TEXT-INDEPENDENT SPEAKER RECOGNITION USING SOURCE BASED FEATURES

JANUARY, 2001

MASTER OF PHILOSOPHY

BRETT RICHARD WILDERMOTH

GRIFFITH UNIVERSITY
AUSTRALIA

Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Abstract

Speech signal is basically meant to carry the information about the linguistic message. But, it also contains the speaker-specific information. It is generated by acoustically exciting the cavities of the mouth and nose, and can be used to recognize (identify/verify) a person. This thesis deals with the speaker identification task; i.e., to find the identity of a person using his/her speech from a group of persons already enrolled during the training phase.

Listeners use many audible cues in identifying speakers. These cues range from high level cues such as semantics and linguistics of the speech, to low level cues relating to the speaker's vocal tract and voice source characteristics. Generally, the vocal tract characteristics are modeled in modern day speaker identification systems by cepstral coefficients. Although, these coefficients are good at representing vocal tract information, they can be supplemented by using both pitch and voicing information.

Pitch provides very important and useful information for identifying speakers. In the current speaker recognition systems, it is very rarely used as it cannot be reliably extracted, and is not always present in the speech signal. In this thesis, an attempt is made to utilize this pitch and voicing information for speaker identification.

This thesis illustrates, through the use of a text-independent speaker identification system, the reasonable performance of the cepstral coefficients, achieving an identification error of 6%. Using pitch as a feature in a straight

forward manner results in identification errors in the range of 86% to 94%, and this is not very helpful.

The two main reasons why the direct use of pitch as a feature does not work for speaker recognition are listed below. First, the speech is not always periodic; only about half of the frames are voiced. Thus, pitch can not be estimated for half of the frames (i.e. for unvoiced frames). The problem is how to account for pitch information for the unvoiced frames during recognition phase. Second, the pitch estimation methods are not very reliable. They classify some of the frames unvoiced when they are really voiced. Also, they make pitch estimation errors (such as doubling or halving of pitch value depending on the method).

In order to use pitch information for speaker recognition, we have to overcome these problems. We need a method which does not use the pitch value directly as feature and which should work for voiced as well as unvoiced frames in a reliable manner. We propose here a method which uses the autocorrelation function of the given frame to derive pitch-related features. We call these features the maximum autocorrelation value (MACV) features. These features can be extracted for voiced as well as unvoiced frames and do not suffer from the pitch doubling or halving type of pitch estimation errors. Using these MACV features along with the cepstral features, the speaker identification performance is improved by 45%.

Acknowledgments

Firstly I would like to thank my loving wife Ruth for all her love and emotional support during the writing of this thesis. I would also like to thank my supervisor Professor K.K. Paliwal for giving me this wonderful opportunity, not to mention his guidance and perseverance. It is through the help of these two special people that this thesis was possible.

I would also like to thank my fellow researchers in the signal processing laboratory for their comments, suggestions and encouragement.

Contents

List of Figures

Chapter 1

Introduction

Speaker recognition is the process of identifying a person on the basis of speech alone. Campbell defines it more precisely as *the use of a machine to recognize a person from a spoken phrase* [9]. It is a known fact that speech is a speaker dependent feature that enables us to recognise friends over the phone.

During the years ahead, it is hoped that speaker recognition will make it possible to verify the identity of persons accessing systems; allow automated control of services by voice, such as banking transactions; and also control the flow of private and confidential data [15].

While fingerprints and retinal scans are more reliable means of identification, speech can be seen as a non-evasive biometric that can be collected with or without the persons knowledge or even transmitted over long distances via telephone. Unlike other forms of identification, such as passwords or keys, a person's voice cannot be stolen, forgotten or lost. Speaker recognition allows for a secure method of authenticating speakers.

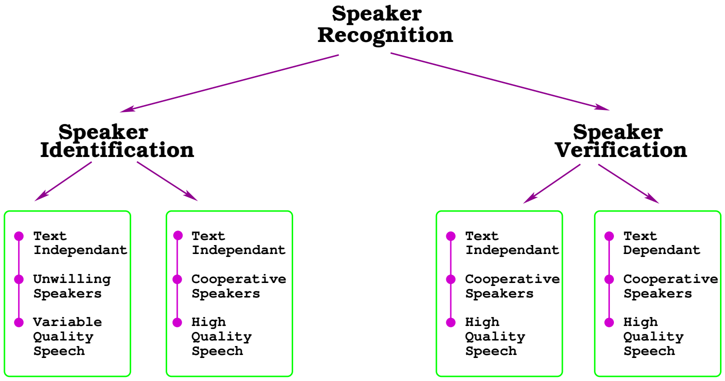
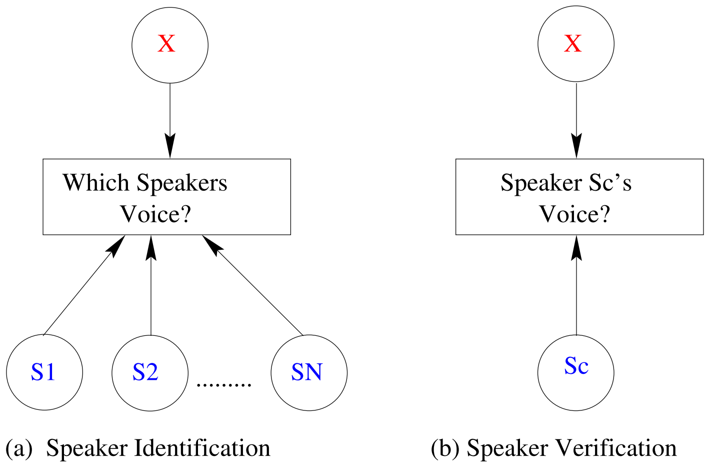


Figure 1.1: The Scope of Speaker Recognition [9].

During the *enrollment phase*, the speaker recognition system generates a *speaker model* based on the speaker's characteristics. The *testing phase* of the system involves making a claim on the identity of an unknown speaker using both the trained models and the characteristics of the given speech. Many speaker recognition systems exist and the following section will attempt to classify the many types of speaker recognition systems.

1.1 Classification of Automatic Speaker Recognition

This section covers the classification of speaker recognition systems (see Fig. 1.1), their differences and how the performance of such systems are accessed. Automatic speaker recognition systems can be divided into two classes depending on their desired function; Automatic Speaker Identification (ASI)



(a) Speaker Identification (b) Speaker Verification

Figure 1.2: Speaker Identification and Speaker Verification [39].

and Automatic Speaker Verification (ASV) systems [49]. ASI systems attempt to answer the question "*who are you?*", while Automatic Speaker Verification systems ask the question "*are you whom you claim to be?*", as shown in Fig. 1.2.

The role of an automatic speaker verification (ASV) system is to justify an identity claim made by the speaker. The decision of the verification system is strictly binary in the form of an accept or a reject [69]. If we represent the probability of a given utterance $\mathbf{x}$ belonging to the i^{th} speaker by $p_i(\mathbf{x})$, then the speaker verification task is simply:

$$\begin{aligned}if\ p_i(\mathbf{x}) > \mathbf{p_{THRESHOLD}} \quad \mathbf{ACCEPT} \\ else\ \ REJECT\end{aligned}$$

Generally, $p_{THRESHOLD}$ is an experimentally derived threshold.

The role of an automatic speaker identification (ASI) system is more complex. It is required to make a claim on the identity of the speaker from the N_S trained speakers in its user database; i.e.,

The speaker is identified as the j^{th} speaker,
if $p_j(\mathbf{x})$ results in the highest score among
all the N_S trained (enrolled) speakers.

Since the system is required to make N_S tests and decisions, the error of the system will increase with N_S, whereas the error of an ASV system is independent of N_S.

The errors of ASV/I systems can be classified into two groups: *false acceptance* (FA), accepting an impostor (ASV) or identifying a wrong person (ASI) and *false rejection* (FR), rejecting a true speaker (ASV) or not matching a speaker (ASI). Since the decision to reject or accept a speaker is defined by a threshold, the system can be designed to minimize the more costly of these two errors. A system designed to protect sensitive information would have a low decision threshold, which in turn would produce a low FA at the expense of a high FR, i.e. unauthorised personnel would be denied access at the expense of inconveniencing authorised personnel.

Since FA and FR are dependent on the threshold, the threshold can be chosen so that these two errors are equal, defined as the *Equal Error Rate* (EER) [7]. The equal error rate can be greatly affected by the utterances used in training and testing the system.

The ASV/I systems can be further classified into text-dependent and text-independent systems. The former requires the speaker to utter sentences based around a set of keywords for both training and recognition (testing) trials.

Due to the nature of text-dependent systems, someone could easily fool the system by playing back the recorded voice of a registered speaker [15]. Experience has also shown that humans function in a text-independent fashion. Therefore more focus is made on text-independent systems.

1.2 Contribution

The speech from a given speaker is modeled using spectral modeling techniques such as cepstral coefficients. These methods are extremely useful in representing vocal tract information. However, supplementary information based on pitch and intensity information is required to improve their performance. It has been shown by Atal [1] that the use of pitch contours as a feature can in fact improve the recognition performance. Furui [14] showed that the direct use of pitch can enable the system to better differentiate speakers, but the results were not very encouraging.

Pitch can not be used directly as a feature in a speaker recognition system because of the following two main reasons: First, the speech is not always periodic; only about half of the frames are voiced. Thus, pitch can not be estimated for half of the frames (i.e. for unvoiced frames). The problem is how to account for pitch information for the unvoiced frames during recognition phase. Second, the pitch estimation methods are not very reliable. They classify some of the frames unvoiced when they are really voiced. Also, they make pitch estimation errors (such as doubling or halving of pitch value depending on the method).

In order to use pitch information for speaker recognition, we have to overcome these problems. We need a method which does not use the pitch value directly as a feature and which should work for voiced as well as unvoiced frames in a reliable manner. We propose here a method which uses the autocorrelation function of the given frame to derive pitch-related features. We call these features the maximum autocorrelation value (MACV) features. These features can be extracted for voiced as well as unvoiced frames and do not suffer from the pitch doubling or halving type of pitch estimation errors. Using these MACV features as supplementary features with the cepstral features, the speaker identification error is reduced by 45%.

1.3 Outline of Thesis

The major goal of this thesis was to derive a set of source-based features that would improve the recognition accuracy of a text-independent speaker identification system.

Chapter 2 outlines the current state of speaker recognition technology. It begins with a brief overview of speech production as a means of explaining feature extraction, leading to an overview of many common classification methods used in modern day speaker recognition systems.

Chapter 3 outlines a text-independent GMM-based speaker identification system developed for evaluating the effectiveness of various feature sets. It begins with an overview of the corpra used during experimentation, an in depth look at the feature extractors used in evaluating the system, and the method of training the system. The chapter concludes with an outline of the system's performance.

The fourth chapter covers the importance of pitch in speaker identification and describes many simple pitch extraction methods used today. The use of pitch as a feature is shown to give extremely poor results. The use of voicing information is proposed and a more complex method, called power difference in spectra of subband (PDSS), is covered. Finally, the MACV features proposed here as a simpler and more effective features for representing pitch and voicing information are described.

The final chapter concludes the thesis with a summary and a list of possible directions for further work in this area.

Chapter 2

Text-independent Speaker Identification

The current day speaker identification system (shown in Fig. 2.1) consists of five different sections: signal acquisition, feature extraction (front-end processing), pattern matching and classification, decision logic and enrollment.

This chapter attempts to explain the current state of speaker identifica-

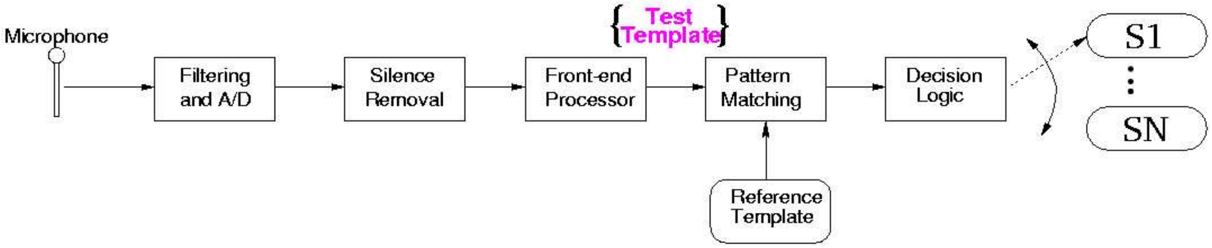


Figure 2.1: A basic speaker identification system.

tion technology. It begins by explaining the process of speech production in humans and addresses key audible cues used by listeners in identifying speakers. Using these relationships it will explain how speaker identification systems are able to use these features in identifying speakers. Several methods used in speaker identification are explained, highlighting the difference between template and stochastic models. The chapter concludes with a brief summary.

2.1 Speech and Feature Extraction

Speech is a complex signal. This section will attempt to shed some light on the subject, by showing how speech is produced and how speech is represented by speaker identification systems in the form of feature vectors.

2.1.1 Speech Production

The vocal system can be thought of as an acoustic tube terminating at the mouth on one end and the vocal cords at the other as can be seen in Fig. 2.2. In the average male, the vocal tract is approximately 17 cm in length, and the cross-sectional area of the vocal tract varies from zero (complete closure) to approximately 20 sq. cm [47]. Speech is produced by acoustically exciting the vocal tract, including the cavities of the mouth and nose (Fig. 2.3). Air enters the lungs via the normal breathing mechanism and is expelled through the trachea which causes the vocal cords to vibrate. The quasi-periodic pulses are then modulated by the pharynx, the mouth cavity, and sometimes the nasal cavity to produce speech [1]. The shape of the vocal tract is continually

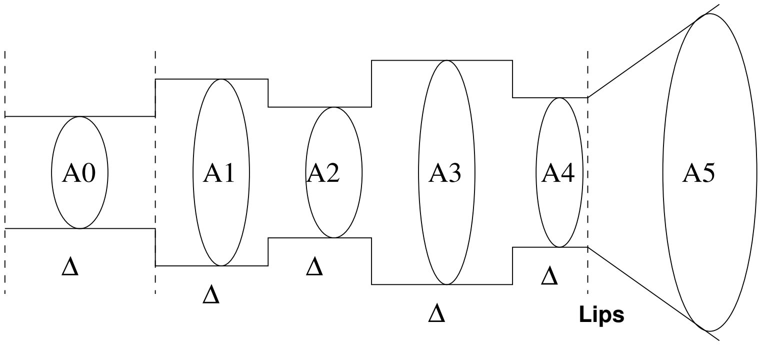


Figure 2.2: Acoustic tube model of speech production.

changing due to the position of the tongue, the jaw and the lips and in most cases all sound is radiated from the lips except for nasal consonants which radiate from the nose [61]. The nasal consonants defined by /m/,/n/, and /η/ are produced by constricting the vocal tract at some point along the oral passage way, lowering the vellum to allow coupling with the nasal cavity and allowing the sound to be radiated via the nostrils. The mouth then acts as a resonant cavity trapping the acoustic energy at natural resonant frequencies.

Sounds are generally classified by their mode of excitation, consisting of three important modes; voiced, unvoiced and plosives. Voiced sounds are produced as a result of exciting the vocal tract with a series of periodic pulses. An example of voiced sounds include vowels, semi-vowels, voiced stops and nasal consonants. The period of excitation is determined by the mass and tension of the vocal cords and is usually in the range of 60 to 400 Hz. Unvoiced sounds are produced when the vocal tract is excited by a noise-like turbulent flow of air at a point of constriction. Examples include

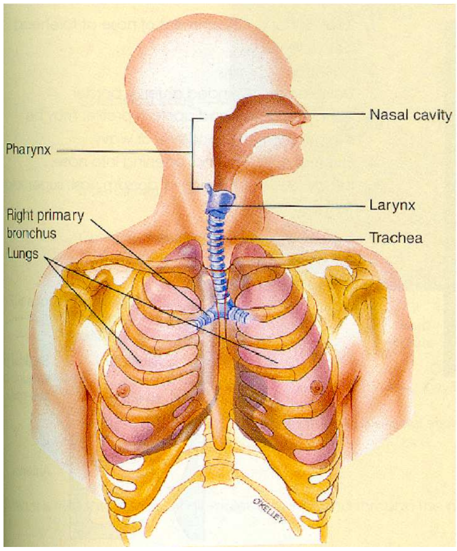


Figure 2.3: Speech production mechanism [78].

fricatives like f, s, sh, etc . Lastly plosives are generated by making a complete closure of the vocal tract, building pressure and releasing it abruptly. Plosives include stop consonants like b, p, g, t.

Listeners use many perceptual cues when recognizing speakers, cues that range from high level cues to low-level cues [51]. High level cues relate to semantics and linguistics of speech, including word usage, pronunciation, and other non-acoustic properties. These are thought to be related to life experiences incorporating place of birth, upbringing and education. These cues are commonly referred to as traits and are not always present in the speech signal. Low level cues are related to the speaker's vocal tract and voice source characteristics. These cues can be extracted from the speech signal through acoustic measurements.

A speaker is identified by both the physiological and behavioral characteristics in their voice. These characteristics are represented by the vocal tract characteristics (spectral envelope) and the voice source characteristics (supra-segmental features) [15]. The vocal tract characteristics are generally represented by linear prediction coefficients or cepstral coefficients.

2.1.2 The Linear Prediction Coefficients

The linear prediction coefficients (LPCs) capture the information about the short-time spectral envelope of speech. Although the LPCs represent important speech characteristics such as formant frequency and bandwidth, they are independent of pitch and intensity information [1]. A study undertaken by Atal [3] shows the effectiveness of LPCs in a speaker identification role. The modern day LP feature extractor, Fig. 2.4, consists of five major sections, preemphasis, frame blocking, windowing, autocorrelation analysis and LPC computation. This section outlines the role of these functional blocks in converting the speech signal to the LP coefficients [47].

Preemphasis Filtering

The digital speech $s(n)$ is captured by an analog-to-digital converter (ADC) at a sampling frequency f_s. The signal is then filtered by a first order FIR filter in the form of

$$H(z) = 1 - \alpha z^{-1} \quad , \tag{2.1}$$

where α typically lies in the range of 0.9 to 1.0 and reflects the degree of preemphasis. Preemphasis has the advantage of spectrally flattening the signal and making it less susceptible to finite precision effects at a later stage.

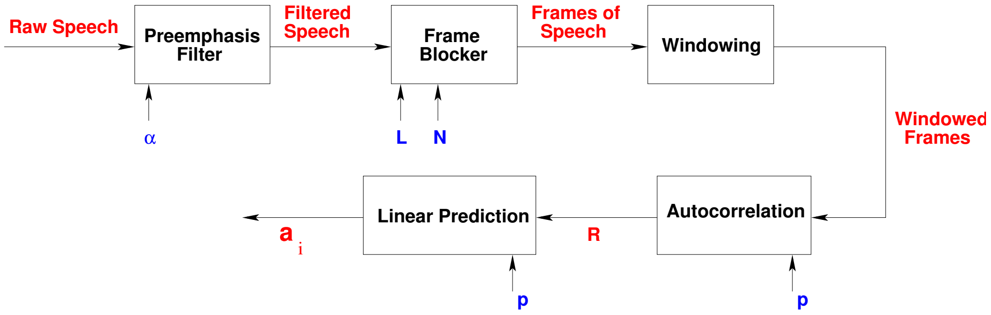


Figure 2.4: The LP feature extractor.

Fig. 2.5 shows the frequency response of a preemphasis filter with $\alpha = 0.95$. It should be noted that at $\omega = \pi$ the filter response is 32 dB higher than at $\omega = 0$. The phase response of the filter is unimportant as it has no effect on the perception of the speech.

The output of the preemphasis filter can be related to the input by the difference equation

$$\hat{s}(n) = s(n) - \alpha s(n-1) \qquad n = 0, 1, 2, \ldots, N-1 \tag{2.2}$$

Preemphasis should generally be applied to voiced speech, but the slight negative effect on unvoiced speech does not warrant limiting preemphasis [69]. LP feature extractors preemphasize the entire speech signal using a constant value of α. However it is also possible to apply preemphasis using a frame dependent value of α. i.e.

$$H(z) = 1 - \alpha(n)z^{-1} \quad , \tag{2.3}$$

where α(n) is a function of frame number. Normally in this scenario the

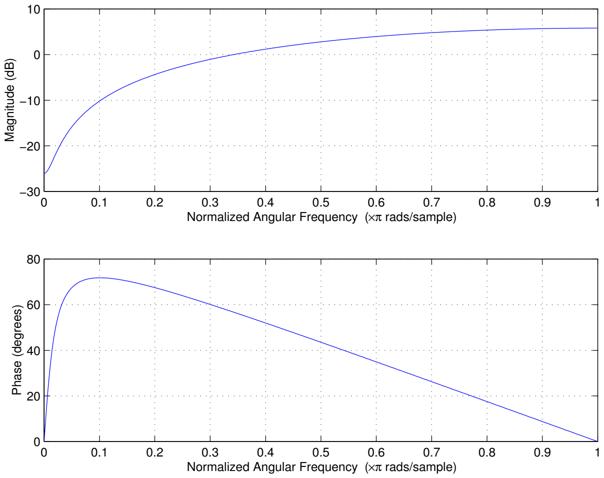


Figure 2.5: The frequency response of the preemphasis filter.

$\alpha(n)$ is dependent on the ratio of the first two autocorrelation values of the current frame.

Frame Blocking

The resulting preemphasized signal is blocked/split into equal frames of length N. The start of each frame is offset from the start of the previous frame by L samples, as illustrated in Fig. 2.6. The start of the second frame begins at L and the third would begin at $2L$ and so on. It can be seen that if $L \leq N$ then adjoining frames will overlap, and the LP spectral estimates will show a high level of correlation. In a system where the sampling frequency is 8 kHz, typical values of L and N are 80 and 160 respectively, which are related to a frame length of 30 ms with an update of 10 ms. If we define x_i as the i^{th} segment of the sampled speech $\hat{s}$ and I frames are required then

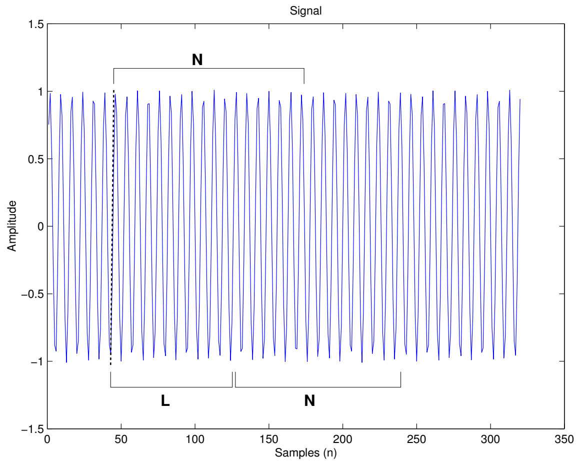


Figure 2.6: How the parameters N and L are utilized in the frame blocker.

the frame blocking process can be described as

$$x_i(n) = \hat{s}(Li + n) \qquad n = 0, 1, \ldots, N-1 \ \ , \ \ i = 0, 1, \ldots, I-1 \tag{2.4}$$

Windowing

In the short-term analysis of the speech signal, a rectangular window is implicitly used. This causes a spectral-leakage type of distortion in spectral analysis. The main reason for this is that the rectangular window has an abrupt discontinuity at the beginning and at the end of a frame. This distortion can be reduced by using a tapered window function $w(n)$. There exist many different windowing functions, Table 2.1 lists some of them. The resulting windowed segment is defined as

$$x(n) = x_i(n)\ w(n) \qquad n = 0, 1, \ldots, N-1 \tag{2.5}$$

Name of window	Time domain Sequence
Rectangular	1
Bartlett	$1-\frac{2\lvert n-\frac{M-1}{2}\rvert}{M-1}$
Blackman	$0.42-0.5cos\frac{2\pi n}{M-1}+0.08cos\frac{4\pi n}{M-1}$
Hamming	$0.54-0.46cos\frac{2\pi n}{M-1}$
Hanning	$\frac{1}{2}\left(1-cos\frac{2\pi n}{M-1}\right)$
Kaiser	$\frac{I_0\left[\alpha\sqrt{\left(\frac{M-1}{2}\right)^2-\left(n-\frac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\frac{M-1}{2}\right)\right]}$
Lanczos	$\left\{\frac{sin\left[2\pi\left(n-\frac{M-1}{2}\right)/(M-1)\right]}{2\pi\left(n-\frac{M-1}{2}\right)/\left(\frac{M-1}{2}\right)}\right\}^L$ $L>0,\ 1,\lvert n-\frac{M-1}{2}\rvert\leq\alpha\frac{M-1}{2},\ 0<\alpha<1$
Tukey	$\frac{1}{2}\left[1+cos\left(\frac{n-(1+\alpha)(M-1)/2}{(1-\alpha)(M-1)/2}\pi\right)\right]$ $\alpha(M-1)/2\leq\left\lvert n-\frac{M-1}{2}\right\rvert\leq\frac{M-1}{2}$

Table 2.1: A summary of some common windowing functions used in the LP feature extractor.

Autocorrelation Analysis

The autocorrelation analysis is used to extract important harmonic and formant properties from the speech. The autocorrelation function is a special case of the cross-correlation function [49] and is defined as:

$$R(i) = \frac{1}{N} \sum_{n=0}^{N-i-1} x(n)x(n+i) \qquad i = 0, \ldots, p \ \ , \tag{2.6}$$

where p is the LP analysis order and typical values range from 8 to 16. The zeroth element of the autocorrelation $R(0)$ provides a measure the of energy of the speech segment, and can be used for discarding silent frames.

LPC computation

The next functional block converts the $p+1$ autocorrelation coefficients into the LP coefficients. If we assume that the vocal tract is excited by a white noise signal having zero mean and unit variance, we can then represent the vocal tract by a p^{th} order auto regressive (all-pole) model of the form

$$H(z) = \frac{G_p^2}{1 + \sum_{k=1}^{p} a_{p,k} z^{-k}} \tag{2.7}$$

The unknowns in this equation ($G_p^2, a_{p,k}$, $\ k = 1, 2, \ldots, p$) are solved using the following two equations

$$R(0) = G_p^2 + \sum_{k=1}^{p} a_{p,k} R(k) \tag{2.8}$$

and

$$R(j) = -\sum_{k=1}^{p} a_{p,k} R(j-k), \qquad j = 1, 2, ..., p. \tag{2.9}$$

These equations (2.8 and 2.9) are commonly referred to as the Yule-Walker equations. It is possible, due to the Toeplitz nature of the matrix of autocorrelation coefficients, to solve these equations using a recursive method. The most popular and well known of these recursive methods is the Levinson-Durbin algorithm.

The Levinson-Durbin algorithm is initialized as:

$$\begin{aligned} a_{1,1} &= -\frac{R(1)}{R(0)} \qquad (2.10) \\ P_1 &= R(0)(1 - a_{11}^2) \qquad (2.11) \end{aligned}$$

and recursively implemented for $m = 2, \dots, p$ by

$$\begin{aligned} a_{m,m} &= -\frac{R(m) + \sum_{i=1}^{m-1} a_{m-1,i} R(m-i)}{P_{m-1}} \qquad (2.12) \\ a_{m,i} &= a_{m-1,i} + a_{m,m} a_{m-1,m-i} \qquad (2.13) \\ P_m &= P_{m-1}(1 - a_{m,m}^2) \qquad (2.14) \end{aligned}$$

On completion of the algorithm, the final solution for the LP coefficients is given as

$$a_i = a_{p,i}, \qquad 1 \leq i \leq p. \qquad (2.15)$$

$$G_p^2 = P_p \qquad (2.16)$$

A by-product of the Levinson-Durbin algorithm are the reflection (or, PARCOR) coefficients $k_m, m = 1, 2, ..., p$, defined as follows:

$$k_i = a_{i,i}, \qquad 1 \leq i \leq p. \qquad (2.17)$$

The reflection or PARCOR (partial correlation) coefficients are directly related to the non-uniform cross-sections of an acoustic tube used to model the vocal tract (see Fig. 2.2). The vocal tract can be considered as a cascade of p cylinders of equal length with various cross-sectional areas $A_1, A_2, \dots, A_p$. When air passes through the tube, the difference in cross-sectional areas causes reflection at the boundaries, where the reflection coefficients are denoted by k_m [47]. The reflection coefficients are related to the LP coefficients in a non-linear fashion, but provide all the information about the all-pole filter similar to the LP coefficients. They have been found to be useful for speech coding.

Other equivalent representations of LP information include log area ratios, inverse sine PARCORs, cepstral coefficients, etc. Among all these LP representations, the cepstral coefficients representation has been found to provide best performance for speech and speaker recognition and, hence, currently used for these applications.

2.1.3 Cepstral Coefficients

The cepstral coefficients provide a better alternative to the LP coefficients for speech and speaker recognition [25, 17, 3]. The cepstral coefficients can be derived either through LP analysis or Mel filter-bank analysis [47]. The former method generates features which are more commonly known as the LP cepstral coefficients. The M LP cepstral coefficients can easily be calculated from the p LP coefficients by

$$c_0 = ln\ G_p^2 \tag{2.18}$$

$$c_m = \frac{-ma_{pm} + \sum_{k=1}^{m-1} a_{pk} c_{m-k}(m-k)}{m}, \qquad 1 \leq m \leq p \tag{2.19}$$

$$c_m = \frac{\sum_{k=1}^{m-1} a_{pk} c_{m-k}(m-k)}{m}. \qquad p < m \leq M. \tag{2.20}$$

Alternatively, the Mel filter-bank cepstral coefficients (MFCCs) [47, 11] are calculated by taking the short-term power spectrum of the signal. The spectrum is then filtered with a set of 20 triangular windows used to simulate critical band filtering. The windows are approximately 300 mels wide (mels define the unit of measure in the mel-scale) and are spaced 150 mels apart and are sometimes referred to as mel scaled filters. However they only weight spectral values, they do not filter time domain signals. The relationship between the mel scale spectrum and the frequency spectrum is approximately given by,

$$Mel(f) = 2595 \; log(1 + \frac{f}{700}) \tag{2.21}$$

If the resulting log energy of the signal obtained from the k^{th} filter is denoted by X_k and M cepstral coefficients are required, then the MFCCs are derived through a discrete cosine transform (DCT) of the form

$$c_n = \sum_{i=1}^{20} X_k cos\left[n\left(k - \frac{1}{2}\right)\right], \qquad n = 1, 2, \ldots, M. \tag{2.22}$$

The complete MFCC extractor is shown in Fig. 2.7. The zeroth cepstral coefficient c_0 represents the average power of the frame and is usually not used; the c_1 coefficient reflects the distribution of energy between the high and low frequencies and the remaining coefficients show the fine spectral detail.

Cepstral Weighting

The performance of the cepstral coefficients can be improved by windowing the M cepstral coefficients with a liftering window w_m given by [29]:

$$w_m = \left[1 + \frac{M}{2}\sin\left(\frac{\pi m}{M}\right)\right], \qquad m = 1, 2, \ldots, M. \tag{2.23}$$

The function of this window is to deemphasize the lower and higher order cepstral coefficients. The lower order coefficients are deemphasized because

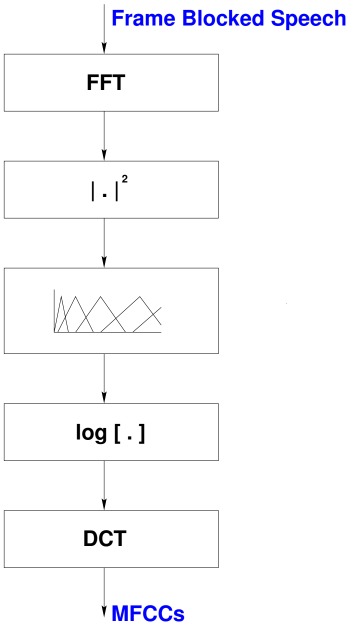


Figure 2.7: The complete MFCC feature extractor.

they get affected by spectral tilt and slowly varying additive noise distortion. The higher order cepstral coefficients are given less weight because they are not as effective for speaker recognition as the low order coefficients.

2.1.4 The Temporal Derivative of Cepstral Coefficients

The cepstral coefficients have been found to be a good representation of local spectral properties. This information can be extended to include temporal information; both first and second derivatives have been known to greatly improve the performance of an ASI system in which there exists a session variation (due to quality or time) between the trained and testing speech [73]. The temporal derivatives capture the information about change over many adjacent feature vectors. Though the cepstral derivative for the n-th frame can be obtained by subtracting the preceding cepstral vector $\mathbf{c}(n-1)$ from the current cepstral vector $\mathbf{c}(n)$, it is not very effective. Instead the first-order derivative is computed as the slope of the least squares linear fit over a finite window [47]; that is

$$\Delta c_m(n) = \frac{\sum_{k=-\Theta}^{\Theta} k c_m(n+k)}{\sum_{k=-\Theta}^{\Theta} k^2} \;, \qquad m = 1, 2, \ldots, M. \tag{2.24}$$

where Θ defines the length of the window (a typical value is $\Theta = 3$). These first-order derivatives of cepstral coefficients are called the delta cepstral coefficients. The second-order derivative of cepstral coefficients $\Delta\Delta\mathbf{c}(n)$ (also known as delta-delta or acceleration coefficients) are computed by taking the first-order derivative of the delta coefficients. The temporal derivatives are used to extend the original feature vector; i.e., the final feature vector $\mathbf{o}(\mathbf{n})$ consists of not only cepstral coefficients but the delta and acceleration coefficients also. Thus, the final $3M$-dimensional feature vector $\mathbf{o}(n)$ for the n-th frame is given by

$$\mathbf{o}'(n) = (\mathbf{c}'(n), \Delta\mathbf{c}'(n), \Delta\Delta\mathbf{c}'(n)), \tag{2.25}$$

where the prime denotes the transpose of a vector.

2.2 Pattern Matching and Classification

Speaker identification is basically a pattern classification problem preceded by a feature extraction stage [49]. Given a sequence of feature vectors representing the given test utterance, it is the job of the classifier to find out which speaker has produced this utterance [9]. In order to carry out this task, the acoustic models are constructed for each of the speakers from its training data. In the classification stage, the sequence of feature vectors representing the test utterance is compared with each acoustic model to produce a similarity measure that relates the test utterance with each speaker. Using this measure, the speaker identification system recognizes the identity of the speaker.

Various types of classifiers have been used for speaker identification. These can be grouped into either template or stochastic based classifiers. This section covers the classifiers in both groups, beginning with template ones.

2.2.1 Template Models

Template model based classifiers are considered to be the simplest of all classifiers. Thus it is understandable that the earliest classifiers belonged to this group. The most common of the template models used dynamic time warping and vector quantization.

Dynamic Time Warping

Dynamic time warping (DTW) was deployed in the earlier classifiers used for speaker identification. It is useful for text-dependent speaker recognition. It was a very popular method used in the 1980s [69]. However it has now been displaced by HMMs.

Due to changes in speaking rate, a speaker speaking the same text twice exhibits timing differences in two utterances. The problem of time alignment is addressed by the DTW algorithm through warping a template (or model) in an attempt to align key similarities between test utterance and training templates. The DTW algorithm also combines both the warping and distance measurement into one simple procedure [65, 66, 70].

Using the Bellman optimality principle, DTW is able to find an optimal path through the numerous possibilities that exists in comparing a test utterance with the training template. Given a reference (training) template R and a test utterance T consisting of N_R and N_T frames respectively, the DTW is able to find a function $m = w(n)$ which maps the time axis n of T to the time axis m of R.

DTW searches frame by frame through T to find the best frame in R in which to make the comparison

$$D = \min_{w(n)} \left[\sum_{n=1}^{T} d(T(n), R(w(n))) \right] \quad , \tag{2.26}$$

where d is a measure of the distance between the n^{th} frame of T and the

$w(n)^{th}$ frame of R, and D is considered as the distance corresponding to the best path or best match.

Given a sequence of feature vectors representing the test utterance, DTW is able to measure the best-match distances of the test utterance from all the reference templates. The system evaluates these distances and makes a decision about speaker identification in favor of the speaker whose reference template produces the lowest distance. DTW is used extensively in text-dependent speaker identification systems [9].

Vector Quantization

The DTW-based method is used for text-dependent speaker recognition. If the aim is to perform text-independent speaker recognition, one possible method could be to use all the feature vectors of a given speaker occurring in the training data to form this speaker's model. However, this is not practical as there are too many feature vectors in the training data for each speaker [61]. Therefore, a method of reducing/compressing the number of training vectors is required. It is possible to compress the training data by using a VQ (Vector Quantization) codebook consisting of a small number of highly representative vectors, that efficiently represent the speaker-specific characteristics [72, 74]. Note that the VQ-based classifiers were popular in earlier days for text-independent speaker recognition, but these days they are replaced by Gaussian mixture model based classifiers.

A formal method of constructing the VQ codebook is given in [34]. A conceptual illustration is included here in Fig. 2.8. Consider a set of I training feature vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I)$ of one particular speaker. These are shown by blue dots in a two dimensional Euclidean space in Fig. 2.8(a). Suppose that we want to represent these vectors in terms of a codebook consisting of N_C code vectors (where $N_C < I$). This is done as follows:

The feature vectors are initially grouped into a single partition, where the border of this partition is shown by a green dashed ellipse (Fig. 2.8(b)). The partition is then assigned a centroid C_p (in red), representing the mean of the partition (Fig. 2.8(c)). i.e.

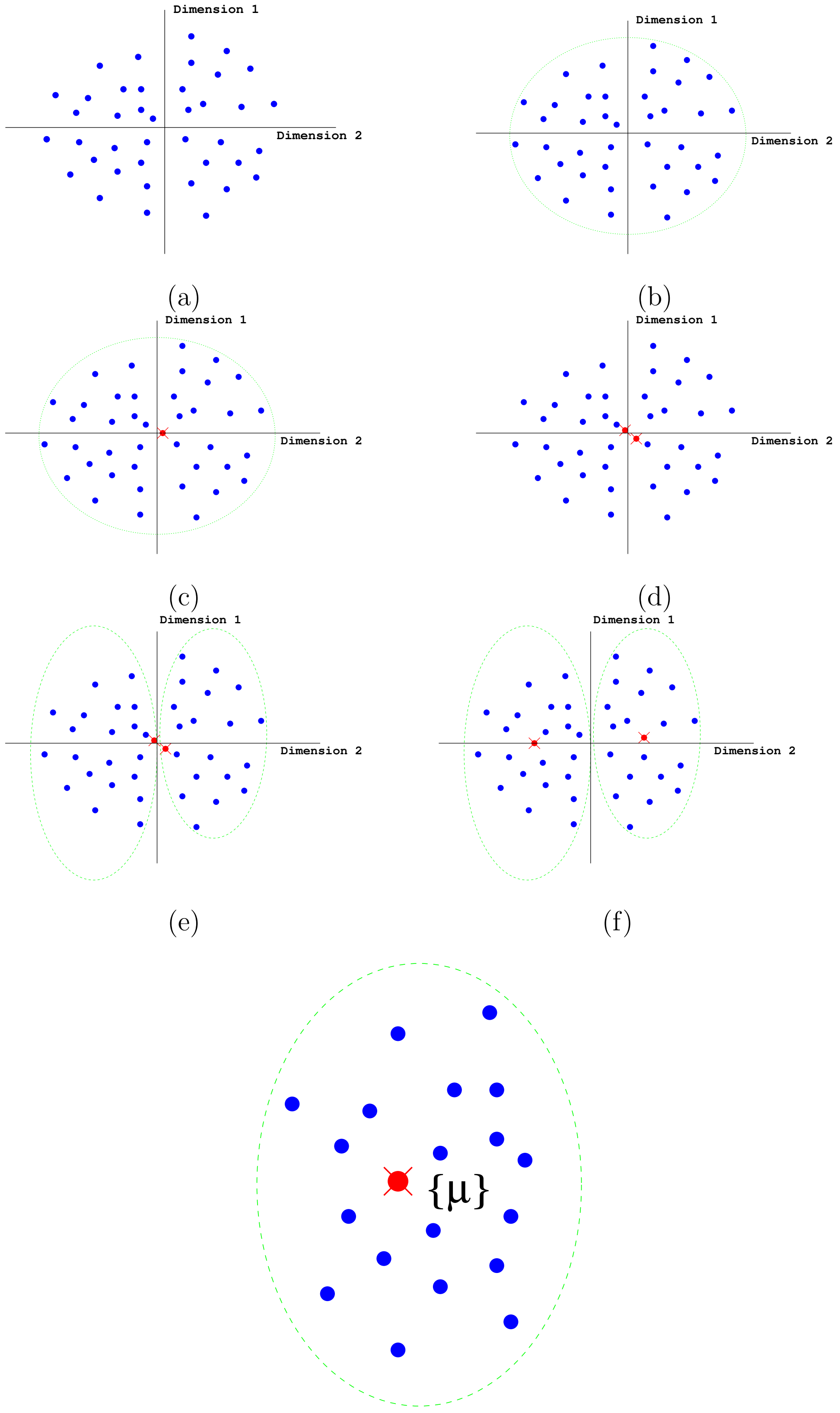
$$C_p = \frac{\sum_{i=1}^{I} \mathbf{x}_i}{I}, \tag{2.27}$$

The centroid is then split to form the following two new code vectors (C_{s1} and C_{s2}) (as shown in Fig. 2.8(d)):

$$\begin{aligned} C_{s1} &= (1-\epsilon)C_p \\ C_{s2} &= (1+\epsilon)C_p \quad , \end{aligned}$$

where ϵ refers to the splitting factor and is typically in the range of 0.001 to 0.0001. Using these two code vectors, the training vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I$ are encoded in terms of these two code vectors using a Euclidean distance measure and two new partitions are created by regrouping the training vectors to each of the two code vectors. Centroids for these two partitions are computed and are used as new code vectors. This process of partitioning and centroid computation is continued until the average distortion between centroids and training vectors is minimized (Fig. 2.8(e)). If the Euclidean distance between the j^{th} code vector $\mathbf{C}_j$ and the i^{th} feature vector $\mathbf{x}_i$ is denoted by $d(\mathbf{x}_i, \mathbf{C}_j)$, then the average distortion of the I feature vectors is computed as follows:

$$D = \frac{1}{I} \sum_{i=1}^{I} \min_{1 \le j \le M} d(\mathbf{x}_i, \mathbf{C}_j) \tag{2.28}$$



(a) (b) (c) (d) (e) (f) (g)

Figure 2.8: The process of VQ codebook generation; the features are shown by blue dots, the group boundary in green and the centroids are in red.

The two code vectors are split into four code vectors and the same process of partitioning and centroid computation is repeated to find four code vectors. Using this binary splitting procedure, we can compute the VQ codebook with N_C code vectors (Fig. 2.8(g)).

One codebook with N_C code vectors is computed for each of the N_S speakers enrolled in the training phase. During the recognition phase, the feature vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L$ representing the test utterance are encoded in terms of their nearest code vectors from the code book of each of the N_S speakers. The total distortion for the i^{th} speaker is computed by

$$D^i = \sum_{l=1}^{L} min_{1 \leq j \leq N_C} d(\mathbf{y}_l, \mathbf{C}_j^i), \tag{2.29}$$

where C_j^i is the j-th code vector of the i-th speaker's code book.

Once these N_S distances are computed, the speaker identification system classifies the test utterance to a speaker whose VQ codebook results in the least distortion; i.e.,

$$i^* = arg\ min_{1 \leq i \leq N_S}\ D^i. \tag{2.30}$$

2.2.2 Stochastic Models

Currently, most of the speaker recognition systems are based on stochastic models. Stochastic models provide better flexibility and more meaningful results in the form of probabilistic scores [69]. In a stochastic model based classifier, the pattern matching procedure requires the computation of the likelihood of a test utterance given the speaker model.

Let λ^s be the stochastic model for the s-th speaker derived from the training data of this speaker. We will have N_S stochastic models for the

N_S speakers (one model for each speaker). Let $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L)$ be the sequence of the feature vectors representing the test utterance (having L frames). Our aim is to identify the speaker who has spoken this test utterance from the group of N_S speakers. This is done by computing the probability

$$p(Y|\lambda^s) = p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L|\lambda^s), \tag{2.31}$$

for $s = 1, 2, ..., N_S$ and deciding the identity of the speaker on the basis of

$$s^* = arg\ max_{1 \leq s \leq N_S}\ p(Y|\lambda^s). \tag{2.32}$$

If there is no correlation between the feature vectors of successive frames (i.e., they are independent), then Eq. (2.31) can be written as follows:

$$p(Y|\lambda^s) = \prod_{i=1}^{L} p(\mathbf{y}_i|\lambda^s), \tag{2.33}$$

Thus, our task is to compute the probability of a test vector given the speaker model; i.e., $p(\mathbf{y}_i|\lambda^s)$.

There are a number of methods recently proposed in the literature to compute this probability. The major ones are the Gaussian Mixture Model [51, 52, 53], the Hidden Markov Model [9], and Neural Networks [10]. These methods are briefly explained in the following sections.

Gaussian Mixture Model

The Gaussian Mixture Model (GMM) based method is used for text-independent speaker recognition. The GMM was developed in 1990 by Reynolds [51], and from the very start it showed promise in obtaining a high level of accuracy in text-independent applications. The motivation for the GMM comes from the need to model the acoustic space of a speaker in terms of a few acoustic classes (each class representing approximately one phoneme) in a simple and reliable manner [53]. This is done by assuming the probability of a feature vector of the n^{th} frame $p(\mathbf{y}_n|\lambda^s)$ to be a linearly weighted mixture of M multidimensional Gaussian probability density functions (PDFs); i.e.,

$$p(\mathbf{y}_n|\lambda^s) = \sum_{i=1}^{M} p_i^s b_i^s(\mathbf{y}_n), \tag{2.34}$$

where $b_i^s(\mathbf{y}_n)$ is the Gaussian PDF associated with the i-th mixture component (or, acoustic class) with mean μ_i^s and covariance matrix Σ_i^s; i.e.,

$$b_i^s(\mathbf{y}_n) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_i^s|}} e^{-\frac{1}{2}(\mathbf{y}_n-\mu_i^s)'(\Sigma_i^s)^{-1}(\mathbf{y}_n-\mu_i^s)}. \tag{2.35}$$

Here D is the dimensionality of the feature space. The mixture weights p_i^s, $i = 1, 2, ..., M$, in Eq. (2.34) satisfy the constraint $\sum_{i=1}^{m} p_i^s = 1$. The covariance matrix used in Eq. (2.35) is assumed to be diagonal. This is done for the following two reasons [53]: 1) It reduces the computational load, and 2) The cepstral features (normally used in speaker recognition systems) show a high degree of independence.

Collectively the s-th speaker's GMM model is represented by M components each consisting of $p_i^s, \mu_i^s, \Sigma_i^s$ (see Fig. 2.9); i.e.,

$$\lambda^s = \{p_i^s, \mu_i^s, \Sigma_i^s\}, \qquad 1 \leq i \leq M. \tag{2.36}$$

Since most of the spoken languages have about 30 to 40 phonemes, the value of M in a GMM is normally taken to be 32. The process of computing the probability of a feature vector given a GMM model is illustrated in Fig. 2.10.

For estimating the speaker model parameters from the training data, the expectation-maximization (EM) algorithm [53, 60, 59] is used. The EM

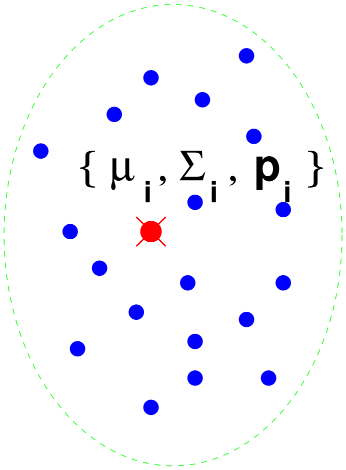


Figure 2.9: One component of a GMM speaker model.

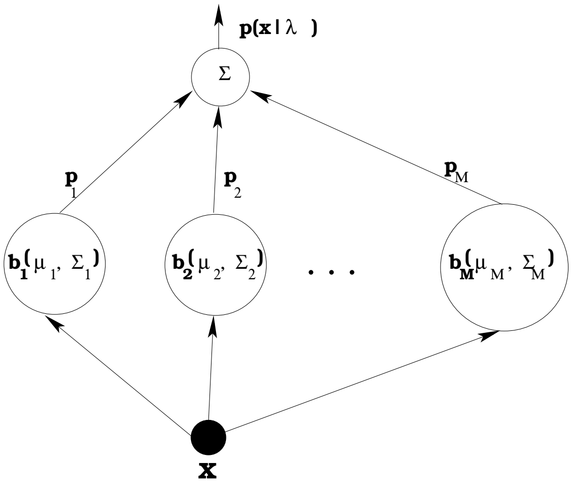


Figure 2.10: The process of computing the probability of a feature vector given a GMM model [53].

algorithm uses a maximum likelihood procedure for computing the GMM model parameters. It consists of two steps: an E-step (Expectation) and a M-step (Maximization). Let us assume that we have I feature vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I$ for a given speaker in the training data. The GMM model parameters for this speaker are initialized using a k-means algorithm, much like the one used in VQ. The EM algorithm for computing the GMM model parameters for the given speaker is given below. Note that we have dropped the speaker specific superscript s for clarity reasons.

The E-Step: Posterior probabilities are calculated for all the training feature vectors of the given speaker using

$$p(i|\mathbf{x}(n), \lambda) = \frac{p_i b_i(\mathbf{x}(n))}{\sum_{k=1}^{M} p_k b_k(\mathbf{x}(n))} \tag{2.37}$$

The M-Step: The M-step uses the posterior probabilities from the E-Step to estimate model parameters as follows:

$$\hat{p}_i = \frac{1}{I} \sum_{n=1}^{I} p(i|\mathbf{x}(n), \lambda), \tag{2.38}$$

$$\hat{\mu}_i = \frac{\sum_{n=1}^{I} p(i|\mathbf{x}(n), \lambda)\mathbf{x}(n)}{\sum_{n=1}^{I} p(i|\mathbf{x}(n), \lambda)}, \tag{2.39}$$

and

$$\hat{\Sigma}_i = \frac{\sum_{n=1}^{I} p(i|\mathbf{x}(n), \lambda)(\mathbf{x}(n) - \mu_i)(\mathbf{x}(n) - \mu_i)'}{\sum_{n=1}^{I} p(i|\mathbf{x}(n), \lambda)} \tag{2.40}$$

Set $p_i = \hat{p}_i$, $\mu_i = \hat{\mu}_i$ and $\Sigma_i = \hat{\Sigma}_i$, and iterate the sequence of E-step and M-step a few times till convergence is reached. On each iteration of the EM algorithm, the variance is limited by a variance floor to reduce singularities in the final model [55]. The iterative process is normally carried out 10 times, at which point the model is assumed to converge to a local maximum.

Hidden Markov Model

A stochastic model usually used for modeling sequences is the Hidden Markov Model (HMM) [47]. The HMM-based classifiers are useful for text-dependent speaker recognition. The HMM consists of two embedded stochastic processes as each observation (feature) vector is also a stochastic function of each state. The underlying stochastic function is not directly observable (it is hidden) and the HMM can only be viewed through another set of stochastic processes that produce the observation [69]. The HMM is a finite-state machine in which each state has an associated PDF for the feature vector.

A simple 5 state HMM is shown in Fig. 2.11. The states are connected by a transition network, which allows transition from one node to any node in the network including itself. The transitional probabilities a_{ij} define the probability of moving from one state to the next. The first and the last states are non-emitting states; having self-transitional probabilities of zero associated with them. For each emitting state, the sum of all possible N transitional probabilities must equal one, i.e. a transition must occur from each state [50]. That is

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.41}$$

When the i-th state is visited at the nth frame (i.e.; $q(n) = i$), it produces a feature vector $\mathbf{y}_(n)$ with probability $p(\mathbf{y}_(n)|q(n) = i)$.

The probability of a sequence of the feature vectors $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L)$ representing the test utterance (having L frames) conditioned on model λ^s is given earlier by Eq. (2.31). This equation can be rewritten for the case of

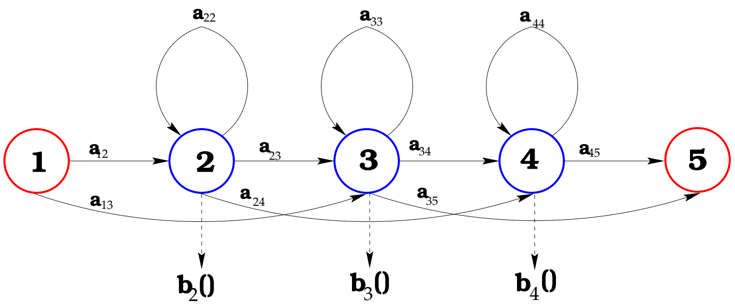


Figure 2.11: A simple left to right HMM

HMM as follows:

$$p(Y|\lambda^s) = \sum_{q(1),q(2),\ldots,q(L)} \prod_{n=1}^{L} p(\mathbf{y}_n|q(n),\lambda^s)p(q(n)|q(n-1),\lambda^s), \qquad (2.42)$$

This probability can be computed in a computationally efficient fashion using the forward-backward algorithm [30, 50]. In the test phase, the Viterbi algorithm is used for this purpose which is even faster than the forward-backward algorithm. HMM-based systems have shown performance better than the conventional template-based text-dependent speaker recognition systems [69].

2.2.3 Neural Networks

Classifiers based on Neural Networks (NN) are used in both text-dependent and text-independent speaker identification and speaker verification systems [4, 5, 44, 43]. The NN is extremely efficient at learning complex mappings between inputs and outputs and is able to approximate posterior probabilities for the trained classes. The NNs are able to approximate non-linear decision surfaces and exhibit a high level of parallelism. The NN consists of small functional units (neurons) that are interconnected to produce the desired global transfer function of the NN. There exist many forms of neural networks, these include the multi-layer perceptron (MLP) [43], the radial bias function (RBF) [44], and learning vector quantiser (LVQ) [22, 4]. The most common of these is MLP.

The MLP consists of an input layer, a number of hidden layers and an output layer (see Fig. 2.12) . The input layer is a non-functional layer responsible for fanning the inputs to all neurons in the hidden layer. The remaining layers however are functional characterised by their weighted inputs and non-linear (activation) functions. Each neuron in the output layer relates directly to a class. Input is fed into the MLP via the input neurons and each output neuron contains the resulting posterior probability for that particular class. The input is then classified into the class whose corresponding output neuron has the highest score.

In speaker recognition, the neural networks can take many forms. The two most common are: 1) a single MLP is trained with N_s output neurons, where N_s is the number of trained speakers, and 2) a MLP is trained for each speaker (a total of N_s MLPs) containing two output neurons relating to the trained speaker and the rest of the population.

The MLP is trained using the error back-propagation algorithm [10]. The error back-propagation algorithm is iterative in nature, where the weights of the MLP are refined during each iteration. Initially the weights of the MLP are set randomly in the range of -0.5 to 0.5. The algorithm is performed in two passes, the forward pass and the backward pass.

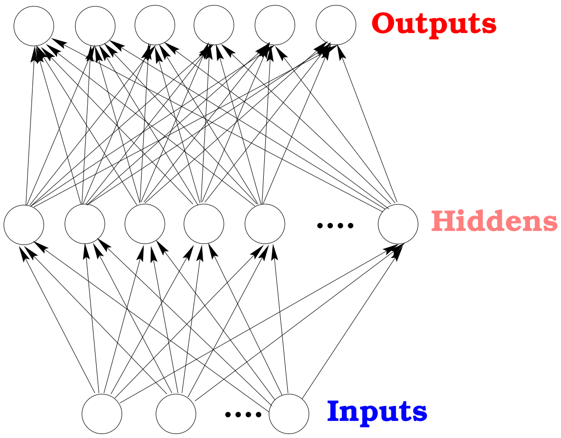


Figure 2.12: A two layered neural network

During the forward pass all the training vectors and there corresponding labels are presented to the MLP and an overall error is found. The output of the k^{th} output neuron given the n^{th} training vector $(\mathbf{y}_n)$ is

$$O_k(n) = f\left(\sum_i w_{ki}\ f\left(\sum_j w_{ij}\mathbf{y}_n(j)\right)\right), \tag{2.43}$$

where the summation j is over all input neurons and summation i is over all hidden neurons. Here w_{ij} is the weight associated with the connection of the i^{th} neuron of given layer with the j^{th} neuron of the of the preceeding layer. The activation function is a sigmoid-like function

$$f(x) = \frac{1}{1+e^{-x}}. \tag{2.44}$$

If we let $d_k(n)$ represent the desired output of the k^{th} output neuron given the n^{th} training vector, then the total-squared error of the MLP is defined as

$$E = \sum_n \sum_k (O_k(n) - d_k(n))^2 \tag{2.45}$$

During the backward pass the weights are modified to minimise the error, starting at the output layer and following into to the hidden layer. The process of calculating E in the forward pass and refining the weights in the backward pass is iterated numerous times until the MLP weights converge to a local minimum error.

During testing, each of the L feature vectors $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_L})$ are given as input to the MLP. For the n^{th} input feature vector $(\mathbf{y}_n)$ the corresponding output of the neural network $O_k(n)$ is generated, this is directly related to the posterior probability $p(\lambda^s|\mathbf{y}_n)$. If we assume that the a *priori* probability of each speaker is same, then the probability of sequence $\mathbf{Y}$ for a given model λ^s is

$$P(\mathbf{Y}|\lambda^s) = \prod_{n=1}^{L} p(\lambda_s|\mathbf{y}_n) \tag{2.46}$$

This probability is then used for speaker identification using Eq. 2.33.

2.3 Summary

The way speech is produced is important in understanding how features represent the cues used by listeners in identifying speakers. Speech is produced by acoustically exciting the vocal tract including the cavities of the mouth and nose. Listeners use many perceptual cues when recognizing speakers, cues that range from high level cues, such as semantics and linguistics of the speech to low level cues represented by both vocal tract properties and voice source characteristics.

In modern day speaker recognition systems, the cepstral coefficients representing the smooth power spectral envelope of speech are used as features. These features can be extracted from the speech signal through Linear prediction (LP) analysis. The linear prediction coefficients (LPCs) are derived from the speech signal using five functional steps: preemphasis, frame blocking, windowing, autocorrelation and LPC computation. The LPCs are then converted to cepstral coefficients using a recursion relation.

Alternatively the cepstral coefficients can be derived using a Mel filter bank analysis. This can be carried out in the following four steps: 1) Compute the energy (or, power) spectrum using the fast Fourier transform algorithm, 2) Warp the frequency axis nonlinearly to mel scale, 3) Construct a bank of triangular-shaped filters covering the mel frequency axis uniformly and apply them to the energy spectrum to get the output energies of individual filters, and 4) Compute the cepstral coefficients through discrete cosine transform (DCT) of the logarithm of the filter-bank energies.

The zeroth cepstral coefficient c_0 represents the average power of the speech segment and is usually not used. Only M cepstral coefficients from c_1 to c_M (where M is typically 10) are used as features. It has been observed that the first and second order temporal derivatives of the cepstral vector sequence are useful for speaker recognition. Therefore, the cepstral coefficients are concatenated with their first and second derivatives and the extended set of $3M$ coefficients are used as features.

Given a sequence of feature vectors representing the test utterance, it is the job of the classifier to carry out speaker identification. For this, it needs to model each of the N_S speakers enrolled at the training phase for speaker identification. This model should represent the feature space of a given speaker in a compact, accurate and reliable manner. These models are computed from the training data. In the test mode, the speaker identification system compares the sequence of feature vectors representing the test utterance with each of the N_S speakers' models and decides the identified speaker to be the one whose model shows maximum similarity.

Depending on the type of models used to represent the speakers, the classifiers used for speaker identification can be grouped into two major types: template-based and stochastic model based classifiers. Template-based classifiers are considered to be the simplest classifiers. The most common template-based classifiers are based on Dynamic Time Warping (useful for text-dependent speaker recognition) and Vector Quantization (useful for text-independent speaker recognition). Stochastic models provide more flexibility and better results. The stochastic model based classifiers use the Gaussian Mixture Model (useful for text-independent speaker recognition), the Hidden Markov model (useful for text-dependent speaker recognition), and Neural Networks to model a speaker's acoustic space.

The Gaussian Mixture Model (GMM) based classifier is used in our thesis for carrying out speaker identification experiments. The details of the GMM-based system and the speaker identification experiments are covered in the following chapter.

Chapter 3

System Description and Performance

This chapter outlines the text-independent speaker identification system developed here, including the training and testing conditions and the performance of the system in identifying speakers.

This chapter will also outline the components of the speaker identification system and their key operating parameters. Initially, speech is transformed by means of signal processing methods into frame-based acoustic features. Each speaker is represented by one Gaussian Mixture Model which is computed from his/her training data. During testing the GMM based models are used to compute speaker likelihoods. These likelihoods are used to classify the speakers so that a decision on the identity of the speaker can be made.

This chapter begins by explaining the corpora used in evaluating the effectiveness of the speaker identification system which is used throughout the remainder of this thesis.

3.1 Corpus

Closed set identification was conducted using TIMIT, NTIMIT and the three IISC databases. These databases were chosen for various reasons. Firstly, the TIMIT databases are widely used and publicly accessible, facilitating our need to compare our results with those of others. Secondly, these databases consist of natural continuous speech which closely resembles real life conditions.

3.1.1 TIMIT Database

TIMIT (Texas Instruments Massachusetts Institute of Technology) [19] database allows identification to be done under almost ideal conditions. Therefore, any recognition errors that occur should only be caused by overlapping speaker distributions [54]. The TIMIT database consists of 630 speakers, 70 % male and 30 % female from 10 different dialect regions in America. Each speaker has approximately 30 seconds of speech spread over ten utterances. The speech was recorded using a high quality microphone in a sound proof booth at a sampling frequency of 16 kHz, with no session interval between recordings.

The speech is designed to have a rich phonetic content, which consists of 2 dialect sentences (SA), 450 phonetically compact sentences (SX) and 1890 phonetically diverse sentences (SI). The dialect sentences developed by SRI are spoken by all speakers and were designed to show the variability introduced by the different dialects of the speakers. The phonetically compact sentences were designed by MIT and their purpose was to provide a good coverage of phoneme pairs. Each speaker reads five of these sentences and each sentence is read by seven speakers. The speakers spoke three phonetically diverse sentences that were directly acquired from existing text sources - Brown Corpus and the Playwrights dialog.

3.1.2 NTIMIT Database

NTIMIT consists of exactly the same speech as TIMIT that has been passed through a local or long distance telephone loop. Through the use of an "*artificial mouth*", each sentence was directly coupled to a carbon button telephone. The speech was then relayed to a local or long distance central office where it was looped back and recorded. The NTIMIT database can be considered to be TIMIT speech suffering from a degradation due to carbon button transducers and actual telephone line conditions.

3.1.3 IISC Database

The IISC database is a new database just recently developed by the Indian Institute of Science [81]. It is a multi-channel database that is able to show the effect of the transmission channel on the accuracy of a speaker identification system. The databases consists of three smaller databases, IISC-Microphone, IISC-Mobile and IISC-Cordless. The original database IISC-Microphone was recorded using a BPL telephone instrument that was placed directly in front of the speaker, approximately 0.5 metres from the mouth. The speech was pre-amplified using both the amplifier in the BPL and the one in a Creative Labs AWE 64 sound card that was used to digitize the speech at a sampling frequency of 16 kHz, with a bit resolution of 16 bits.

Each of the IISC databases consists of 80 speakers, 37 female and 43 male, from four different regions in India. The database is subdivided further into words and sentences. The words directory consists of three repetitions of a 77 word vocabulary spoken in isolation. The sentences directory consists of 10 sentences, each approximately 1 minute long containing 20 independent sentences borrowed directly from TIMIT.

3.2 Speaker Recognition System Parameters

All experiments used 24 seconds of speech to train the system, during TIMIT experiments the SX and SI files were concatenated to produce one 24 second utterance containing 8 sentences for each speaker. The remaining two SA files were used as two independent tests segments. Unless otherwise stated only one dialect directory of the TIMIT/NTIMIT databases was used, due to time constraints. The training for the IISC databases was achieved using the first twenty four seconds of the first speech file in each speaker's directory and the first three seconds of the second and third file were used for testing.

Having acquired the testing or training utterances, it is now the role of the feature extractor to extract the acoustic features from the speech.

3.2.1 Feature Extraction and Parameter Estimation

In this thesis, we investigate the use of the following two feature sets for speaker identification: 1) the LPCC feature set and 2) the MFCC feature set. This section will briefly describes the extraction of these features from the speech waveform.

LPCC Feature Extractor

To reiterate, the LPCC feature extractor consists of six functional blocks [47]: preemphasis, frame blocking, windowing, autocorrelation and LPC computation, and lastly cepstral conversion.

Preemphasis

The preemphasis section is implemented by the difference equation

$$\hat{s}(n) = s(n) - \alpha s(n-1) \quad , \tag{3.1}$$

where α is the preemphasis coefficient and is set here to 0.95.

Frame Blocker

The frame blocker splits the entire speech utterance into segments to carry out frame-wise analysis. The signal for the i-th frame is given by

$$x_i(n) = s(iL + n) \qquad n = 0, 1, \ldots, N-1 \tag{3.2}$$

The parameters for this section are N and L, where N specifies the length of the frame and L specifies the offset between adjacent frames. In order to maintain a constant frame size of 30ms and an update of 10ms, the values of L and N are database dependent due to inconsistent sampling frequencies between databases. The parameters used are shown in Table 3.2.1.

Windowing

By tapering the start and the end of each frame using a windowing function ($w(n)$), it is possible to reduce the effect of spectral leakage caused by the

Database	L (10ms)	N (30ms)
TIMIT	160	480
NTIMIT	80	240
IISC-Microphone	160	480
IISC-Mobile	160	480
IISC-Cordless	80	240

Table 3.1: The database dependent values of the L an N parameters of the frame blocker.

discontinuities present at the ends of the framed speech. The window is applied to the segmented speech through

$$\hat{x}_i(n) = x_i(n)\ w(n) \qquad n = 0, 1, \ldots, N-1 \tag{3.3}$$

We use a Hamming window function for this purpose. It is given by

$$w(n) = 0.54 - 0.46\ cos\left(\frac{2\pi n}{N-1}\right) \qquad n = 0, 1, \ldots, N-1 \tag{3.4}$$

Autocorrelation analysis

The windowed signal is used to compute the autocorrelation coefficients.

$$R(m) = \sum_{n=0}^{N-m-1} \hat{x}_i(n)\hat{x}_i(n+m) \qquad m = 0, 1, \ldots, p. \tag{3.5}$$

(Note that we have dropped the frame subscript here to simplify the notation.)

LPC computation

The $p+1$ autocorrelations are converted to LPC coefficients using the Levinson-Durbin algorithm, included for completeness as

For m=1:

$$\begin{aligned} a_{m,m} &= -\frac{R(1)}{R(0)} && (3.6) \\ P_m &= R(0)(1-a_{11}^2) && (3.7) \end{aligned}$$

For m=2,3,. . .,p:

$$\begin{aligned} a_{m,m} &= -\frac{R(m)+\sum_{i=1}^{m-1} a_{m-1,i}R(m-i)}{P_{m-1}} && (3.8) \\ a_{m,i} &= a_{m-1,i} + a_{m,m}a_{m-1,m-i} && (3.9) \\ P_m &= P_{m-1}(1-a_{m,m}^2) && (3.10) \end{aligned}$$

Cepstral Conversion

The LPCs are converted to cepstral coefficients using the relation,

$$c_0 = ln\ P_p, \qquad (3.11)$$

$$c_m = \frac{-ma_{pm} + \sum_{k=1}^{m-1} a_{pk}c_{m-k}(m-k)}{m}, \qquad 1 \leq m \leq p, \qquad (3.12)$$

$$c_m = \frac{\sum_{k=1}^{m-1} a_{pk}c_{m-k}(m-k)}{m}. \qquad p < m \leq M. \qquad (3.13)$$

The zeroth cepstral coefficient c_0 represents the average power of the speech segment and is usually not used. Only M cepstral coefficients from c_1 to c_M are used as features. We set $M = p$.

Table 3.2 lists all the key parameters and the value chosen in implementing our system.

MFCC feature extractor

The MFCC feature extractor [47] converts an utterance into a sequence of MFCC feature vectors. It consists of exactly the same preemphasis, frame

Section	Parameter	Value used
preemphasis	α	0.95
frame blocker	N	480 (TIMIT)
		240 (NTIMIT)
		480 (IISC-Microphone)
		480 (IISC-Cordless)
		240 (IISC-Mobile)
	L	160 (TIMIT)
		80 (NTIMIT)
		160 (IISC-Microphone)
		160 (IISC-Cordless)
		80 (IISC-Mobile)
windowing	w(n)	$0.54 - 0.46cos\frac{2\pi n}{M-1}$
autocorrelation and LPC analysis	p	8, 10, 12
cepstral conversion	M	p

Table 3.2: Summary of operational parameter for the LPCC feature extractor.

Database	Frame Size	FFT order
NTIMIT	240	256
TIMIT	480	512
IISC-Mobile	240	256
IISC-Microphone	480	512
IISC-Cordless	480	512

Table 3.3: FFT order used for various databases.

blocking and windowing section explained in the LPCC feature extractor. The windowed speech segment is converted to power spectrum via an FFT algorithm and the the number of points used in the FFT algorithm is taken as the power of 2 greater than the frame size. Table 3.3 shows the length of the frames generated by the frame blocker and the corresponding FFT order used for each database. The resulting power spectrum is windowed by a set of 20 triangular filters equally spaced by 150 mels and each 300 mels wide. The power for each window is calculated, denoted by E_k, where k is the window number. A discrete cosine transform is then applied as

$$\hat{c}(n) = \sum_{k=1}^{K} (\log \ E_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right] \tag{3.14}$$

resulting in L cepstral coefficients, typically values for the cepstral order used were 8, 10, and 12.

3.2.2 Speaker Modeling

Each speaker is modeled using one Gaussian Mixture Model (GMM) with 32 mixture components. Each mixture component is characterized by its weight,

Triangular window #	Lower Cutoff	Center Frequency	Upper Cutoff
1	0	100	200
2	100	200	300
3	200	300	400
4	300	400	500
5	400	500	600
6	500	600	700
7	600	700	800
8	700	800	900
9	800	900	1000
10	900	1000	1149
11	1000	1149	1320
12	1149	1320	1516
13	1320	1516	1741
14	1516	1741	2000
15	1741	2000	2297
16	2000	2297	2639
17	2297	2639	3031
18	2639	3031	3482
19	3031	3428	4000
20	3482	4000	4595

Table 3.4: The triangular filters used in MFCC calculation.

mean vector and (diagonal) covariance matrix. As described in Section 2.2.2, the GMMs are trained using the EM algorithm [53] with an original model (λ_0) derived by a k-means algorithm. Ten iterations of the EM algorithm were used and a variance floor [55] of 0.3 is enforced on the models during each iteration. These models are then used to identify the speaker from the given test utterance.

3.3 Performance of the Speaker Recognition System

This final section provides results for the MFCC and LPCC coefficients when used with TIMIT and IISC databases.

Since this is a speaker identification system and we are ultimately concerned with its ability to identify speakers, the performance of the system is measured using the identification error. The identification error can be described as

$$\%\ identification\ error = \frac{\#\ incorrectly\ identified\ segments}{total\ \#\ of\ segments} \times 100\% \tag{3.15}$$

The system is tested using the following parameters. Their effect on the performance of the system is shown by the identification error.

- **The Number of Gaussian Mixture Components**

  The system is tested using 8, 16, 32, and 64 mixture components per speaker using 10 LPCC coefficients, the results are shown in Table 3.5.

Database	8 mixtures	16 mixtures	32 mixtures	64 mixtures
IISC-microphone	20%	18.75 %	13.75 %	11.875 %
IISC-mobile	18.125 %	14.375 %	10.625 %	8.13 %
IISC-cordless	10.625 %	8.75 %	7.5 %	6.875 %
NTIMIT	46.06 %	34.22 %	27.64 %	23.69 %
TIMIT	0 %	0 %	0 %	0 %

Table 3.5: The effect of the mixture size on the performance of the system.

Table 3.5 shows that the mixture size greatly affects the ability of the GMM system to successfully separate speakers under non-ideal conditions. The TIMIT/NTIMIT experiments used only 36 speakers from the same dialect region, with TIMIT producing an identification error of 0% in all cases. The performance of the system using 32 mixtures, and including the entire 630 speakers of the TIMIT database, the identification error was found to be 0.46%. A result that is confirmed by Reynolds [54].

- **Type of Features Used**

  The system is finally tested using the two feature extractors (LPCC and MFCC) separately. The performance of the LPCCs and MFCCs are shown in Table 3.6 and Table 3.7, respectively. The LPCC features are seen to be better suited for speaker identification. The number of the features had a significant effect on the performance of the system. Even though the MFCC features are not as suited as the LPCC features they do perform adequately in this role. An increase in the number of

Database	8 LPCCs	10 LPCCs	12 LPCCs
IISC-microphone	23.12 %	13.75 %	13.75 %
IISC-mobile	15.63 %	10.63 %	8.13 %
IISC-cordless	13.75 %	7.5 %	6.25 %
NTIMIT	32.9 %	27.64 %	26.32 %
TIMIT	1.42 %	0 %	0 %

Table 3.6: The system performance using LPCC coefficients.

Database	8 MFCCs	10 MFCCs	12 MFCCs
IISC-microphone	26.86 %	20.63 %	15.25 %
IISC-mobile	14.48 %	12.5 %	12.5 %
IISC-cordless	13.13 %	10 %	6.88 %
NTIMIT	31.58 %	26.32 %	22.58 %
TIMIT	5.39 %	5.39 %	5.39 %

Table 3.7: The system performance using MFCC coefficients.

the features has a positive effect on the system's performance when used on low quality speech.

To summarize the performance of the system, increasing the mixture size reduces the identification error, the LPCC feature outperforms the MFCC feature and the order of the feature has a significant effect on the system. The system showed an exact performance match with the system implemented by Reynolds [54].

3.4 Summary

Using the speech included in corpora such as TIMIT, NTIMIT, and the IISC databases, it was possible to conduct closed-set speaker identification experiments using a text-independent Gaussian Mixture Model System. The databases were chosen due to their availability and variation in speech quality. The system was trained using 24 seconds of speech and two independent tests of three seconds of speech was used for each speaker.

The performance of the speaker identification system is investigated for two different feature sets (LPCC and MFCC). Also, the effect of mixture size in GMMs is evaluated.

Since we are only concerned with the ability of the system to identify speakers, the performance of the system was measured using the identification error. The system showed that the identification error was inversely proportional to an increase in both the mixture size and the number of the features used. The system also showed that the LPCC coefficients functioned better for speaker identification than MFCCs and the order of both features had no effect when used on clean speech. The speaker identification system showed an identification error of 0.46% over the entire TIMIT database, the result obtained by Reynolds [54].

Even though the LPCC coefficients performed reasonably well, certain additions can be made to these features to increase the performance of the system. The following chapter covers the use of pitch and voicing information in a speaker identification system and illustrates the performance achieved.

Chapter 4

Using Pitch and Voicing Information

So far we have used the information contained in the power spectral envelope (in the form of cepstral coefficients) for speaker identification. Though the cepstral coefficients perform quite well for speaker recognition, we can improve the performance by using source related features such as pitch period, shape of the glottal pulse, etc. When speech signal is voiced (or, periodic), the pitch frequency corresponds to the frequency of vocal chord vibrations. Its value is low for adult male speakers and high for female speakers and children.

The pitch frequency is an extremely important property of speech. Also known as the fundamental frequency (f_0), it defines the periodicity of a speech signal [46]. Pitch is considered to be one of the important properties used by humans in identifying a person from his/her voice. In addition to pitch value, amount of voicing (or, periodicity) in the speech signal is also important for determining the identity of a speaker. Pitch has an important advantage over spectral information that it does not suffer due to the frequency characteristics of the transmission system. In this thesis, an attempt is made to utilize this pitch and voicing information for automatic speaker identification.

Most of the current speaker recognition systems use the cepstral coefficients as features. The use of pitch and voicing information is not very common in the current speaker recognition systems. This is because pitch information cannot be reliably extracted, and is not always present in the speech signal. A few studies have been reported in the literature where the pitch value has been used in a straight-forward manner for speaker identification. When speech utterances are carefully selected so that all the frames in an utterance are voiced, the pitch feature works reasonably well for speaker identification [1]. But, when the speech utterances have both voiced and unvoiced frames, the speaker identification performance reported by Furui [14] as well as shown in this thesis is very poor.

The two main reasons why the direct use of pitch as a feature does not work for speaker recognition are listed below. First, the speech is not always periodic; only about half of the frames are voiced. Thus, pitch can not be estimated for half of the frames (i.e. for unvoiced frames). The problem is how to account for pitch information for the unvoiced frames during recognition phase. Second, the pitch estimation methods are not very reliable. They classify some of the frames unvoiced when they are really voiced. Also, they make pitch estimation errors (such as doubling or halving of pitch value depending on the method).

In order to use pitch information for speaker recognition, we have to overcome these problems. We need a method which does not use the pitch value directly as feature and which should work for voiced as well as unvoiced frames in a reliable manner. We propose here a method which uses the autocorrelation function of the given frame to derive pitch-related features. We call these features the maximum autocorrelation value (MACV) features. These features can be extracted for voiced as well as unvoiced frames and do not suffer from the pitch doubling or halving type of pitch estimation errors. Using these MACV features along with the cepstral features, the speaker identification performance is improved by 45%.

The goal of this chapter is to illustrate ways of incorporating pitch and voicing information into a speaker recognition system. It begins with pitch period used in a straight-forward manner as feature for speaker identification. For this, we experiment with a number of pitch estimation methods and provide speaker identification results. Then, we explore ways to extract the pitch and voicing information in an indirect manner. For this, Hayakawa et al. [23] have recently proposed a method which tries to determine harmonicity in a number of subbands from the spectrum of the LP residual signal. The resulting features are called the Power Difference of Spectra in Subband (PDSS) features. We describe this method briefly and use it for speaker recognition. Finally, the method of extracting MACV features is described. These features are used for speaker identification and the results are described.

4.1 Using Pitch Value as a Feature

In this section, we use the pitch period (or, frequency) in a straight-forward manner as a feature for speaker identification. For this, we experiment with a number of pitch estimation methods and provide speaker identification results.

In 60's, the pitch extraction methods involved low pass filtering the speech to remove all the higher harmonics and measure the pitch frequency using simple means from the first harmonic. These methods had numerous problems, main one being the absence of first harmonic in certain speech signals like telephone-speech signal which is band-limited to 200-3400 Hz and may not have the first harmonic [46]. Since then, more sophisticated pitch extraction methods have been developed. These include the autocorrelation method, the cepstrum method, the absolute magnitude difference function (AMDF) method, the harmonic peak method, the maximum likelihood method, etc. Here we use only the autocorrelation and the AMDF methods for estimating the pitch period. The pitch period is used as a feature for speaker identification and the results are reported.

4.1.1 Autocorrelation method

In the absence of the fundamental frequency, it is quite common to search the signal for periodicities using the autocorrelation function [46]. The autocorrelation function of signal $s(n)$ is defined as:

$$R_{ss}(k) = \sum_{n=-\infty}^{\infty} s(n)\ s(n+k) \tag{4.1}$$

If the signal is periodic with period equal to P samples, then the autocorrelation function will show a peak at a lag equal to P. The autocorrelation function of a periodic signal will also be periodic with the same period.

Other important properties of the autocorrelation function include [49]:

- It is an even function; i.e., $R_{ss}(k) = R_{ss}(-k)$.
- The maximum value of the autocorrelation function is at $k = 0$; i.e., $R_{ss}(k) \leq R_{ss}(0)$.
- $R_{ss}(0)$ is equal to the energy for deterministic signals and the average power for random and periodic signals.

The way the autocorrelation function represents periodicity is an important feature. Computing the autocorrelation function of a signal with a period P will result in maxima at samples $0, \pm P, \pm 2P, \pm 3P, \ldots$. The autocorrelation function does not depend on the phase of the signal. The period of the signal can be found by finding the location of the first maximum (see Fig. 4.1) [81] [48]. The simplest autocorrelation-based pitch extractor can therefore be outlined as

1. Given a speech segment s(n), $n = 0, 1, 2, \ldots, N-1$, compute its autocorrelation function.

$$R_{ss}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} s(n)s(n+k), \qquad k = 0, 1, ..., N-1.$$

2. Normalize the autocorrelation function by the power of the segment ($R_{ss}(0)$); i.e.,

$$\bar{R}_{ss}(n) = \frac{R_{ss}(n)}{R_{ss}(0)} \quad n = 0, 1, \ldots, N-1. \tag{4.2}$$

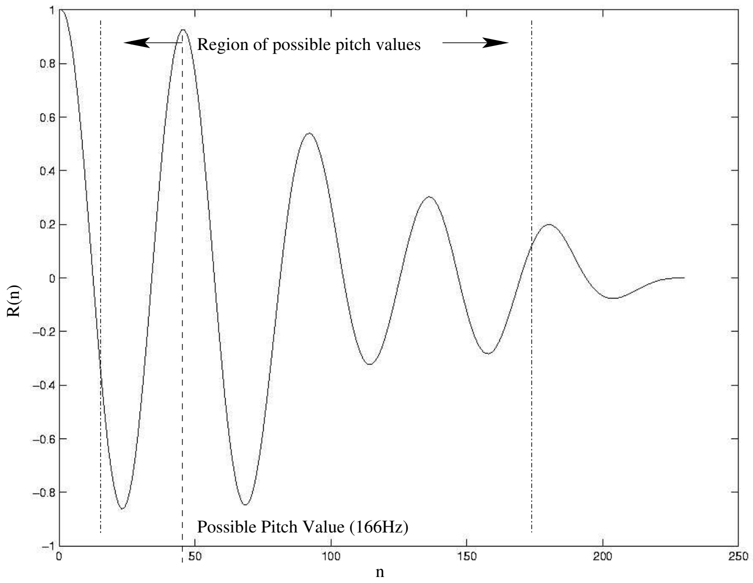


Figure 4.1: The autocorrelation function of 30 ms segment of vowel sound /i/.

Feature	IISC-Microphone	IISC-Cordless	IISC-Mobile	TIMIT	NTIMIT
Used	Id. Error (%)	Id. Error (%)	Id. Error (%)	Id. Error (%)	Id. Error (%)
Pitch (1)					
Auto. method	91.3	90.0	91.9	86.9	92.1

Table 4.1: The performance of the autocorrelation method.

3. Discard the portion of the autocorrelation function outside valid pitch values; i.e.,

$$\hat{R}_{ss} = \bar{R}_{ss}(2ms \leq k \leq 16ms) \tag{4.3}$$

4. Using the remainder of the segment locate the maximum peak. The location of this peak is the estimate of pitch period.

5. The normalized amplitude of the pitch peak is compared to an experimentally derived threshold. If the peak exceeds this threshold, then it is considered to be a voiced frame and the pitch value is kept, otherwise the pitch value is discarded.

Using the text-independent GMM system outlined in Chapter 3, the performance of the autocorrelation method was evaluated on the TIMIT, NTIMIT and IISC databases. The pitch values were used to both train and test the system, which consisted of 8 Gaussian mixtures per speaker. The results of the experiments are shown in Table 4.1.

The system performance is extremely poor while using the autocorrelation method. The identification error does not fall below 86% and ranges up to 92%.

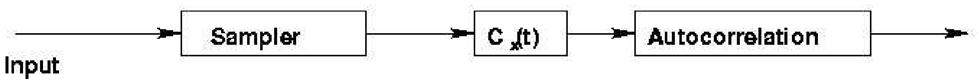


Figure 4.2: The use of a clipper in the autocorrelation pitch extraction method.

The performance of the autocorrelation method can be improved through the use of non-linear processing. Non-linear processing enables the removal of information that would normally confuse the pitch extraction method, like the low amplitude sections of the speech signal. The high amplitude portions of the speech tend to contain most of the pitch information, therefore any non-linear processing that can remove the lower portions, should in fact improve the performance of the pitch extractor. Non-linear processing is achieved in the time domain with center clipping and cubing. We investigate the center clipping and cubing operations for improving the pitch estimation performance and use the resulting pitch values for evaluating the speaker identification performance.

Center Clipping: As the name suggests, low portions of the speech signal are removed through center clipping [46] [71] [12]. The peak amplitude of the speech signal determines the clipping point and all portions of speech below this threshold (T) are removed. The clipped speech is then passed to an autocorrelation (see Fig. 4.2), whose values are typically zero for most lag times, with large peaks at the pitch periods. The clipper is incorporated into the system as shown in Figure 4.2.

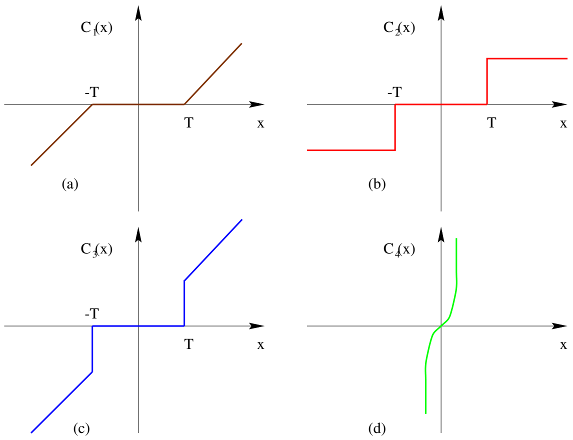


Figure 4.3: The center clipping functions used throughout our experiments are shown in (a),(b) and (c), (d) illustrates the transfer function used in the cubing method.

The performance of three such clippers were evaluated, these included clippers C_1, C_2, and C_3, whose input-output transfer functions were defined as (see Fig. 4.3)

$$C_1(x) = \begin{cases} x - T & x > T \\ 0 & |x| \leq T \\ x + T & x < -T \end{cases} \tag{4.4}$$

$$C_2(x) = \begin{cases} 1 & x > T \\ 0 & |x| \leq T \\ -1 & x < -T \end{cases} \tag{4.5}$$

Feature	IISC-Microphone	IISC-Cordless	IISC-Mobile	TIMIT	NTIMIT
Used	Id. Error (%)	Id. Error (%)	Id. Error (%)	Id. Error (%)	Id. Error (%)
Base	91.3 %	90.0	91.9	86.9 %	92.1 %
C_1	91.25 %	88.23 %	94.38 %	90.79 %	85.53 %
C_2	88.23 %	88.23 %	91.25 %	86.85 %	88.16 %
C_3	89.37 %	88.23 %	91.25 %	89.48 %	84.21 %
$Cubing$	91.63 %	89.37 %	92.5 %	90.79 %	88.16 %

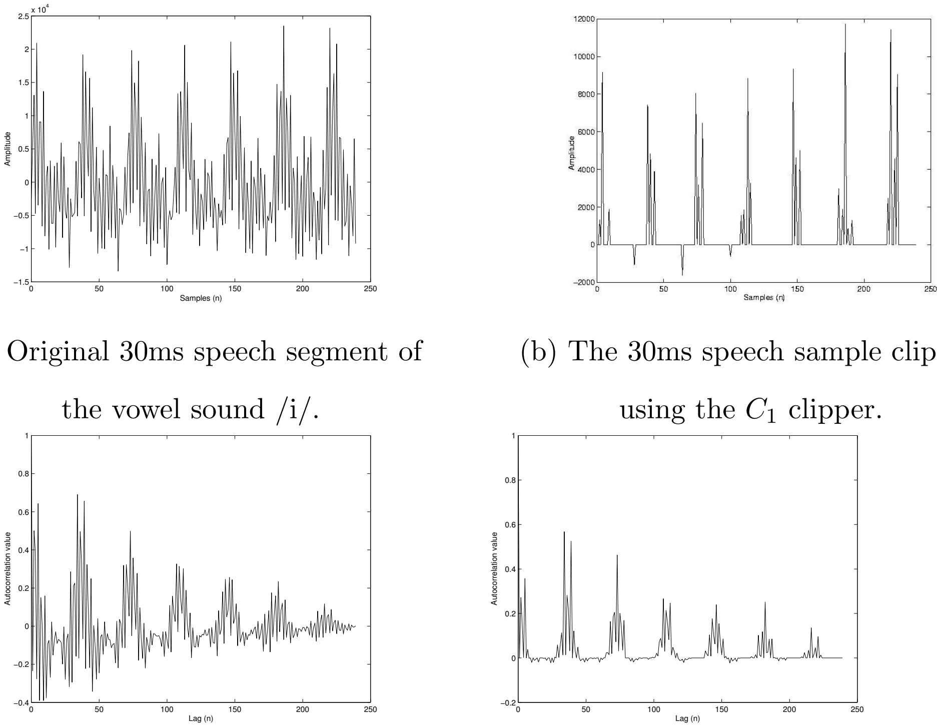
Table 4.2: The performance of the enhancements to autocorrelation method.

$$C_3(x) = \begin{cases} x & x > T \\ 0 & |x| \leq T \\ x & x < -T \end{cases} \tag{4.6}$$

Figure 4.4 includes an example of the clipper C_1, whose clipping point was set at 50% of the peak amplitude. Figure 4.4 (a) and (b) show the 30 ms speech segment of sound /i/ preceding and following clipping, and (c) and (d) show the corresponding autocorrelation function of the unclipped and clipped speech segment respectively.

The performance of these clippers was evaluated using the text-independent GMM system outlined in Chapter 3. The performance over NTIMIT, TIMIT and IISC databases was evaluated.

The clipping point for all clippers was set to 30% of the peak amplitude of the speech segment. Each speech segment was windowed by a 30 ms Hamming window, which had an update of 10 ms. The speech was clipped using either the C_1, C_2 or C_3 clippers and a 30 ms autocorrelation function was derived. The maximum peak locations (pitch values) were used as single element features that were used to train eight GMM mixtures per speaker. The performance of the clippers is shown in Table 4.2 compared with the results of the base autocorrelation function.



(a) Original 30ms speech segment of the vowel sound /i/.

(b) The 30ms speech sample clipped using the C_1 clipper.

(c) The autocorrelation function of the original speech sample.

(d) The autocorrelation function of the clipped speech sample.

Figure 4.4: An example of center clipping, showing the effect on the autocorrelation function.

Table 4.2 shows that nonlinear clipping methods have limited success and no method is able to perform well over all databases. The last row of Table 4.2 shows the performance of the cubing method.

Cubing: The cubing operation is realized by passing the speech signal through a non-linear system, whose transfer function is defined by $y(t) = x^3(t)$, shown in Fig. 4.3(d). It enhances the high amplitude portions of the speech with respect to low amplitude portions, without the need to maintain an adjustable threshold [1]. The speaker recognition performance is evaluated using the pitch value obtained by the autocorrelation method with the cubing operation. The results are shown in Table 4.2. Like the clipping operation, the cubing operation does not show any improvement in speaker identification performance.

4.1.2 Average Magnitude Difference Function Method

The average magnitude difference function (AMDF) method, first considered by Moorer [41] and Ross (et al) [64] in 1974, is thought to be more efficient than the autocorrelation function as it removes the need for costly multiplications [49].

The AMDF function can be used for pitch extraction in the following manner:

1. Given a speech segment s(n), $n = 0, 1, 2, \ldots, N-1$, compute its AMDF function.

$$\gamma_{ss}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} |s(n) - s(n+k)|, \qquad k = 0, 1, ..., N-1.$$

2. Normalize the AMDF function by the power of the segment; i.e.,

$$\bar{\gamma}_{ss}(n) = \frac{\gamma_{ss}(n)}{\sum_{n=0}^{N} s(n)^2} \tag{4.7}$$

3. Keep only the portion of the AMDF function relating to valid pitch values; i.e.,

$$\hat{\gamma}_{ss} = \gamma_{ss}(2ms \leq k \leq 16ms) \tag{4.8}$$

4. Using the remaining segment find the location of the minimum dip, which is considered as an approximation of the pitch frequency.

5. The normalized amplitude of the pitch dip is compared to an experimentally derived threshold. If the dip is lower than this threshold then it is considered to be a voiced frame and the pitch value is kept, otherwise the pitch value is discarded and the frame is considered unvoiced.

The performance of the AMDF pitch extractor was evaluated using the GMM speaker identification system outlined in Chapter 3. The speech was preemphasized and split into 30 ms frames with a 10 ms update. The speech was then windowed with a Hamming window and the pitch values extracted using the method outlined above. The pitch values were used to train eight Gaussian mixtures per speaker and the results of the experiment are shown in Table 4.3.

The performance of the AMDF method was comparable with the autocorrelation method, resulting in identification errors in the range of 90% to 79%.

Feature Used	IISC-Microphone Id. Error (%)	IISC-Cordless Id. Error (%)	IISC-Mobile Id. Error (%)	TIMIT Id. Error (%)	NTIMIT Id. Error (%)
Pitch (1) AMDF method	90.0	98.3	93.1	82.9	79.0

Table 4.3: The performance of AMDF pitch extractor.

4.2 The Use of Voicing Information

This chapter up to now has covered the use of pitch in automatic speaker identification. However, using pitch as a feature is plagued with problems. Pitch is an unreliable feature, which is illustrated in the preceding section by the poor speaker identification performance for the autocorrelation and AMDF methods.

The two main reasons why the direct use of pitch as a feature does not work for speaker recognition are listed below. First, the speech is not always periodic; only about half of the frames are voiced. Thus, pitch can not be estimated for half of the frames (i.e. for unvoiced frames). The problem is how to account for pitch information for the unvoiced frames during recognition phase. Second, the pitch estimation methods are not very reliable. They classify some of the frames unvoiced when they are really voiced. Also, they make pitch estimation errors (such as doubling or halving of pitch value depending on the method).

In order to use pitch information for speaker recognition, we have to overcome these problems. We need a method which does not use the pitch value directly as feature and which should work for voiced as well as unvoiced frames in a reliable manner. We propose in this thesis a method which captures the periodicity characteristics of speech signal in an indirect manner in the form of voicing information. We use the autocorrelation function of the given frame to derive this information. In the literature [23, 77], various methods to extract this type of voicing information have been proposed and successfully applied for speaker recognition. One such method based on the LPC residual spectrum is PDSS (Power Difference of Spectra in Subband) features [23]. In order to put our method in proper perspective, we compare its speaker recognition performance with that of the PDSS method. We describe the PDSS method first, followed by our method.

4.2.1 Power Difference of Spectra in Subband

The spectrum of the LPC residual signal is obtained by applying the inverse LPC filter to the speech segment. The harmonic structure of the LPC residual spectrum is captured by the PDSS features. The procedure for extracting the PDSS features from the speech signal is outlined below.

1. Generate the LPC residual signal using p (=10) linear prediction coefficients.

2. Increase the frequency resolution of the spectrum by appending an efficient number of zeros to the signal prior to calculating the power spectrum.

3. Subdivide the spectrum into M bands, each having a bandwidth of about 500 Hz.

4. For each band, calculate the ratio of the geometric to arithmetic mean and subtract it from one as follows:

$$V(i) = 1.0 - \frac{\left[\prod_{k=L_i}^{H_i} P(k)\right]^{\frac{1}{N_i}}}{\frac{1}{N_i}\sum_{k=L_i}^{H_i} P(k)}, \qquad i = 1, 2, ..., M, \tag{4.9}$$

where $V(i)$ is the i^{th} element of the PDSS feature set, P represents the power spectrum, L_i is the lower boundary of the i^{th} subband and H_i the upper boundary, and $N_i = H_i - L_i$.

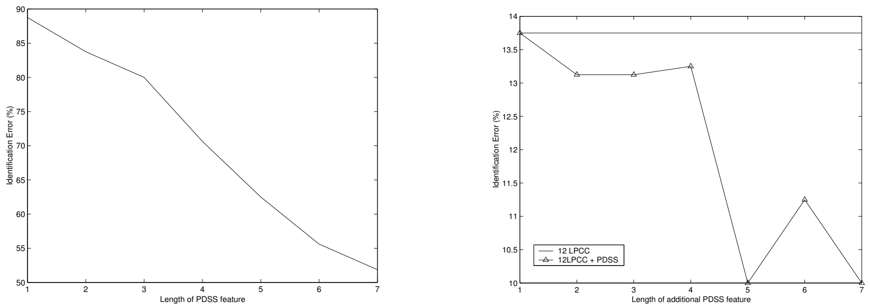
The ratio of geometric to arithmetic mean is a measure of spectral flatness. For example, if the spectrum is flat then,

$$\left[\prod_{k=L_i}^{H_i} P(k)\right]^{\frac{1}{N_i}} = \frac{1}{N_i}\sum_{k=L_i}^{H_i} P(k) \tag{4.10}$$

and $V(i)$ will be equal to zero, otherwise the values of the PDSS can exist anywhere in the range of 0 to 1. Thus, the PDSS features represent the dynamic range of individual bands. It is conjectured in [23] that when speech is periodic, the dynamic range within a band will be large. Thus, the PDSS features capture the voicing information in an indirect manner.

Using the speech contained in the IISC-Mobile database, this feature set was evaluated using the system discussed in Chapter 3. The PDSS features were derived from a 30 ms framed section of speech that was updated every 10 ms. Using the above algorithm, the PDSS feature set was evaluated for speaker identification for values of M ranging from 1 to 7. The performance of the feature was shown with and without the presence of LPCC coefficients in Fig. 4.5(a) and Fig. 4.5(b), respectively.

From Fig. 4.5(a), it can be seen that the PDSS feature set works extremely well in this application, as shown by the abrupt linear decrease in the identification error. However, the improved performance is not as obvious when the PDSS is used in conjunction with LPCC coefficients. Figure 4.5(b) shows a minor difference of only 3.7% when an additional 5 PDSS features are used.



(a) (b)

Figure 4.5: The speaker identification error when, (a) the PDSS feature is used alone and (b) with LPCC coefficients.

The last test performed on this feature set related to its computational complexity. The time taken to generate the feature set was average over all speakers. It took on the average 600 ms seconds to compute the PDSS features from the one second utterance, using a Dell Dimension PIII-800 system. Thus, this method is computationally very expensive.

4.2.2 The Maximum Autocorrelation Value (MACV) Features

We propose here a simple and reliable method for extracting voicing information from the speech signal. We use the autocorrelation function of the given frame to derive pitch-related features. We call these features the maximum autocorrelation value (MACV) features [81]. These features can be extracted for voiced as well as unvoiced frames and do not suffer from the pitch doubling or halving type of pitch estimation errors.

The MACV features are derived from the speech signal as follows:

1. Given the speech signal $s(n)$, compute the autocorrelation function $R_{ss}(n)$ using,
$$R_{ss}(n) = \frac{1}{N} \sum_{k=0}^{N-1-n} x(k)x(n+k) \tag{4.11}$$

2. Normalize the autocorrelation function by its mean power, (the autocorrelation value at n=0); i.e.,
$$r_{ss}(n) = \frac{R_{ss}(n)}{R_{ss}(0)} \tag{4.12}$$

3. Since the lower section of the autocorrelation function is used to generate the cepstral coefficients and the upper section is considered unreliable, we maintain only the portion of the normalized autocorrelation function between the values of 2 ms and 16 ms. With the reduced autocorrelation sequence, we perform the following steps:

   i. Divide the remaining autocorrelation into N equal segments.

   ii. For each of the N segments locate the maximum value.

   iii. These N maximum values correspond to the N elements of the MACV feature set.

This process is best illustrated in Fig. 4.6. The MACV features can be thought of as an N point approximation of the mid-section of the autocorrelation function.

The MACV feature set was used for speaker identification using the TIMIT, NTIMIT and IISC databases. The speech was split into 30 ms segments with an update of 10 ms, it was windowed and a corresponding 30 ms autocorrelation function derived.

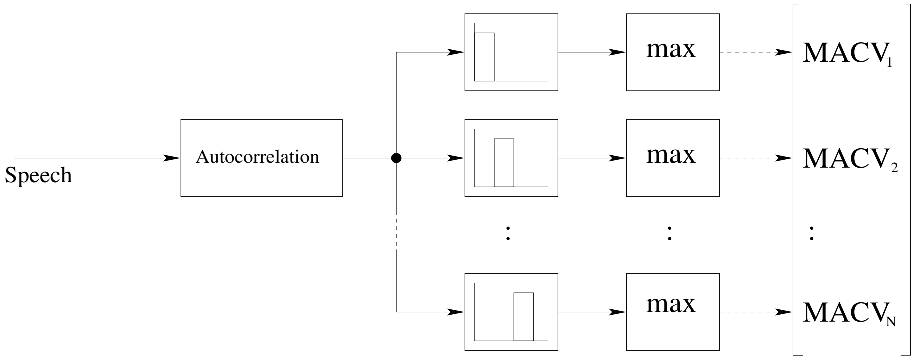


Figure 4.6: The MACV feature extractor.

In order to show that the MACV features contained additional information not present in the LPCC coefficients, a test was conducted to show the effect of using MACVs as an additional feature to the LPCC coefficients. The identification error for the system was first found using only 12 LPCC coefficients and repeated with the addition of 5 MACV coefficients on all of the above mentioned databases.

To show the improved performance when using MACV features the reduction in the identification error with respect to the error obtained using only 12 LPCCs was calculated. This reduction was defined as,

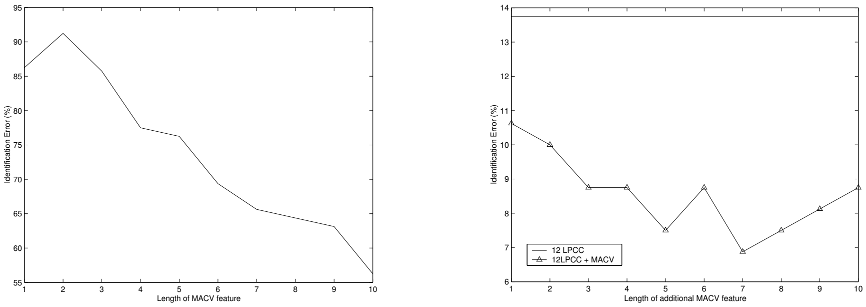
$$\%\ reduction = \frac{\%\ identification\ error_{LPCC} - \%\ identification\ error_{MACV}}{\%\ identification\ error_{LPCC}} \times 100\% \tag{4.13}$$

The results for this experiment are shown in Table 4.4.

Table 4.4 illustrates that the addition of the MACV feature improves the

Database Used	12 LPCC Id. Error (%)	12 LPCC + 5 MACV Id. Error (%)	Reduction in Id. Error (%)	5 MACV Id. Error (%)
IISC-Microphone	13.8	7.5	45.5	76.3
IISC-Mobile	6.3	5.6	10.0	80.6
IISC-Cordless	21.3	15.6	26.5	90.0
NTIMIT	21.6	13.2	39.2	61.8

Table 4.4: The performance of the MACV and LPCC features.



(a) (b)

Figure 4.7: The speaker identification error when, (a) MACVs are used alone and (b) with LPCC coefficients.

identification error by up to 45%, and performs consistently well across all the databases.

We varied the number of the MACV features from 1 to 10 and its effect on the identification error with and without the assistance of the LPCC coefficients is shown in Figure 4.7(a) and Fig. 4.7(b), respectively. The identification error of the system when using 12 LPCCs is shown in Fig. 4.7(b) by a straight line at 13.7 %. The jagged line shows the added benefit of including the MACV features. With only one MACV element the system performance is increased by 3%, an improvement of 21 %. At the optimal configuration of 5 MACVs, the performance of the system is increased by 45%. As can be expected, this feature set incurs very little computational cost, requiring only 80 ms of computation time to generate features for one second of speech.

4.3 Summary

Pitch is an important property of speech as it represents the periodicity of a speech signal. Pitch is a very important feature used by listeners in identifying speakers. However, it is very rarely used in the current speaker recognition systems as it cannot be reliably extracted, and is not always present in the speech signal. In this chapter, we have tried to utilize this pitch and voicing information for speaker identification.

In our experiments, we first investigated the use of the pitch value directly as a feature. We have used the following methods for estimating the pitch value:

- The autocorrelation method,
- The autocorrelation method incorporating nonlinear functions such as center clipping and cubing,
- and the AMDF method.

These methods results in identification errors in the range of 86% to 94%, which are very high. Thus, the use of pitch as a feature in a straight forward manner does not provide a satisfactory solution.

The two main reasons why the direct use of pitch as a feature does not work for speaker recognition are listed below. First, the speech is not always periodic; only about half of the frames are voiced. Thus, pitch can not be estimated for half of the frames (i.e. for unvoiced frames). The problem is how to account for pitch information for the unvoiced frames during recognition phase. Second, the pitch estimation methods are not very reliable. They classify some of the frames unvoiced when they are really voiced. Also, they make pitch estimation errors (such as doubling or halving of pitch value depending on the method).

In order to overcome these problems, we have proposed here a simple and reliable method which captures the periodic property of speech signal indirectly in the form of voicing information. We have used the autocorrelation function of the given frame to derive pitch-related features. We have called these features the maximum autocorrelation value (MACV) features. These features can be extracted for voiced as well as unvoiced frames and do not suffer from the pitch doubling or halving type of pitch estimation errors. Using these MACV features along with the cepstral features, we have shown that the speaker identification performance is improved by 45%. We have compared the speaker recognition performance of our method with another method (PDSS method) recently reported in the literature. Our method compares favorably with respect to this method.

Chapter 5

Conclusions

This chapter summarises the key issues and results covered in this thesis, and a few suggestions are made for possible directions for future research in this area. In this thesis, we have focused on feature extraction for speaker identification. We have addressed their limitations and suggested possible solutions for improving them

5.1 Summary

In this thesis, the Gaussian Mixture Models (GMMs) have been used for text-independent speaker identification. The GMM based systems are most commonly used and have shown great success in this area. Each speaker in the training set is represented by one GMM. GMMs are created using a k-means clustering algorithm, optimised by the expectation-maximisation (EM) algorithm. Closed-set identification experiments are conducted using the GMM models on speech data from the TIMIT, NTIMIT and IISC databases. These databases are chosen because of the large amount of continuous speech they contain under a wide variety of conditions. The TIMIT databases are especially chosen due to their wide use and availability, serving as a means to compare our results with those of others.

The performance of the GMM system has been evaluated using two feature sets: LPCC and MFCC. The LPCC features have achieved an identification error of 0.46 % over all dialects in the TIMIT database and 0 % over a single dialect. However other databases have shown that the feature had room for improvement with identification errors ranging from 6 % to 26 %.

The MFCC features have been found not to be as good as the LPCC features. They have resulted in a slight degradation in system performance with an identification error of 5 % for a single dialect directory in TIMIT and a range of 7 % to 22 % for other databases. However, an improvement in identification error is noticed when using the NTIMIT database, this suggested that the MFCC coefficients are more suited to non-ideal conditions. However their performance is limited using clean microphone quality speech.

The LPCC and MFCC features were used extensively in the current speaker recognition systems, due to their ability to capture the spectral properties of the vocal tract. The performance of these features can be improved using additional information such as pitch and voicing. The pitch frequency is an extremely important property of speech and is utilized by human listeners to recognise speakers.

The performance of the pitch value as a feature has been evaluated using two common methods: the autocorrelation method and the average magnitude difference method. The pitch period is extracted from a 30 ms segment of speech, updated every 10 ms. A corresponding 30 ms autocorrelation/AMDF function is computed and the pitch location extracted. The pitch values are used to generate 8 Gaussian mixtures per speaker model and closed set speaker identification is performed. The use of pitch as a feature has resulted in limited success, (even when nonlinearities such as center clipping and cubing functions are used in conjunction with the autocorrelation method of pitch estimation). The identification errors achieved are extremely large in the range of 86 % to 92 %.

Using pitch as a feature is plagued with problems: pitch is an unreliable feature to extract; it is not always present in speech; and it suffers from voiced-to-unvoiced and unvoiced-to-voiced classification errors. The use of voicing is a more worthwhile approach, since voicing information can be easily and reliably extracted. When speech is periodic the amount of voicing will be high; when it is not periodic, it will be small. The following two methods have been used for extracting voicing information: the PDSS method and the MACV method.

The PDSS method is based on the LPC residual spectrum of the signal and represents the harmonic structure of the spectrum. For deriving PDSS features, the speech signal is analysed with a 30 ms segment with a 10 ms update and transformed to its power spectrum by using a 8192 point FFT. The PDSS features have shown promising results when used alone in achieving an identification error as low as 45 %. However they provide a minimal improvement when used with LPCC coefficents, only increasing the performance of the feature by 21 %.

The MACV method uses the autocorrelation function computed from the speech signal. Using a 30 ms speech segment with a 10 ms update, a 30 ms autocorrelation function is derived. The lower portion of the autocorrelation function is already used in the current speaker identification systems in the form of LPCC of MFCC features. The higher portion (2 ms to 16 ms) contains information about voicing and periodicity. The upper portion of the autocorellation function is split into N segments (where N = 5). The maximum values in each segment of the autocorrelation function is picked, and these form the MACV features. The performance of these features when used alone has been found comparable with PDSS. However, when used in conjunction with LPCC coefficients, the performance is improved by 45%. Not only are the MACV features more reliable than current methods, they can be extracted from the speech signal in an extremely simple manner.

Thus, we can conclude that though cepstral coefficients as features are reasonably suited for speaker identification, their performance can be improved through the addition of voicing information.

5.2 Future Direction

The current methods of feature extraction, even though they appear to function reasonably well, are inadequate in representing speech. The cepstral coefficients are extremely good at representing vocal tract properties. However, they are unable to represent voicing information. This thesis has shown the poor performance of pitch as a feature, due to the problems associated with extracting it reliably. The use of the more general voicing information offers a possible solution to these problems. However, more extensive work on voicing features is needed. In this thesis, the autocorrelation function has been used to represent voicing information. The cepstrum function also provides this voicing information and should be investigated in future. The MACV features employed in this thesis use the maximum value of N = 5 segments of the upper portion of the autocorrelation function for extracting voicing information. Alternative ways of representing the upper portion of the autocorrelation function can also be investigated.

Bibliography

[1] B. S. Atal, "Automatic recognition of speakers form there voices," *Proc. IEEE,* vol. 64, pp. 460-475, 1976.

[2] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoustic. Soc. Amer.,* vol. 52, pp. 1687-1697, 1972.

[3] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and recognition," *J. Acoustic. Soc. Amer.,* vol. 54, no. 6, pp. 1304-1312, 1974.

[4] Y. Bennani, F. Fogelman, and P. Gallinari, "A connectionist approach for automatic speaker recognition," *Proc. ICASSP'90,* (Albuquerque, New Mexico), April 1990, pp. 265-268.

[5] Y. Bennani and P. Gallinari, "A connectionist approach for automatic speaker recognition," *Workshop on Automatic Speaker Recognition Identification Verification,* (Martigny, Switzerland), April 1994, pp. 95-102.

[6] D. P. Bogert et al., "The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking,"

in *Proc. Symp. Time series analysis,* M. Rosenblatt, Ed. New York: Wiley, pp. 209-243, 1963.

[7] J. T. Buck, D. K. Burton, and J. E. Shore, "Text dependent speaker recognition using vector quantisation," *Proc. ICASSP'85,* (Tampa, Florida), March 1985, pp. 391-394.

[8] J. P. Campbell, "Testing with the YOHO CD-ROM voice verification corpus," *Proc. ICASSP'95,* (Detroit, Michigan), May 1995, pp. 341-344.

[9] J. P. Campbell, "Speaker recognition: a tutorial,"*Proc. IEEE,* vol. 85, no. 9, pp. 1437-1462, 1997.

[10] M. Chester, *Neural networks: a tutorial,* New York: Prentice Hall Publishers, 1993.

[11] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-28, no. 4, pp. 357-366, August 1980.

[12] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, "Real-time digital hardware pitch detection," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-24, no. 1, pp. 2 - 8, February 1976.

[13] K. R. Farrel, R. J. Mammone, and K. T. Assaleh, "Speaker recognition using neural networks and conventional classifiers," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-2, no. 1, pp. 194-204, January 1994.

[14] T. Matsui and S. Furui, ”Text-independent speaker recognition using vocal tract and pitch information,” *Proc. ICSLP’90,* (Kobe, Japan), pp. 137 - 140, November 1990.

[15] S. Furui, ”An overview of speaker recognition technology,” in *Automatic Speech and Speaker Recognition,* C. H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Boston: Kluwer Academic, pp. 31-56 ,1996.

[16] S. Furui, ”Recent advances in and perspectives of speaker recognition technology,” *Proc. of International Workshop on Human Interface Tech. 94,* (Aizu, Japan), September 1994, pp. 35-42.

[17] S. Furui, ”Cepstral analysis technique for automatic speaker recognition,” *IEEE Trans Acoustics, Speech and Signal Processing,* vol. ASSP-29, no. 1, pp. 254-272, April 1981.

[18] S. Furui, ”Research on individuality features in speech waves and automatic speaker recognition techniques,” *Speech Communication*, vol. 5, no. 1, pp. 183 - 197, 1986.

[19] J. S. Garofolo et al, *DARPA TIMIT: Acoustic-Phonetic Continuous Speech Corpus,* New Jersey: NIST Publications, 1993.

[20] Y. Lavner, J. Rosenhouse, and I. Gath, ”The prototype model in speaker identification,” *Proc. EuroSpeech ’99,* (Budapest, Hungry), September 1999, pp. 771-774.

[21] H. Gish, M. Krasner, W. Russell, and J. Wolf, ”Methods and experiments for text-independent speaker recognition over telephone channels,” *Proc. ICASSP’86,* (Tokyo, Japan), April 1986, pp. 865-868.

[22] P. Gillinari and X. Driancourt, ”A speech recognizer optimaly combining learning vector quantization, dynamic programming and multi-layer perceptron,” *Proc. ICASSP'92,* (San Francisco, California), March 1992, pp. 609-612.

[23] S. Hayakawa, K. Takeda, and F. Itakura, ”Speaker identification using harmonic structure of LP-residual spectrum,” *Proc. AVBPA'97*, (Crans-Montana, Switzerland), March 1997, pp. 253-260.

[24] M. H. Hayes, *Statistical Digital Signal Processing and Modeling,* New York: John Wiley & Sons, 1996.

[25] H. Hermansky, ”Perceptual linear predictive (PLP) analysis of speech,”*J. Acoustic. Soc. Amer.,* vol. 87, no. 1, pp. 1738-1752, 1990.

[26] J. Makhoul, ”Linear Prediction: A tutorial review,” *IEEE Proc.,* vol. 63, pp. 561 - 580, 1975.

[27] A. L. Higgins and R. E. Wohlford, ”A new method for text-independent speaker recognition,” *Proc. ICASSP'86,* (Tokyo, Japan), April 1986, pp. 869-872.

[28] B. H. Juang, D. Y. Wong, and A. H. Gray Jr., ”Distortion performance of VQ for LPC voice coding,” *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-34, no. 1, pp. 52-59, February 1986.

[29] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, ”On the use of bandpass liftering in speech recognition,” *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-35, no. 7, pp. 947-954, July 1987.

[30] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine,* pp. 4-16, January 1986.

[31] Y. H. Kao, J. S. Baras, and P. K. Rajasekaran, "Robustness study of free-text speaker identification and verification," *Proc. ICASSP'93,* (Minneapolis, Minnesota), April 1993, pp. II-379-II-382.

[32] Unknown, "The Linguistic Data Consortium," *http://www.ldc.upen.edu*, April 2000.

[33] K. P. Li and E. H. Wrench, "An approach to text-independent speaker recognition with short utterances," *Proc. ICASSP'83,* (Boston, Massachusetts), April 1983, pp. 555-558.

[34] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantisation," *IEEE Trans. Communications,* vol. COM-28, no. 1, pp 84-95, January 1980.

[35] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust speaker recognition," *IEEE Signal Processing Magazine,* pp. 58-71, September 1996.

[36] J. Markel and A. H. Gray Jr, *Linear Prediction of Speech,* Berlin: Springer-Verlag, 1976.

[37] A. Martin and M. Przybocki, "The NIST 1999 speaker recognition evaluation - an overview," *Digital Signal Processing,* vol. 10, pp. 1-18, June 2000.

[38] T. Matsui and S. Furui, ”A text-independent speaker recognition method robust against utterance variations,” *Proc. ICASSP’91,* (Toronto, Canada), May 1991, pp. 377-380.

[39] T. Matsui and S. Furui, ”Speaker Recognition Technology,” *NTT Review*, vol. 7, no. 2, pp. 40-48, 1995.

[40] T. Matsui and S. Furui, ”Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs,” *Proc. ICASSP’92,* (San Francisco, California), March 1992, pp. II-157-II-160.

[41] J. A. Moorer, ”The optimum comb method of pitch period analysis of continuous digitized speech,” *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. 22, no. 5, pp. 330 - 338, October 1974.

[42] J. M. Naik, ” Speaker verification: a tutorial,” *IEEE Communications Magazine,* pp. 42-48, January 1990.

[43] J. Oglesby and J. S. Mason, ”Optimization of neural models for speaker identification,” *Proc. ICASSP’90*, (Albuquerque, New Mexico), April 1990, pp. 261-264.

[44] J. Oglesby and J. S. Mason, ”Radial basis function network for speaker recognition,” *Proc. ICASSP’91,* (Toronto, Canada), May 1991, pp. 393-396.

[45] K. K. Paliwal, ”Speech processing techniques.” in *Advances in Speech, Hearing and Language Processing,* W. A. Ainsworth, Ed. London: JAI Press, pp. 1-78, 1990.

[46] T. W. Parsons, *Voice and Speech Processing,* New York: Mc Graw-Hill Book Company, pp. 197-219, pp. 138-141, 1976.

[47] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition,* New Jersey: Prentice Hall, pp. 14-17, pp. 52-65, pp. 112-117, pp. 183-191, 1993.

[48] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-25, no. 1, pp. 24-33, February 1977.

[49] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals,* New Jersey: Prentice Hall, pp. 141-161, pp. 314-322, pp. 476-485, 1978.

[50] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE,* vol. 77, pp. 257-286, February 1989.

[51] D. A. Reynolds, "Automatic speaker recognition using gaussian mixture speaker models," *Lincoln Laboratory Journal*, vol. 8, no. 2, pp. 173-192, 1995.

[52] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. Speech and Audio Processing,* vol. SAP-3, no. 1, pp. 72-83, January 1995.

[53] D. A. Reynolds, "A gaussian mixture modeling approach to text-independent speaker identification," Ph.D. Thesis, Georgia Institute of Technology, 1992.

[54] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1, pp. 91-108, 1995.

[55] D. A. Reynolds, "Experimental evaluation of features for robust speaker identification," *IEEE Trans. Speech Audio Processing*, vol. SAP-2, no. 4, pp. 639-643, 1992.

[56] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing,* vol. 10, pp. 19-41, June 2000.

[57] D. A. Reynolds, M. A. Zissman, T. F. Quateri, G. C. O'Leary, and B. A. Carlson, "The effects of telephone transmission degradations on speaker recognition performance," *Proc. ICASSP'95,* (Detroit, Michigan), May 1995, pp. 329-332.

[58] D. A. Reynolds and B. A. Carlson, "Text-dependent speaker verification using decoupled and integrated speaker and speech recognizers," *Proc. EuroSpeech'95,* (Madrid, Spain), September 1995, pp. 647-650.

[59] R.C. Rose, J. Fitzmaurice, E. M. Hofstetter, and D. A. Reynolds, "Robust speaker identification in noisy environments using adaptive speaker models," *Proc. ICASSP'91,* (Toronto, Canada), May 1991, pp. 401-404.

[60] R. C. Rose and D. A. Reynolds, "Text-independent speaker identification using automatic acoustic segmentation," *Proc. ICASSP'90,* (Albuquerque, New Mexico), April 1990, pp. 293-296.

[61] A. E. Rosenberg, and F. K. Soong, "Recent research in automatic speaker recognition," in *Advances in Speech Signal Processing,* S. Furui, M. Sondhi, Eds. New York: Marcel Dekker Inc., pp. 701-737, 1992.

[62] A. E. Rosenberg, "Automatic speaker verification: a review," *Proc. IEEE,* vol. 64, pp. 457-487, April 1976.

[63] A. E. Rosenberg, C. H. Lee, and F. K. Soong, "Sub-word unit talker verification using hidden markov models," *Proc. ICASSP'90,* (Albuquerque, New Mexico), April 1990, pp. 269-272.

[64] M. J. Ross et al, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-22, no. 5, pp. 353-362, October 1974.

[65] H. Sakoe and S. Chiba, "Dynamic programming optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-26, no. 1, pp. 43-49, January 1992.

[66] H. Sakoe, "Dynamic programming-based speech recognition," in *Advances in Speech Signal Processing,* S. Furui, M. Sondhi, Eds., New York: Marcel Dekker Inc., pp. 487-507, 1992.

[67] R. M. Schroeder, " Period histogram and product spectrum: new methods for fundamental-frequency measurement," *J. Acoustic. Soc. Amer.,* vol. 45, no. 1, p. 316(A), January 1969.

[68] D. O'Shaughnessy, "Speaker recognition," *IEEE ASSP Magazine,* pp. 4-17, October 1986.

[69] D. O'Shaughnessy, *Speech Communications - Man and Machine,* New York: IEEE Press, 2nd Ed., p. 199, pp. 437-458 , 2000.

[70] H. Silverman and D. Morgan, "The application of dynamic programming to connected speech segmentation," *IEEE Acoustics, Speech and Signal Processing Magazine,* vol. 7, no. 3, pp. 7-25, 1990.

[71] M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio and Electro - Acoustics,* vol. AU-16, no. 2, pp. 262 - 266, June 1968.

[72] F. K. Soong, A. E. Rosenberg, and B. H. Juang, "A vector quantization approach to speaker recognition," *AT & T Journal*, vol. 66, no. 2, pp. 14-26, 1987.

[73] F. K. Soong and A. E. Rosenberg, "Use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-36, no. 6, pp. 871-879, 1988.

[74] F. K. Soong, A. E. Rosenberg, and B. H. Juang, "A vector quantization approach to speaker recognition," *Proc. ICASSP'85*, (Tampa, Florida), March 1985, pp. 387-390.

[75] F. K. Soong and A. E. Rosenberg, "Use of instantaneous and transitional spectral information in speaker recognition," *Proc. ICA'88,* (Montreal, Canada), June 1988, pp. 877-880.

[76] P. de Souza, B. Ramabhadran, Y. Goa, and M. Picheny, "Enhanced likelihood computation using regression," *Proc. EuroSpeech '99,* (Budapest, Hungry), September 1999, pp. 1699-1702.

[77] P. Thevenaz and H. Hugli, "Usefulness of the LPC-residue in text-independent speaker verification," *Speech Communication*, vol. 17, no. 1, pp. 145 - 157, 1995.

[78] G. J. Tortora and S. R. Grabowski, *Principles of Anatomy and Physiology,* (8th ed.) New York: Harper Collins, p. 709, 1996.

[79] D. Tran and M. Wagner, "A proposed likelihood transformation for speaker verification," *Proc. ICASSP 2000,* (Istanbul, Turkey), April 2000, pp. 1069-1072.

[80] C. K. Un and S. C. Yang, "A pitch extraction algorithm based on LPC inverse filtering and AMDF," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-25, no. 6, pp. 565-572, December 1977.

[81] B. Wildermoth and K. K. Paliwal, "Use of voicing and pitch information for speaker recognition,", *Proc. Speech Science and Technology 2000,* (Canberra, ACT), December 2000, pp. 324 - 328.

[82] J. D. Wise et al, "Maximum likelihood pitch estimation," *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. ASSP-24, no. 5, pp. 418-423, October 1976.

[83] S. Young, "A review of large-vocabulary continuous-speech recognition,"*Signal Processing Magazine,* pp. 45-58, September 1996.