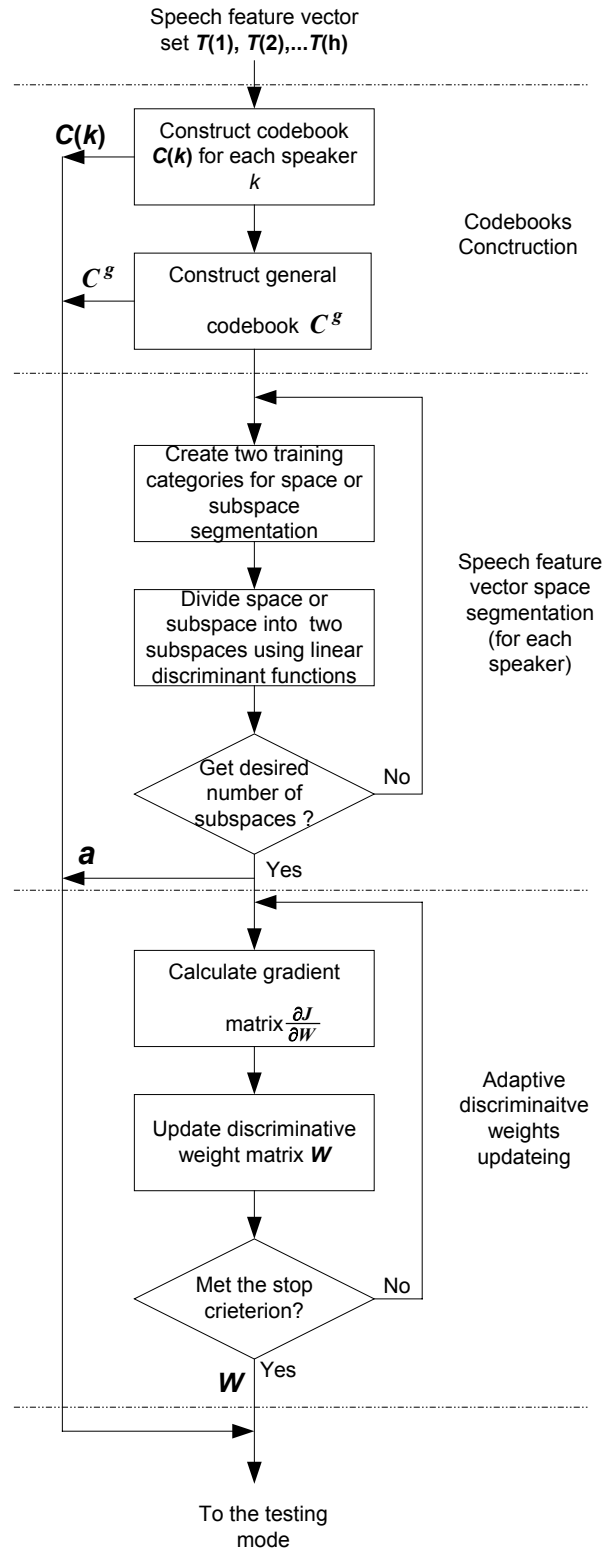


Figure 23. The diagram of the training mode of ADVQSI

### 5.2.2. The Testing Mode

In the testing mode, testing waveforms from unknown speakers in the SI group are presented for speaker identification. For each testing waveform  $R$ , a testing speech feature vector set  $\mathbf{T}(\mathbf{R})$  is created from waveform  $R$ . In this mode, for each testing waveform, discriminative weighted average VQ distortions are calculated. Then, the SI decision is made based on these weighted average VQ distortions.

For each testing waveform  $R$ , the discriminative weighted average VQ distortion  $d_{dis}(R, k)$  for speaker  $k$  is given by

$$d_{dis}(R, k) = \frac{\mathbf{W}(\mathbf{k})' \mathbf{N}(\mathbf{R}, \mathbf{k}) \mathbf{D}(\mathbf{R}, \mathbf{k})}{\mathbf{W}(\mathbf{k})' \mathbf{n}(\mathbf{R}, \mathbf{k})} \quad (5.6)$$

where

$$\mathbf{D}(\mathbf{R}, \mathbf{k}) = [d_1(R, k), d_2(R, k), \dots, d_m(R, k)]'$$

$$d_j(R, k) = d_{gj}(R, k) - d_{kj}(R, k).$$

$$\mathbf{N}(\mathbf{R}, \mathbf{k}) = \text{diag}[n_1(R, k), n_2(R, k), \dots, n_m(R, k)]$$

$$\mathbf{n}(\mathbf{R}, \mathbf{k}) = [n_1(R, k), n_2(R, k), \dots, n_m(R, k)]'$$

$n_j(R, k)$  is the number of the feature vectors in  $\mathbf{T}_j^R(\mathbf{k})$ ,  $d_{kj}(R, k)$  is the average VQ distortion of  $\mathbf{T}_j(\mathbf{R}, \mathbf{k})$  quantized by  $\mathbf{C}(\mathbf{k})$ , and  $d_{gj}(R, k)$  is the average VQ distortion of  $\mathbf{T}_j(\mathbf{R}, \mathbf{k})$  quantized by  $\mathbf{C}^g$ , and  $\mathbf{T}_j(\mathbf{R}, \mathbf{k})$  is the set for all speech feature vectors of  $\mathbf{T}(\mathbf{R})$  located in subspace  $j$  of speaker  $k$ .

$d_{dis}(R, k)$  is the similarity matching score between the testing waveform  $R$  and the speech template of speaker  $k$ . The larger the  $d_{dis}(R, k)$ , the better the matching. The definition of  $d_{dis}(R, k)$  in (5.6) is similar to the definition of  $d_{dis}(k1, k2)$  in (5.1), except the former uses the testing

speech feature vector set and the latter considers the training speech feature vector set. The definitions of  $d_{dis}(k1, k2)$  in the training mode and  $d_{dis}(R, k)$  in the testing mode are consistent.

The SI decision rule is expressed as follows: the unknown waveform  $R$  comes from speaker  $i$ , if  $d_{dis}(R, i) = \max_{k=1,2,\dots,h} d_{dis}(R, k)$ . The testing waveform is classified to the speaker, whose template most closely matches the testing waveform.

### **5.3. Experimental Results**

In this section, experiments are given to evaluate the effectiveness of the proposed ADVQSI approach. Speech records are obtained from the CSLU (Center for Spoken Language Understanding, Oregon Health & Science University) Speaker Recognition V1.1 corpus. For each speaker, the speech records collected on different collection dates are packaged into different recording sessions. There are mismatches between the speech utterances taken from different speakers. Also, there are mismatches due to different recording sessions of the same speaker. All the speech files in the corpus were sampled at 8 kHz and 8-bits per sample.

Thirty-five speakers are used in the text-independent SI experiments. Four spontaneous speeches for each speaker are used in the training mode. Two other spontaneous speeches, taken about one year after the training speech waveform for each speaker, are used in the testing mode. Each speech waveform lasts about 4 seconds.

Silenced and unvoiced segments are discarded based on an energy threshold. The analysis Hamming window size is 32ms, 256 samples, with 24ms overlapping [49]. The feature vector used in the experiment is composed of 15 Mel Frequency Cepstral Coefficients (MFCCs) [60].

The codebook sizes of VQSI, DVQSI, DVQSI-U and ADVQSI are 64. In this work, the speech feature vector space is divided into 4 subspaces, i.e.,  $m=4$ . All the codebooks are constructed by the Generalized Lloyd algorithm [18, 32]. The initial values of codebooks are obtained by using the splitting algorithm [18, 32]. The parameters for the adaptive discriminative weights updating are  $\alpha=0.3$ ,  $\beta=9$ , and  $\Gamma=0.05$ . The initial value for all discriminative weights is 100.

Table 8 shows the SI accuracy results employing VQSI, DVQSI, DVQSI-U, and ADVQSI. It is observed that ADVQSI and DVQSI-U result in the highest SI accuracies. The SI accuracy of DVQSI is better than that of VQSI. The discussions and simulation results of the parameters selection for DVQSI/DVQSI-U are given in previous chapters. Compared with VQSI, DVQSI/DVQSI-U and ADVQSI exploit interspeaker variations between different speakers (or speaker groups). The ADVQSI approach employs adaptive techniques to find optimal discriminative weights, whereas the DVQSI/DVQSI-U approach obtains discriminative weights by trial and error.

Table 8. The SI accuracy rates employing VQSI, DVQSI, and ADVQSI

Technique	DVQSI	DVQSI	DVQSI-U	ADVQSI
SI accuracy	62.9%	68.6%	71.4%	71.4%

For simplification, in ADVQSI experiments, the subspaces are ranked from the highest interspeaker variation to the lowest interspeaker variation for all speakers. Table 9 shows the average values of  $d(v)$  for the speech feature vector space segmentation of the first speaker. The average values of  $d(v)$  for different subspaces are not equal. This means that different subspaces



have various interspeaker variations between speaker 1 and all speakers in the SI group, i.e., the lower the average value of  $d(v)$ , the higher the interspeaker variation in the subspace. The feature vector space segmentation of ADVQSI is based on the interspeaker variation between each speaker and all speakers in the SI group.

Table 9. The average  $d(v)$  for the first speaker in the speech feature vector space segmentation

For all the training feature vectors	For feature vectors in subspace 1	For feature vectors in subspace 2	For feature vectors in subspace 3	For feature vectors in subspace 4
0.6763	0.4110	0.6420	0.7028	0.8862

The mean value of the discriminative weights for all the speakers in each subspace versus the number of adaptive iterations is presented in Figure 24. From Figure 24, it is seen that the subspaces with higher interspeaker variations increase their discriminative weights as the adaptive algorithm converges. In contrast, the adaptive algorithm reduces discriminative weights of subspaces, which have lower interspeaker variations. As a result, the subspaces with higher interspeaker variations play more important roles in the SI decision than the ones with lower interspeaker variations by assigning different discriminative weights to different subspaces. Though the mean values of the discriminative weights in different subspaces are different at the end of the discriminative weights updating, all of them are positive. This means that all the subspaces play positive roles in SI.

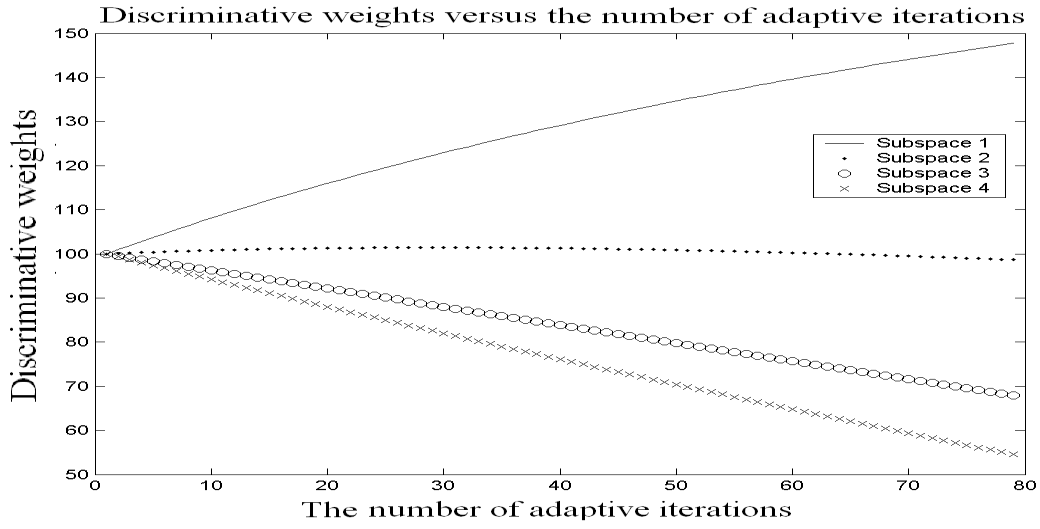


Figure 24. The average discriminative weights for different subspaces versus the number of adaptive iterations

The value of the cost function  $J$  in (5.2) versus the number of adaptive iterations is given in Figure 25. The value of the cost function decreases as the adaptive algorithm converges. The average value of  $h_{dis}(k1,k2)$  for all possible speaker pairs versus the adaptive iteration number is given in Figure 26. This value increases when the number of adaptive iterations increases. The results confirm that the adaptive algorithm converges successfully.

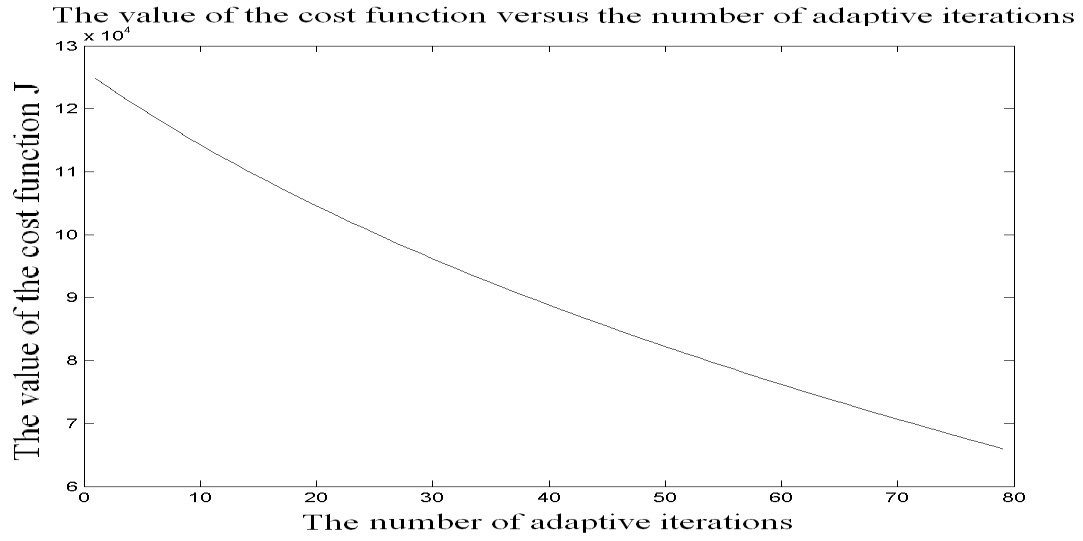


Figure 25. The value of the cost function  $J$  in (5.2) versus adaptive the number of adaptive iterations

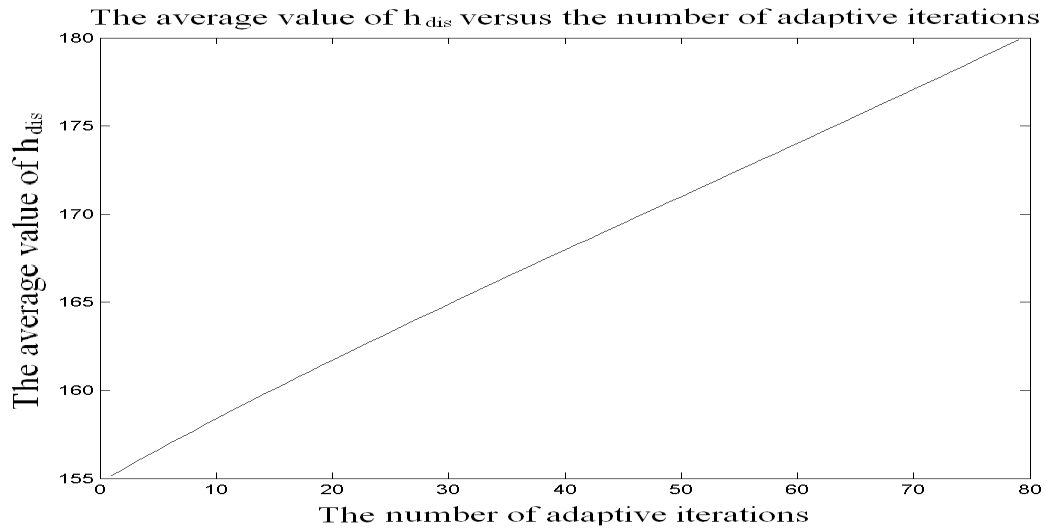


Figure 26. The average value of  $h_{dis}(k1,k2)$  for all speaker pairs versus the number of adaptive iterations

#### **5.4. Conclusions**

In this work, a new SI approach based on Adaptive Discriminative VQ is developed and presented. The ADVQSI technique takes advantage of the interspeaker variation between each individual speaker and all speakers in the SI group. In the training mode of this technique, for each speaker, the speech feature vector space is divided into a number of subspaces, based on interspeaker variation between this speaker and all speakers. Then, an optimal discriminative weight is adaptively trained for each speaker and each subspace in order to maximize the template differences between different speakers for SI. In the test mode of ADVQSI, discriminative weighted average VQ distortions are used as similarity measures between speakers' templates and each testing waveform. The testing waveform is classified to the speaker whose template leads to the highest similarity score.

The effectiveness of the ADVQSI approach is demonstrated experimentally. It is shown that the proposed technique yields better SI accuracy than the VQSI approach.

Compared with recently reported DVQSI/DVQSI-U approach, ADVQSI determines discriminative weights by using adaptive techniques instead of trial and error. Because ADVQSI considers each speaker instead of each speaker pair, the computational requirement of ADVQSI is considerably reduced relative to DVQSI/DVQSI-U, in which discriminative weights are assigned for each speaker pair.

Although the ADVQSI technique is applied to SI, this technique can be gainfully extended to other pattern identification applications, such as handwritten character identification and face identification.

## **CHAPTER SIX: COMPATIBLE PROBABILITY MEASURES FOR THE OUTPUTS OF THE TEMPLATE-BASED SPEAKER IDENTIFICATION CLASSIFIER FOR DATA FUSION**

### **6.1. Introduction**

Data fusion is a popularly used and promised technique in pattern classification [1, 21, 26, 55, 63]. Basically, it combines the results of different pattern classification classifiers together, in order to improve the classification accuracy. Data fusion techniques should take advantage of each classifier and avoid each classifier's limitations. Data fusion systems can be categorized into three levels based on the type of the raw output information of classifiers [63]. At the abstract level, the output of each classifier is a unique class label. Each classifier ranks the candidate classes from highest to lowest likelihood at the rank level. At the measurement level, a similarity score is assigned for each candidate class by each classifier.

Speaker models in SI are constructed from the speech features extracted from the speech signal. There are two kinds of speaker models, template models and stochastic models. Correspondingly, there are two kinds of SI classifiers, the template-based classifiers and the stochastic-based classifiers. The raw outputs of template-based approaches are distortions between the testing speech waveform and speakers' templates. While the raw outputs of stochastic-based approach are the likelihood between the testing speech waveform and speakers' stochastic models. It is apparent that the outputs of most SI classifiers are similarity scores. It is

preferable to consider the data fusion problem of SI at the measurement level in order to fully use the raw information of classifiers' outputs.

However, distortion outputs of template-based methods are generally incompatible with probability measures of stochastic-based methods. Even for different SI classifiers in the same category, the outputs of various SI classifiers often have different scales. The existing combination techniques for the data fusion at the measurement level, such as the linear opinion pools technique and the log opinion pools technique, require the results of different SI classifiers are compatible. In order to apply the existing combination techniques at the measurement level, the raw outputs of different classifiers need to be converted into some compatible measures, which are typically in terms of probability. In stochastic-based approaches, compatible measures can be easily obtained by converting the likelihood outputs into the posteriori probabilities. In the template-based approach, the existing technique, which transfers the distortion output  $d$  into likelihood  $L$ , is given by [4, 9, 11]

$$L=e^{-\alpha d} \quad (6.1)$$

where  $\alpha$  is a positive constant need to be estimated. In this technique, the distortion outputs are assumed to be proportional to the log likelihood. There are no experiment results or theories to support this assumption, and the proper estimation of the parameter  $\alpha$  is often difficult.

In this chapter, a novel approach, which transfers the distortion outputs of each template-based SI classifier into compatible probability measures, is presented. In the proposed approach, for each classifier, a large set of training utterances is needed for each speaker in the SI group. The stochastic models for the distortion outputs of each classifier are estimated first in this technique. In the estimation, the exact same feature extraction and pattern matching techniques employed in the classifier are used. All but one training utterances for each speaker are used to

construct a reference template for this speaker. Then, for each possible speaker pair, a distortion is calculated for the remaining training utterance of one speaker and the corresponding constructed reference template of the other speaker. This process is repeated  $n$  times, where  $n$  is the number of training utterances for each speaker. Based on distortions obtained in the estimation, for each speaker, given that the unknown utterance comes from this speaker, stochastic models for distortion outputs of the classifier are estimated. Next, for each classifier, the posteriori probabilities of the unknown utterance belonging to each speaker are calculated based on the corresponding stochastic models of distortion outputs. Finally, compatible probability measures are assigned based on the posteriori probabilities.

This chapter is organized as follows. Section 6.2 presents the proposed compatible probability measures for the distortion outputs of the template-based SI classifier for data fusion. Experimental results to evaluate the proposed approach are given in Section 6.3. Section 6.4 contains conclusions.

## **6.2. Compatible Probability Measures for the Outputs of the Template-based SI Classifier for Data Fusion**

For each speaker in the SI group, the number of the training utterances available for each speaker is a large integer and denoted by  $n$ .

In each template-based SI classifier, a reference template  $T(j)$  for each speaker  $j \in \Lambda$  is constructed in the training mode by all  $n$  training speech utterances of speaker  $j$ , where  $\Lambda$  is the closed set of the speakers for SI and  $\Lambda = \{\text{speaker 1, speaker 2, ..., speaker } N\}$ . For simplification, speaker  $j \in \Lambda$  is represented by  $j \in \Lambda$ . Then, in the testing mode, for each unknown utterance  $R$ , the

distortion  $d(R, j)$  between  $R$  and each  $T(j)$  is calculated.  $d(R, j)$  is the distortion output of the template-based SI classifier. For each individual classifier, SI is performed by finding the reference template  $T(h)$  ( $h \in \Lambda$ ) and its corresponding speaker  $h$ , which gives the smallest distortion  $d(R, h)$ , to represent the unknown utterance  $R$ .

The key factor of the proposed technique is to obtain the stochastic model  $m(k, j)$  of  $d(R, j)$ , given that  $R \in k$ , for each  $j \in \Lambda$  and  $k \in \Lambda$ , where  $R \in k$  denotes the unknown utterance  $R$  belonging to speaker  $k$ .

First, the template  $T_i'(j)$  of each speaker  $j$ , constructed by all but the  $i$ th training utterances of speaker  $j$  is calculated for each  $i$ , where  $i$  is an index of the training utterances for each speaker ( $i=1, 2, \dots, n$ ).  $T(j)$  and  $T_i'(j)$  are the templates for the same speaker  $j$  and constructed by the same technique. Since the number of the training utterances  $n$  for each speaker is a large integer, most training vectors for  $T(j)$  are used in the construction of  $T_i'(j)$ . This leads to  $T(j)$  and  $T_i'(j)$  becoming similar. Then, the distortion  $d_i'(k, j)$  between the  $i$ th training utterances of each speaker  $k$  and  $T_i'(j)$  is obtained for each  $i$  and  $j$ . A stochastic model  $m(k, j)$  of  $d(R, j)$ , given that  $R \in k$ , is estimated by the distribution of distortions  $d_i'(k, j)$  ( $i=1, 2, \dots, n$ ) for each  $k$  and  $j$ . In this work,  $m(k, j)$  is assumed to follow the Gaussian distribution. The flow chart of the estimation of  $m(k, j)$  is shown in Figure 27.



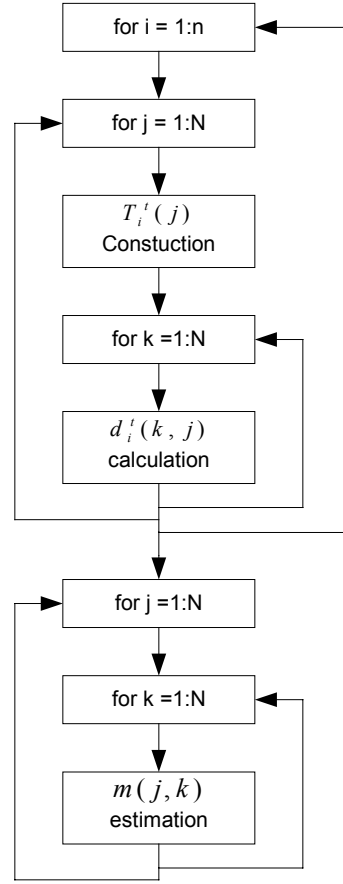


Figure 27. The flow chart of the estimation of  $m(j, k)$

Since the number of training utterances  $n$  for each speaker is large, it is reasonable to assume that the mismatches between training utterances and testing utterances can be represented by the stochastic models of the corresponding mismatches between different training utterances. If the above assumption is true, since  $T(j)$  is similar to  $T_i'(j)$  as mentioned before, the stochastic model of  $d_i'(k, j)$  is similar to the stochastic model of  $d(R, j)$  for  $R \in k$ .

Let  $M = \{m(k, j)\}$ , for  $k = 1, 2, \dots, N$  and  $j = 1, 2, \dots, N$ , be a collection of  $m(k, j)$ , a conditional probability  $P[d(R, j) | R \in k, M]$  is obtained for each  $R, k$  and  $j$ .

The posteriori probability  $P[d(R, j) | R \in k, M]$  is calculated by using Bayes rule, i.e.

$$\begin{aligned}
P[R \in k | d(R, j), M] &= \frac{P[d(R, j) | R \in k, M]P[R \in k | M]}{p[d(R, j) | M]} \\
&= \frac{P[d(R, j) | R \in k, M]P[R \in k | M]}{\sum_{h=1}^N P[d(R, j) | R \in h, M]P[R \in h | M]}
\end{aligned} \tag{6.2}$$

For most SI systems, no prior knowledge of  $P[R \in h | M]$  ( $h=1, 2, \dots, N$ ) can be obtained, which leads to

$$P[R \in h | M] = 1/N \tag{6.3}$$

Substituting from (6.3) into (6.2) yields

$$P[R \in k | d(R, j), M] = \frac{P[d(R, j) | R \in k, M]}{\sum_{h=1}^N P[d(R, j) | R \in h, M]} \tag{6.4}$$

Compatible probability measure  $O(R, k)$  for  $R \in k$  is based on the posteriori probabilities.

In this work, it is given by

$$O(R, k) = P[R \in k | d(R, k), M] \text{ for } k=1, 2, \dots, N \tag{6.5}$$

### **6.3. Experimental Results**

In this section, sample experimental results are given to illustrate the effectiveness of the proposed technique. Speech records are obtained from the CSLU (Center for Spoken Language Understanding, Oregon Health & Science University) Speaker Recognition V1.1 corpus. For each speaker, the speech records collected on different collection dates are packaged into different recording sessions. There are mismatches between the speech utterances taken from different speakers. Also, there are mismatches due to different recording sessions of the same speaker. All the speech files in the corpus were sampled at 8 kHz and 8-bits per sample.

Thirty-five speakers are used in the text-dependent SI experiments. Eleven text-dependent speech phrases for each speaker are used in the training. A text-dependent speech phrase for each speaker is used in the testing mode. Each speech utterance in the training mode lasts about 2 to 3 seconds.

Thirty-five speakers are used in the text-independent SI experiments. Eleven spontaneous speeches for each speaker are used in the training. A spontaneous speech for each speaker is used in the testing mode. Each speech utterance in the training mode lasts about 5 to 8 seconds.

Silenced and unvoiced speech segments are discarded based on an energy threshold. The analysis Hamming window size is 32 milliseconds (256 samples) with 16 milliseconds overlapping between successive windows.

Two template-based classifiers are used in the experiments. The feature vector used in SI Classifier 1 is composed of 15 Mel Frequency Cepstral Coefficients (MFCC's) [60]. The other SI classifier uses Linear Predictive Coding \_ Log Area Ratios (LPC\_LAR) as the feature vector. For both classifiers, the Vector Quantization (VQ) method is used for pattern matching. All VQ codebooks are constructed by the Generalized Lloyd algorithm with the splitting algorithm for the initial values [18, 32, 58]. The linear opinion pool combination function with equal weights for both classifiers is used in data fusion experiments of this work.

The results of text-dependent and text-independent experiments based on the proposed technique are given in Table 10. SI data fusion results employing the likelihood technique given by (6.1) [4, 9, 11], are shown in Figure 28 for comparison.

Table 10. The SI accuracies rate by employing individual classifiers and data fusion techniques

	Text-dependent	Text-independent
--	----------------	------------------

	SI accuracy based on distortion outputs	SI accuracy based on probability outputs	SI accuracy based on distortion outputs	SI accuracy based on probability outputs
SI Classifier 1	77.1%	82.9%	77.1%	74.3%
SI Classifier 2	82.9%	77.1%	77.1%	77.1%
Data fusion-based SI	*	88.6%	*	82.9%

\* indicates that SI results of Classifier 1 and 2, which are based on distortion outputs, cannot be combined together directly.

Table 10 shows that the SI accuracies based on the distortion outputs and the corresponding compatible probability measures are comparable for both classifiers. Data fusion-based SI leads to higher SI accuracy than either individual classifier. The data fusion results based on the proposed technique, Table 10, are comparable to the best data fusion results in Figure 28. The results in Figure 28 employ the technique given in (6.1) and the best results in Figure 28 are obtained by trial and error [4, 9, 11].

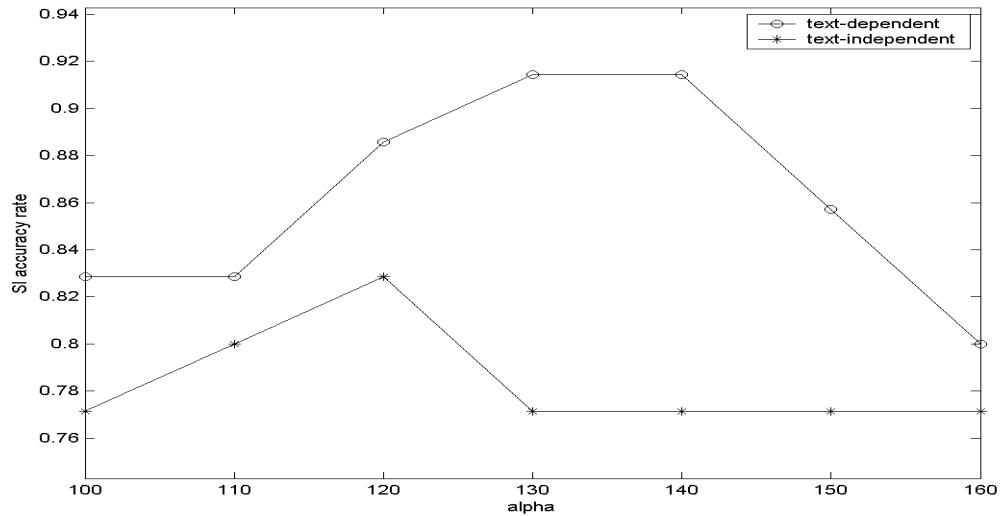


Figure 28. The SI accuracies versus  $\alpha$  of the second classifier by employing Eq. (6.1) in the data fusion, where  $\alpha$  of the first classifier is set to 1

–O– for the text-dependent experiment and –\*– for the text-independent experiment.

#### **6.4. Conclusions**

In this chapter, a novel approach, which converts the distortion outputs of the template-based SI classifier into compatible probability measures for SI data fusion at the measurement level, is presented. In the proposed approach, the stochastic models of the distortion outputs of the SI classifier for each speaker are estimated, given that the unknown utterance comes from this speaker. Then the posteriori probabilities of the unknown utterance belonging to each speaker are calculated. Compatible probability measures of the distortion outputs of SI classifiers are assigned based on the posteriori probabilities.

From experimental results, the SI accuracies employing compatible probability measures are comparable to those obtained by using the corresponding distortion outputs. The SI accuracy employing data fusion-based SI is better than those obtained from individual classifiers. The data fusion results based on the proposed technique are comparable to the best data fusion results by using the existing technique, which converts the distortion outputs into the likelihood and gets the best results by trial and error.

## **CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK**

In this contribution, novel Discriminative Vector Quantization (DVQ) based techniques and data fusion based techniques for Speaker Identification (SI) are developed and presented. This chapter summarizes contributions of the research and concludes with suggestions for future research.

### **7.1. Main Contributions**

The research presented in this dissertation contains the following contributions.

In Chapter three, the DVQ technique for SI (DVQSI) is presented and its parameters selection is discussed. The DVQSI technique takes advantage of the interspeaker variation between each possible speaker pair in the SI group. The speech feature vector space is segmented into subspaces for all speaker pairs. For each speaker pair, different subspaces of the speech feature vector space play different roles in SI by assigning various discriminative weights. Discriminative weighted average VQ distortions instead of equally weighted average VQ distortion are used for the SI decision. The existing VQ technique for SI (VQSI) can be considered a special case of the DVQSI technique, where only one subspace, which equals the speech feature vector space, is used. The advantages of the DVQSI technique are confirmed by experiments and the results are reported in [66, 67].

An enhanced approach of DVQSI, DVQSI with Unique speech feature vector space segmentation for each speaker pair (DVQSI-U), is investigated in Chapter four. One of the key techniques of DVQSI is the speech feature vector space segmentation. In DVQSI, all the speaker

pairs in the SI group share the same space segmentation. However, in DVQSI-U, each speaker pair has its individual space segmentation based on this speaker pair's interspeaker variation. In addition, in the testing mode of DVQSI-U, an improved technique is presented to calculate the discriminative weighted average VQ distortions for speaker pairs. The new technique ignores the subspaces that may lead to wrong SI decisions. The comparison between DVQSI and DVQSI-U is provided in [68]. DVQSI-U leads to higher SI accuracies than DVQSI, at the price of much higher computational complexity.

A novel DVQ based technique, Adaptive DVQ technique for SI (ADVQSI), is introduced in Chapter five. DVQSI and DVQSI-U presented in previous chapters assign discriminative weights for each speaker pair in the SI group and appropriate discriminative weights are selected by trial and error. In ADVQSI, discriminative weights are obtained for each speaker in the SI group by using adaptive techniques. The computational burden of ADVQSI is significantly reduced, compared with DVQSI and DVQSI-U, while SI accuracies of DVQSI-U and ADVQSI are comparable. The improvements of ADVQSI over DVQSI and DVQSI-U are presented in [70-72]

Chapter six derives a technique, which converts the raw outputs of template-based SI classifiers into compatible probability measures. This technique makes data fusion at the measurement level applicable to SI. It is shown that SI accuracies employing our technique are comparable with the best results of previous reported approaches, which obtain its parameters by trial and error [69].

## **7.2. Areas of Future Research**

The presented research work can be extended in the following directions.

DVQ based techniques can be gainfully extended to the general area of pattern recognition, where a number of feature vectors can be obtained to describe each object for recognition. Although DVQ is only applied to SI in this dissertation, the technique can be employed in other pattern recognition applications, such as image registration, image recognition, face recognition, optical character recognition, and wafer surface inspection

In the data fusion approach for SI presented in Chapter six, raw outputs of various template-based SI classifiers need to be converted in the compatible probability measures, before the application of the linear opinion pool technique. If an optimal weight can be found for each classifier, the linear opinion pool technique can be employed directly without the probability conversion. This will considerably reduce the computational requirement. In future research, adaptive techniques used in Chapter five for the ADVQSI method can be used to decide the optimal weights of template-based SI classifiers for the application of the linear opinion pool technique.



## LIST OF REFERENCES

- [1] K. Al-Ghoneim and B.V.K. Vijaya Kumar, "Unified decision combination framework," *Pattern Recognition*, vol. 31, no. 12, pp. 2077-2089, 1998.
- [2] U. Bayazit and W.A. Pearlman, "Variable-length constrained-storage tree-structured vector quantization," *IEEE Transactions on Image Processing*, vol. 8, pp. 321-331, 1999.
- [3] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 955-966, 1995.
- [4] J.P. Campbell, "Speaker recognition: A tutorial," *Processing of IEEE*, vol. 85, pp. 1437-1462, 1997.
- [5] W.M. Campell, K.T. Assaleh and C.C. Broun, " Speaker recognition with polynomial classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 4, pp. 205-211, 2002.
- [6] C.C. Chibelushi, F. Deravi and J.S.D. Mason, "A review of speech-based bimodal recognition," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 23-37, 2002.
- [7] S. Davis and P. Mermelstein, "Comparison of parameteric representations for monosyllabic word Recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28, pp. 357-366, 1980.
- [8] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2001.
- [9] K.R. Farrell, S. Kosonocky and R. Mammone, "Neural tree network/vector quantization probability estimators for speaker recognition," *Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing*, pp. 279 –288, 1994.

- [10] K.R. Farrell, R.J. Mammone and K.T. Assaleh, "Speaker recognition using neural networks and conventional classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 194-205, 1994.
- [11] K.R. Farrell, R.P. Ramachandran and R.J. Mammone, "An analysis of data fusion methods for speaker verification," *ICASSP-98*, pp. 1129 –1132, 1998.
- [12] G. Feng and E. Castelli, "Some Acoustic features of nasal and nasalized vowel: A target for vowel nasalization," *J. Acoust. Soc.*, pp. 3728-3737, 1996.
- [13] T.E.F. Filho, R.O. Messina and Jr. E.F. Cabral, "Learning vector quantization in text-independent automatic speaker recognition," *1998 Proceedings. Vth Brazilian Symposium on Neural Networks*, pp. 135 –139, 1998.
- [14] S. Furui, "Recent advance in speaker recognition," *Pattern Recognition Letters*, vol. 18, pp. 859-872, 1997.
- [15] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, Marcel Dekker, Inc., New York, 2001.
- [16] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Mag.*, vol. 11, pp. 18–32, 1994.
- [17] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, John Wiley & Sons, NY, 2000.
- [18] A. Gresho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publisher, Boston, 1991.
- [19] J. He, L. Liu and G. Palm, "A discriminative training algorithm for VQ-based speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 7 pp. 353 –356, 1999.

- [20] A. Higgins, L. Bhaler and J. Porter, "Voice identification using nearest neighbor distance measure," *ICASSP-93*, pp. 375–378, 1993.
- [21] T.K. Ho, J.J. Hull and S.N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66 –75, 1994.
- [22] Y.H. Hu, S. Palreddy and W.J. Tompkins, "A patient-adaptable ECG beat classifier using a mixture of experts approach," *IEEE Trans. Biomed. Eng.*, vol. 44, pp. 891-900, 1997.
- [23] W. Hwang and H. Derin; "Multistage storage- and entropy-constrained tree-structured vector Quantization," *IEEE Transactions on Signal Processing*, vol. 44, pp. 1810-1810, 1996.
- [24] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, pp. 3043-3054, 1992.
- [25] J.C. Junqua and J.P. Haton, *Robustness in Automatic Speech Recognition: Fundamentals and Application*, Kluwer Academic Publishers, Bonston, 1996.
- [26] J. Kittler, M. Hatef, R.P.M. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226-239, 1998.
- [27] T. Kohonen, "The self-organizing map," *Processings of IEEE*, vol. 78, pp. 1464-1480, 1990.
- [28] T. Koski, *Hidden Markov Models for Bioinformatics*, Kluwer Academic Publishers, Boston, 2001.
- [29] L.I. Kuncheva, J.C. Bezdek and R.P.W. Dulin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, pp.299-314, 2001

- [30] V. Krishnan and W.B. Mikhael, "Efficient code excited linear predictor using redundant vector quantiser representations," *Electronics Letters*, pp. 1370–1372, 2001.
- [31] I. Lapidot, H. Guterman, and A. Cohen, "Unsupervised speaker recognition based on competition between self-organizing maps," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, 877-887, 2002.
- [32] Y. Linde, A. Buzo and R.M. Gray, "A algorithm for vector quantizer Design," *IEEE Transactions on Communication*, vol. 28, pp 702-710, 1980.
- [33] B. Mak, "A mathematical relationship between full-band and multiband mel-frequency cepstral coefficients," *IEEE Signal Processing Letters*, vol. 9, pp. 241-244, 2002.
- [34] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, pp. 561–580, 1975.
- [35] R. Mammone, X. Zhang, and R. Ramachandran, "Robust speaker recognition—A feature-based approach," *IEEE Signal Processing Mag.*, vol. 13, pp. 58–71, 1996.
- [36] T. Matsui and S. Furui, "A text-independent speaker recognition method robust against utterance variations," *ICASSP-91*, pp. 377–380, 1991.
- [37] T. Matsui and S. Furui, "Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs," *ICASSP-92*, pp.157-160, 1992.
- [38] T. Matsui and S. Furui, "Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMM's," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 456-459, 1994.
- [39] W.B. Mikhael and V. Krishnan, "Multiple transform domain split vector quantization," *Electronics Letters*, pp. 538–539, 2001.

- [40] W.B. Mikhael and V. Krishnan, "Energy\_based split vector quantizer employing signal representation in multiple transform domains," *Digital Signal Processing*, pp. 359-370, 2001.
- [41] W.B. Mikhael and P. Premakanthan, "Speaker identification employing redundant vector quantisers," *Electronics Letters*, vol. 38, pp. 1396 –1398, 2002.
- [42] W.B. Mikhael and P. Premakanthan, "An improved speaker identification technique employing multiple representations of the linear prediction coefficients," *Proceedings of the 2003 International Symposium on Circuits and Systems*, pp. 584-587, 2003.
- [43] S. Molau, M. Pitz, R. Schluter and H. Ney, "Computing Mel-frequency cepstral coefficients on the power spectrum," *ICASSP2001*, pp.73-76, 2001
- [44] T. Moriya, "Processing of LPC Cepstrum for Speech Coding," *IEEE Workshop on Speech Coding for Telecommunication*, pp. 83-84, 1995.
- [45] J. Naik, "Speaker verification: A tutorial," *IEEE Commun.Mag.*, pp. 42–48, 1990.
- [46] J. Oglesby and J.S. Mason, "Optimization of neural models for speaker identification," *ICASSP-90*, 261-264, 1990.
- [47] M.D. Plumpe, T.F. Quatieri, and D.A. Reynolds, "Modeling of the glottal flow derivative waveform with application to speaker identification" *IEEE Transactions on Speech and Audio Processing*, pp. 569-586, 1999.
- [48] T.F. Quatieri, *Discrete-Time Speech Signal Processing: Principal and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, 2002.L. Rabiner, and B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, London, 1993.
- [49] L. Rabiner, and B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, London, 1993.

- [50] R.P. Ramachandran, K.R. Farrell, R. Ramachandran and R.J. Mammone, "Speaker recognition-general classifier approaches and data fusion methods," *Pattern Recognition*, vol. 35, pp. 2801-2821, 2002.
- [51] D.A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91-108, 1995.
- [52] D.A. Reynolds, "An overview of automatic speaker recognition technology," 2002 *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. IV-4072 -IV-4075, 2002.
- [53] K. Rose, D. Miller and A. Gersho, "Entropy-constrained tree-structured vector quantizer design," *IEEE Transactions on Image Processing*, vol. 5, pp. 393-398, 1996.
- [54] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 43-49, 1978
- [55] H. Saranli and M. Demirekler, "A stochastic unified framework for rank-based multiple classifier decision combination," *Pattern Recognition*, vol. 34, pp. 865-88, 2001.
- [56] D. O'Shaughnessy, *Speech communication, human and machine*, IEEE Press, NY, 2000.
- [57] F.K. Song and B.H. Juang, "Optimal Quantization of LSP coefficients," *IEEE Trans. Speech and Audio Processing*, pp. 15-23, 1993.
- [58] F.K. Soong, A.E. Rosenberg, L.R. Rabiner and B.-H Juang, "A vector quantization approach to speaker recognition," *ICASSP-85*, pp. 387-390, 1985.
- [59] N.Z. Tishby, "On the application of mixture AR hidden Markov models to text independent speaker recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 563-570, 1990.

- [60] R. Vergin, D. O'Shaughnessy and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 525 –532, 1999.
- [61] R. Viswanathan and J. Makhoul "Quantization properties of transmission parameters in linear predictive systems," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, pp. 309-321, 1975.
- [62] D.M. Weber and J.A. du Preez, "A comparison between hidden Markov models and vector quantization for speech independent speaker recognition," *Proceedings of the 1993 IEEE South African Symposium on Communications and Signal Processing*, pp. 139 –144, 1993.
- [63] L. Xu, A. Krzyzak and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 22, pp. 418-435, 1992.
- [64] K. Yu, J. Mason and J. Oglesby, "Speaker recognition using hidden Markov models, dynamic time warping and vector quantization," *IEE Proceedings- Vision, Image and Signal Processing*, vol. 142, no. 5, pp. 313-318, 1995.
- [65] Y. Zhang, D Zhang and X. Zhu, " A novel text-independent speaker verification method based on the global speaker model," *IEEE Transactions on Systems, Man and Cybernetics-Part A*, vol. 30, no. 5, pp. 598 –602, 2000.
- [66] G. Zhou, W.B. Mikhael and B. Myers "A novel discriminative vector quantization approach for speaker identification," to appear, *Journal of Circuits, Systems and Computers*, 2005.

- [67] G. Zhou and W.B. Mikhael, "Speaker identification based on discriminative vector quantization," *the 46th IEEE International Midwest Symposium on Circuits and Systems*, Cairo, Egypt, December 2003.
- [68] G. Zhou and W.B. Mikhael, "Analysis of discriminative vector quantization approach for speaker identification," *the 8<sup>th</sup> World Multi-Conference on Systemic, Cybernetics and Information*, Orlando, Florida, USA, pp. IV 479-483, July 18-21, 2004.
- [69] G. Zhou and W.B. Mikhael, "Compatible probability measures of the outputs of template-based speaker identification classifiers for data fusion," *2004 IEEE International Symposium on Circuits and Systems*, Vancouver, British Columbia, Canada, pp. III 473-476, May 23-26, 2004.
- [70] G. Zhou and W.B. Mikhael, "Speaker identification based on vector quantization with adaptive discriminative techniques," accepted by *the 48<sup>th</sup> IEEE International Midwest Symposium on Circuits and Systems*, Cincinnati, Ohio, August 7-10, 2005.
- [71] G. Zhou and W.B. Mikhael, "Speaker identification based on adaptive discriminative vector quantization," submitted to *IEE Proceedings-Vision, Image & Signal Processing*.
- [72] G. Zhou and W.B. Mikhael, "Adaptive discriminative vector quantization for speaker identification accuracy enhancement," submitted to *IEE Electronics Letters*.
- [73] X. Zhu, B. Millar, J. Macleod, M. Wagner, F. Chen, and S. Ran, "A comparative study of mixture-Gaussian VQ, ergodic HMMs and left-to-right HMMs for speaker recognition," *ISSIPNN '94*, pp. 618 –621, 1994.