Speaker segmentation and verification

Zhong, Haishan

2008

Zhong, H. (2008). Speaker segmentation and verification. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2604>

<https://doi.org/10.32657/10356/2604>

Nanyang Technological University

Downloaded on 26 Apr 2023 23:20:50 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**SPEAKER SEGMENTATION AND
VERIFICATION**

ZHONG HAISHAN

SCHOOL OF COMPUTER ENGINEERING

2008

SPEAKER SEGMENTATION AND VERIFICATION

ZHONG HAISHAN

2008

Speaker Segmentation and Verification

Zhong Haishan

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2008

Acknowledgements

I would like to thank my previous supervisor, Professor Graham Leedham who spent much of his precious time guiding me and my research, as well as repeatedly proofreading my thesis. I would also like to thank my current supervisor, Assistant Professor David Cho for guiding my research during the last few months. An additional thank you goes to Assistant Professor Vladimir Pervouchine for helping me in my research and providing me with invaluable knowledge. I would like to thank Michael Magnin for his encouragement and his help with proofreading my thesis. Finally, I would like to thank my family and my friends who helped and supported me during the pursuit of my Master's Degree.

Contents

Acknowledgements	1
Abstract	5
1 Introduction	7
1.1 Background	7
1.1.1 Speaker identification and speaker verification	8
1.1.2 Text-dependent and text-independent speaker recognition	12
1.2 Motivation and problems of speaker recognition	14
1.2.1 Speaker segmentation	15
1.2.2 Features for speaker recognition	17
1.3 Productions of this thesis	18
1.4 Organisation of this thesis	19
2 Review of features and techniques used in speaker recognition	21
2.1 Features for speaker recognition	21
2.1.1 LPC (Linear Prediction Coefficient)	22
2.1.2 Linear prediction residual phase	23
2.1.3 HOCOR (Haar Octave Coefficients of Residue)	24
2.1.4 MFCC (Mel frequency cepstral coefficients)	27
2.1.5 MLSF (Mel Line Spectral Frequencies)	28
2.1.6 pH (Hurst Parameter Related Features)	29
2.1.7 MVDR (Minimum Variance Distortionless Response)	30
2.1.8 MFCC FrFT, MECB, DMECB (Features based on Fractional Fourier Transform)	32
2.1.9 Short-time energy and short-time average zero-crossing rate	34
2.1.10 Other features	35
2.2 Techniques for speaker recognition	36

2.2.1	Vector quantization (VQ)	37
2.2.2	Gaussian Mixture Model (GMM)	39
2.2.3	Hidden Markov Models (HMM)	42
2.2.4	Support Vector Machine (SVM)	45
2.3	Identification of research issues	49
3	Speaker change detection	51
3.1	Performance evaluation methods	52
3.2	Evaluation data	53
3.3	Feature of frames	53
3.4	Speaker changing points detection using BIC	53
3.4.1	Variance BIC value	55
3.4.2	Search for speaker changing points	56
3.4.3	Experimental results	59
3.5	Speaker segmentation using SVM	61
3.5.1	SVM feature generation	61
3.5.2	Labels of frames	62
3.5.3	Scaling SVM feature vectors	63
3.5.4	Peak tracking	64
3.5.5	Analysis of experimental results	64
3.6	Summary of speaker changing point detection	65
4	Testing features in speaker verification	66
4.1	Speaker verification based on GMM	67
4.2	Different features' performance in speaker verification	68
4.2.1	Performance of mel line spectral frequencies (MLSF)	68
4.2.2	Performance of hurst parameter related features (pH)	69
4.2.3	Performance of linear prediction residual phase	71
4.2.4	Features based on fractional Fourier transform	72
4.2.5	HOCOR and MVDR	75
4.2.6	Summary of results for studied features	77
4.2.7	Combination of features	78
4.3	Summary of feature testing in speaker verification	79

5	Conclusion and future of speaker recognition	82
5.1	Conclusion	82
5.2	Future of speaker recognition	85
	References	87

Abstract

Speaker recognition has recently become popular in biometric security applications because voice is easier to obtain and speaker recognition is easier to apply remotely on networks or telephones. Speaker recognition can be classified into speaker identification and speaker verification, as well as text-dependent and text-independent recognition.

Speaker changing point detection which is a key pre-requisite to speaker recognition is investigated in this thesis. Speaker changing point is defined as the moment one individual stops speaking and another starts speaking within a single continuous audio signal. Speaker changing point detection then is the process of identifying these points within a given audio signal. In this thesis, speaker changing point detection is investigated under the following conditions: there is no a-prior knowledge of the number of speakers or speakers' identities and it must be a fully automated detection process. First, Bayesian information criterion was used to detect speaker changing point. Different kinds of speech features were applied and compared in this system. Second, a method using support vector machine to combine different speech features for speaker changing point detection was developed. It is valuable to combine features and make use of the different speaker information they contain. Instead of just concatenating different long vector features, their Bayesian information criterion values were combined. The result shows that the proposed method improves the F -score performance which is calculated from Recall (RCL) and Precision (PRC) percentage provided by Ajmera, McCowan, and Boulard (2004) from a base-line of 0.6151 to 0.6414.

Speaker verification is used more frequently in security application because its performance is better in one to one matching when compared to speaker identification. Speech features are one of the key factors affect the performance of speaker verification. The performance of different speech features on speaker verification system

were investigated in this thesis. Different features' performance was compared under the same conditions. A Gaussian mixture model was used to build the speaker verification system for testing features' performance. In this thesis, the following acoustic features are tested: classic MFCC features (Gish & Schmidt, 1994), Mel Line Spectral Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst Parameter Related Features (pH) (Sant'Ana, Coelho, & Alcaim, 2006), Linear Prediction Residual Phase (rPhase) (Murty & Yegnanarayana, 2006), Haar Octave Coefficients of Residue (HO-COR) (Zheng & Ching, 2004), Minimum Variance Distortionless Response (MVDR) (Huang & Hansen, 2006), and features based on the fractional Fourier transform (FrFT) (Almeida, 1994) that include MFCC features calculated using FrFT and mean energy within critical bands (MECB). The performance of these features was compared to base line MFCC feature. In addition different features were combined to achieve better result than using MFCC features alone.

Chapter 1

Introduction

1.1 Background

Research in speaker recognition has been investigated by people for some four decades (Fernando & Jeffrey, 2001). Speaker recognition is the process of automatically recognizing who is speaking on the basis of information extracted from the speech signal. The process is not concerned with what the speaker is saying but who is speaking.

Speech recognition and speaker recognition have similar characteristic and challenges, they co-evolved together. Speaker recognition is different from speech recognition in that speaker recognition mainly considers automated methods of identifying a person or verifying the identity of a person based on a physiological or behavioral characteristic (Fernando & Jeffrey, 2001). The speaker recognition relies on features influenced by both the physical structure of an individual's vocal tract and the behavioral characteristics of the individual. Originally speaker recognition systems used the average output of several analog filters to perform matching (*Speaker recognition*, 2006). In the mid 1980s, the National Institute of Standards and Technology (NIST) develop the NIST speech Group to study and promote the use of speech processing technologies. Since 1996, NIST has been coordinating Speaker Recognition Evaluation.

Speaker recognition system can verify claimed identity or identify a speaker, and

can be text-dependent or text-independent. In the following sections, these two classifications are described further.

1.1.1 Speaker identification and speaker verification

Speaker identification and speaker verification are different according to the purpose for which they are used. In speaker identification, utterances from an unknown speaker are used to determine whether the speaker is one of a set of known speakers, while in speaker verification, utterances from a speaker are used to determine whether the speaker is who he/she claims to be.

Figure 1.1: Closed-set Speaker Identification

Figures 1.1 and 1.2 provide details of a sample speaker identification system. The input speech is pre-processed and features are extracted from the speech signal. The model or the reference template of each speaker is represented using these features. When the speech of the claimed speaker is input into the speaker identification system, features are extracted using the same feature extraction scheme and these features

Figure 1.2: Open-set Speaker Identification

are sent through each speaker model or template. The similarity of the input speech to each speaker model or template in the database is evaluated, and according to the evaluation results, the identification decision is made. Usually the speaker model or template which has the maximum similarity to the claimed speaker is selected.

There are closed-set speaker identification and open-set speaker identification system. In closed-set identification, training and testing are done within the database. Therefore the number of the different speakers is known, and all the speech utterances in a segment of speech to be analysed are from the speaker group in the database. In closed-set identification, the task is to match the unknown speech to one speaker in the database. Conversely, in the testing stage of open-set identification, testing is not done within the database as there may be an utterance from a speaker that is not existing in the database. A reference model for this unknown speaker may not exist in database, which means a new decision has to be made whether this utterance should be classified into the new group and marked as a new speaker. In this open-

set identification, a threshold is defined to decide whether a new speaker has been encountered and should be added into the database. The population of speakers may grow with time in the open-set identification.

Figure 1.3: Speaker Verification

Speaker verification is the process of accepting or rejecting the identity claim of a speaker. The main difference between speaker identification and speaker verification is the number of decisions that needs to be made. For speaker verification (Figure 1.3), there are only two decision alternatives: either to accept that the person is whom he/she claims to be or reject the claim. It is a one-to-one and not a one-to-many match. The input speech is processed to extract features for speaker verification. In order to make an identity verification of an unknown speaker, an utterance of the speaker is compared with the model for the speaker whose identity is claimed. A

decision threshold is needed in speaker verification. If the score of the match is above the decision threshold, the identity claim is accepted. In this case, a low threshold will make it easy for impostors to be accepted by the system (False Accept), but a high threshold is at the risk of rejecting the true customer (False Reject). In order to set the threshold at a desired level of true customer rejection and impostors acceptance, it is necessary to know the distribution of customer and impostors scores.

DET (Detection Error Tradeoff) curve is one of the effective ways to evaluate speaker-verification systems. DET curve (Figure 1.4) is a means of representing the performance of a classification task that involves a tradeoff between two error types: i.e., False Acceptance Rate and False Rejection Rate. The DET curve is preferred in speaker recognition systems to the ROC (Receiver Operating Characteristics) curve in which false alarm rate is plotted on the horizontal axis and the correct detection rate is plotted on the vertical axis (Martin et al., 1997). In the DET curve, the error rates are plotted on both axes. The effects on performance from both types of error are considered in the plot. By using a scale, it is possible to produce curves that are close to linear, which makes the tradeoff between false acceptance and false rejection clearly visible.

Although depending on the application, speaker identification and speaker verification are different, they share the same technologies. For example, feature extraction and classification techniques are common to both systems. However, speaker verification is used more frequently in security application because its performance is better in one to one matching. As an example, biometric security measures applied to door security or bank account verification are more focused on verifying a person's identity. Speaker identification has fewer application compared to verification. In this thesis, an examination of features which improves the speaker verification system is

Figure 1.4: An example of a DET curve

performed.

1.1.2 Text-dependent and text-independent speaker recognition

Speaker recognition can also be classified into text-dependent and text-independent speaker recognition. In text-dependent speaker recognition, a person's specific utterance or sentence must be exactly the same as the text utterance used in the training process. The recognition system then compares the similarity of the whole speech segment. In text-independent speaker recognition, the system does not rely on the specific text that is spoken but focuses on extracting features which uniquely identify a speaker.

The methods used for text-dependent speaker recognition are usually based on template matching techniques. The template matching approach is a technique in digital signal processing which is used for finding small parts of utterances which match a template utterance. For example, dynamic time warping (DTW) (Shahin & Botros, 1998) is a commonly used template matching algorithm. Hidden Markov Modelling (HMM) is also used in text-dependent recognition as an extension of the

DTW-based methods and have achieved better recognition accuracies than DTW, because it can efficiently model statistical variation in spectral features (Melin & Castillo, 2005).

Without the requirement to input specific text into the recognition system, text-independent speaker recognition is more convenient to apply. For example, the speaker recognition process can be applied at any time without customers having to be cooperative and offer a specific utterance or be aware that speaker verification is in progress. Different challenges are brought in text-independent speaker recognition, in that there is no fixed text and no hint to predict the speech that a particular customer is going to speak. In this thesis, only text-independent speaker verification is considered.

For text-independent verification, methods that are based on higher level speech characteristics have to be investigated. Examples of higher level speech characteristics are: prosody, semantics and pronunciation. Higher level characteristics can possibly be combined with the underlying low level spectral information to improve the performance of text-independent speaker recognition systems (Reynolds et al., 2003). In the early stage, long-term statistics-based analysis is used in text-independent speaker recognition. The long-term statistics-based method uses long-term sample statistics of various spectral features, such as mean and variance of spectral features over a series of utterances. The disadvantage of long-term spectral averages is it lacks discrimination ability and is sensitive to the channel effect. Model based methods are popular in text-independent speaker recognition recently, such as Vector quantization (VQ). VQ codebooks consist of a small number of representative feature vectors which are used as an efficient means of characterizing speaker-specific features. More techniques for text-independent speaker recognition are being investigated such as HMM, GMM and neuro networks.

1.2 Motivation and problems of speaker recognition

Speaker recognition is useful in application for forensic and security purposes (Kunzel, 1994). In banking transactions over the telephone or other networks, speaker verification can help to determine whether it is the authorised person asking for the transaction service. Other applications using secure access controlled by voice also find speaker recognition quite helpful, such as meeting or teleconference monitoring, receiving and browsing, voice mail or customizing service by voice. Compared to other biometrics such as finger prints and iris scanning, speaker recognition is easier to be accepted by customers, because special action is not needed to collect data for recognition and it is applicable remotely over telephone lines.

Speaker recognition and speech recognition share a number of technologies (refers to section 1.1), although the purposes of speaker recognition and speech recognition are different. Speaker-recognition systems pay more attention to “who is speaking”, while speech recognition systems are more interested in identifying “what is being said”, regardless of the identity of the speaker. Despite this difference, it is still possible to achieve improved performance by sharing newly developed techniques. For example, speaker-independent speech recognition methods may ignore the characteristic of each speaker, therefore many useful information about the speaker are lost. As a result, they still cannot achieve the same level of performance as the speaker-dependent speech recognition systems. In this case, if the speaker adaptation techniques are introduced into the speech recognition system (Furui et al., 1991), they will narrow down the search-space of the feature parameter, accelerate the recognition speed, and improve the accuracy of the system. In the other case, such as text-prompted speaker recognition, it is very helpful to create speaker specific phoneme models that include

enough information related to each phoneme and to each speaker. The performance of both speech and speaker recognition can be simultaneously improved by automatically adjusting speaker-independent phoneme models to each new speaker. Consequently the techniques of speaker recognition are investigated further and will also become useful in speech recognition algorithm.

The performance of speaker recognition systems today still needs to be improved by solving some existing problems, such as in-channel noise, distortion due to microphones, lack of robust speech features and so on. This thesis concentrates on two tasks necessary in speaker recognition: one is to investigate speaker segmentation, the other is to investigate speech features in text-independent speaker verification.

1.2.1 Speaker segmentation

The process of speaker segmentation aims to find acoustic events within an audio stream, for example finding the speaker changing point in the continuous speech files according to different speakers' characteristics. Automatic segmentation and classification of an audio stream according to speaker identities and environmental conditions are gaining increasing importance as we get more and more information from TV programs and movies (Chen & Gopalakrishnan, 1998). Since some speech files are obtained from a telephone conversation or recorded from a meeting, usually they contain more than one person speaking, music and other sounds mixed with the audio recording. In this case, it is necessary to perform speaker segmentation automatically to obtain separate speech streams where each contains only one speaker's speech, before performing speaker recognition.

Depending on the source of the audio, the speaker segmentation can be divided into multi-channel or mono-channel speaker segmentation. In multi-channel speaker segmentation, each speaker is supposed to speak in a separate microphone during the

meeting. Therefore the information from the channel can be used to separate speakers. Multi-channel recordings of meetings provide information on speaker locations in the timing and level differences between microphones (Ellis & Liu, 2005). Some methods that use this channel delay information to separate the speakers were proposed by Lathoud and McCowan (2003). Mono-channel speaker segmentation can only use the information from one channel. Therefore it is a more challenging problem.

In previous work on the automatic speaker segmentation problem, it is assumed that the number of the speakers or even the names of the speakers in the data base are known. This assumption makes the segmentation process easier. A report on unsupervised speaker identification (Sugiyama et al., 1993) presented a study of a simple case in which the number of the speakers to be clustered was known. Another example is the study by Wilcox et al. (1994) where an agglomerative clustering method was used to solve the problem with unknown information but the number of the speakers was limited to two. However the problems in speaker segmentation become more challenging when the number of speakers in a conversation is unknown, and the speakers speak simultaneously (André et al., 2002). It is more useful in application because the number of the speakers are automatically detected.

Another problem in speaker segmentation systems is the need for real-time processing. Previous efforts to solve the problem of unsupervised speaker clustering consist of clustering audio segments into homogeneous clusters according to speaker identity, background conditions or channel conditions. Most of these models are based on VQ (Mori & Nakagawa, 2001), GMM (Gaussian mixture model) (Wu et al., 2003) and the HMM model (Cohen & Lapidus, 1996). A drawback of these models is that iterative operations are unavoidable, which makes these algorithms very time consuming. Therefore these models are extremely difficult to process in real time without

expensive hardware (Lu & Zhang, 2002). The real-time system will need a more reliable algorithm that requires less calculation and a system that can be trained while applying segmentation.

Furthermore, the noise problem persists (e.g. to classify the speech with background music or a singing voice with music). Many methods claim they can achieve good results, but they still encounter problems in the noisy environment (Kwon & Narayanan, 2002).

The thesis focuses on finding speaker changing point. The experiment was conducted under the following assumptions:

- No two speakers are speaking simultaneously in the audio files;
- It is a mono-channel speaker segmentation;
- No prior knowledge of speakers is known, such as the number of speakers and names of the speakers;
- The noise problem is not considered, which means the testing database contains clean speech and there is no background music or non-speech sounds.

1.2.2 Features for speaker recognition

Speaker samples are usually presented as waveforms with time on the horizontal axis and loudness on the vertical access (*Speaker recognition*, 2006). Usually the speaker recognition systems do not analyse the waveforms directly because they are changing in the time scale. Instead, the speaker recognition systems analyse the frequency content of the speech and compare characteristics such as quality, duration, intensity, dynamics and pitch of the signal. These can be called speech features. Speech features are the transformations of speech waveforms and they can distinguish the difference in the acoustic properties of the speech waveforms.

Speech can be analysed at several different levels such as semantic, linguistic or

acoustic level. For speaker recognition the efficient features that capture the speaker characteristic are of more interest. The differences between speakers are the result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits of different individuals (Campbell, 1997). The most commonly used features are the LPC (linear prediction coefficients) and the MFCC (mel frequency cepstral coefficients). These two kinds of feature are also used in speech recognition, which means they contain not only speaker information but also speech information, which may disturb the recognition system and make accuracy of speaker recognition drop. Therefore more research is needed into speech features to extract robust speaker-related feature for speaker recognition. In cases when noise background exists, many speaker features become unreliable. The robust features, which are less affected by noise, also need to be investigated.

Besides generating new features, feature combination is another way to improve the performance of the speaker recognition system. Different features may contain different information related to the speaker identity. Combining those features may result in performance improvement if the features compliment each other.

In this thesis, to determine the efficient features, different kinds of speech features have been investigated under the GMM speaker verification system. The performance is compared to the baseline MFCC features. In addition, a combination of the decision results from different kinds of features is performed to achieve improvement in speaker verification.

1.3 Productions of this thesis

The products of this thesis are the following:

- Graham Leedham, Vladimir Pervouchine and Haishan Zhong-Seeking patterns

in the forensic analysis of handwriting and speech, Chapter in Pattern Recognition Technologies and Applications: Recent Advances (Eds. Brijesh Verma and Michael Blumenstein), In press 2007.

- Haishan Zhong, Vladimir Pervouchine, Graham Leedham, David Cho, “Combining novel acoustic features using SVM to detect speaker changing points,” International Conference on Bio-inspired Systems and Signal Processing, Portugal, 28-31 January 2008.
- Vladimir Pervouchine, Haishan Zhong, Graham Leedham, David Cho and Haizhou Li, “Comparative study of several novel acoustic features for speaker recognition,” International Conference on Bio-inspired Systems and Signal Processing, Portugal, 28-31 January 2008.

1.4 Organisation of this thesis

The thesis concentrates on speaker segmentation and testing the performance of different kinds of features in speaker recognition systems. The thesis is organised in the following chapters.

Chapter 2 is the literature review. It is a survey of features and methods in the research of speaker recognition. In the beginning of this chapter, different kinds of features in speaker recognition are introduced. In each section the introduction of a feature, followed by the usage and extraction process are presented. There are 10 features used here in speaker recognition which are investigated. Later, some of the other methods used in speaker recognition systems are investigated. Some of the methods are used in the experiments in the later chapters.

Chapter 3 describes the speaker segmentation algorithm. In this chapter, the segmentation algorithm using BIC is presented first. Based on this, different kinds of speech features are tested and the performances are compared to each other. Later,

the speaker segmentation system using SVM is introduced. The proposed system combines different kinds of speech features based on the BIC speaker segmentation algorithm. The performance of the SVM speaker segmentation algorithm is improved compared to the BIC speaker segmentation algorithm.

Chapter 4 describes the testing performance of different features in a speaker verification system. The system and database are introduced first. The result of each feature is analysed separately later. In the final section, the combinations of features are investigated to improve the performance of the system.

The final chapter is the conclusion. The conclusions of the experiments on speaker segmentation and the experiment of testing features in speaker verification are made. The research contribution of the thesis is presented. Finally recommendations for future work in research of speaker recognition are proposed.

Chapter 2

Review of features and techniques used in speaker recognition

Speaker recognition is a biometric modality that uses an individual's voice for recognition purpose. The overall performance of speaker recognition systems depend on a number of sub-components. Feature extraction is one of the important components in speaker recognition. In the first part of the literature review, different kinds of features used in speaker recognition are investigated. In the second part of this chapter, the techniques that are frequently used in speaker recognition are introduced.

2.1 Features for speaker recognition

A straight forward way of representing speech is the time-amplitude waveform which is constantly changing while people are speaking. Speech features extracted from the speech waveform represent the characteristic of the speech and speaker. The speaker recognition system is to classifying individual speaker characteristics in the feature space.

There are different levels of features: acoustic features such as cepstra, prosodic features such as pitch and energy, phonetic features such as phone binary trees and phone N-grams and lexical features (Reynolds et al., 2003). Among these features, acoustic features are most commonly used. Acoustic features are based on the spectrogram. The speech spectrum has been shown to be very effective for speaker iden-

tification (Campbell, 1997). Because the spectrum reflects a person's vocal tract structure, the predominant physiological factor which distinguishes one person's voice from others, such as LPC, MFCC, LSP (Line spectral pairs).

There are no distinguishing speaker features discovered so far. It is possible to improve the accuracy of the speaker recognition system by studying efficient features. In the following section, acoustic features that are mostly used in recent years are discussed. They are Mel Line Spectral Frequencies (MLSF), Hurst Parameter Related Features (pH), Linear Prediction Residual Phase, Haar Octave Coefficients of Residue (HOCOR), Minimum Variance Distortionless Response (MVDR), features based on Fractional Fourier Transform (MFCC FrFT, MECB, DMECB). The prosodic features that are discussed are short-time energy and short-time average zero-crossing rate.

To extract short-term spectral features, the speech file is segmented into speech frames of about 20-30 ms, so that the speech signal can be treated as approximately stationary within each frame. The frame size depends on the feature extraction scheme. For example, the frame size in MFCC feature extraction is usually 30 milliseconds. The feature extraction is applied on each frame. Therefore the features for speaker recognition are $m \times n$ vectors, where m is the number of frames in the speech file, and n is the length of feature in each frame.

2.1.1 LPC (Linear Prediction Coefficient)

The human vocal tract, which consists of the pharynx, the mouth and nose cavities, works like a musical instrument to produce a sound. Different vocal tract shapes would generate a different sound. Linear Prediction is used to model the vocal tract as an infinite impulse response (IIR) that produces the speech signal. Assume that the present sample of the speech is predicted by the past p samples of the speech as

follows:

$$\hat{s}(n) = - \sum_{k=1}^p a_k s(n-k) \quad (2.1)$$

where $\hat{s}(n)$ is the prediction of $s(n)$, $s(n-k)$ is the k -th step previous sample, and a_k are called the linear prediction coefficients. The error between the actual sample and the predicted one can be expressed as

$$r(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^p a_k s(n-k) \quad (2.2)$$

The sum of the square error to be minimized is:

$$E = \sum_n r^2(n) \quad (2.3)$$

To estimate the linear prediction coefficients a_k , usually the Levinson-Durbin algorithm (Brillinger, 2001) is used. It is an iterative method which uses autocorrelation sequence to calculate the linear prediction coefficients.

Linear prediction is of major importance in many speech processing applications, such as spectral peak estimation, fundamental frequency estimation.

2.1.2 Linear prediction residual phase

Linear prediction (LP) filtering can be used to approximate the vocal tract by a channel with different cross-sections. Thus, linear prediction coefficients (LPC) are speaker-specific features to characterize the vocal tract of a person (see section 2.1.1). There are other features which are derived from LPC, such as reflection coefficients (RC), log-area ratios (LAR), Line spectral pairs (LSP) frequencies. Linear prediction residual phase is generated based on the linear prediction error (Islam & Kabal, 2000).

If the vocal-tract-transfer function of the speech production model can be characterized by the predictive coefficients, the prediction error (Eq.2.2), referred to as the LP residue, contains mostly information about the excitation source (Murty & Yeg-

nanarayana, 2006). Thus the LP residue can be used to derive the source information that has additional information useful for speaker recognition.

One way to extract the speaker-related information of the linear prediction residual is to extract the phase information. The residual signal is $r(n)$ (Eq.2.2), then the analytic signal is given by:

$$r_a(n) = r(n) + jr_h(n) \quad (2.4)$$

where $r_h(n)$ is the Hilbert transform of $r(n)$ and is given by

$$r_h(n) = F^{-1}[R_h(\omega)] \quad (2.5)$$

$$R_h(\omega) = \begin{cases} -jR(\omega), & 0 \leq \omega < \pi \\ jR(\omega), & 0 > \omega \geq -\pi \end{cases} \quad (2.6)$$

Here $R(\omega) = F[r(n)]$ is the Fourier transform of $r(n)$. The magnitude of the analytic signal $r_a(n)$ is given by

$$h_e(n) = |r_a(n)| = \sqrt{r^2(n) + r_h^2(n)} \quad (2.7)$$

The cosine of the phase of the analytic signal $r_a(n)$ is given by

$$\cos(\theta(n)) = \frac{\text{Re}(r_a(n))}{|r_a(n)|} = \frac{r(n)}{h_e(n)} \quad (2.8)$$

The phase can be calculated from short segments of speech of around 5 ms duration, which is approximately the period of the bursts in the excitation source. Figure 2.1 shows the feature extraction process.

2.1.3 HOCOR (Haar Octave Coefficients of Residue)

Another way of extracting the speaker-related information from the LP residue is to capture burst information using the wavelet decomposition of the residual signal (Campbell, 1997; André et al., 2002; Soong & Rosenberg, 1986). HOCOR (Haar Octave Coefficients of Residue) is obtained by applying Haar Transformed to the LP residue.

Figure 2.1: Extraction of residual phase

The discrete Haar transform of a signal and its inverse transform are formulated as

$$\begin{cases} X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)H(k, n), & k = 0, 1, \dots, N-1 \\ x(n) = \sum_{k=0}^{N-1} X(k)H(k, n), & n = 0, 1, \dots, N-1 \end{cases} \quad (2.9)$$

The $H(k, n)$ is called the Haar function and is a completely orthogonal function set of rectangular waveforms

$$H(0, n) = 1, 0 \leq n \leq N-1 \quad (2.10)$$

$$H(k, n) = H(2^{i-1} + j - 1, n) = \begin{cases} \sqrt{2^{i-1}}, & \frac{j-1}{2^{i-1}}N \leq n \leq \frac{j-1/2}{2^{i-1}}N \\ -\sqrt{2^{i-1}}, & \frac{j-1/2}{2^{i-1}}N \leq n \leq \frac{j}{2^{i-1}}N, \\ 0, & \text{elsewhere} \end{cases} \quad (2.11)$$

where $i = 1, 2, \dots$ denotes an octave subset having a zero-crossing in a given width $\frac{N}{2^{i-1}}$, and $j = 1, 2, \dots, 2^{i-1}$ gives the position of the function within this subset. The Haar spectrum is defined as

$$G(k) = |X(k)|, k = 0, 1, \dots, N-1 \quad (2.12)$$

Then the generation of the HOCOR of order α is following:

$$H_i^\alpha = \{H_i^\alpha(j) | j = 0, 1, \dots, 2^\alpha - 1\}, (\alpha \leq i - 1) \quad (2.13)$$

where

$$H_i^\alpha(j) = \left\{ G(k) \left| \begin{array}{l} k = 2^{i-1} + j2^{i-1-\alpha}, \dots, \\ 2^{i-1} + (j+1)2^{i-1-\alpha} - 1 \end{array} \right. \right\} \quad (2.14)$$

And the α th-order HOCOR is given by

$$HOCOR_\alpha = \left\{ \sum_{G(k) \in H_i^\alpha(j)} G^2(k) \left| \begin{array}{l} i = 3, \dots, \log_2 N \\ \hat{\alpha} = \min(i - 1, \alpha) \\ j = 0, \dots, 2^{\hat{\alpha}} - 1 \end{array} \right. \right\} \quad (2.15)$$

Figure 2.2 shows the diagram of HOCOR feature extraction. The first difference of HOCOR can also be used as additional features for speaker recognition.

Figure 2.2: HOCOR feature extraction

The properties of HOCOR is the following (Zheng & Ching, 2004):

- Because the LP residue is orthogonal to the vocal tract system, HOCOR is uncorrelated to the LPC.
- The rectangular base function and the time-frequency properties of the Haar transform result in better spectral decomposition of the residual signal with burst.
- $HOCOR_\alpha$ with $\alpha > 0$ represents pitch and harmonics as well as their phase information of the vocal source, which will be useful for speaker recognition.

2.1.4 MFCC (Mel frequency cepstral coefficients)

MFCC is a popular acoustic feature for the speech research and it has been demonstrated to work well for speaker recognition and also speech recognition. Unlike LPC, Linear Prediction analysis is not used in MFCC features. The generation process is not as complicated (Gish & Schmidt, 1994).

The generation of MFCC features is based on frame level analysis. The first step is to perform fast Fourier transform (FFT) of the speech frame. The second step is to take the magnitude. The third step is to warp the frequencies according to the mel scale. Mel scale is based on the nonlinear human perception of the frequency of the sounds. Therefore the warping transforms the frequency scale to place less emphasis on high frequencies. In this step by setting the mel scale window number, the number of MFCC features is decided. The fourth step is to take the logarithmic value of the mel scale result. The final step is to take the inverse FFT. Finally the MFCC feature vectors for each frame are obtained.

Figure 2.1.4 shows an example of mel transform. In this example the number of MFCC features is 40. Usually 13, 26, 39 MFCC are chosen.

$\Delta MFCC$ or $\Delta\Delta MFCC$ are also generated as the first and second derivative (calculated as difference) from MFCC features. The $\Delta MFCC$ and $\Delta\Delta MFCC$ are used because they capture short-term speech dynamics (interval of 50-100 ms). It is thought that short-term speech dynamics can additionally characterize a person's vocal tract (Oppenheim & Schaffer, 2004).

There are some parameters that can be varied during the generation of MFCC features:

- Window type, length and overlap to frame the speech wave.
- The number of FFT points for the MFCC.

Figure 2.3: Example of Mel transform

- The number of Mel-scaling windows
- Upper frequency

2.1.5 MLSF (Mel Line Spectral Frequencies)

Line Spectral Frequencies (Ribeiro & Trancoso, 1999) generated from LPC are used in speaker adaptation on the context of phonetic coding. They are used in speaker recognition and results show that they contain signification speaker information.

Mel line spectral frequencies (MLSF) are generated from Line Spectral Frequencies. However, the idea is to take advantage of mel frequency warping and emphasise the information in lower frequencies (Cordeiro & Ribeiro, 2006). To overcome the drawback of line spectral frequencies features in that they do not take advantage of the properties of the human ear, such as using mel filter bank to reduce the information in high frequencies, MLSF are computed from the mel-spectrum energies.

The diagram in Figure 2.4 shows the sequence of procedures to extract MLSF

features.

Figure 2.4: MLSF feature extraction

Similar to differences between consecutive LSF coefficients, the differences between the MLSF features contain the speaker-specific information. Therefore the differential mel-LSF (DMLSF) features are generated as the extended features for MLSF features.

2.1.6 pH (Hurst Parameter Related Features)

The physiological features are not robust to the channels acoustic distortion because they model the spectral characteristic of the human vocal mechanism. The statistical feature pH is a vector of Hurst parameters obtained from the windowed short-time segments of speech (Sant'Ana et al., 2006). It expresses the time-dependence of a stochastic process. Since it models the stochastic behavior of the speech signal, it is robust to channel distortion. Because it does not relate to the transfer function of the vocal tract, the extraction methods of pH are less complex and pH can be extracted in real time.

The Hurst parameter is defined by the decaying rate of the autocorrelation coefficient function (ACF): $\rho(k)$ ($-1 < \rho(k) < 1$) as $k \rightarrow \infty$. If the speech signal is $X(t)$

with the finite variance, the autocorrelation coefficient is: $\rho(k) = \frac{Cov[X(t), X(t+k)]}{Var[X(t)]}$ where the $\rho(k)$ belongs to $[-1, 1]$ and $\lim_{k \rightarrow \infty} \rho(k) = 0$.

The asymptotic behaviour of $\rho(k)$ is given by

$$\rho(k) \approx H(2H - 1)K^{2(H-2)} \quad (2.16)$$

To analyse the speech time-dependence, accurate methods are required to estimate H . Several schemes are available for the estimation of the Hurst parameter. The one used in the current work uses wavelet based AV (Abry-Veith) estimator (Veith & Abry, 1998). The AV estimator is calculated from the discrete wavelet decomposition of a signal. The procedure of AV estimation is as follow:

1. Division of the speech signal into frames.
2. Wavelet decomposition of a frame: the discrete wavelet transform (DWT) is applied to the sample data generating the detail coefficients $d(j, k)$, where j is the scale and k is the coefficient index.
3. Variance estimation of the detail coefficients: for each scale j , the variance is $\sigma_j^2 = (\frac{1}{n_j}) \sum_k d(j, k)^2$ is evaluated, where n_j is the number of available coefficients for each scale j .
4. Hurst parameter estimation: use weighted linear regression to get the slope of α the curve $y_j = \log_2(\sigma_j^2)$ versus j . The parameter H is estimated as $(1 + \alpha)/2$.
5. Estimation of Hurst parameter for details of the signal $d(j, k)$.
6. All estimated parameters together give the pH feature vector for the frame.

2.1.7 MVDR (Minimum Variance Distortionless Response)

MVDR features were reported to be useful for speaker segmentation (Huang & Hansen, 2006) as well as for in-vehicle speech recognition (Yapanel & Hansen, 2003). High-order minimum variance distortionless response (MVDR) models provide better

upper envelope representations of the short-term speech spectrum than MFCC (Yapanel & Hansen, 2003). Furthermore, it has been shown that the MVDR spectrum can be simply obtained from a non-iterative computation of the linear prediction (LP) coefficients.

In the MVDR spectrum estimation method, the signal power at a frequency, ω_i is determined by filtering the signal by a specially designed FIR filter, $h(n)$, and measuring the power at its output. The FIR filter, $h(n)$, is designed to minimize its output power subject to the constraint that its response at the frequency of interest, ω_i , has unity gain. The Q^{th} order MVDR spectrum can be parametrically written as

$$P_{MV}(\omega) = \frac{1}{\sum_{k=-Q}^Q \mu(k)e^{-j\omega k}} = \frac{1}{|B(e^{j\omega})|^2} \quad (2.17)$$

The parameters $\mu(k)$ can be obtained from a modest non-iterative computation using the LP coefficients a_k and prediction error variance P_ϵ

$$\mu(k) = \begin{cases} \frac{1}{P_\epsilon} \sum_{i=0}^{Q-k} (Q+1-k-2i)a_i a_{i+k}, & k : 0, \dots, Q \\ \mu(-k), & k : -Q, \dots, -1 \end{cases} \quad (2.18)$$

The MVDR extraction algorithm is as follow:

1. Obtain the perceptually warped FFT power spectrum.
2. Compute the “perceptual autocorrelations” by utilizing the inverse FFT on the bark-warped power spectrum.
3. Perform a Q -th order LP analysis via Levinson-Durbin recursion using perceptual autocorrelation lags.
4. Calculate the Q -th order MVDR spectrum from the LP coefficients.
5. Obtain the final cepstrum coefficients using the straightforward FFT-based approach.

The block diagram of the MVDR feature extraction process is shown in Figure 2.5. An important trait of PMVDR is that it does not require an explicit filter bank

Figure 2.5: MVDR extraction

analysis of the speech signal. For the application of speaker segmentation, the order of the LP model is increased to reflect more speaker dependent information in the features. A detailed Bark frequency warping is also applied for better results.

2.1.8 MFCC FrFT, MECB, DMECB (Features based on Fractional Fourier Transform)

The fractional Fourier transform is a generalization of the Fourier transform. If the conventional Fourier transform is F^n , where n can only be an integer ($n = 1$ for the direct transform, $n = -1$ for the inverse transform, etc.), the fractional Fourier transform can be thought as F^p , where p is a real number. Thus, it is said to transform a function to an intermediate domain between time and frequency (Almeida, 1994).

The fractional Fourier transform can be given as:

$$\begin{aligned}
 F^a[s(x_1)] &= s(x) \\
 &= \frac{\exp(i(\frac{\pi}{4} - \frac{\pi}{2}))}{\sqrt{2\pi \sin \alpha}} \exp\left(-\frac{i}{2}x^2 \cot \alpha\right) \int_{-\infty}^{\infty} \exp\left(-\frac{i}{2}x_1^2 \cot \alpha - \frac{ix_1 x}{\sin \alpha}\right) s(x_1) dx_1
 \end{aligned} \tag{2.19}$$

and the inverse fractional Fourier transform can be given as:

$$F^{-a}[s(x_1)] = \frac{\exp - i(\frac{\pi}{4} - \frac{\pi}{2})}{\sqrt{2\pi \sin \alpha}} \exp \left(+ \frac{i}{2} x^2 \cot \alpha \right) \int_{-\infty}^{\infty} \exp \left(+ \frac{i}{2} x_1^2 \cot \alpha - \frac{i x_1 x}{\sin \alpha} \right) s(x_1) dx_1 \quad (2.20)$$

where $\alpha = a\pi/2$

When a is different, the cases are different:

1. When $\alpha = \pi/2$, $a = 1$

$$F^a[s(x_1)] = F^1[s(x_1)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(x_1) \exp(-ix_1 x) dx_1 \quad (2.21)$$

is the ordinary Fourier transform.

2. When $\alpha = 0$, $a = 0$, the transform kernel reduces to identity operation:

$$\lim_{\epsilon \rightarrow 0} \frac{1}{\sqrt{i\pi\epsilon}} \exp(-\frac{x^2}{i\epsilon}) = \delta(x) \quad (2.22)$$

so we have,

$$F^0[s(x_1)] = \int_{-\infty}^{\infty} \delta(x - x_1) s(x_1) dx_1 = s(x_1) \quad (2.23)$$

A similar procedure can be applied to the case.

3. When $\alpha = \pi$, $a = 2$ and the result turns out to be

$$F^2[s(x_1)] = \int_{-\infty}^{\infty} \delta(x + x_1) s(x_1) dx_1 = s(-x_1) \quad (2.24)$$

So for an angle from 0 to π , the values of a are from 0 to 4.

MFCC fractional Fourier transform features based on the fractional Fourier transform of order p , denoted as $MFCC_p$, were extracted just as conventional MFCC features ($MFCC_{1,0}$) with the difference that the fractional Fourier transform of order p was used.

MECB features (Wang & Wang, 2005) were calculated as

$$E_i = \frac{\int_{f_{i-1}}^{f_i} |F^p(f)|^2 df}{f_i - f_{i-1}} \quad (2.25)$$

where f_i , $i = 0 \dots N$ define frequency bands. The frequency range from $f_0 = 0\text{Hz}$ to $f_N = 4\text{kHz}$ was warped according to Bark scale. This equation gave $MECB_p$ features.

Differential $MECB$ and $MFCC$ ($DMECB_{p1-p2}$ and $DMFCC_{p1-p2}$) were calculated as

$$DMECB_{p1-p2} = MECB_{p1} - MECB_{p2} \quad (2.26)$$

and

$$DMFCC_{p1-p2} = MFCC_{p1} - MFCC_{p2} \quad (2.27)$$

Since conventional MFCC works well in speaker recognition, it makes sense to try MFCC fractional Fourier transform features in speaker recognition as well.

2.1.9 Short-time energy and short-time average zero-crossing rate

Short-time energy provides a convenient representation of the amplitude variation over time (Zhang & Kuo, 2001). The short-time energy function of an audio signal is defined as:

$$E_n = \frac{1}{N} \sum_m [x(m)w(n-m)]^2 \quad (2.28)$$

where $x(m)$ is the discrete time audio signal; n is the time index of the short time energy; $w(m)$ is the rectangle window of the length. The major reasons for using the short-time energy feature in speaker recognition are:

1. For speech signals, it provides a basis for distinguishing voiced speech components from unvoiced speech components, because values of E_n for the unvoiced components are, in general, significantly smaller than those of the voiced components, as can be seen from the peaks and troughs in the energy curve.
2. It can be used as the measurement to distinguish audible sounds from silence when the SNR is high.

3. Its change pattern over time may reveal the rhythm and periodicity properties of sound.

The short-time average zero-crossing rate (ZCR) can be used as another measure to distinguish between voiced and unvoiced speech signals (Lu & Zhang, 2002; Zhang & Kuo, 2001), because unvoiced speech components normally have much higher ZCR values than voiced ones. The speech ZCR curve has peaks and troughs from unvoiced and voiced components, respectively. Note also that the ZCR curve has a relatively low and stable baseline with high peaks above it.

For discrete-time signals, a zero-crossing is said to occur if successive samples have different signs. The rate at which zero crossings occur is a simple measure of the frequency content of a signal. The short-time average zero-crossing rate is defined as:

$$Z_n = \frac{1}{2} \sum_m |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| w(n-m) \quad (2.29)$$

$$\text{where } \text{sgn}[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) < 0 \end{cases}$$

and $w(n)$ is a rectangle window.

2.1.10 Other features

Besides the features described in the previous sections, others features have been investigated, such as short-time fundamental frequency (Zhang & Kuo, 2001) which is calculated based on the peak detection from the spectrum of sound. The problem of short-time fundamental frequency is it can change while the same person pronounces different sounds. Spectral Peak Track (Zhang & Kuo, 2001; Keum & Lee, 2005) is the feature that reveals the characteristics of the type of sound. For example, sounds from musical instruments normally have spectral peak tracks which remain at the same frequency level and last for a certain period of time. The parameters that are generated from GMM can also be used as features, which can be input into a classifier

such as SVM. More novel and efficient features are still needed to be investigated because there are no universally distinguishing features discovered so far.

The thesis here mainly concentrates on the acoustic features. In Chapter 4, Mel Line Spectral Frequencies (MLSF), Hurst Parameter Related Features (pH), Linear Prediction Residual Phase, Haar Octave Coefficients of Residue (HOCOR), Minimum Variance Distortionless Response (MVDR), Features based on Fractional Fourier Transform (MFCC FrFT, MECB, DMECB) are investigated further for use in a speaker verification system.

2.2 Techniques for speaker recognition

The techniques for speaker recognition can be categorized into three major approaches. The first approach is to use long-term averages of acoustic features such as spectrum representations or pitch. It is also the earliest way used to extract information from a speech waveform. Through averaging out other factors influencing the acoustic features, such as the phonetic variations, the speaker dependent component remains. For spectral features, the long-term average represents a speaker's average vocal tract shape. However the averaging process lowers the speaker-dependent information. The stable long-term statistics can only be derived from long sections of speech audio.

The second and mostly used approach is to build a model of each speaker using the speaker-dependent features. By comparing acoustic features extracted from phonetic sounds in a test utterance with speaker-dependent acoustic features extracted from similar phonetic sounds, the comparison measures speaker differences rather than textual difference.

There are different methods of building probabilistic models for speakers. One of them is Vector Quantization (VQ) (Soong et al., 1985). For the VQ, each speaker is represented by a codebook of spectral templates representing the phonetic sound

clustered in speech. But the VQ is limited in its ability to model the possible variability encountered in an unconstrained speech task although this technique has good performance on limited vocabulary tasks. HMM's is another popular technique used as probabilistic speaker models for speaker recognition (Poritz, 1982; Matsui & Furui, 1994). The HMM models build not only the models of the underlying speech sounds, but also the temporal sequencing among these sounds. Temporal structure modelling is advantageous for text-dependent tasks, so HMM is used both in text-dependent and text-independent speaker recognition. The Gaussian mixture speaker model (Reynolds & Rose, 1995) is also used in speaker recognition. GMM is trying to build a probabilistic model of the underlying sounds of a person's voice, it does not impose any Markovian constraints between the sound classes. After well-trained models of speakers are obtained, the models can be implemented on a real-time speaker recognition application.

The third approach to speaker recognition is to use discriminate methods, such as neural networks (NN), support vector machine (SVM). SVM is a binary classifier that constructs a decision boundary to separate the two classes. SVM has gained much attention since the experimental results indicate that it can achieve a generalisation performance comparable to other classifiers on smaller training data (Wan & Campbell, 2000). Speaker verification can be treated as a binary decision task: the classifier must decide whether or not a speaker is the one he claims to be. The speaker identification can be an extension of the verification task. By constructing n SVM classifiers for n speakers, SVM can be used in speaker identification.

2.2.1 Vector quantization (VQ)

The idea of Vector Quantization is to generate a speaker-based VQ codebook that approximates the distribution of the discrete set of feature vectors (Soong et al.,

1985). Based on the features vectors $a_1, a_2, \dots, a_I$ which characterize the variability of speaker, a partitioning of the feature vector space S , which is $[S_1, S_2, \dots, S_M]$, have to be constructed. For a particular speaker space S , the whole feature space is represented as $S = S_1 \cup S_2 \cup \dots \cup S_M$. Each partition, S_i , forms a convex, non-overlapping region and every vector inside S_i is represented by the corresponding centroid vector, b_i of S_i . The partitioning is done in a way to minimize the average distortion over the whole training set. The average distortion is:

$$D = \frac{1}{I} \sum_{i=1}^I \min_{1 \leq j \leq M} d(a_i, b_j) \quad (2.30)$$

where $d(a_i, b_j)$ is the distortion distance between the vectors a_i and b_j .

For example if the VQ uses the LPC features, the distortion measure is the LPC likelihood ratio distortion measure between two LPC vectors a and b as follow:

$$d_{LR}(a, b) = \frac{b^T R_a b}{a^T R_a a} - 1 \quad (2.31)$$

where R_a is the autocorrelation matrix of the speech input data associated with vector a . The VQ codebook training algorithm proposed by Linde et al. (1980) can be used to train the VQ codebook.

Figure 2.6 illustrates VQ speaker recognition.

In the training stage, a speaker-specific codebook is generated by clustering the training feature vectors of each speaker. In the recognition stage, an input utterance is vector-quantized using the codebook of each reference speaker. And the VQ distortion which is accumulated over the entire input utterance is used to make the final recognition decision. The average distortion with respect to the i th codebook of speaker is:

$$D^i = \frac{1}{L} \sum_{l=1}^L \min_{1 \leq j \leq M} d(a_l, b_j^i) \quad (2.32)$$

Figure 2.6: Diagram of VQ speaker recognition

The N average distortions are compared to find the minimum. The final speaker recognition decision is given by:

$$i^* = \operatorname{argmin}_{1 \leq i \leq N} D^i \quad (2.33)$$

The study on the VQ-based speaker recognition reported that it is simple (Soong et al., 1985) and it is robust against utterance variations even when only a short utterance is available (Matsui & Furui, 1994).

2.2.2 Gaussian Mixture Model (GMM)

GMM is a method to approximate the probability density of each speaker features. A Gaussian mixture density is a weighted sum of M component densities (Douglas et

al., 2000):

$$p(\vec{x}|\lambda) = \sum_i^M p_i b_i(\vec{x}) \quad (2.34)$$

where $\vec{x}$ is a D -dimensional feature vector and $p_i, i = 1, \dots, M$ are the mixture weights. The mixture weights satisfy the constraint that $\sum_{i=1}^M p_i = 1$.

The density is a weighted linear combination of M Gaussian densities, $b_i(\vec{x}), i = 1, \dots, M$, and each parameterized by a mean $D \times 1$ vector, $\vec{\mu}_i$, $D \times D$ covariance matrix, Cov_i . The Gaussian function is as follow:

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |Cov_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)' \sum_i^{-1} (\vec{x} - \vec{\mu}_i) \right\} \quad (2.35)$$

The complete GMM is parameterized by the mean vectors, covariance matrix and mixture weights. Therefore the parameters of the density model are denoted as $\lambda = \{p_i, \vec{\mu}_i, Cov_i\}, i = 1, \dots, M$.

Figure 2.7: Example of GMM for speaker models. There are three components Gaussian mixture density represent speakers in 2 dimension

For speaker recognition, each speaker is represented by a GMM (Figure 2.7) and is referred to by the model parameters λ . The GMM can have several different forms depending on the choice of covariance matrix. Some GMM can have one covariance matrix per Gaussian component, and it is called nodal covariance. Some GMM has

one covariance matrix for all Gaussian components in a speaker model which is called grand covariance, or a single covariance matrix shared by all speaker models which is called global covariance. The covariance matrix can also be full or diagonal. The nodal and diagonal covariance matrices are used more often for speaker models. The same modelling of the full or non-diagonal covariance GMM can be equally achieved by using diagonal covariance GMM and the diagonal covariance matrix is more computationally efficient than full element covariance. The experimental results also indicate that diagonal covariance have better identification performance (Douglas et al., 2000).

The feature vectors extracted from the speech waveform which are supposed to represent the speaker features are used for GMM training. Training means to estimate the parameters of the GMM that best match the distribution of the speaker characteristic. There are many different kinds of techniques available for estimating the parameters of a GMM. The iterative expectation-maximisation (EM) algorithm is often used to estimate the Maximum likelihood (ML) while training the parameters of GMM. The idea of ML estimation is to determine the model parameters which maximise the likelihood of the GMM. For example, the training feature vectors are $X = \{\vec{x}_1, \dots, \vec{x}_T\}$, the GMM likelihood is:

$$p(X|\lambda) = \prod_{t=1}^T p(\vec{x}_t|\lambda) \quad (2.36)$$

This expression is a nonlinear function of parameters of λ , so the EM algorithm is to compute the maximum. The process of EM is introduced in the following:

1. initialize GMM model parameters λ and set iteration time $k = 1$.
2. for each iteration $k + 1$, generate new parameters λ and make ML bigger,
 $p(X|\lambda^{k+1}) > p(X|\lambda^k)$.
3. go to step 2 and iterate until the convergence threshold is reached.

To identify the claim of the speaker, the first step is to build up GMMs for each speaker with the parameters $\lambda_1, \lambda_2, \dots, \lambda_n$. The second step is to find out which of these GMM best fit the claimed speaker, which means to find the biggest probability for the features that is from the claimed speaker.

$$\hat{S} = \arg \max_{1 \leq k \leq S} Pr(\lambda_k | X) = \arg \max_{1 \leq k \leq S} \frac{p(X | \lambda_k) Pr(\lambda_k)}{p(X)} \quad (2.37)$$

To maximize the probability, assuming that $Pr(\lambda_k)$ is not changing, $P(X)$ is the same for all the speaker models, the maximisation becomes:

$$\hat{S} = \arg \max_{1 \leq k \leq S} p(X | \lambda_k) \quad (2.38)$$

And after applying the logarithm, it becomes:

$$\hat{S} = \arg \max_{1 \leq k \leq S} \sum_{t=1}^T \log p(\hat{x}_t | \lambda_k) \quad (2.39)$$

2.2.3 Hidden Markov Models (HMM)

Statistical methods of Markov source or hidden Markov modelling (HMM) was introduced during the 1960s and 1970s (Lawrence, 1989) and have been applied in many different areas. Hidden Markov models were introduced for the first time into speech areas by Baker (1975) at CMU.

Assuming $s_i, 1 \leq i$ are the states in Markov model, when all possible transitions from s_i to s_j are allowed, the model is called ergodic Markov (Figure 2.8). When only state sequences in chronological order are allowed (upper-triangular transition matrix) the model is called left-to-right Markov (Figure 2.9). Ergodic Markov modelling is usually applied in text independent speaker verification, whereas left-to-right Markov modelling is often applied in the text dependent case. A hidden Markov model differs from a Markov model in the sense that the state sequence cannot be observed directly and only the observation sequence is known.

Figure 2.8: Example of 4-states ergodic Markov model

Figure 2.9: Example of 4-states left-to-right Markov model

An HMM is characterized by the following elements:

1. The number of states in the model N . Although the states are hidden for HMM, for many practical applications there are ways to know the number of the states in the model according to some physical significance.
2. The number of distinct observation symbols per state M . The observation symbols correspond to the physical output of the system being modelled.
3. The state transition probability distribution $A = a_{ij}$ where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N$.
4. The observation symbol probability distribution in state j , $B = b_j(k)$.
5. The initial state distribution $\pi = \pi_i$.

Given appropriate values of the HMM parameters, HMM can be used as a generator to give an observation sequence. For convenience, $\lambda = (A, B, \pi)$ is used to indicate the complete parameter set of the HMM model.

To determinate the optimal model parameters, given a set of training vectors from a particular speaker, there exist approaches to train an HMM: the Baum-Welch algorithm (Lance, 2003) and the Viterbi algorithm (Forney, 1973). Both of these techniques use the Maximum Likelihood (ML) criterion. The Baum-Welch algorithm evaluates all possible state sequences, which could have produced the training observations, while the Viterbi algorithm only considers the most probable state sequence. The most general implementation of these algorithms update all parameters of the HMM, including covariances, means, weighting coefficients and transition probabilities.

In text-dependent speaker recognition, each speaker is modelled by a set of HMMs describing all speech units in the speaker's vocabulary. These speech units can be words, sub-words, or phonemes. The model of a speech unit u by speaker S is trained with the speech material of speaker S labeled as speech unit u in the initial segmentation of the speech of S . During the recognition phase, the likelihood that the claimed speaker has spoken a given sequence of speech units is computed, using the test data and the corresponding sequence of speech unit models, with optional leading, inter-word and trailing silences.

In text-independent speaker recognition, because there is no knowledge about the linguistic content of the speech utterances, except knowledge about the silence-speech segmentation of the utterance, each speaker is modelled by a single HMM describing the speaker in the non-silence parts of the training material. In the recognition phase, the likelihood of the test utterance for a claimed speaker is computed by aligning the test data with the speaker model alternated with silence. Figure 2.10 shows speaker recognition for both text-dependent and text-independent model.

In order to calculate the likelihood in Figure 2.10, a model describing all other

Figure 2.10: HMM speaker recognition system

speakers has to be built up besides the models describing each speaker. The reason is that the likelihood statistic $p(S_i|O)$, which determines the probability of speaker S_i given the observations O , cannot be evaluated directly. But it can be approximated to:

$$p(S_i|O) \approx \frac{p(O|S_i)}{\sum_{j \in L} p(O|S_j)} \quad (2.40)$$

Here L is the group of testing speakers. Therefore by evaluating the model of all speakers in the testing group, the likelihood can be computed.

The experiment in Matsui and Furui (1994) had showed that the HMM is robust as a VQ-distortion method against utterance variations.

2.2.4 Support Vector Machine (SVM)

SVM (support vector machine) is a binary classifier that makes its decisions by constructing a linear decision boundary or hyper-plane that optimally separates the two classes (Christopher, 1998).

The hyper-plane (Figure 2.11) is defined by $\vec{x} \cdot \vec{w} + b = 0$, where $\vec{w}$ is the normal to the plane. The linearly separable data is labeled as:

$$\{\vec{x}_i, y_i\}, \vec{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}, i = 1 \dots N \quad (2.41)$$

The optimal hyper-plane is chosen according to the maximum margin criterion,

Figure 2.11: A linear hyper-plane SVM that maximun the margin

i.e. by choosing the separating plane that maximizes the Euclidean distance to the nearest data points on each side of that plane. This is achieved by minimizing the square of the L_2 -norm of $\vec{w}$, $\|\vec{w}\|_2^2$ subject to the inequalities $(\vec{x}_i \cdot \vec{w} + b)y_i \geq 1$ for all i . The solution for the optimal hyper-plane, $\vec{w}_0$, is a linear combination of a small subset of data, $\vec{x}_s$, $s \in \{1 \dots N\}$, known as the support vectors. These support vectors also satisfy the equality $(\vec{x}_s \cdot \vec{w}_0 + b)y_s = 1$.

The data that is from daily life is usually not linearly separable, then there is no hyper-plane exists which can separate all points and satisfy the inequality above ideally. To overcome this problem slack variables, ξ_i , are introduced to relax the constraint slightly, so that some points are allowed to lie within the margin or even be misclassified completely. The resulting problem is then to minimize

$$\frac{1}{2} \|\vec{w}\|_2^2 + C \sum_i L(\xi_i) \quad (2.42)$$

subject to $(\vec{x}_i \cdot \vec{w} + b)y_i \geq 1 - \xi_i$, where ξ_i is the empirical risk associated with the marginal or misclassified points, L is the loss function and C is a hyper-parameter used to specify the balance between minimizing the empirical risk and maximizing the margin.

The most commonly used loss is the linear-error cost function. The dual formulation, which is more conveniently solved, with $L(\xi_i) = \xi_i$ is

$$\max_{\alpha} \left(\sum_i \alpha_i + \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \right) \quad (2.43)$$

subject to:

$$0 \leq \alpha_i \leq C \quad (2.44)$$

$$\sum_i \alpha_i y_i = 0 \quad (2.45)$$

in which α_i is the Lagrange multiplier of the i^{th} constraint in the primal optimization problem. The dual can be solved using standard quadratic programming techniques.

The orientation of the optimal plane, $\vec{w}_0$, is given by :

$$\vec{w}_0 = \sum_i \alpha_i y_i \vec{x}_i \quad (2.46)$$

And $\vec{w}_0$ is a linear combination of all points in the feature space that have $\xi_i > 0$ as well as those that lie on the margin (i.e. $\alpha_i \neq 0$).

To extend the linear hyper-plane to non-linear boundaries, kernels are used. The idea of kernels is to map each data point onto a manifold embedded in some feature space which may be of higher dimension than the input space. The feature space is defined implicitly by the kernel. The hyper-plane is constructed in the feature space and intersects with the manifold creating a non-linear boundary in the input space (Figure 2.12).

In practice, the mapping is achieved by replacing the value between two data points in the input space with the value that results when the same product is carried out in the feature space. The product in the feature space is expressed conveniently by the kernel as some function of the two data points in the input space. Polynomial kernels and radial basis function (RBF) kernels are used commonly. The polynomial

Figure 2.12: Non-linear input space is mapped into feature space

kernel is as following:

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^n \quad (2.47)$$

where n is the order of the polynomial.

And the RBF kernel is:

$$K(\vec{x}_i, \vec{x}_j) = \exp \left[-\frac{1}{2} \left(\frac{\|\vec{x}_i - \vec{x}_j\|}{\sigma} \right)^2 \right] \quad (2.48)$$

where σ is the width of the radial basis function.

The dual for the non-linear case thus becomes:

$$\max_{\alpha} \left(\sum_i \alpha_i + \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \right) \quad (2.49)$$

subject to

$$0 \leq \alpha_i \leq C \quad (2.50)$$

$$\sum_i \alpha_i y_i = 0 \quad (2.51)$$

The use of kernels means that an explicit transformation of the data into the feature space is not required. There are other kernels such as sigmoid, but the linear kernel, polynomial kernel and RBF kernel are used more often.

While SVMs are used in speaker verification, both positive and negative examples for training are required. That means not only examples of the user's speech are required, but also sufficient samples from some impostor speakers. This ensures the classifier does not misclassify an impostor who was not seen by the classifier in the training stage. Increasing the number of impostors in the training set is a way to prevent this from occurring.

However, SVMs do not scale well to very large training sets. When the data is inseparable, the number of support vectors that parameterise the solution grows with the size of the training set. This arises from the fact that points that lie inside the margin or are incorrectly classified are included as support vectors in the solution. A larger solution requires more storage space and significantly increases the amount of computation required during both training and classification. Thus it is more desirable to train an SVM on relatively small amounts of data so one must be especially careful when selecting training data to avoid under-training. Vector quantization can be used as a simple method that provides a small training set that is still representative of the full training set with only a small effect on the overall performance.

It is possible to extend the SVM speaker verification to the speaker identification task. The task of speaker identification is to determine the identity of a speaker from a group of speakers. One of the simplest methods to construct classifiers which can separate each speaker from all of the others is to construct n classifiers if there are n speakers.

2.3 Identification of research issues

There are other feature extraction methods for speaker recognition besides the features that are mentioned in the previous section. Different features may contain different information of speaker characteristics. It is necessary to evaluate them under the

same system. Therefore in the later chapters, some of the features are tried out and compared both in the speaker segmentation system and speaker verification system in the later chapters.

In the chapter of speaker segmentation algorithm, SVM is used for finding speaker changing points. A novel way of combining different kinds of features is proposed for SVM training.

GMM is chosen for the speaker verification system in the later chapter. GMM acts as a hybrid between Gaussian models and VQ by using a discrete set of Gaussian functions, each with their own mean and covariance matrix, to allow a better modelling capability. GMM can perform smooth approximations to arbitrarily-shaped densities.

Chapter 3

Speaker change detection

The audio signal obtained from conversations in meetings usually contains the speech of different speakers. The audio from broadcast or TV shows can be a mixture of human speech, music and noise. Audio segmentation aims at finding acoustic events such as speaker changing points within an audio stream, music or other non-speech events in the audio files. Speaker changing point detection is a special case of speaker segmentation. It is a key pre-requisite to speaker tracking and speaker adaptation (Kwon & Narayanan, 2002). Its task is to detect the points where a speaker identity changes in a multi-speaker audio stream. In this chapter, methods of speaker change detection are investigated.

Figure 3.1 shows an example of speaker changing point detection. The arrows are the labels of speaker changing points in the continuous speech file. The speaker changing point detection in this chapter is performed under the assumptions:

- No more than two speakers speak simultaneously;
- Speakers use the same channel to record voice, which means it is a mono-channel speaker segmentation task;
- There is no prior knowledge about the number of speakers;
- The speech file is assumed to be clean speech, thus no noise deduction scheme is considered.

Figure 3.1: Example of speaker change detection

3.1 Performance evaluation methods

To evaluate the performance of the speaker changing point detection, two criterions are considered: the precision of speaker changing points that are found and the number of missing changing points. If, among all the speaker changing points that are found, only a small number of changing points are correct, the performance of the system is poor. The precision scale is used to measure this performance. On the other hand, although most of the points that are found are correct, many true speaker changing points are missing, the performance of the system is still poor. For this situation, the recall scale is used. The performance depends on these two scales. In this study the evaluation criterion are the *F – Score* which is calculated from Recall (RCL) and Precision (PRC) percentage as follow (Ajmera et al., 2004):

$$PRC = \frac{\text{Number of correctly found changes}}{\text{Total number of changes found}} \quad (3.1)$$

$$RCL = \frac{\text{Number of correctly found changes}}{\text{Total number of correct changes}} \quad (3.2)$$

$$F - \text{Score} = \frac{2 \times PRC \times RCL}{PRC + RCL} \quad (3.3)$$

3.2 Evaluation data

The database used in this chapter for testing is the broadcast news in English from test Data in 1997 Hub-4E Evaluation Plan of NIST (*NIST HUB-4E Broadcast News Evaluation*, 1997). The data was obtained from the audio component of broadcast news sources and each audio file consists of approximately one hour of speech in English. The audio files include the speech of many speakers. The data consists of a single monophonic channel of audio.

3.3 Feature of frames

Acoustic features are used in speaker changing point detection in this chapter. Acoustic features are generated based on spectrogram. Figure 3.2 shows an example of the speech spectrogram which is taken from a 72 millisecond speech file with the sampling rate of 16 kHz. The spectrogram is generated based on frames of speech. The frame size in Figure 3.2 is 30 milliseconds. The colour bar shows the magnitude.

The frame size is decided according to feature extraction scheme, for example usually 20 millisecond or 30 millisecond are used for MFCC features. The frames can be overlapping. Because the overlapping can compensate the loss of signal data at the border of frames, which results from application of a hamming window to the signal frames. Figure 3.3 shows how each speech frame corresponds to feature vectors.

3.4 Speaker changing points detection using BIC

Generally there are three main techniques for detecting the speaker change: decoder guided, metric based and model based. The decoder guided segmentation only places

Figure 3.2: Speech spectrogram

boundaries at silence locations, that is it has no direct connection with the acoustic changes in the data (Chen & Gopalakrishnan, 1998). The metric based algorithm calculates the distance between the segments (Siegler et al., 1997). For example, Euclidean and Mahalanobis distances can be used to calculate the distance between the segments. Because it just uses the threshold of the measurements, it does not need a large training data and prior knowledge of the speakers. For model based segmentation methods, different models, such as Gaussian mixture model, are built for a set of acoustic classes (Bakis, 1997). The input audio stream is classified by maximum likelihood selection over a sliding window. The segmentation is made at the location where there is a change in the acoustic class.

In this chapter, BIC (Bayesian Information Criterion) proposed by Chen and Gopalakrishnan (1998) is used for the speaker change detection. It is effective for the audio file in which the speaker change is not so frequent, such as broadcast news where speech segments have long duration. BIC speaker segmentation does not need

Figure 3.3: Speech frame to feature frame

training and has low computational cost, so it can be used in real-time segmentation.

3.4.1 Variance BIC value

BIC (Bayesian Information Criterion) is a criteria for model selection and is a likelihood criterion penalized by the model complexity: the number of the parameters in the model. The BIC speaker changing point detection assumes that each segment of acoustic features can be modelled as a single Gaussian distribution. Then the distance of different acoustic feature segment is calculated based on variances between segments.

Let $X = x_i : i = 1, \dots, N$ be the acoustic features of speech that are being modelled and let $M = M_i : i = 1, \dots, K$ be the candidates of the desired parametric models.

Assuming the likelihood functions have to be maximized separately for each model M : $L(X, M)$. Denote $N(M)$ as the number of the parameters in the model M . The BIC criterion is defined as:

$$BIC(M) = \log L(X, M) - \lambda \frac{1}{2} N(M) \times \log(N) \quad (3.4)$$

here λ is the penalty weight. The BIC is to choose the model for which the BIC criterion is maximized.

The Variance BIC (Nishida & Kawahara, 2003) is developed from BIC and it is used to represent the feature vector distance between two speech segments. Variance BIC is formulated using the following function:

$$\begin{aligned} \Delta BIC_{variance}^i = & -\frac{n_1 + n_2}{2} \log_i |\Sigma_0| \\ & + \frac{n_1}{2} \log_i |\Sigma_1| + \frac{n_2}{2} \log_i |\Sigma_2| \\ & + \alpha \frac{1}{2} (d + \frac{1}{2} d(d + 1)) \log(n_1 + n_2) \end{aligned} \quad (3.5)$$

where Σ_0 is a covariance of the whole segment, Σ_1 is a covariance of the first segment, and Σ_2 is a covariance of the second segment. n_i is the number of frames for the segment and d is the dimension of the acoustic features. The bigger the variance BIC of two segments is, the greater probability that there is a speaker changing point between these two segments. By searching for the peak points in variance BIC value of whole speech, the speaker changing points are marked. Figure 3.4 shows the main process of speaker segmentation.

3.4.2 Search for speaker changing points

Figure 3.5 shows an example of speaker changing point of two segments of frames. Variance BIC values are used to compare the distance between two segments of features. In the first place, there is no prior knowledge, searching for the speaker changing points has to be applied from the beginning of the speech file. Just by comparing two

Figure 3.4: The process of speaker segmentation

Figure 3.5: Segmentation of audio file

frames, there is insufficient information to detect a speaker changing point, because each frame gives one feature vector which is calculated from about 30 milliseconds of speech. Concatenated frames are grouped as segments for analysis. Speaker changing point detection is performed between two segments.

To search speaker changing point in the whole speech file, a sliding window was applied from the beginning of the audio file to the end of the audio file. Inside this sliding window, there is a detecting point. This detecting point moves from the beginning of the sliding window to the end of the window on the step of one frame. The left part of detecting point is segment A, the right part of the detecting point is segment B. The variance BIC value of these two segments is calculated at each detecting point. The scheme of moving the sliding window was as follows:

Figure 3.6: The figure of variance BIC Value. Each frame has a variance BIC Value. The circle stands for a speaker changing point in that frame

1. Initialize the sliding window $[a, b]$: $a = 0$, $b = \text{Minimun Window size}$;
2. Find the change point in $[a, b]$ according to variance BIC value;
3. If (no change point in $[a, b]$): $b = b + \text{More Frames}$; else if (t is the change point) $a = t + 1$, $b = a + \text{More Frames}$;
4. If $(b - a > \text{Maximun Window size})$, $a = b - \text{Maximun Window size}$;
5. If b is not the end of the audio file, Go to step (2); else stop searching.

In the step 2, if variance BIC value at $point(t)$ is bigger than a set threshold, a potential speaker changing point is found. Figure 3.6 shows the variance BIC values of a speech file. On the waveform, the points where circles are marked are the true speaker changing points.

3.4.3 Experimental results

In this chapter, Mel Frequency Cepstral Coefficients (MFCC) (Oppenheim & Schaffer, 2004) features are used in a BIC speaker changing point detection algorithm. In addition, several novel acoustic features are also used: Mel Line Spectrum Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst parameter features (pH) (Sant’Ana et al., 2006), Haar Octave Coefficients of Residue (HOCOR) (Zheng & Ching, 2004), and features based on fractional Fourier transform (MFCCFrFT). The features were extracted from speech signals sampled at 16 kHz.

In the experiment 12-dimensional MFCC vectors are extracted from 30 millisecond long frames without overlap. The feature values were normalised by subtracting the mean and dividing by the standard deviation. First order difference features are also used.

Fractional Fourier transform (FrFT) is a generalization of the ordinary (integer) Fourier transform and was introduced in signal processing by (Almeida, 1994).

MLSF are similar to Line Spectrum Frequencies calculated from Linear Predictor coefficients and were proposed in the context of the speaker verification problem (Cordeiro & Ribeiro, 2006). Mel spectrum was generated via Fast Fourier Transform (FFT) and mel filter bank applied to 30 millisecond frames. Inverse Fourier transform was applied to calculate the mel autocorrelation of the signal, from which MLSF features were then calculated via Levinson-Durbin recursion. Linear prediction of order 16 was used refers to Cordeiro and Ribeiro (2006). The feature values were normalised by subtracting the mean and dividing by the standard deviation.

Hurst parameter features have also been proposed for speaker recognition problem (Sant’Ana et al., 2006). The feature vector is a vector of Hurst parameters calculated for frames of a speech signal via Abry-Veitch Estimator using discrete wavelet

transform (Veith & Abry, 1998). In the current study a frame length of 60 milliseconds was used, and Daubechies wavelets with four, six, and twelve coefficients were tried giving rise to pH_4 , pH_6 , and pH_{12} features. The depth of wavelet decomposition was chosen to be 5, 4, and 3 for pH_4 , pH_6 , and pH_{12} correspondingly, thus resulting in 5-, 4-, and 3-dimensional feature vectors.

Haar Octave Coefficients of Residue (HOCOR) features were extracted by applying Haar transform to the residual signal. The features were originally proposed in Zheng and Ching (2004) for the purpose of speaker recognition. In the current study the LP of order 12 was applied (refers to (Zheng & Ching, 2004)) to 30 millisecond frames. HOCOR_α features of order α 1, 2, 3, and 4 were extracted (Zheng & Ching, 2004).

Feature	d	F -score	Precision	Recall
MFCC	26	0.6151	0.6081	0.6223
MLSF	10	0.4211	0.2857	0.8000
pH_4	5	0.5213	0.6675	0.4276
pH_6	4	0.5272	0.6667	0.4359
pH_{12}	3	0.5496	0.6806	0.4609
HOCOR_1	6	0.4215	0.5389	0.3461
HOCOR_2	5	0.3704	0.4729	0.3045
HOCOR_3	4	0.3100	0.3885	0.2579
HOCOR_4	3	0.3026	0.3804	0.2512
$\text{FrFTMFCC}_{0.9}$	12	0.6148	0.7259	0.5641

Table 3.1: F -score, precision and recall for different features. d is the dimensionality of the acoustic feature vectors

Table 3.1 shows the speaker changing point detection results achieved when different acoustic features were used to calculate the variance BIC. The peak detection algorithm was used to detect speaker changing points from the BIC values. Among these features, pH features gives F -scores comparable to those when MFCC features are used, even though the dimensionality of feature vectors of pH features is far less than those of MFCC. This suggests that pH features may be a better choice when the training data set is small.

3.5 Speaker segmentation using SVM

Support vector machine (SVM) is a binary classifier that makes its decisions by constructing a linear decision boundary or hyper-plane that optimally separates the two classes (Christopher, 1998). It requires less training data to achieve better performance than other classifiers. Speaker segmentation can be treated as a binary decision task: the system must decide whether or not a speech frame is the speaker changing point. This chapter proposes a SVM speaker changing point detection scheme which combines different speech features in a novel way. Figure 3.7 shows the structure of SVM speaker segmentation.

Figure 3.7: SVM speaker segmentation system

3.5.1 SVM feature generation

Different speech features may contain different speaker information. The proposed SVM speaker changing point detection is to make use of different features together. The features that are used for SVM training in this section were not the normal

acoustic features such as MFCC, they are the combination of several acoustic features. Most of the acoustic features are of high dimensionality, and simple concatenation of the feature vectors will result in a feature vector of even higher dimensionality, which, in turn, will require too many training samples to be trained reliably. Instead, in the current study the variance BIC value is calculated from each of the acoustic features as described above. The variance BIC values give the information of distance of two speech segments. The combination of variance BIC values can make speaker changing point detection robust.

Figure 3.8 shows the scheme of combination of variance BIC value vectors. Each frame will have the variance BIC values from different acoustic features. The length of the SVM feature vector is the number of acoustic features that are used.

Figure 3.8: Combination of variance BIC value vectors

3.5.2 Labels of frames

The two classes in speaker segmentation are: the frames that contain speaker changing points and the frames which do not contain speaker changing points. In SVM training, the frames which contain speaker changing point are labelled as -1 , the frames without speaker changing point are labelled as 1 . In the thesis the acceptable error range of the found speaker changing points is 0.5 second, which means the

frames that are 0.5 second before a speaker changing point and the frames that are 0.5 second after a speaker changing point are all labelled as -1 . Figure 3.9 shows an example of labelling -1 frames for 2 seconds speech file. The speech frames that are not in shape are considered as the frames without speaker changing points, and they are labelled as 1.

Figure 3.9: Labels of frames

3.5.3 Scaling SVM feature vectors

The variance BIC values obtained from different acoustic features are in different scales, it is necessary to scale them before training and testing using SVM. The scaling applied on each variance BIC value vector is as follow:

$$ScaleFeature_i^j = \frac{Feature_i^j - \text{mean of } Feature_i}{\text{standar variance of } Feature_i} \quad (3.6)$$

where i represent different acoustic feature, j is the frame number of the i th feature.

3.5.4 Peak tracking

The SVM classifier returns two values for each frame that are related to the distance to the separating hyperplane (either of them can be monotonically mapped into the conditional class probability). The values sum to one and indicate to what extent the frame belongs to class -1 (or 1). The -1 class value is analysed to determine the true speaker changing point. A peak searching algorithm was used to determine the local maxima of -1 class value as we moved along the frames. The peak searching algorithm uses adaptive threshold in an attempt to eliminate small peaks due to noise and find only true local maxima.

3.5.5 Analysis of experimental results

The database used in this experiment is also the broadcast news in English from test Data in 1997 Hub-4E Evaluation Plan of NIST (*NIST HUB-4E Broadcast News Evaluation*, 1997). $1/4$ data was used as SVM training data and the rest of the data was used for testing.

The following acoustic features were used to calculate variance BIC value for SVM training: Mel frequency cepstral coefficients (MFCC) (Oppenheim & Schafer, 2004), Mel Line Spectral Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst Parameter Related Features (pH) (Sant’Ana et al., 2006), Haar Octave Coefficients of Residue (HOCOR) (Zheng & Ching, 2004), Features based on Fractional Fourier Transform (MFCC FrFT) (Almeida, 1994).

Feature	F -Score	Precision	Recall
SVM	0.6414	0.7214	0.5774

Table 3.2: F -score, precision and recall for SVM speaker change detection

The features used for SVM are the 10 variance BIC values resulting from the 10 acoustic features. The bottom row in Table 3.2 shows that the proposed SVM speaker

changing point detection scheme improves the speaker change point detection performance as compared to each of the individual acoustic features, with a higher F -score of 0.6414. This means that other acoustic features, which were originally proposed for speaker recognition problem, can be used for the problem of speaker segmentation as well. This study demonstrates that new features do carry additional information about speaker differences, and some of them also have attractiveness because of their low dimensionality. Further study may find better ways of how to integrate complementary information about speaker differences contained in the new features with traditional features such as MFCC and LPCC.

3.6 Summary of speaker changing point detection

In this chapter, first speaker changing point detection using BIC is investigated. Different kinds of acoustic features are used in this BIC speaker changing point detection. And a dynamic threshold is applied to search for speaker changes. By comparing the F -Score result from Hub4 broadcast audio files, the MFCC feature get the best performance. BIC speaker change detection algorithm works well in the audio file in which the speaker change is not so frequent. But it misses many speaker changes where speakers changes occur in short period.

The idea of the proposed SVM speaker change detection is to combine different acoustic features. Instead of concatenating each feature, the variance BIC values are combined, which denotes the distance of speech segments. And it overcomes the problem of some features whose vector length is too long. The F -Score of the proposed system shows that it improves the performance of speaker change detection system which only use one acoustic features. Beside combination of variance BIC values, other features or other feature combination schemes may be used in the SVM speaker changing point detection system to improve the performance.

Chapter 4

Testing features in speaker verification

Speech data is easily obtained as it is a natural signal and no separate or intrusive step is required to collect the speech data. Most of the commercialization has focused on using speaker verification as a biometric to control access to information service and computer accounts, for example telephone banking.

In this chapter, besides the classic MFCC features, six types of novel speech features were tested separately in the speaker verification system which was built up using a Gaussian mixture Model (GMM). They are: Mel Line Spectral Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst Parameter Related Features (pH) (Sant'Ana et al., 2006), Linear Prediction Residual Phase (rPhase) (Murty & Yegnanarayana, 2006), Haar Octave Coefficients of Residue (HOCOR) (Zheng & Ching, 2004), Minimum Variance Distortionless Response (MVDR) (Huang & Hansen, 2006), and features based on the fractional Fourier transform (FrFT) (Almeida, 1994) that include MFCC features calculated using FrFT and mean energy within critical bands (MECB).

The combinations of these features with classic MFCC were also investigated. The performance results collected from the same testing environment were compared with the classic MFCC results while MFCC was used alone in the verification.

Figure 4.1: GMM for speaker verification

4.1 Speaker verification based on GMM

The GMM is used to build the speaker models for testing the performance of different speech features in speaker verification. The use of the Gaussian mixture density for speaker identification is motivated by two interpretations (Reynolds & Rose, 1995). First, the individual Gaussian components in a speaker-dependent GMM are interpreted to represent some broad acoustic classes. These acoustic classes reflect some general speaker-dependent vocal tract configurations that are useful for modelling speaker identity. Second, a Gaussian mixture density is shown to provide a smooth approximation to the underlying long-term sample distribution of observations obtained from utterances by a given speaker. Figure 4.1 shows the structure of a GMM speaker verification system.

In Figure 4.1 universal background models (UBM) are trained to represent the

background of the all the speech audio. The equal error rate (ERR) was used to measure the performance of the verification. The ERR of a verification system when operating threshold for the accept/reject decision helps to adjust system such that the probability of false acceptance and that of false rejection become equal. DET curve was used as presentation of the ERR.

4.2 Different features' performance in speaker verification

The feature testing experiments were conducted using the NIST 2001 speaker recognition evaluation database, 1-speaker detection audio files. They are from telephone conversation speech, which are encoded in 16-bit PCM using .wav file format. The audio files were sampled at 8 kHz. After being pre-emphasized with filter coefficient of 0.97, the speech signal is divided into frames of different lengths for extraction of different features mentioned in the following section (from section 4.2.1 to 4.2.5) using the Hamming window. A GMM classifier with 512 multivariate normal distributions with diagonal covariance matrices was used to build speaker models in MATLAB. Speech samples from 82 male and 56 female speakers, in which everyone spoke for 2 minutes, were used for training. In the evaluation stage, speech samples from 174 speakers are used. The number of testing speech segments is 22418, with 1 genuine segment per every 10 impostor segments. The average length of a speech sample is approximately 20 seconds. The DET curves are shown along with the ERR.

4.2.1 Performance of mel line spectral frequencies (MLSF)

MLSF features (Cordeiro & Ribeiro, 2006) are an improved version of LSF features. The MLSF features for speaker verification in this chapter were extracted from frames of 30 milliseconds length with 1/3 frame overlap (according to the parameters provided

Figure 4.2: DET curves and EER values for MLSF features with first and second order differences

by Cordeiro and Ribeiro (2006)). Individual frame pre-emphasis is applied on each frame too.

In the speaker verification experiment, first and second order differences were added. It is found that first order differences improve EER from 18.6% to 16.5%. While adding the second order differences, it improves the accuracy further, with EER equal to 15.9%. The DET curves are shown in Figure 4.2.

The scores obtained from the classifier using MLSF features were used in classifier fusion later (section 4.2.7) and resulted in improvement of speaker verification accuracy.

4.2.2 Performance of hurst parameter related features (pH)

Hurst parameter related features (pH) can be extracted using Daubechies wavelets of different order for audio signal decomposition. As was suggested in the original paper Sant'Ana et al. (2006), the frame length was 30 milliseconds in this chapter, wavelets of orders 4, 6, and 12 (db_4 , db_6 , and db_{12}) were used in experiments, giving rise to pH_4 ,

Figure 4.3: DET curves and EER values for single pH feature vectors and concatenated, frame length 80 ms, frame overlap 1/3

pH_6 , and pH_{12} feature vectors correspondingly. The length of each feature vector is defined by the maximum level of decomposition, which is bound by frame length.

Different frame length and overlapping were tried in the experiment. Frame length can be: 30 milliseconds, 60 milliseconds, 80 milliseconds, and 100 milliseconds with frame overlaps of 1/3, 1/2, and 2/3. The feature vectors from different wavelet decompositions were also concatenated to gain more information, giving $pH_4 + pH_6 + pH_{12}$ feature vector for each frame. The experiment result is showed in Figure 4.3.

Concatenation of individual feature vectors resulted in improvement of speaker verification accuracy, with EER dropping from 26.5%, 28.7%, and 29.0% for pH_4 , pH_6 , and pH_{12} correspondingly to 20.8% for frame overlap of 1/2 and 20.4% for frame overlap of 2/3. The results are presented in Figure 4.4.

As seen from the experiment, the best frame length and frame overlapping parameters are chosen. According to the experiment, the final extraction settings for Hurst parameter related features in NIST data based can be selected as following:

Figure 4.4: DET curves and EER values for concatenated pH features, different frame lengths

- Frame length of 80 milliseconds with 2/3 frame overlap.
- Decomposition order of 5 for db_4 and 4 for db_6 and db_{12} .
- Concatenation of feature vectors calculated from decompositions using different wavelets.
- No individual frame pre-emphasis.

The scores obtained from the classifier using pH features under this parameters are used in classifier fusion later section (section 4.2.7).

4.2.3 Performance of linear prediction residual phase

Residual phase is calculated from LP error 2 and thus is supposed to carry additional information not captured by LPC features (Murty & Yegnanarayana, 2006). In the experiments described in this chapter, the feature extraction parameters to be set were:

- Frame length and frame overlap;
- LP order;

LP order	ERR, %	Frame length (milliseconds)	ERR, %	Chunk size, in fraction of frame length	EER, %
6	24.8	20	26.8	1/30	28.1
8	26.6	30	28.0	1/15	27.6
10	28.0	40	27.4	1/10	27.3
16	27.4	50	27.9	1/16	28.0
20	27.9			1/3	27.1
				1/2	26.6

Table 4.1: Equal error rate for varying parameters for linear prediction residual phase extraction

- Number of feature vectors per frame. This parameter reflects division of the LP error signal into equal pieces of smaller size which is called chunk size later and calculating phase within each of them.

While varying parameters of the LP order, frame length, and chunk size, different results were obtained, see Table 4.1.

Figure 4.5 shows the ERR obtained using residual phase features with different extraction parameters such as LP orders, chunk sizes, and frame lengths. It is seen that EER for frame length of 6 millisecond frames is higher than that for 3 millisecond frames, and also better than the EER obtained for 12 millisecond frames. Also, LP order of 6 appears to give better performance. The lowest EER values are also produced when the chunk size is equal to the frame length.

The scores obtained from the classifier using residual phase features are also used in classifier fusion later section (section 4.2.7).

4.2.4 Features based on fractional Fourier transform

About the features based on fractional Fourier transform, a few different feature were extracted under different parameter p . First, Mean energy within critical bands ($MECB_p$) were extracted for fractional Fourier transform (FrFt) of orders p , $p = 0.1, 0.2, \dots, 1.0$, when $p = 1.0$ equals to mean energy of classic MFCC features. Also, differential $MECB$ ($DMECB_{p1-p2}$) were calculated as a difference between

Figure 4.5: EER values for residual phase features, different frame lengths, LP orders and number of feature vectors per frame

$MECB$ extracted using FrFT of order $p_1 = 1.0$ and orders of $p_2 = 0.1, 0.2, \dots, 0.9$. This differential $DMECB_{1.0-p_2}$ represents the different between $MECB$ where $p = 0.1, 0.2, \dots, 0.9$ from mean energy of classic MFCC features $MECB$ with $p = 1.0$.

Audio speech was cut into frames of 30 milliseconds with $1/3$ overlap according to classic MFCC features parameters that are used often. The frequency range from 0 to 4 kHz was warped according to mel scale and divided into 17 bands, thus giving 17 feature values per frame calculated for $MECB$ and $DMECB$ correspondingly.

DET curves for $MECB$ and $DMECB$ features are shown in Figure 4.6 and Figure 4.7. $MECB_{0.9}$ shows the highest classification accuracy among $MECB$ of different orders, while $DMECB$ features of different orders demonstrates similar performance, with $DMECB_{1.0-0.9}$ being slightly better than others.

$MFCC_p$ features were also extracted using FrFT F^p of orders $p = 0.1, 0.2, \dots, 1.0$ in place of conventional Fourier transform ($FrFT_{1.0}$). All the rest of parameters

Figure 4.6: DET curves with EER values for MECB features extracted using different order fractional Fourier transform

Figure 4.7: DET curves with EER values for DMECB features calculated as difference between MECB_{1.0} and MECB of other orders

Figure 4.8: DET curves and EER values for MFCC features extracted using fractional Fourier transform of different orders.

are made equal to those of baseline *MFCC*. In a similar way to *MECB* and *DMECB* features, *DMFCC*_{*p*1–*p*2} features were calculated between *p*1 = 1.0 and *p*2 = 0.1, 0.2, . . . , 0.9.

DET curves for *MFCC*_{*p*} features for different orders *p* are shown in Figure 4.8. DET curves for *DMFCC* of different orders are shown in Figure 4.9. As seen from the figure, conventional *MFCC* (*MFCC*_{1.0}) still gives the best performance.

The features that were attained in this section were also used in combination with MFCC features in the section 4.2.7.

4.2.5 HOCOR and MVDR

The HOCOR and MVDR features were also tried in the GMM speaker verification system under the same condition. The result shows that both of them demonstrated poor performance.

HOCOR feature is based on LP analysis of the frame followed by wavelet decom-

Figure 4.9: DET curves and EER values for DMFCC features calculated as differences between conventional MFCC and MFCC of different orders

position of the residual signal. To extract the feature, the following parameters are required:

- Frame length.
- Frame overlap.
- LP order.
- HOCOR order.

The performance of HOCOR features in speaker verification is poor. For example, by using HOCOR features of order 1 ($HOCOR_1$), the EER that was obtained was 45.0%.

For the MVDR features, the following parameters were used as suggested by the original papers:

- Frame length of 30 ms.
- Frame overlap of 1/3.

Feature type	Detail	ERR, %
Baseline MFCC	12 MFCC	12.7
	MFCC+ Δ MFCC+ $\Delta\Delta$ MFCC	9.6
MLSF	MLSF only	18.6
	MLSF+ Δ MLSF	16.5
	MLSF+ Δ MLSF+ $\Delta\Delta$ MLSF	15.9
PH	PH_4	29.0
	PH_6	28.7
	PH_{12}	29.0
	$PH_4 + PH_6 + PH_{12}$	20.8
HOCOR	$HOCOR_1$	45.0
	$HOCOR_1, HOCOR_3, HOCOR_4$	100.0
Linear prediction residual phase		21.5
MVDR	MVDR	50.0
	MVDR+ Δ MVDR+ $\Delta\Delta$ MVDR	51.0
$MFCC_p$	$MFCC_{0.9}$	13.8
$DMFCC_{p1-p2}$	$DMFCC_{1.0-0.9}$	13.1
$MECB_p$	$MECB_{0.9}$	20.0
$DMECB_{p1-p2}$	$DMECB_{1.0-0.8}, DMECB_{1.0-0.7}, DMECB_{1.0-0.6}$	20.6

Table 4.2: Best achieved equal error rates for different features studied

- Bar spectrum warping.
- LP order of 22.
- Feature mean subtraction and normalisation by standard deviation.

The equal error rate (EER) of 51% is achieved for MVDR features.

Since MVDR and HOCOR feature achieve poor performance compared to other features in the GMM speaker verification system, they were not used in the later combination section. The conclusion that they have no useful information for speaker verification can not be made. The extraction parameters have to be adjusted in the future experiment and the model of speaker verification system may need to be improved.

4.2.6 Summary of results for studied features

Table 4.2 presents the best results achieved in terms of the EER for different features studied as well as for the baseline MFCC features.

Figure 4.10: Combining GMM scores from different acoustic features into a score feature vector

4.2.7 Combination of features

Combining different acoustic features can be performed in a number of ways. One way is to simply concatenate the feature vectors. To do this, the number of feature vectors per audio file should be the same for all feature types. A drawback of the concatenation method is feature vectors of very high dimensionality that result from concatenation of several high-dimensional feature vectors, which means more data is required for reliable training of a classifier.

A GMM enables modelling the conditional probability density functions in the feature space for each class. The GMM speaker verification system in this chapter returns a score for each given speaker, which is an estimation of the log likelihood ratio for the hypothesis that the speaker is who he claims to be. These scores returned by GMM speaker verification system using different acoustic features can be used again as features, as shown in Figure 4.10. These resulting score feature vectors were used in an SVM classifier.

To make the results comparable to those of acoustic features alone, a 5-fold cross-validation scheme was applied. The test set of speakers was divided into 5 approximately equal parts. Every time one different part was left for testing and four others were used for training the SVM, resulting in 5 experiments in total. The SVM was

Figure 4.11: DET curves and EER values for baseline MFCC and linear combination of scores from MFCC and new features that were found useful

designed to produce a soft decision (a value that depends on the distance from a pattern to the separating hyper plane).

The scores produced from SVM testing were then combined by choosing the best score from different features. It was decided to combine the following features: MFCC with the first and second differences, MLSF with the first and the second differences, Residual Phase with LP filter of order 6 and first differences, pH features and first differences, $MECB_{0.9}$, and $DMECB_{1.0-0.9}$, $MFCC_{0.9}$, $DMFCC_{1.0-0.9}$ and first differences. The rationale for choosing these features was to choose one feature from each group that gives the best performance. The results of the combination of the features using an SVM are presented in Figure 4.11.

4.3 Summary of feature testing in speaker verification

In this Chapter, nine features have been studied. Among them two did not give good performance and the remaining seven may carry additional information to MFCC

features and thus may improve speaker verification accuracy when are used together with MFCC.

Two features that give poor performance in speaker verification system also demonstrated poor results when combined with other features. They are Haar octave coefficients of residue (HOCOR) and minimum variance distortionless response (MVDR). Since the HOCOR features have been reported to improve performance of speaker recognition systems (Zheng & Ching, 2004), and the MVDR features have been reported to be used successfully for speaker segmentation and speech recognition, it is possible that the features are very sensitive to the right choice of feature extraction parameters and more experiments are required to evaluate them.

Seven features were found to contribute in improvement of speaker recognition accuracy:

- Hurst parameter features (pH).
- Mel Line Spectral Frequencies (MLSF).
- Residual Phase features. These features are complimentary to LPC (and MLSF) features and are aimed at characterising a person's vocal excitation source.
- $MFCC_p$ extracted using fractional Fourier transform of different orders p .
- $DMFCC_{p1-p2}$, the difference between $MFCC_{p1}$ and $MFCC_{p2}$ features.
- Mean Energy within Critical Bands ($MECB_p$) extracted using fractional Fourier transform of different orders p .
- $DMECB_{p1-p2}$, the difference between $MECB_{p1}$ and $MECB_{p2}$.

Even though it is possible to reduce error of speaker verification using a number of features and combining scores obtained from GMM speaker verification system, the improvement in accuracy is not dramatic. One issue to be investigated here is the speaker verification classifier. It is possible that GMM classifier is good for MFCC

features, but for different features either a different classifier or different classifier parameters may be needed. For example, Sant'Ana et al. (2006) use a new classifier for Hurst parameter features, which, they argue, is more suitable for this type of features than GMM.

Although according to the final result of the experiment, it is found that none of the features that are studied, when used alone, achieve speaker recognition accuracy better than with MFCC features, their combination with MFCC features do result in improvement of the speaker recognition accuracy. This shows that these features do contain additional speaker-related information as compared to MFCC. A 10% reduction of the EER is achieved in comparison to the baseline test from MFCC features (see Figure 4.11).

Chapter 5

Conclusion and future of speaker recognition

5.1 Conclusion

To understand how humans recognize different people by voice is important for a speaker recognition system. By analyzing the spectral content of people's voice, different kinds of speech features were generated. It is believed that the speech features contains the identifying information of different speakers. Speaker models were built using the speech features. By comparing the speaker models, speaker recognition was performed. In this thesis, two areas of speaker recognition were investigated, speaker changing point detection and speaker verification.

The task of speaker changing point detection is to automatically detect the points where the speaker changes in a continuous audio file. In the thesis, the task is performed under the following conditions. There are no more than two speakers speaking simultaneously in the audio file. There is no prior knowledge about the number of speakers in the file. In the thesis only a mono-channel speaker segmentation task is considered and the data for testing is assumed to be clean speech, thus no noise reduction scheme is considered. The thesis has concentrated on how different kinds of speech features affect the speaker changing point detection system and how to make use of them together in the system to improve the performance. In chapter 3, Mel Fre-

quency Cepstral Coefficients (MFCC) (Oppenheim & Schafer, 2004) features, Mel Line Spectrum Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst parameter features (pH) (Sant'Ana et al., 2006), Haar Octave Coefficients of Residue (HOCOR) (Zheng & Ching, 2004), and features based on fractional Fourier transform (MFCCFrFT) were investigated individually in the BIC speaker changing point detection system. The result shows that MFCC produces the best result. Other features do have extra speaker information compare to MFCC features. In chapter 3 a novel speaker changing point detection system using SVM was provided to combine different features mentioned above. The performance is improved in the proposed algorithm.

Besides combining different features to improve the speaker segmentation system, a feature selection scheme can be developed to improve the performance in future work. Selection scheme can be achieved by selecting the effective feature points in one kind of feature. It is not true that good performance of speaker recognition system can be achieved by using all the feature points. For example, the result of using 26 MFCC features may not always be better than the one using 13 MFCC features. By selecting the useful or effective feature points out of all feature points can reduce the computational complexity. More experiments are needed to test the performance of selecting feature points.

In the future work, other feature combination schemes can be considered. In the thesis, the BIC values of each feature are combined to low down the feature dimension. Other combination schemes such as weighting features and non-linear feature combination can be tried in the future work. Weighting feature and non-linear feature combination can be carried out within one kind of features. For example, MFCC feature points can be weighted before they are used in speaker segmentation system. Non-linear combination can be multiplying several feature points within one

kind of features. Similarly, weighting feature and non-linear feature combination can be applied between different kinds of features.

In Chapter 4 speaker verification was investigated. Speaker verification belongs to the area of speaker recognition technology. It is used more often since its performance is better in one-to-one matching when compared to speaker identification which is a one-to-many matching problem. In this thesis, attempts have been made to find effective speech features. Various speech features have been proposed by a number of authors and their performance evaluated; however, since the experimental conditions differed in all the cases, their performance cannot easily be compared. This required the speaker verification system using GMM to be built with the intention of testing different speech features under consistent experimental conditions. The following speech features were tested: Mel Line Spectral Frequencies (MLSF) (Cordeiro & Ribeiro, 2006), Hurst Parameter Related Features (pH) (Sant'Ana et al., 2006), Linear Prediction Residual Phase (rPhase) (Murty & Yegnanarayana, 2006), Haar Octave Coefficients of Residue (HOCOR) (Zheng & Ching, 2004), Minimum Variance Distortionless Response (MVDR) (Huang & Hansen, 2006), and features based on the fractional Fourier transform (FrFT) (Almeida, 1994) that include MFCC features calculated using FrFT and mean energy within critical bands (MECB). After analysing the result of each feature, MFCC was still found to be the best feature in speaker verification when using GMM. The later experiment which combined the result of each feature under SVM improved the equal error rate (ERR) performance of speaker verification system from 9.6% to 8.7% (see Chapter 4).

Results from the experiment indicate that MFCC is still the best individual feature among other speech features when used alone. The performance can be improved by combining different speech features. In the future work more effective speech features

are to be investigated. Research on feature selection and combination algorithms are also needed. According to the experiment result, although other features are not better than MFCC while using GMM models or BIC speaker changing point detection, they may be better while using different speaker models. The combination of features improves the performance indicates that these features contain useful speaker identity information.

5.2 Future of speaker recognition

The commercial application of speaker recognition has been steadily increasing since the mid-1980s, with a large number of companies currently offering this technology. Compared to other biometric approaches, such as fingerprint, iris and face recognition, speaker recognition has strengths. According to the study by the United Kingdom's Communications-Electronics Security Group (CESG), by comparing the false rejection rate and false acceptance rate of each biometric approach, voice verification performed reasonably well compared to other biometrics in certain applications (Mansfield et al., 2001).

The other strength of using speaker recognition is that it relies on a speech signal that is natural and unobtrusive to produce and can be obtained easily from almost anywhere using familiar tools, such as a telephone or the internet. Speaker recognition can be used with on remote users and applications already applying a speech interface. Speaker recognition is easy to use compared to some other biometrics such as face recognition, and a iris recognition. For example, speaker recognition can be embedded in smartcards because its coding is not complicated.

The problem with speaker recognition is that speech is not a consistent signal. It can be affected by illness of the speaker, emotion and the speaker's language. Additionally the channels that people speaker into may affect the speech signal by

introducing noise. Consequently a robust method for speaker recognition which can overcome variability is needed.

The future of speaker recognition is to solve the aforementioned problem and to improve the accuracy of system, making it more reliable in applications. The first thing that can be improved are the speech features for speaker recognition. After testing different kinds of acoustic features where were generated from the spectrum, the result shows that there are still no outstanding features. In addition to the low-level spectrum features, the speech signal conceives rich information in higher level, such as prosodic idiolect. It is necessary to investigate these features to improve the accuracy and robustness of the system.

The trend of speaker recognition is also to focus on the real world robustness and emphasis on unconstrained tasks. For example, it is necessary to collect different speech data in various environments. The unconstrained tasks include the speaker recognition in variable channels and noise conditions, text-independent speaker recognition. To invent a new model of speaker recognition is another approach, such as building a model of human ear and simulating the process of speaker recognition performed by this.

References

- Ajmera, J., McCowan, I., & Boulard, H. (2004, August). Robust speaker change detection. *IEEE Signal Processing Lett.*, 11(8).
- Almeida, L. (1994). The fractional fourier transform and time-frequency representations. In *IEEE Transactions on Signal Processing* (pp. 3084–3091).
- André, G., Sachin, S., & Hynek, H. (2002, May). A new speaker change detection method for two-speaker segmentation.
- Baker, L. (1975, February). The dragon system and overview. In *IEEE Trans. Acoust. Speech Signal Processing* (pp. 24–29).
- Bakis, R. (1997). Transcription of broadcast news shows with the ibm large vocabulary speech recognition system. In *Proceedings of the Speech Recognition Workshop* (pp. 67–72).
- Brillinger, D. (2001). *Time series: data analysis and theory*. Society for Industrial and Applied Mathematics.
- Campbell, J. (1997, September). Speaker recognition: a tutorial. In *Proceedings of IEEE* (Vol. 85).
- Chen, S., & Gopalakrishnan, P. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *DARPA Speech Recognition Workshop* (pp. 127–132).
- Christopher, J. (1998). A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery* (pp. 121–167).
- Cohen, A., & Lapidus, V. (1996, November). Unsupervised speaker segmentation in telephone conversations. In *Nineteenth Convention of Electrical and Electronics Engineers in israel* (pp. 102–105).
- Cordeiro, H., & Ribeiro, C. (2006, June). Speaker characterization with mlsf. In *Odyssey 2006: The Speaker and Language Recognition Workshop*. San Juan, Puerto Rico.
- Douglas, A., Thomas, F., & Robert, B. (2000). Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing* (pp. 19–41).
- Ellis, D., & Liu, J. (2005). speaker turn segmentation based on between-channel differences.
- Fernando, L., & Jeffrey, S. (2001). Biometric authentication technology: from the

- movies to your desktop. In *National Science and Technology Council*.
- Forney, G. (1973, March). The viterbi algorithm. In *Proceedings of the IEEE* (pp. 268–278).
- Furui, S., Sondhi, M., & Dekker, M. (1991). Speaker-independent and speaker-adaptive recognition techniques. In *Advances in Speech Signal Processing* (pp. 597–622).
- Gish, H., & Schmidt, M. (1994, April). Text-independent speaker identification. *IEEE Signal Processing Magazine*, 18–32.
- Huang, R., & Hansen, J. (2006, May). Advances in unsupervised audio classification and segmentation for the broadcast news and ngsu corpora. In *IEEE Transactions on Audio, Speech, and Language Processing* (pp. 907–919).
- Islam, T., & Kabal, P. (2000, September). Partial-energy weighted interpolation of linear prediction coefficients. In *IEEE Workshop on Speech Coding* (pp. 105–107).
- Keum, J., & Lee, H. (2005, December). Speaker change detection based on spectral peak track analysis for korean broadcast news. In *Fifth International Conference on Information, Communications and Signal Processing* (pp. 724–728).
- Kunzel, H. (1994). Current approaches to forensic speaker recognition. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification* (pp. 135–141).
- Kwon, S., & Narayanan, S. (2002). Speaker change detection using a new weighted distance measure. In *ICSLP2002*.
- Lance, C. (2003, December). *Hidden markov models and the baum-welch algorithm* (No. 4).
- Lathoud, G., & McCowan, I. (2003, April). Location based speaker segmentation. In *Acoustics, Speech, and Signal Processing, 2003 Proceedings. (icassp '03). 2003 IEEE International Conference on* (pp. 176–179).
- Lawrence, R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE* (pp. 257–286).
- Linde, Y., Buzo, A., & Gray, R. (1980, January). An algorithm for vector quantizer design. In *IEEE Transactions on Communications* (pp. 84–95).
- Lu, L., & Zhang, H. (2002). Real-time unsupervised speaker change detection.
- Mansfield, T., Kelly, G., Chandler, D., & Kane, J. (2001, March). *Biometric product testing final report*.
- Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The det curve in assessment of detection task performance. In *Proc. Eurospeech '97*.
- Matsui, T., & Furui, S. (1994, July). Comparison of text-independent speaker recognition methods using vq-distortion and discrete/continuous hmm's. In *IEEE Transactions on Speech and Audio Processing* (pp. 456–459).

- Melin, P., & Castillo, O. (2005). In *Hybrid intelligent systems for pattern recognition using soft computing* (pp. 223–240). Springer Berlin / Heidelberg.
- Mori, K., & Nakagawa, S. (2001). Speaker change detection and speaker clustering using vq distortion for broadcast news speech recognition. In *ICASSP01* (pp. 413–416).
- Murty, K., & Yegnanarayana, B. (2006, January). Combining evidence from residual phase and mfcc features for speaker recognition. In *IEEE Signal Processing Letters* (pp. 52–55).
- Nishida, M., & Kawahara, T. (2003). Unsupervised speaker indexing using speaker model selection based on bayesian information criterion. In *ICASSP*.
- NIST HUB-4E Broadcast News Evaluation*. (1997).
- Oppenheim, A., & Schafer, R. (2004, September). From frequency to quefrency: a history of the cepstrum. *Signal Processing Magazine, IEEE*(5), 95–106.
- Poritz, A. (1982, May). Linear predictive hidden markov models and the speech signal. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 1291–1294).
- Reynolds, D., Andrews, W., Campbell, J., Navratil, J., Peskin, B., Adami, A., et al. (2003, April). The supersid project: exploiting high-level information for high-accuracy speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 784–787).
- Reynolds, D., & Rose, R. (1995, January). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1), 72–83.
- Ribeiro, C., & Trancoso, I. (1999). Speaker adaptation in a phonetic vocoding environment. In *IEEE Workshop on Speech Coding*. Finland.
- Sant’Ana, R., Coelho, R., & Alcaim, A. (2006, May). Text-independent speaker recognition based on the hurst parameter and the multidimensional fractional brownian motion model. In *IEEE Transactions on Audio, Speech and Language Processing*.
- Shahin, I., & Botros, N. (1998, April). Speaker identification using dynamic time warping with stress compensation technique. In *Southeastcon’98 Proceedings IEEE* (pp. 65–68).
- Siegler, M., Jain, U., Ray, B., & Stern, R. (1997). Automatic segmentation, classification and clustering of broadcast news audio. In *Proceedings of the Speech Recognition Workshop* (pp. 97–99).
- Soong, F., & Rosenberg, A. (1986). On the use of instantaneous and transitional spectral information in speaker recognition. In *Proc. IEEE ICASSP* (pp. 877–880).
- Soong, F., Rosenberg, A., Rabiner, L., & Juang, B. (1985, April). A vector quanti-

- zation approach to speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 387–390).
- Speaker recognition*. (2006).
- Sugiyama, M., Murakami, J., & Watanabe, H. (1993). Speech segmentation and clustering based on speaker features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.
- Veith, D., & Abry, P. (1998, March). A wavelet-based joint estimator of the parameters of long-range dependence. *IEEE Trans. Inform. Theory*, 45(3), 878–897.
- Wan, V., & Campbell, W. (2000, December). Support vector machines for speaker verification and identification. In *Neural Networks for Signal Processing x, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop* (pp. 775–784).
- Wang, J., & Wang, J. (2005, October). Speaker recognition using features derived from fractional fourier transform. In *IEEE International Conference on Automatic Identification Advanced Technologies* (pp. 95–100).
- Wilcox, L., Chen, F., Kumber, D., & Balasubramanian, V. (1994). Segmentation of speech using speaker identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.
- Wu, T., Lu, L., Chen, K., & Zhang, H. (2003). Ubm-based real-time speaker segmentation for broadcasting news. In *ICASSP03*.
- Yapanel, U., & Hansen, J. (2003, September). A new perspective on feature extraction for robust in-vehicle speech recognition. In *Eurospeech* (pp. 1281–1284). Geneva, Switzerland.
- Zhang, T., & Kuo, C. (2001, May). Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transaction on Speech and Audio Processing*(4).
- Zheng, N., & Ching, P. (2004, May). Using haar transformed vocal source information for automatic speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 77–80). Montreal, Canada.

Appendices

The products of this thesis are the following:

- Graham Leedham, Vladimir Pervouchine and Haishan Zhong-Seeking patterns in the forensic analysis of handwriting and speech, Chapter in Pattern Recognition Technologies and Applications: Recent Advances (Eds. Brijesh Verma and Michael Blumenstein), In press 2007.
- Haishan Zhong, Vladimir Pervouchine, Graham Leedham, David Cho, “Using SVM to detect speaker changing points,” Submitted to International Conference on Bio-inspired Systems and Signal Processing, Portugal, 28-31 January 2008.
- Vladimir Pervouchine, Haishan Zhong, Graham Leedham, David cho and Haizhou Li, “Comparative study of several novel acoustic features for speaker recognition,” Submitted to International Conference on Bio-inspired Systems and Signal Processing, Portugal, 28-31 January 2008.