School of Computer Science and Information System

Master's Dissertation

Assessing the discriminative power of Voice

Submitted by Pasupathy Naresh Trilok

Supervised by Dr. Sung-Hyuk Cha
Dr. Charles Tappert

Defense Date 14th January, 2004

We hereby certify that this dissertation, submitted by Pasupathy Naresh Trilok, satisfies the dissertation requirements of the Masters degree of Computer Science and has been approved.

-

Dr. Sung-Hyuk Cha
Supervisor
Chairperson of Dissertation Committee

Date

-

Dr. Charles Tappert
Dissertation Committee Member

Date

-

Dr. Narayan Murthy, Chair, CS Dept.
Dissertation Committee Member

Date

-

Dr. Susan M. Merritt
Dean, School of Computer Science
and Information Systems

Date

Abstract

This study establishes the individuality of voice, using the discriminative power of biometric data quantitatively. The task of establishing voice modality to discriminate every person is difficult because there are a large number of classes for the entire population. The paper proposes a methodology that is statistically inferable. A many class problem is transformed into a dichotomy by using distances between measurements of intra and inter-person classes. This establishes a thorough distinction of classes and thereby validates distinct individuality. This model remains statistically inferable even when it does not observe all the classes.

Acknowledgments

I would like to thank Drs. Cha and Tappert for their continued guidance and support during my work on this thesis.

Chapter 1

Introduction

The task considered is that of establishing the individuality of the voice of each individual in a population. This task of establishing individuality is the same as showing the distinctiveness of classes with a very small error rate in discrimination. Individuality in handwriting has been shown in [26]. This paper proposes to give a validation of the methodology used in [26] for individuality in voice, and also to generalize the results to other domains. The same model has been recently shown to establish individuality in fingerprints [18].

Motivation

Speech recognition is the field of computer science that deals with designing computer systems that can recognize spoken words. Although handwriting, fingerprints, face, etc have been recognized as distinct per individual and used for verification purposes, the voice of the speaker has not been used with this model. Current voice recognition systems are based on the polychotomy principle that has a distinct disadvantage of being statistically non-inferential and thereby requiring many more observable instances of the same class in the training data. This paper proposes to show the individuality of voice by dichotomy, which has the advantage of being statistically inferential.

Individuality

The task of showing individuality is the same as showing the distinctiveness of the classes with a very small error rate in discrimination.

Statistical Inference

Statistical inference infers a conclusion about the population of interest from a sample. If the error rate of the random sample set the same as the error in the universe, the procedure is said to be statistically inferential. Inferential statistics is the measure of reliability of individuality about the entire population based on data obtained from a sample drawn out of that population.

Problem statement

Two audio inputs will be taken from speakers and used for the biometric determination of speaker individuality by determining whether the two inputs come from the same person or from different people.

Hypotheses

1. The individuality of the speaker can be shown when the speech is normal.
2. The individuality of the speaker can be shown when the speech is disguised.

1.1 Basic definitions

The human speech conveys different types of information. The primary type is the meaning or words, which speaker tries to pass to the listener. But the other types that are also included in the speech are information about language being spoken, speaker emotions, gender and identity of the speaker. The goal of automatic *speaker recognition* is to extract, characterize and recognize the information about speaker identity [21]. Speaker recognition is usually divided into two different branches, *speaker verification* and *speaker identification*. Speaker verification task is to verify the claimed identity of person from his voice [3,16]. This process involves only binary decision about claimed identity.

1.2 Applications

Practical applications for automatic speaker identification are obviously various kinds of security systems. Human voice can serve as a key for any security objects, and it is not so easy in general to lose or forget it. Human voice can also be used to prove identity during access to any physical facilities by storing speaker model in a small chip, which can be used as an access tag, and used instead of a pin code. Another important application for speaker identification is to monitor people by their voices. For instance, it is useful in information retrieval by speaker indexing of some recorded debates or news, and then retrieving speech only for interesting speakers. It can also be used to monitor criminals in common places by identifying them by voices. In fact, all these examples are actually examples of real time systems.

1.3 Thesis Description

Nowadays, speaker verification is not anymore just a theory. Applications based on it are widely used around the word and found their appropriate places in the industry. But even though a lot of work has already done in this field [3,5,11], it is still not a solved problem. The research in the area of speaker verification still continues and at present there are a few basic techniques that have shown their effectiveness in practice and called “classical” by scientists. The goal of this work is to make general overview of these techniques and then propose a new approach for binary decision making for speaker verification purposes.

To give a better understanding, we start from the very beginning. In Chapter 2, we study the fundamentals of digital signal processing theory used in speaker verification, and model of biometric characteristics of human speech production organs. This model will serve us as a basis for techniques described in the next chapters. In Chapter 3, we study the popular method for the extraction of the speaker characteristics from speech signal. In Chapter 4, we discuss classifications and ways for modeling of extracted characteristics and methods, used to calculate the dissimilarity value between unknown speech sample and the stored speaker models. In Chapter 5, we discuss the approaches used for the verification problem. In Chapter 6, we evaluate the proposed approach with experiments and showcase the results of the experiment. Finally, we finish this work by giving short discussion and conclusions in Chapter 7.

Chapter 2

Verification Background

In this chapter we discuss theoretical background for speaker verification. We start from the digital signal processing theory. Then we move to the anatomy of human voice production organs and discuss the basic properties of the human speech production mechanism and techniques for its modeling. This model will be used in the next chapter when we will discuss techniques for the extraction of the speaker characteristics from the speech signal.

2.1 DSP Fundamentals

According to its abbreviation, *Digital Signal Processing (DSP)* is a part of computer science, which operates with special kind of data – *signals*. In most cases, these signals are obtained from various sensors, such as microphone or camera. DSP is the mathematics, mixed with the algorithms and special techniques used to manipulate with these signals, converted to the digital form [24].

2.1.1 Basic Definitions

By *signal* we mean here a relation of how one parameter is related to another parameter. One of these parameters is called *independent parameter* (usually it is time), and the other one is called *dependent*, and represents what we are measuring. Since both of these parameters belong to the continuous range of values, we call such signal *continuous signal*. When continuous signal is passed through an *Analog-To-Digital converter (ADC)* it is said to be *discrete* or *digitized* signal. Conversion works in the following way: every time period, which occurs with frequency, called *sampling frequency*, signal value is

taken and *quantized*, by selecting an appropriate value from the range of possible values. This range is called *quantization precision*, and usually represented as an amount of bits available to store signal value. Based on the *sampling theorem*, proved by Nyquist in 1940 [24], digital signal can contain frequency components only up to one half of the sampling rate. Generally, continuous signals are what we have in nature while discrete signals exist mostly inside computers. Signals that use time as the independent parameter are said to be in the *time domain*, while signals that use frequency as the independent parameter are said to be in the *frequency domain*.

One of the important definitions used in DSP is the definition of *linear system.* By *system* we mean here any process that produces *output* signal in a response on a given *input* signal. A system is called linear if it satisfies the following three properties: *homogeneity*, *additivity* and *shift invariance* [24]. Homogeneity of a system means that change in the input signal amplitude corresponds to the change in the output signal. Additivity means that the output of the sum of two signals results in the sum of the two corresponding outputs. And finally, shift invariance means that any shift in the input signal will result in the same shift in the output signal [5,19,24].

2.1.2 Convolution

An *impulse* is a signal composed of all zeros except one non-zero point. Every signal can be decomposed into a group of impulses, each of them then passed through a linear system and the resulting output components are synthesized or added together [24]. The resulting signal is exactly the same as obtained by passing the original signal through the system.

Every impulse can be represented as a shifted and scaled *delta function*, which is a *normalized* impulse, that is, sample number zero has a value of one and all other samples have a value of zero. When the delta function is passed through a linear system, its output is called *impulse response*. If two systems are different they will have different impulse responses. According to the properties of linear systems every impulse passed through it will result in the scaled and shifted impulse response and scaling and shifting of the input are identical to the scaling and shifting of the output [19,24]. It means that knowing systems impulse response we know everything about the system [5,19,24].

Convolution is a formal mathematical operation, which is used to describe relationship between three signals of interest: input and output signals, and the impulse response of the system. It is usually said that the output signal is the input signal convolved with the system's impulse response. Mathematical equation of convolution for discrete signals is represented in the following (convolution is denoted as a star):

$$y[i] = x[i] * h[i] = \sum_{j=0}^{M-1} h[j]x[i-j] \qquad \textbf{(2.1)}$$

Where *y [i]* is the output discrete signal, *x [i]* is the input discrete signal and *h [j]* is *M* samples long system's impulse response *flipped left-for-right*. Index *I* goes through the size of the output signal. Mathematics behind the convolution does not restrict how long the impulse response is. It only says that the size of the output signal is the size of the input signal plus the size of the impulse response minus one.

Convolution is very important concept in DSP. Based on the properties of linear systems it provides the way of combining two signals to form a third signal. A lot of mathematics behind the DSP is based on the convolution. In detail it is described in [5,19,24].

2.1.3 Discrete Fourier Transform

Fourier transform belongs to the family of linear transforms widely used in DSP based on decomposing signal into sinusoids (sine and cosine waves). Usually in DSP we use the *Discrete Fourier Transform* (DFT), a special kind of Fourier transform used to deal with aperiodic discrete signals [24]. Actually there are an infinite number of ways how signal can be decomposed but sinusoids are selected because of their *sinusoidal fidelity* that means that sinusoidal input to the linear system will produce sinusoidal output, only the amplitude and phase may change, frequency and shape remain the same [24].

Discrete Fourier Transform changes an *N* point input signal into two *N/2+1* point output signals. The output signals represent the amplitudes of the sine and cosine components scaled in a special way that is represented by the equations:

$$C_k[i] = \cos(2 \cdot k \cdot i \cdot \pi / N)$$
$$S_k[i] = \sin(2 \cdot k \cdot i \cdot \pi / N) \qquad \textbf{(2.2)}$$

Where, *Ck* are *N/2+1* cosine functions and *Sk* are *N/2+1* sine functions, index *k* runs from zero to *N/2*. These functions are called *basis functions*. Actually zero samples in resulting signals are amplitudes for zero frequency waves, first samples for waves which make one complete cycle in *N* points, second for waves which make two cycles and so on. Signal represented in such a way is called to be in *frequency domain* and obtained

coefficients are called *spectral coefficients* or *spectrum*. Frequency domain contains exactly the same information as the time domain and every discrete signal can be moved back to the time domain, using operation called *Inverse Discrete Fourier Transform (IDFT)*. Because of this fact, the DFT is also called *Forward DFT* [24]. Schematically DFT is represented in Figure 2.1.

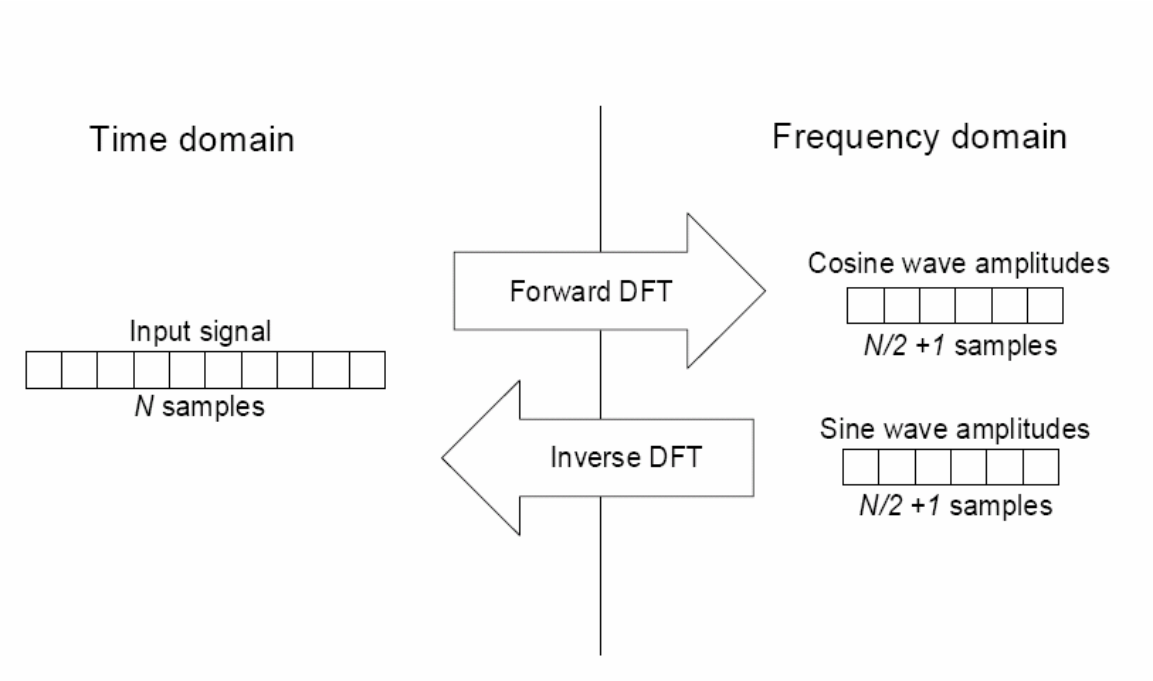


Figure 2.1 Discrete Fourier Transform

The amplitudes for cosine waves are also called *real part* (denoted as *Re[k]*) and for sine waves are called *imaginary part* (denoted as *Im[k]*). This representation of frequency domain is called *rectangular* notation. Alternatively, the frequency domain can be expressed in the *polar* notation. In this form, real and imaginary parts are replaced by magnitudes (denoted as *Mag[k]*) and phases (denoted as *Phase[k]*) respectively [24]. The equations for conversion from rectangular notation to the polar notation are as follows:

$$Mag[k] = \sqrt{(\mathrm{Re}[k]^2 + \mathrm{Im}[x]^2)}$$

$$Phase[k] = \arctan\left(\frac{\mathrm{Im}[k]}{\mathrm{Re}[k]}\right)$$

(2.3)

There are two main reasons why DFT became so popular in DSP. First is *Fast Fourier Transform (FFT)* algorithm [24], developed by Cooley and Tukey in 1965, which opened a new era in DSP because of the efficiency of the FFT algorithm. The second reason is the *convolution theorem* [24], which states that convolution in time domain is a multiplication in frequency domain and vice versa. This makes possible to use high-speed convolution algorithm, which convolves two signals by passing them through the Fast Fourier Transform, multiplying and using Inverse Fourier Transform computing convolved signal. More details about Fourier Transform can be found in [5,19,24].

2.1.4 Filters

By *filter* we mean here a method to manipulate with signals defined as a linear system. There are two main uses for filters: signal *separation* and signal *restoration*. Signal separation is needed when the signal was interfered with the other not useful signals or noise. Signal restoration is needed when the signal was distorted for example due to the transform through a long wire or bad quality recording. There are two main types of filters: *analog* and *digital*. Analog filters are cheap and have a large dynamic range in frequency and amplitude. However, digital filters can achieve thousands better performance [24].

Easiest way to implement a digital filter is to convolve the input signal with the filters impulse response. Based on the length of its impulse responses, filters are usually divided into *Infinite Impulse Response (IIR)* filters and *Finite Impulse Response (FIR)* filters. There are also few types of responses: *step response* and *frequency response*. Each of these responses can be used to completely define filter. Step response is the output signal of the filter when input is a *step function*, which is defined as a transition from one level of signal to another. This type of responses can be used to define filters, which are able to divide signal into regions with similar characteristics. The frequency response can be found by taking discrete Fourier transform of the impulse response. It can be useful to define filters, which are able to block undesirable frequencies in input signals or separate one band of frequencies from another, such as high-pass, band-pass and band-reject filters.

Digital filter theory is important in speaker identification, since it allows by a given signal to analyze origin of it or in this case the unknown speaker. There are also few minor uses for filters like a noise removal or other types of filtering to achieve better results in signal analyzing. More details about filter design and implementation can be found in [5,19,24].

2.2 Human Speech Production Model

The ability to speak is the most important way for humans to communicate between each other. Speech conveys various kind of information, which is essentially the meaning of information speaking person, wants to impart, individual information representing speaker and also some emotional filling. Speech production begins with the initial formalization of the idea which speaker wants to impart to the listener. Then speaker

converts this idea into the appropriate order of words and phrases according to the language. Finally, his brain produces motor nerve commands, which move the vocal organs in an appropriate way [9]. Understanding of how human produce sounds forms the basis of speaker verification.

2.2.1 Anatomy

The sound is an acoustic pressure formed of compressions and rarefactions of air molecules that originate from movements of human anatomical structures [11]. Most important components of the human speech production system are the *lungs* (source of air during speech), *trachea* (windpipe), *larynx* or its most important part *vocal cords* (organ of voice production), *nasal cavity* (nose), *soft palate* or *velum* (allows passage of air through the nasal cavity), *hard palate* (enables consonant articulation), *tongue*, *teeth* and *lips*. All these components, called *articulators* by speech scientists, move to different positions to produce various sounds. Based on their production, speech sounds can also be divided into consonants and voiced and unvoiced vowels [5,11].

From the technical point of view, it is more useful to think about speech production system in terms of an acoustic filtering operation that affect the air going from the lungs. There are three main cavities that comprise the main acoustic filter. According to [5] they are *nasal*, *oral* and *pharyngeal* cavities. The articulators are responsible for changing the properties of the system and form its output. Combination of these cavities and articulators is called *vocal tract*. Its simplified acoustic model is represented in Figure2.2.

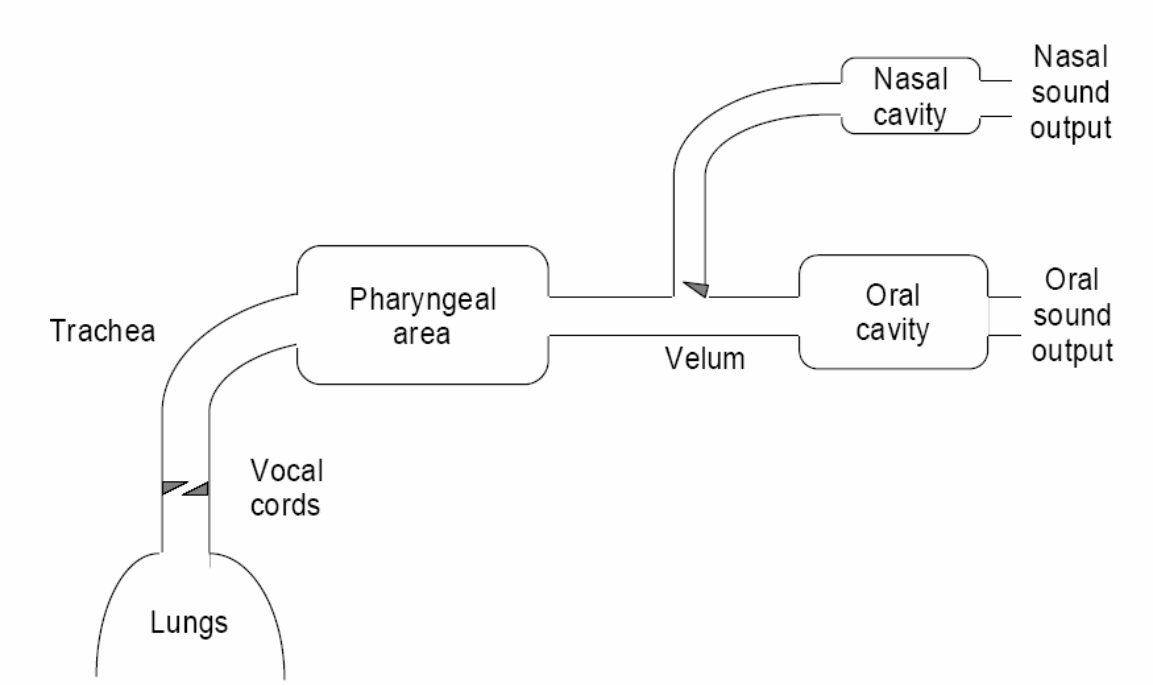


Figure 2.2 Vocal tract model

Speech production can be divided into three stages: first stage is the sound source production, second stage is the articulation by vocal tract, and the third stage is sound radiation or propagation from the lips and/or nostrils [9]. A v*oiced sound* is generated by vibratory motion of the vocal cords powered by the airflow generated by expiration. The frequency of oscillation of vocal cords is called the *fundamental frequency*. Another type of sounds - *unvoiced sound* is produced by turbulent airflow passing through a narrow constriction in the vocal tract [3,5].

In a speaker recognition task, we are interested in the physical properties of human vocal tract. In general it is assumed that vocal tract carries most of the speaker related information [3,5,11,20]. However, all parts of human vocal tract described above can serve as speaker dependent characteristics [3,5,20]. Starting from the size and power of lungs, length and flexibility of trachea and ending by the size, shape and other physical characteristics of tongue, teeth and lips. Such characteristics are called *physical*

distinguishing factors. *A*nother aspects of speech production that could be useful in discriminating between speakers are called *learned factors*, which include speaking rate, dialect, and *prosodic* effects [3].

2.2.2 Vocal Model

In order to develop an automatic speaker identification system, we should construct reasonable model of human speech production system. Having such a model, we can extract its properties from the signal and, using them, we can decide whether or not two signals belong to the same model and as a result to the same speaker.

Modeling process is usually divided into two parts: the excitation (or source) modeling and the vocal tract modeling [5]. This approach is based on the assumption of independence of the source and the vocal tract models [3,5]. Let us look first at the *continuous-time* vocal tract model called *multitube lossless model* [5], which is based on the fact that production of speech is characterized by changing the vocal tract shape. Because the formalization of such a time-varying vocal-tract shape model is quite complex, in practice it is simplified to the series of concatenated lossless acoustic tubes with varying cross-sectional areas [5], as shown in Figure 2.3.

This model consists of a sequence of tubes with cross-sectional areas *Ak* and lengths *Lk*. In practice the lengths of tubes assumed to be equal [5]. If a large amount of short tubes is used, then we can approach to the continuously varying cross-sectional area, but at the cost of more complex model. Tract model serves as a transition to the more general *discrete-time* model, also known as *source-filter model*, which is shown in Figure 2.4 [5].

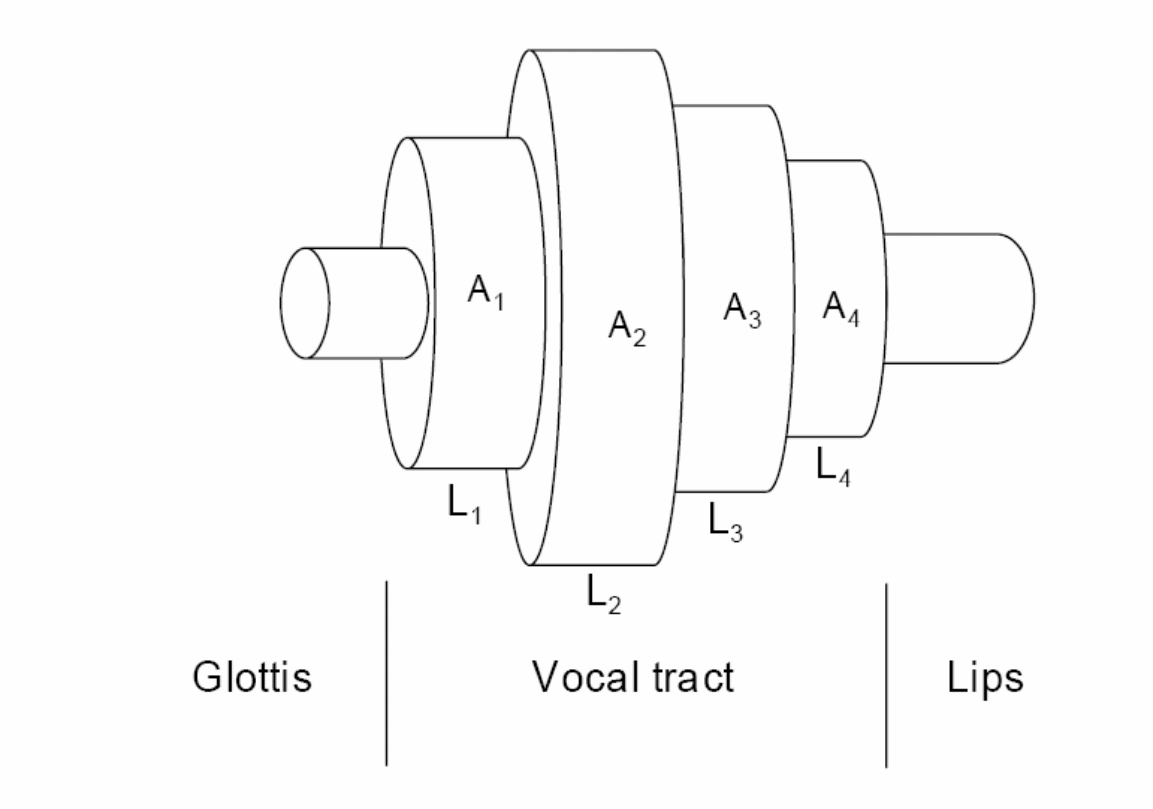


Figure 2. 3 Multitube lossless model

In this model, the voice source is either a periodic pulse stream or uncorrelated white noise, or a combination of these. This assumption is based on the evidence from human anatomy that all types of sounds, which can be produced by humans, are divided into three general categories: voiced, unvoiced and combination of these two (2.2.1). Voiced signals can be modeled as a basic or fundamental frequency signal filtered by the vocal tract and unvoiced as a white noise also filtered by the vocal tract. Here *E(z)* Represents the *excitation function*, *H(z)* represents the *transfer function*, and *s(n)* is the output of the whole speech production system [5].

Finally, we can think about vocal tract as a digital filter, which affects source signal and about produced sound output as a filter output. Then based on the digital filter theory we can extract the parameters of the system from its output.

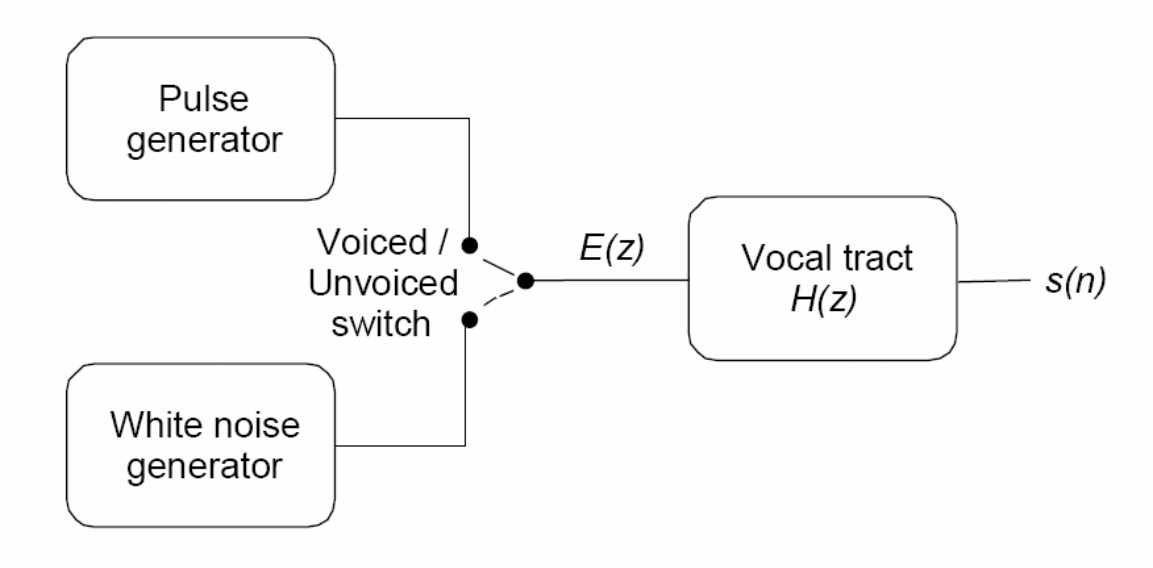


Figure 2.4 Source-filter model

The issues described in this chapter serve as a basis for developing speaker identification techniques described in the next chapter. More details about speech production system modeling can be found in [3,5,11,20].

Chapter 3

Feature Extraction

In this chapter we discuss the most widely used way of extracting speaker discriminative characteristics from speech signal.

3.1 Introduction

The acoustic speech signal contains different kind of information about speaker. This includes “high-level” properties such as dialect, context, speaking style, emotional state of speaker and many others [16]. A great amount of work has been already done in trying to develop identification algorithms based on the methods used by humans to identify speaker. But these efforts are mostly impractical because of their complexity and difficulty in measuring the speaker discriminative properties used by humans [16]. More useful approach is based on the “low-level” properties of the speech signal such as *pitch* (fundamental frequency of the vocal cord vibrations), *intensity*, *formant frequencies* and their *bandwidths*, *spectral correlations*, *short-time spectrum* and others [1].

From the automatic speaker recognition task point of view, it is useful to think about speech signal as a sequence of *features* that characterize both the speaker as well as the speech. It is an important step in recognition process to extract sufficient information for good discrimination in a form and size, which is amenable for effective modeling [10]. The amount of data, generated during the speech production, is quite large while the essential characteristics of the speech process change relatively slowly and therefore, they require less data. According to these matters *feature extraction* is a process of reducing data while retaining speaker discriminative information [5,10].

Based on the issues described above, we can define requirements that should be taken into account during selection of the appropriate speech signal characteristics or features [26,16]:

- discriminate between speakers while being tolerant of intra-speaker variabilities
- easy to measure
- stable over time
- occur naturally and frequently in speech
- change little from one speaking environment to another
- not be susceptible to mimicry.

Practically, it is not possible to meet all of these criteria and there will be always a trade-off between them, based on what is more important in the particular case. The speech wave is usually analyzed based on spectral features. There are two reasons for it. First is that the speech wave is reproducible by summing the sinusoidal waves with slowly changing amplitudes and phases. Second is that the critical features for perceiving speech by humans ear are mainly included in the magnitude information and the phase information is not usually playing a key role [9].

3.2 Short-Term Analysis

Because of its nature, the speech signal is a slowly varying signal or *quasi-stationary*. It means that when speech is examined over a sufficiently short period of time (20-30 milliseconds) it has quite stable acoustic characteristics [5]. It leads to the useful concept of describing human speech signal, called "*short-term analysis*", where only a portion of the signal is used to extract signal features at one time. It works in the following way: predefined length window (usually 20-30 milliseconds) is moved along the signal with an

overlapping (usually 30-50% of the window length) between the adjacent frames. Overlapping is needed to avoid losing of information. Parts of the signal formed in such a way are called *frames.* In order to prevent an abrupt change at the end points of the frame, it is usually multiplied by a *window function*. The operation of dividing signal into short intervals is called *windowing* and such segments are called *windowed frames* (or sometime just *frames*). There are several window functions used in speaker recognition area [9], but the most popular is *Hamming window function*, which is described by the following equation:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2n\pi}{N-1}\right) \qquad \textbf{(3.1)}$$

where *N* is the size of the window or frame. A set of features extracted from one frame is called *feature vector*. Overall overview of the short-term analysis approach is represented in Figure 3.1.

More details about feature selection and extraction can be found in [1,5,9,10,16,20, 26].

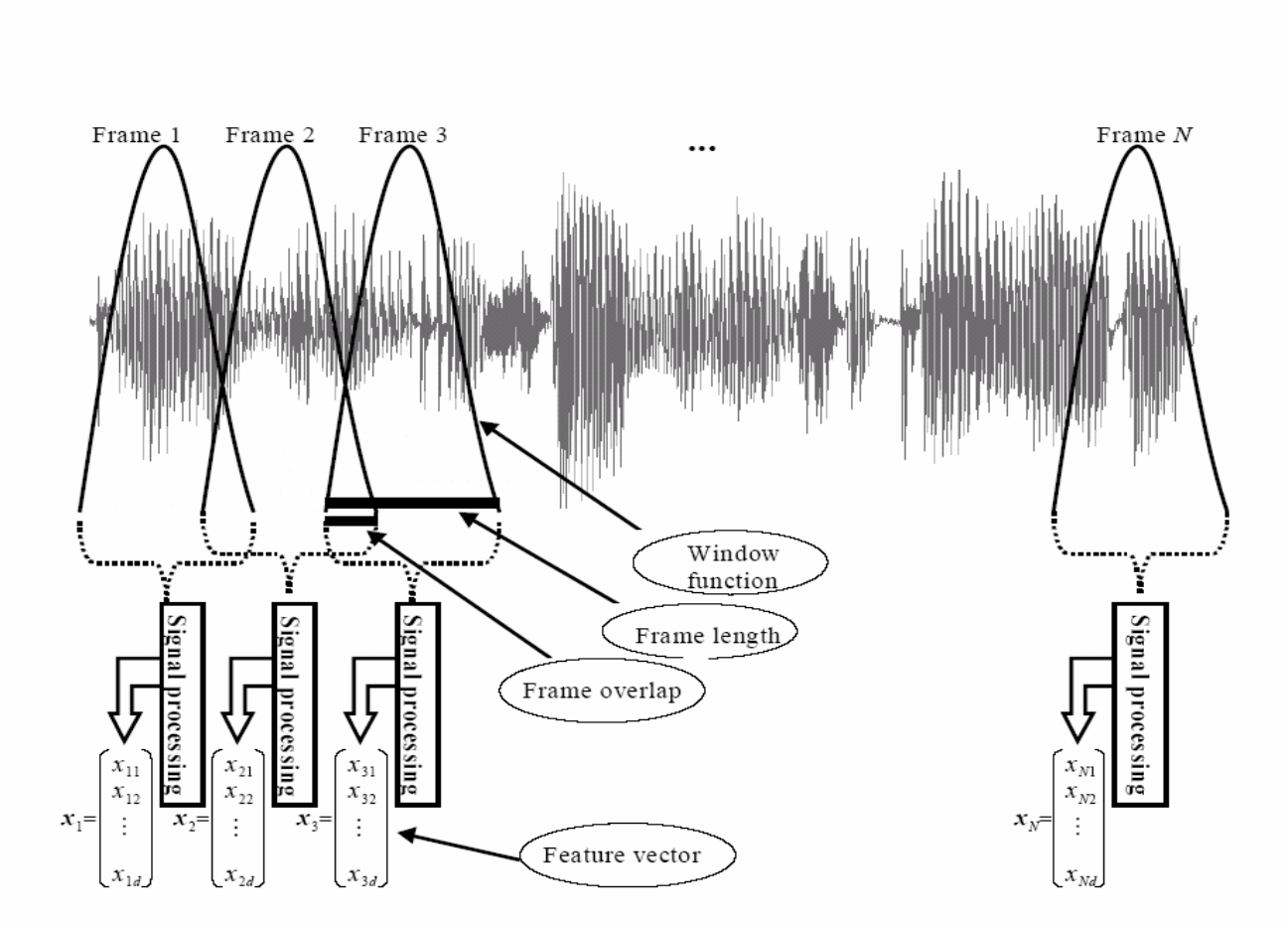


Figure 3.1 Short-Term Analysis

3.3 Cepstrum

According to the issues described in the subchapter (2.2.2), the speech signal *s(n)* can be represented as a "quickly varying" source signal *e(n)* convolved with the "slowly varying" impulse response *h(n)* of the vocal tract represented as a linear filter [5]. We have access only to the output (speech signal) and it is often desirable to eliminate one of the components. Separation of the source and the filter parameters from the mixed output is in general difficult problem when these components are combined using not linear operation, but there are various techniques appropriate for components combined linearly. The *cepstrum* is representation of the signal where these two components are

resolved into two additive parts [5]. It is computed by taking the inverse DFT of the logarithm of the magnitude spectrum of the frame. This is represented in the following equation:

$$cepstrum(frame) = IDFT\,(\log(|DFT(frame)|)) \quad \textbf{(3.2)}$$

Some explanation of the algorithm is therefore needed. By moving to the frequency domain we are changing from the convolution to the multiplication. Then by taking logarithm we are moving from the multiplication to the addition. That is desired division into additive components. Then we can apply linear operator inverse DFT, knowing that the transform will operate individually on these two parts and knowing what Fourier transform will do with quickly varying and slowly varying parts. Namely it will put them into different, hopefully separate parts in new, also called *quefrency* axis [5]. Let us look at the speech magnitude spectrum in Figure 3.2 [5].

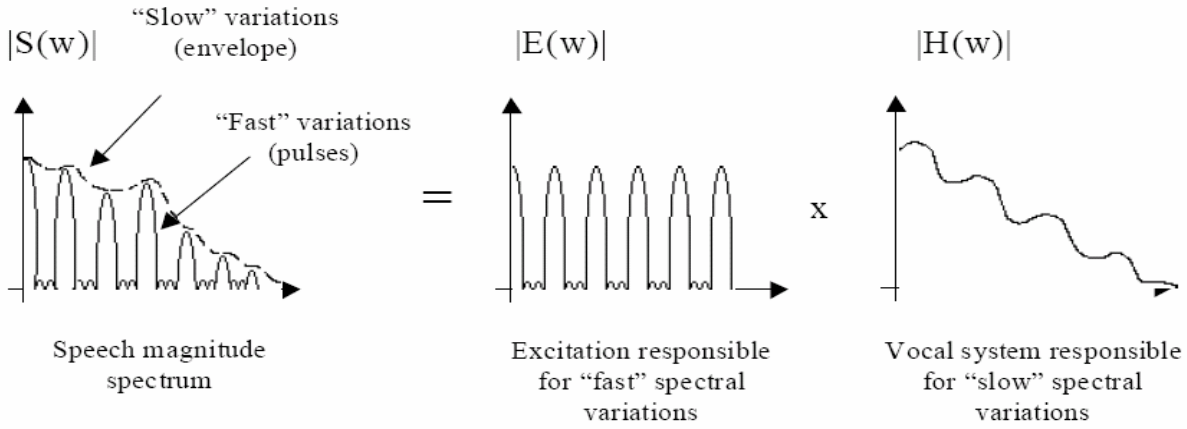


Figure 3.2 Speech magnitude spectrum

From the Figure 3.2 we can see that the speech magnitude spectrum is combined from slow and quickly varying parts. But there is still one problem: multiplication is not a

linear operation. We can solve it by taking logarithm from the multiplication as described earlier. Finally, let us look at the result of the inverse DFT in Figure 3.3 [5].

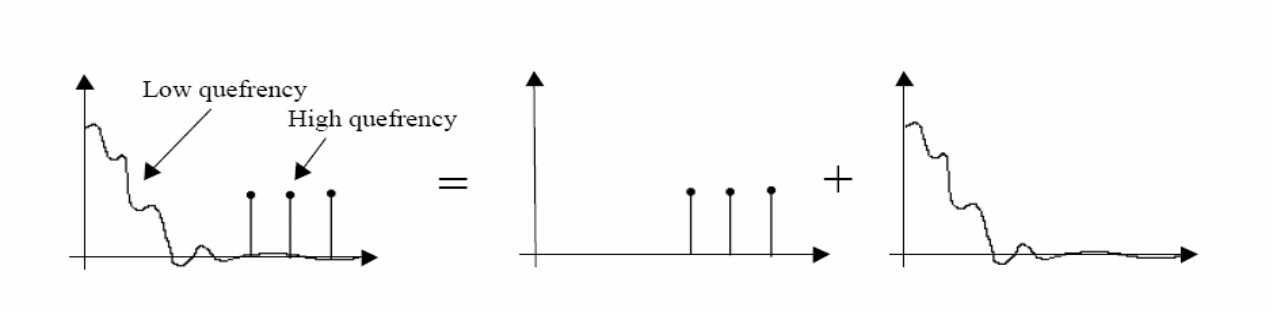


Figure 3.3 Cepstrum

From this figure we can see that two components are clearly distinctive now. Cepstrum is explained in more details in [5,10,20].

3.4 Mel-Frequency Cepstrum Coefficients

Mel-frequency cepstrum coefficients (MFCC) are well known features used to describe speech signal. They are based on the known evidence that the information carried by low-frequency components of the speech signal is phonetically more important for humans than carried by high-frequency components [5]. Technique of computing MFCC is based on the short-term analysis, and thus from each frame a MFCC vector is computed.

MFCC extraction is similar to the cepstrum calculation except that one special step is inserted, namely the frequency axis is warped according to the mel-scale. Summing up, the process of extracting MFCC from continuous speech is illustrated in Figure 3.4.

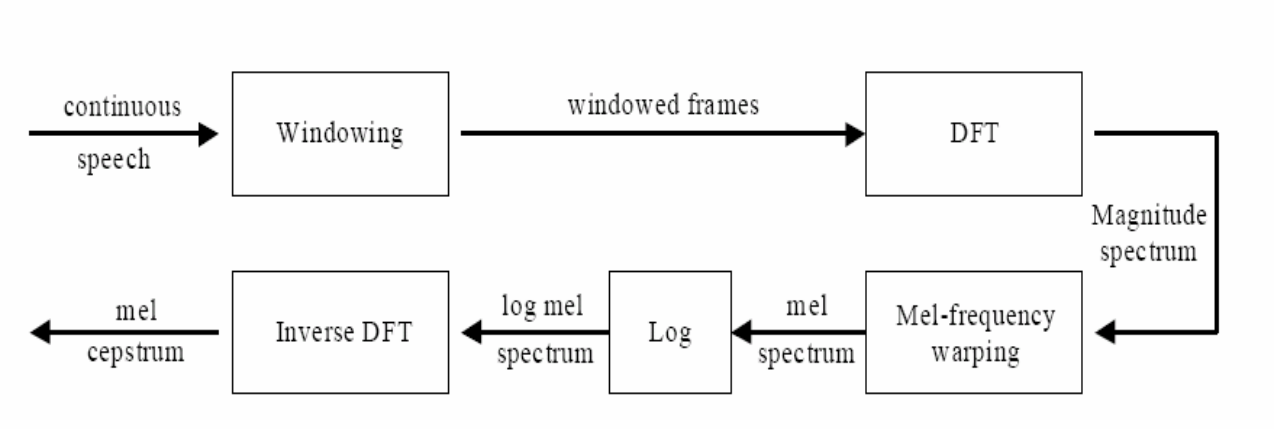


Figure 3.4 Computing of mel-cepstrum

As described above, to place more emphasize on the low frequencies one special step before inverse DFT in calculation of cepstrum is inserted, namely mel-scaling. A "*mel*" is a unit of special measure or scale of *perceived pitch* of a tone [5]. It does not correspond linearly to the normal frequency; indeed it is approximately linear below 1 kHz and logarithmic above [5]. This approach is based on the psychophysical studies of human perception of the frequency content of sounds [5,20]. One useful way to create mel-spectrum is to use a filter bank, one filter for each desired mel-frequency component. Every filter in this bank has triangular band pass frequency response. Such filters compute the average spectrum around each center frequency with increasing bandwidths, as displayed in Figure 3.5.

This filter bank is applied in frequency domain and therefore, it simply amounts to taking these triangular filters on the spectrum. In practice the last step of taking inverse DFT is replaced by taking *discrete cosine transform (DCT)* for computational efficiency.

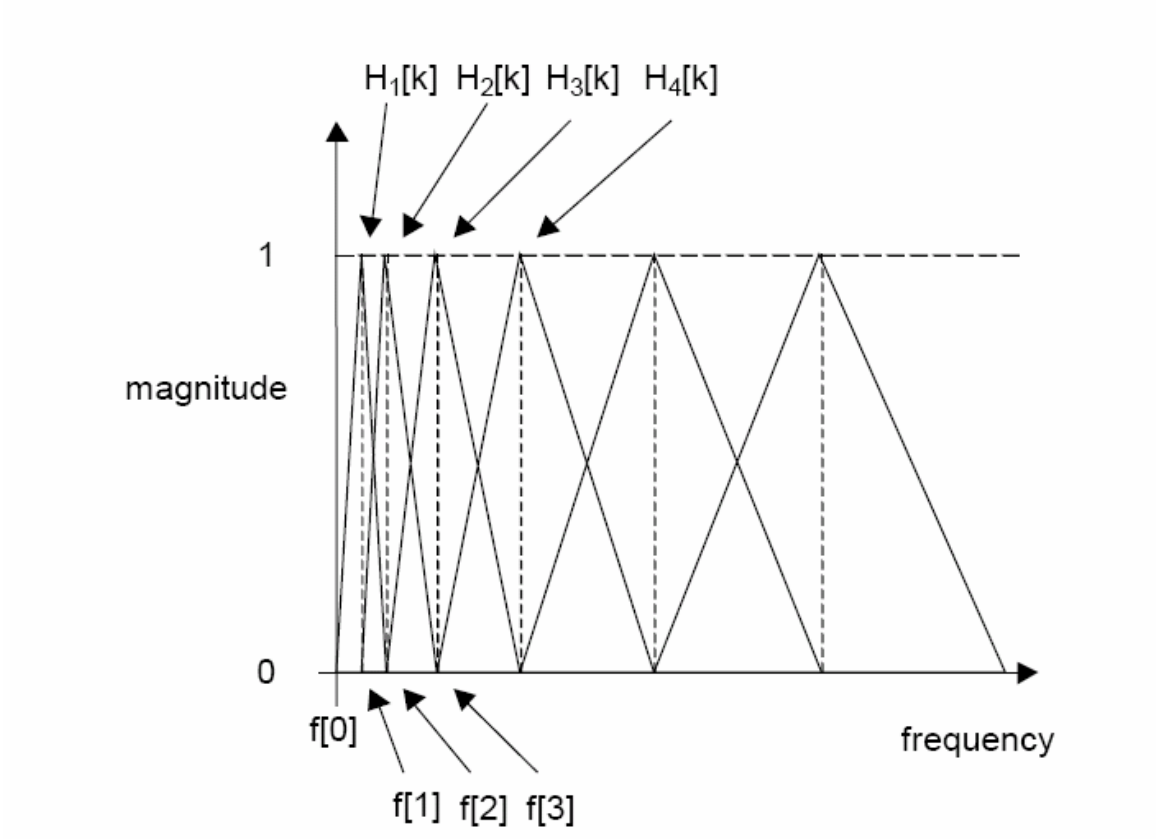


Figure 3.5 Triangular filters used to compute mel-cepstrum

The number of resulting mel-frequency cepstrum coefficients is practically chosen relatively low, in the order of 12 to 20 coefficients. The zeroth coefficient is usually dropped out because it represents the average logenergy of the frame and carries only a little speaker specific information.

3.5 MFCC Features

- Compact – The same information can be represented with fewer parameters. High-order cepstra can be discarded since they represent high-frequency variations in log-spectrum.

- Uncorrelated – The cepstral coefficients are approximately uncorrelated. In fact, for Speech signals, DCT (Discrete Cosine Transform is an approximation that makes it uncorrelated)
- Gain Independent – Only the zeroth cepstral value (a function of power) is dependent on energy (power) of the signal

3.6 Conclusion

Cepstrum representation of the speech signal has shown to be useful in practice. However, it is not without drawbacks. The main disadvantage of the cepstrum is that it is quite sensitive to the environment and noise [5]. Therefore, in practice speech signal is usually preprocessed to achieve more precise representation. This process usually includes noise removal [5,23] and pre-emphasis [5,28]. One approach for separating speaker information and environment can be found in [23]. More details about cepstrum and other feature extraction methods can be found in [1,5,9,11,10,20,21,22,26].

Chapter 4

Classification and Modeling

In this chapter we discuss techniques for modeling of features extracted from the speech signal, and methods, which are allowing computing dissimilarity between speech samples.

4.1 Introduction

In the previous chapter we were discussing so called *measurement* step in the speaker identification where a set of speaker discriminative characteristics is extracted from the speech signal. In this chapter, we go through the next step called *classification*, which is a decision making process of determining the author of a given speech signal based on the previously stored or learned information [1]. The methods used in classification could be categorized as ***Geometric, Topological and probabilistic***. The three methods are best illustrated when the test and reference patterns are viewed as points in a multi-dimensional space. The methods are explained with an example in a 2-dimensional space as in Fig 4.1

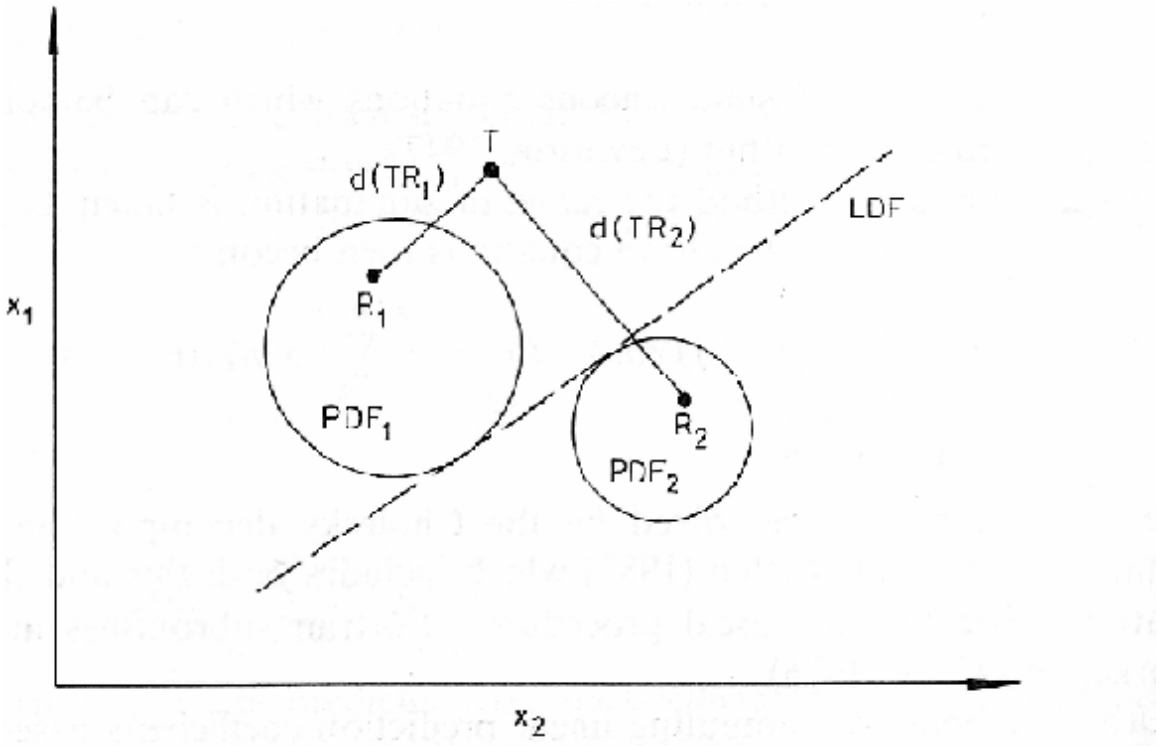


Figure 4.1 2-dimensional space of training vectors

Geometric method divides space into regions (with each class in one region) with boundaries. These boundaries are defined by *Linear Discriminant Functions*. In Fig.2 T is classified as R1, because it lies on the same side of the linear discriminant function (LDF) as R1.

In topological method**,** one, or more points in the space represent each class. The distance between the test vector point and each class is determined and the test vector is assigned to the class with the shortest distance. T is classified as R1 because the distance form T to R1 is less than distance to R2.

In probabilistic method a *probability density function* is defined for each point in the space. The test pattern is assigned to the class, which has the greatest PDF at that point. T

is classified as R1 because the probability density function PDF1 at T is greater than PDF2.

4.2 Nearest Neighbour

As the name suggests the test pattern is assigned to the nearest reference pattern in an verification/Identification problem. Hence this is a topological method. In verification, the distance between the test vector and the speaker vector is determined and if it is within a threshold then the claim is verified; else it is rejected [15]

4.3 Vector Quantization

Vector quantization (VQ) is a process of mapping vectors from a vector space to a finite number of regions in that space. These regions are called *clusters* and represented by their central vectors or *centroids*. A set of centroids, which represents the whole vector space, is called a *codebook*. In speaker identification, VQ is applied on the set of feature vectors extracted from the speech sample and as a result, the speaker codebook is generated. Such codebook has a significantly smaller size than extracted vector set and referred as a speaker model. Actually, there is some disagreement in the literature about approach used in VQ. Some authors [3] consider it as a template matching approach because VQ ignores all temporal variations and simply uses global averages (centroids). Other authors [13,16] consider it as a stochastic or probabilistic method, because VQ uses centroids to estimate the modes of a probability distribution [10]. Theoretically it is possible that every cluster, defined by its centroid, models particular component of the speech. But practically, however, VQ creates unrealistically clusters with rigid boundaries in a sense that every vector belongs to one and only one cluster.

Mathematically a VQ task is defined as follows: given a set of feature vectors, find a partitioning of the feature vector space into the predefined 30 number of regions, which do not overlap with each other and added together form the whole feature vector space. Every vector inside such region is represented by the corresponding centroid [25]. The process of VQ for two speakers is represented in Figure 4.2.

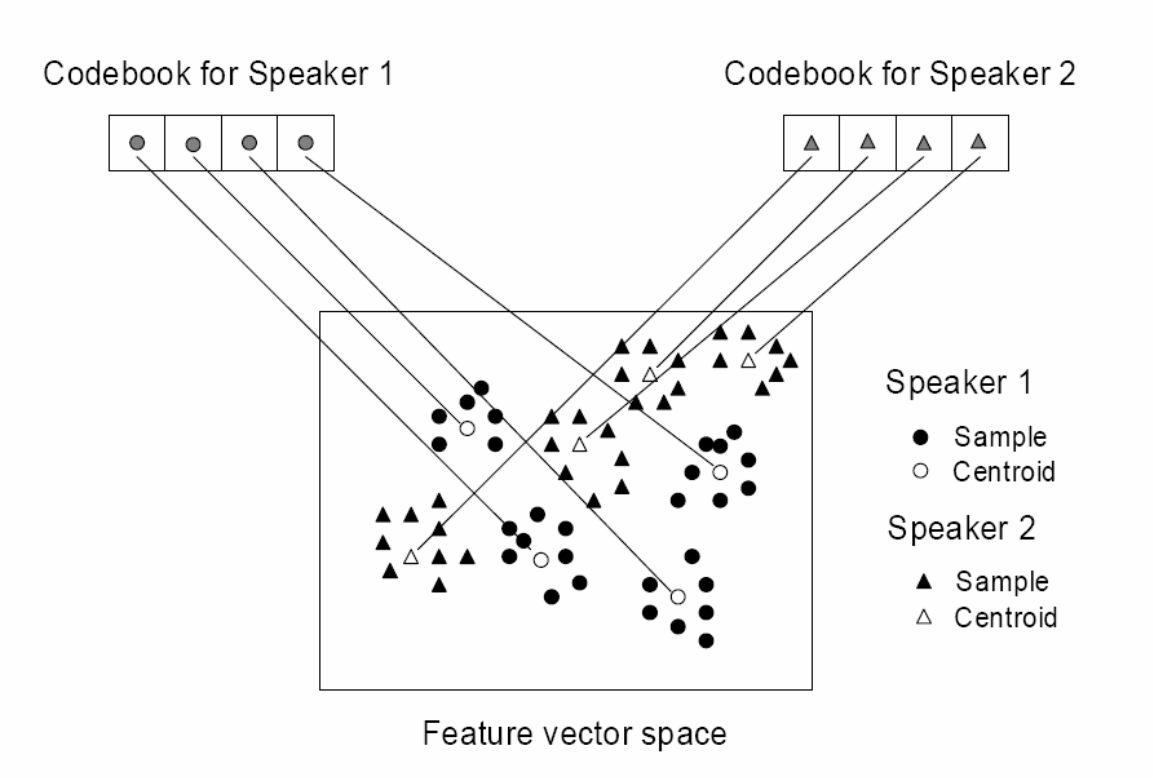


Figure 4.2 Vector quantization of two speakers

There are two important design issues in VQ: the method for generating the codebook and codebook size [12]. Known clustering algorithms for codebook generation are [12]:

- *Generalized Lloyd algorithm (GLA),*
- *Self-organizing maps (SOM),*
- *Pairwise nearest neighbor (PNN),*
- *Iterative splitting technique (SPLIT),*

- *Randomized local search (RLS).*

According to [12], iterative splitting technique [7] should be used when the running time is important but RLS [8] is simpler to implement and generates better codebooks in the case of speaker identification task. Codebook size is a trade-off between running time and identification accuracy. With large size, identification accuracy is high but at the cost of running time and vice versa [12]. Experimental result obtained in [12] is that saturation point choice is 64 vectors in codebook. The *quantization distortion* (quality of quantization) is usually computed as the sum of squared distances between vector and its representative (centroid) [8]. The well-known distance measures are *Euclidean, city block distance*, *weighted Euclidean* and *Mahalanobis* [3,17]. They are represented in the following equations:

$d_C(x,y) = \sum_{i=1}^{N} \lvert x_i - y_i \rvert$	City block distance	**(4.1)**
$d_E(x,y) = (x-y)^T \cdot (x-y) = \sum_{i=1}^{N} (x_i - y_i)^2$	Euclidean distance	
$d_M(x,y) = (x-y)^T \cdot D^{-1} \cdot (x-y)$	Weighted Euclidean distance	

where *x* and *y* are multi-dimensional feature vectors and *D* is a weighting matrix [3,17]. When *D* is a covariance matrix weighted Euclidean distance also called *Mahalanobis distance* [3,17]. A set of observation was made in [17] concerning the choice of distance for speaker identification task. Their conclusion is that weighted Euclidean distance

where *D* is a diagonal matrix and consists of diagonal elements of covariance matrix is more appropriate, in a sense that it provides more accurate identification result. The reason for such result is that because of their nature not all components in feature vectors are equally important [4] and weighted distance might give more precise result.

During the matching a matching score is computed between extracted feature vectors and every speaker codebook enrolled in the system. Commonly it is done as a partitioning extracted feature vectors, using centroids from speaker codebook, and calculating matching score as a quantization distortion. Another choice for matching score is *mean squared error (MSE)*, which is computed as the sum of the squared distances between the vector and nearest centroid divided by number of vectors extracted from the speech sample. MSE formula is represented in the following:

$$MSE(X,C) = \frac{1}{N}\sum_{i=1}^{N}\min_{j}(d(x_i, c_j))^2 \qquad \textbf{(4.2)}$$

where *X* is a set of *N* extracted feature vectors, *C* is a speaker codebook, *xi* are feature vectors, *ci* are codebook centroids and *d* is any of distance functions. However, these methods are not adapted to the speaker identification. More realistic approaches are proposed in [13], which are based on the assigning of weights to the code vectors according to their discrimination power or the correlations between speaker models in the database.

4.4 Decision

The next step after computing of matching scores for every speaker model enrolled in the system is the process of assigning the exact classification mark for the input speech. This process depends on the selected matching and modeling algorithms. In template matching, decision is based on the computed distances, whereas in stochastic matching it is based on the computed probabilities. This process is represented in Figure 4.3.

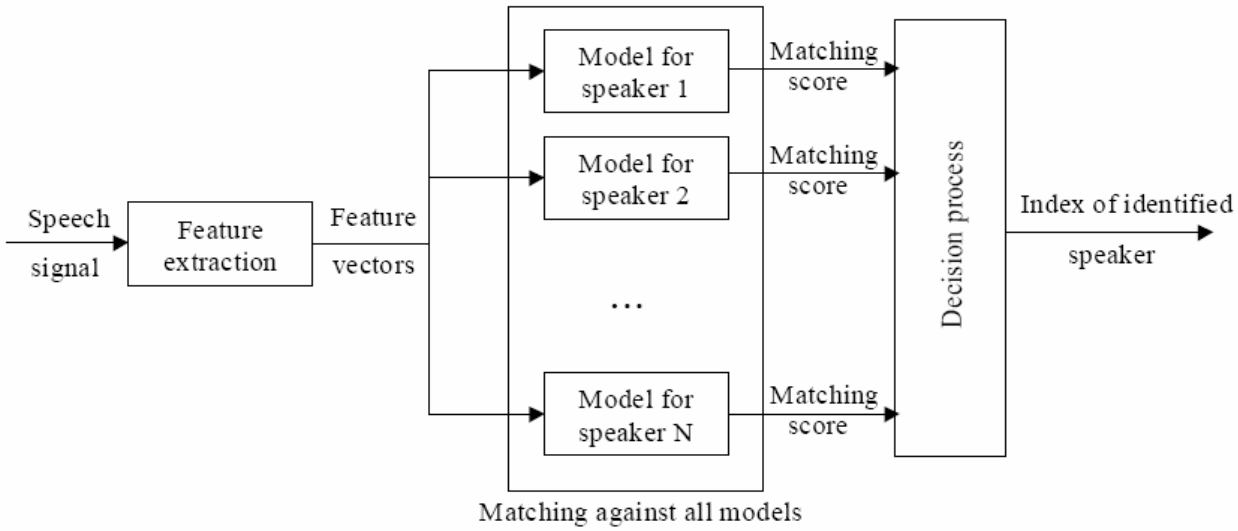


Figure 4.3 Decision process

Practically, decision process is not so simple and for example for so called open-set identification problem the answer might be that input speech signal does not belong to any of the enrolled speaker models. More details about decision process can be found in [3,10].

4.5 Alternatives and Conclusions

The issues described in this chapter actually fall into the more general topic, namely *pattern recognition*, which aims to classify object of interest into one of a number of

classes [27]. Therefore, the methods applicable for pattern recognition are applicable for speaker identification as well. Nearest Neighbor and VQ are the most well studied techniques for speaker verification. Both of these methods aim to produce reasonable model for high accuracy verification. However, VQ works mostly as a quantifier rather than modeler and therefore, in practice it produces reduced number of feature vectors rather than speaker model [6].

4.7 Remarks

In chapters 2,4,4 we were discussing about general techniques used in speaker verification area. These methods serve as a basis for future investigations and ideas behind them still lead researchers to the new discoveries. Nowadays it is obvious that it is possible to recognize speakers from their voices using computers, at least under laboratory environments and within small speaker populations. Nowadays research in speaker verification area is mostly concentrated on the developing fast and robust algorithms, which can work in difficult, from the identification task point of view, conditions, such as in noise or using poor environments. The motivation for future work is driven by practical and economical applications of automatic speaker recognition. In the next chapters we judge these basic techniques from the real-time speaker identification task point of view and also propose few solutions for this kind of identification problems.

Chapter 5

Approaches

In this Chapter we shall see in brief the approaches that are employed in the process of speaker verification.

5.1 Polychotomy

Consider a multiple class problem with a small number of classes where one can observe many instances of each class. This is an easy and a valid procedure, but is limited to classes that have substantial number of instances available. However, without knowing the geometrical distribution of the unseen classes (populations), the true error of the entire population (universe) cannot be drawn from the error estimate of the sample population. Hence this approach remains statistically non-inferential. Details of this approach are found in [14]

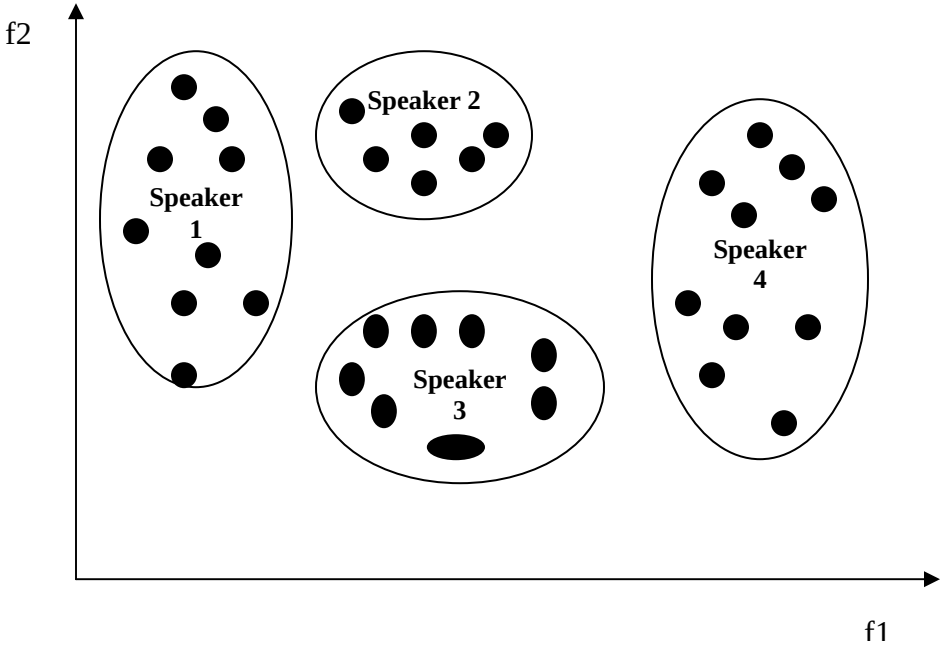


Figure 5.1 Polychotmy, Multiple class Problem. Statistically non-inferential

5.2 Dichotomy

Consider a many class problem where the number of classes is too large to be observed. The classification technique as mentioned in the previous paragraph cannot be applied to establish individuality because the number of classes is too large or unspecified. Many pattern identification problems especially in forensic sciences for establishing individuality fall under this category of many class problems.

The Identification Model is claimed to be not statistically inferable for a many class problem. In a many class problem, a population is all the biometric data samples of each person and is a very large or unspecified number. Samples from every single individual must be observed so that a conclusion could be drawn. This is a tedious and usually an impossible task. To draw statistical inference, the knowledge of the geometry of the

unseen classes is a basic requirement. Since there are unseen classes, the error estimate of a sample population cannot infer the true error estimate of the entire population.

The alternative approach to be taken is that of transforming the many class problem, into a dichotomy by taking the "distance" two samples of the same class and those of two different classes [4]. This model allows inferential classification although there is no requirement for all the classes to be observed. In this model, two patterns are categorized into only one of the two classes; they either belongs to the same class or are from two different classes.

Given two biometric data samples, the distance between the two samples is computed first. This distance value is used as data to be classified as positive or negative. Positive value of distance is intra-variation, within a person or identity and negative value is inter-variation, between different people or non-identity.

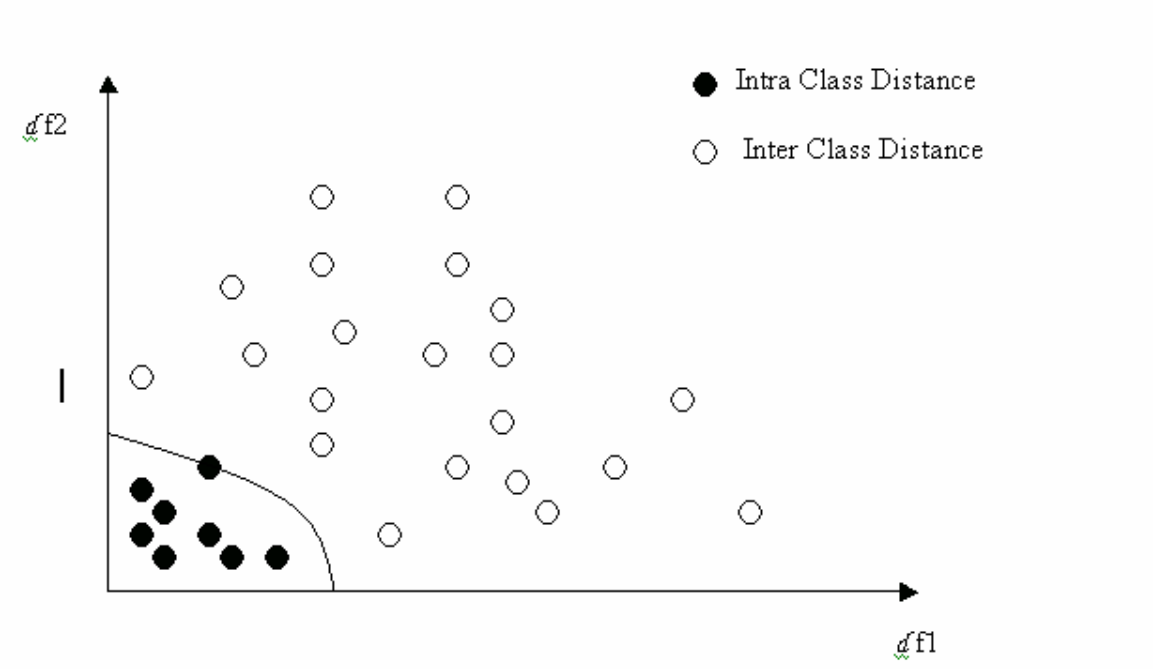


Figure 5.2 Dichotomy for a particular Speaker X

Details of this approach are found in [26,18,4]

Chapter 6

Experiment

In this Chapter, we discuss in detail the experiment that we performed using the Dichotomy Approach as discussed in [5.2]. The Experiment involved collecting voice data from subjects, Segmentation of collected data, Visual representation of collected data, Feature Extraction, Nearest Neighbor Experiment results, Artificial Neural Network Experiment Results.

6.1 Data Collection

Speech samples were collected from 10 subjects. Each subject was asked to repeat the utterance "MY NAME IS …" 10 times normally and 5 times in a disguised manner. In total there are therefore 100 samples of normal speech and 50 samples of disguised speech. The speech samples were collected over a standard microphone (*Cyber Acoustics OEM AC-200 Stereo Speech Headset and Microphone*) attached to a PC (*Dell Dimension™ 2400, with a Pentium IV Processor and 256MB Ram*) running the *Windows XP* Operating System. The software used to collect the speech is "*Sound Recorder*" (*Microsoft Sound Recorder*), which comes as the part of the aforesaid Operating System.

A database of the speakers and the speech samples was implemented in *MySQL*- an open source Relational Database Managements System. The database included two tables, one for holding information about the speaker and one for holding information about the sample provided. The Entity Relationship Diagram is as shown in Figure 6.1.

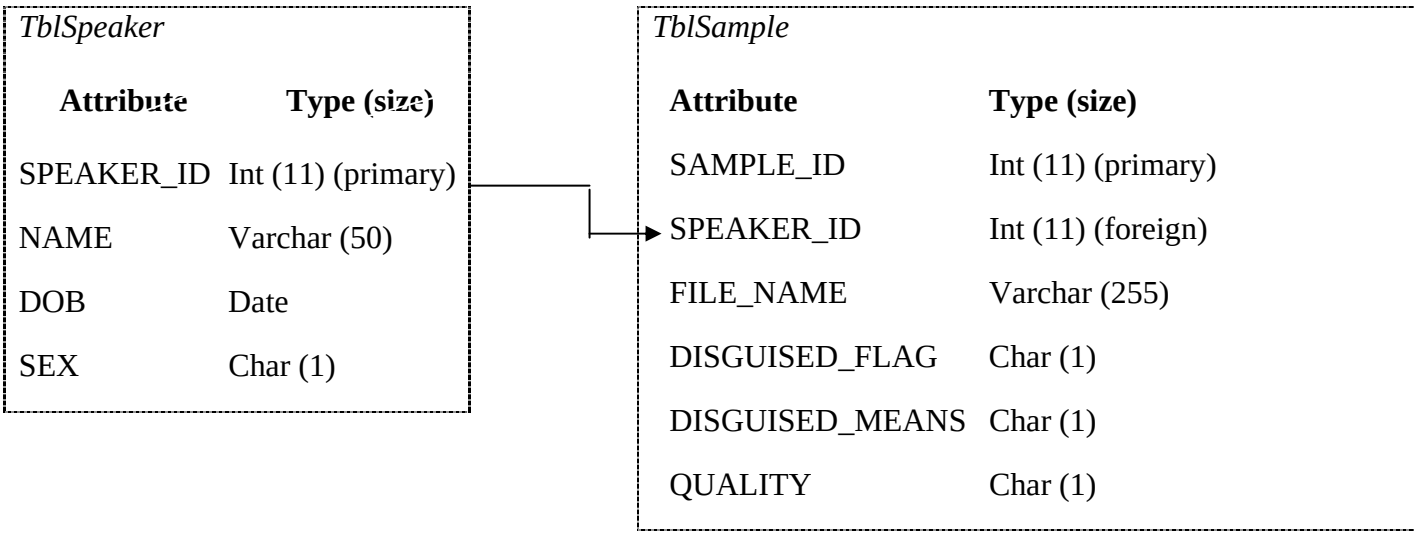


Figure 6.1 ERD of the Speech Data

where, in the tblSample, FILE_NAME attribute contains the entire path and the name of the sample wave file, DISGUISED_FLAG attribute when set means that the sample is a disguised sample and the DISGUISED_MEANS attribute gives information about the manner in which the speaker tried to disguise the voice.

The speakers used one of the following standard means to disguise their samples.

- Increase in Pitch
- Decrease in Pitch
- Talk at a different speed
- Spoke far away from the microphone
- Induced an accent in the sample

6.2 Segmentation

The segmentation problem was to isolate that part of the speech utterance, which is common to all of the collected samples. The common portion was from the beginning of the utterance, the start of the [9] sound of “My”, to the end of the high-frequency [28] sound in the word “is” before the person’s name. Hence, the segmented part of the speech consisted of just the phrase “My name is” which was common to all of the speech samples collected.

6.2.1 Tools used for Segmentation

The tool used was *Free Wave Editor* (Editionv3.0, Code-it Software), a freeware application downloaded from the *internet* [29]. The entire package was downloaded as a zip file, unzipped and installed on a PC. The way to launch the Wave Editor Application was to open Free Wave Editor, the executable program. The advantage of using this application was that one could view the waveform in both the time, as *Time Waveform*, as well as in the frequency domain as *Spectrograph*. The Spectrograph provides a much better view for manual segmentation of the waveform because it clearly shows the different bands that indicate the start of the utterance as well as the high frequency [28] sound produced by “is” in the input sample sentence “*My name is*”. The Application window is shown in Figure 6.2.

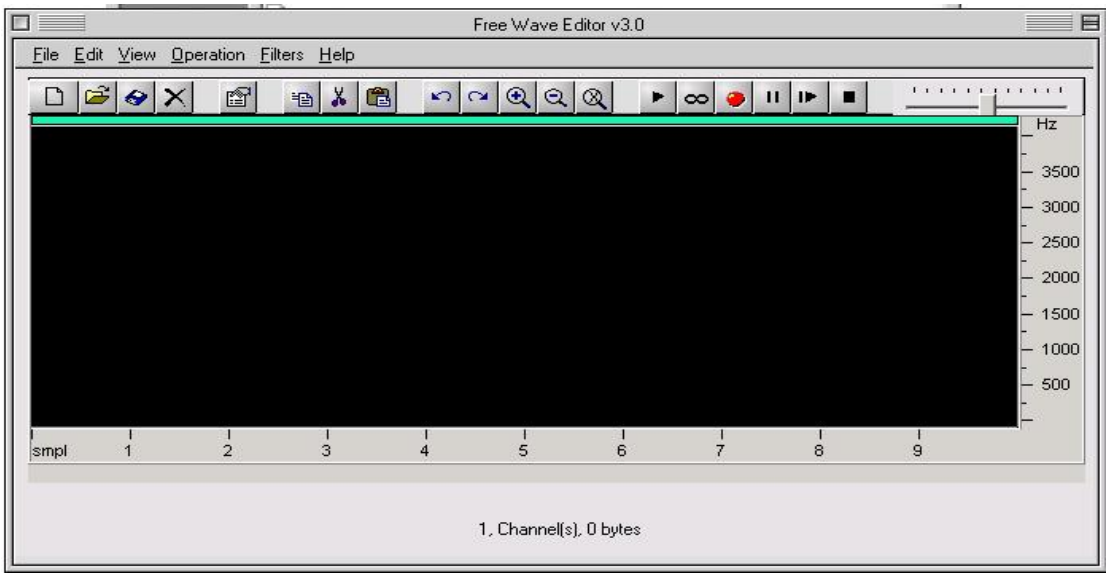


Figure 6.2 Free Wave Editor Window

The application has an open command that opens a dialogue box to take the input .wav file for segmentation (Figure 6.2) An example loaded .wav file, in the Time domain is as shown in Figure 6.3.

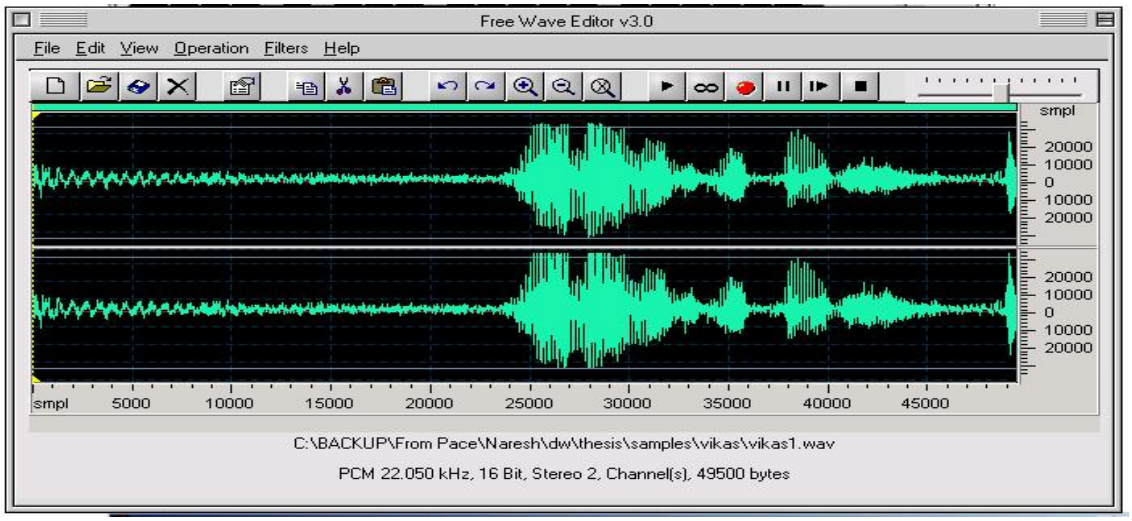


Figure 6.3. Free Wave Editor with a loaded waveform in time domain

The Spectrograph view of the same loaded file can be obtained by changing the view settings. The spectrographic view is shown in Fig. 6.4

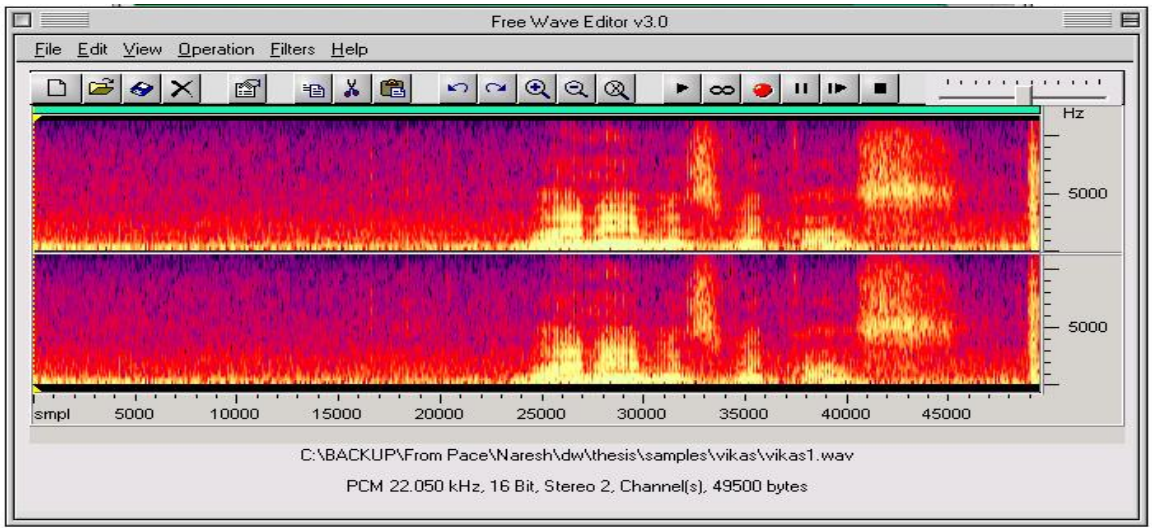


Figure 6.4. Free Wave Editor showing the Waveform in Frequency Domain (Spectrograph)

Segmentation takes place by left clicking at the start of the phrase to get a dotted yellow line and right clicking at the end of the word “is” to get a shaded blue area between the lines (Figure 6.5). These lines can be adjusted after playing the selected portion to get the required segmentation before saving the selected part as a separate .wav file.

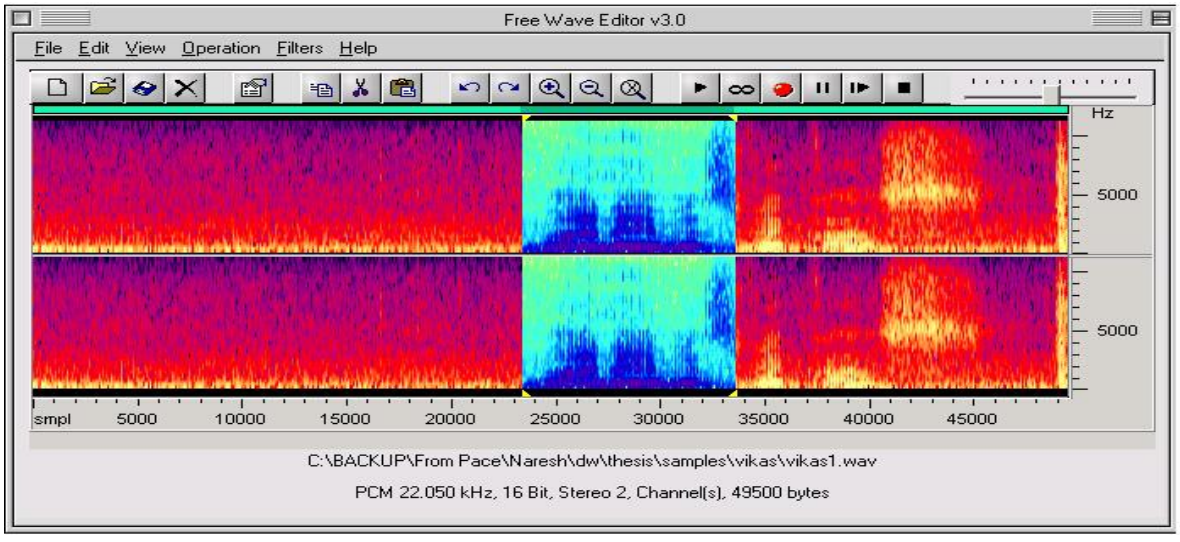


Figure 6.5 Free Wave Editor showing the Segmented Portion of the Waveform

The front and the back boundaries can be adjusted by listening to that part of the waveform by using the play button. Once the segmentation is completed, the segmented part of the .wav file can be saved as shown in Figure 6.6.

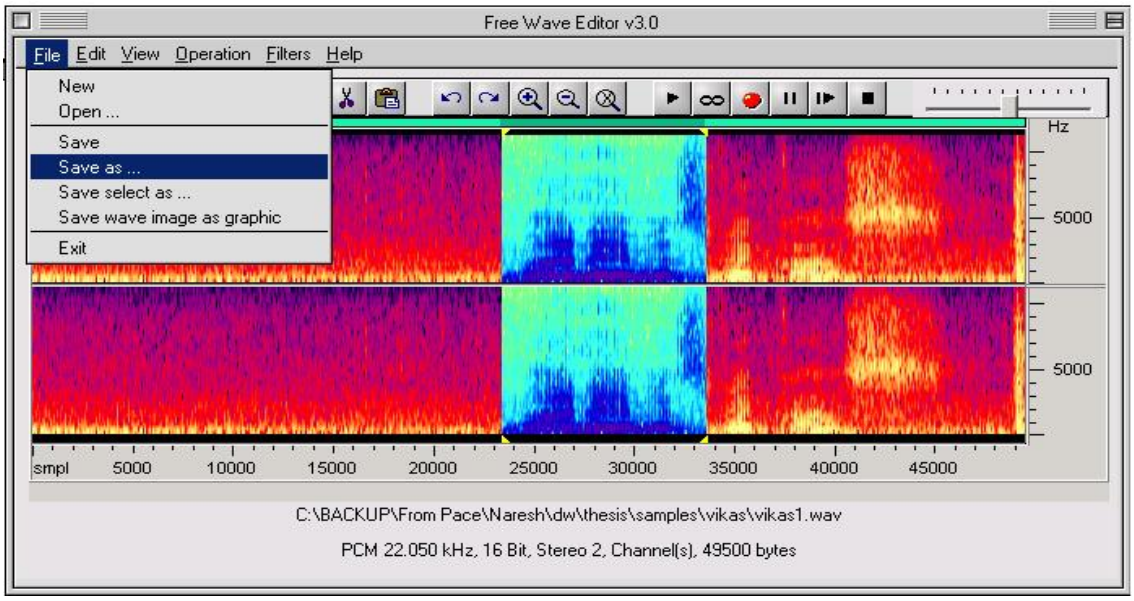
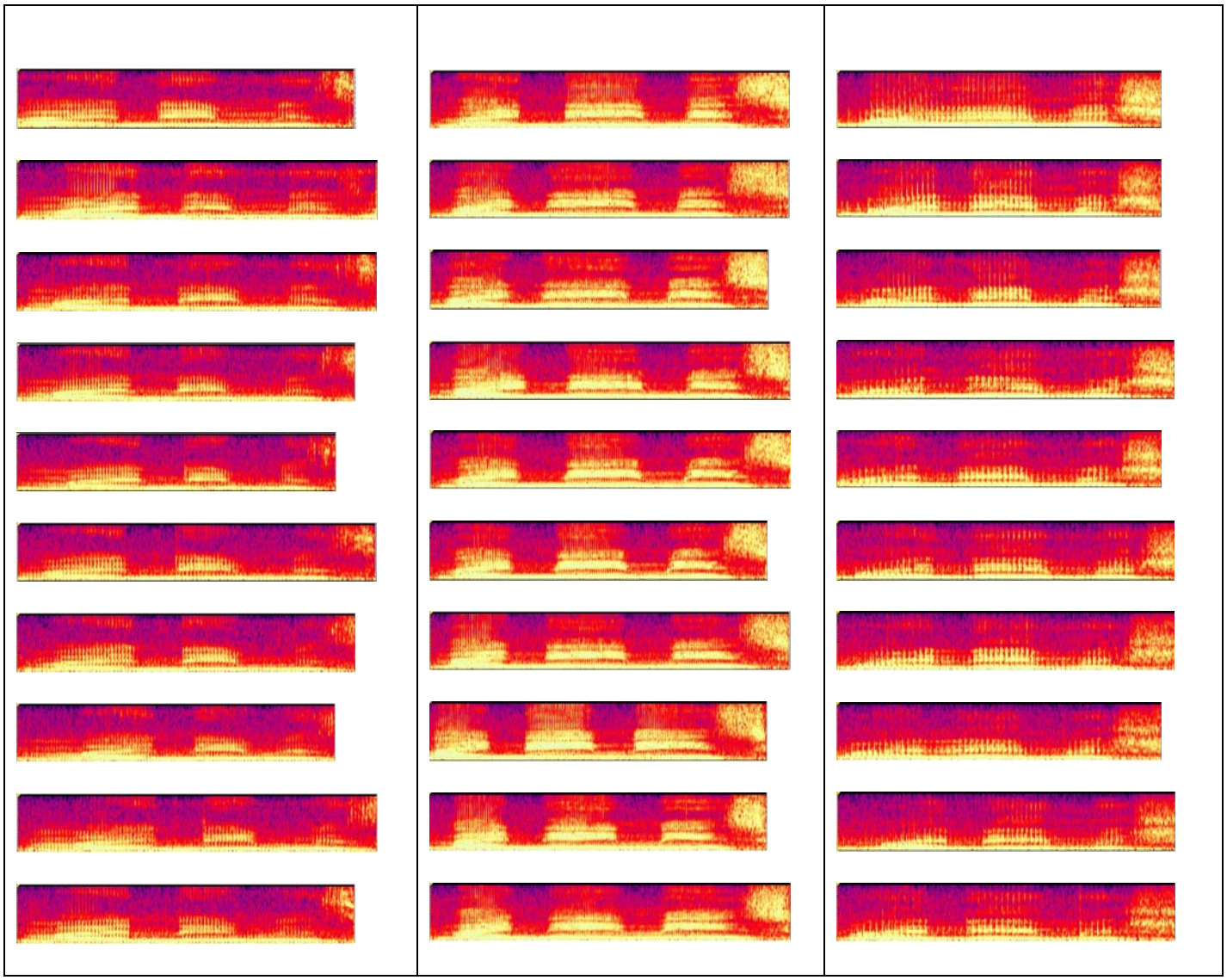


Figure 6.6. Free Wave Editor saving the Segmented waveform as a separate file

Now the selected region is saved as a new file and the process of segmentation is complete. The new file is saved in the appropriate place and its location is changed in the database. This file has the required part of the wave sample, i.e. “My name is”, which is common to all the samples. These Segmented new files are used as input for Feature Extraction.

6.3 Visualization of Spectrographs

The Spectrographs provide a very nice visualization of the audio data. Visualization is important as the human eye has a much higher recognition of pattern. Also the Spectrographs can be easily printed on a sheet of paper and easily viewed. The Figure 6.7 shows the 10 audio samples, collected from 3 different subjects.



Samples of Speaker 1 (Female)	Sample of Speaker 2 (Female)	Sample of Speaker 3 (Male)

Figure 6.7. Spectrographs of the samples collected from three different speakers

From the visualization, it is clearly visible that the Female Subjects have higher pitch as against the male subject.

6.4 Variable length of the audio data.

One of the key issues that exist in any speaker related experiment is, there is a clearly marked difference in the time taken for the utterance of a sentence, even when repeated the same subject. Figure 6.8 shows two samples of same utterance (“MY NAME IS”) collected from the same subject and the different time taken by two samples.

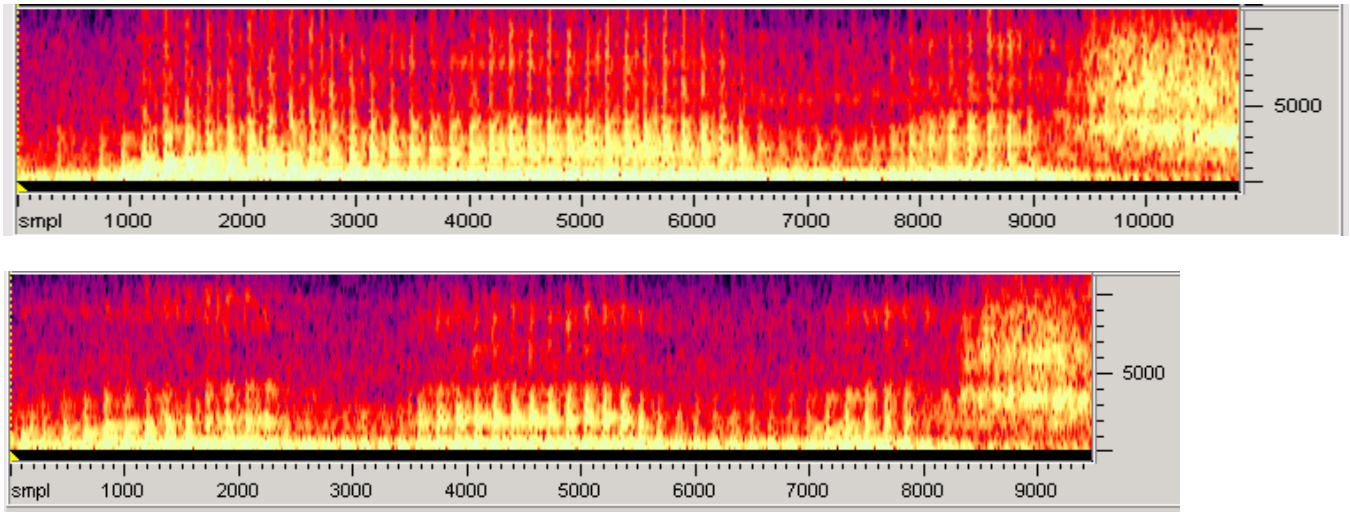


Figure 6.8 Two samples of same speaker for same utterance taking different time

6.5 Normalization

We performed two different normalizations to compensate for the variable length of the speech data. The first was to take the Means and Variances along the entire x-axis to arrive at a fixed number of feature points in the feature extraction. The second was to groups the similar utterances of phonemes into 7 groups. The input utterance (“My Name is ”) was divided to seven phonemes that form the utterance. The division is tabulated below in Figure 6.9. Figure 6.10 shows the Spectrograph divided to 7 parts according to the utterance of the Phonemes.

MY		NAME			IS	
m	ai	n	ae	m	i	z

Figure 6.9 Table showing the Phonemes in the utterance

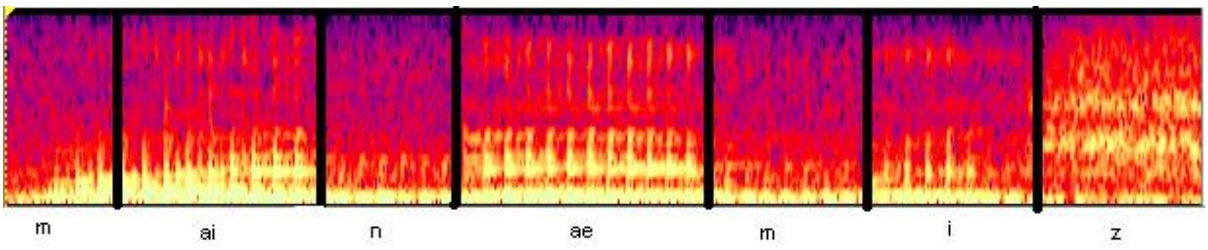


Figure 6.10 Spectrograph broken into 7 parts based on Phonemes

6.6 Feature Extraction

In our experiments, we used feature vectors composed from 12 lowest mel-frequency cepstral coefficients (MFCC) computed using 40 mel-spaced filters. 13 of the filters were spaced linearly at 133.33 Hz between central frequencies and 27 filters placed logarithmically, separated by a factor of 1.0711703 in frequency. The 0-th coefficient was excluded, because it carries a little of speaker specific information. Analysis frame was windowed by 30 milliseconds Hamming window with 10 milliseconds overlapping. The signal was pre-emphasized by the filter $H(z)=1-0.97 \cdot z^{-1}$ and silence frame were removed before the feature extraction. All sample durations in these experiments refer to the silence-removed speech. Figure 6.11 shows the filter placements.

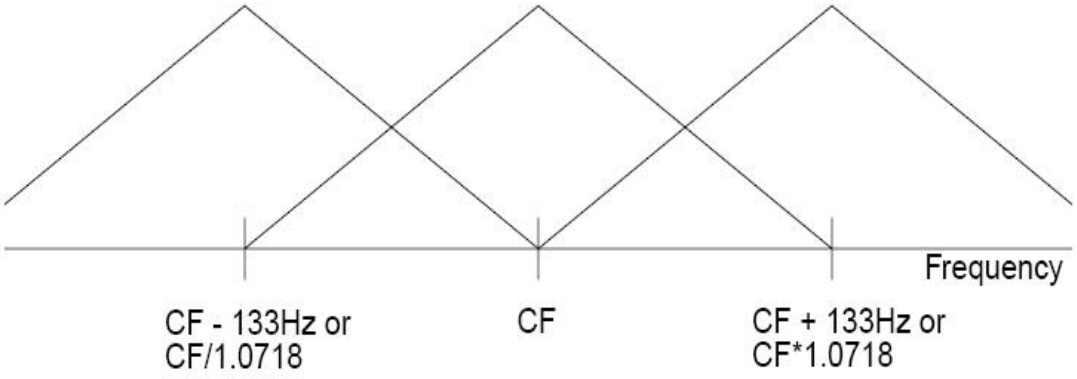


Figure 6.11 Filter placements

The following figure 6.12 shows the frequency response of the forty filters.

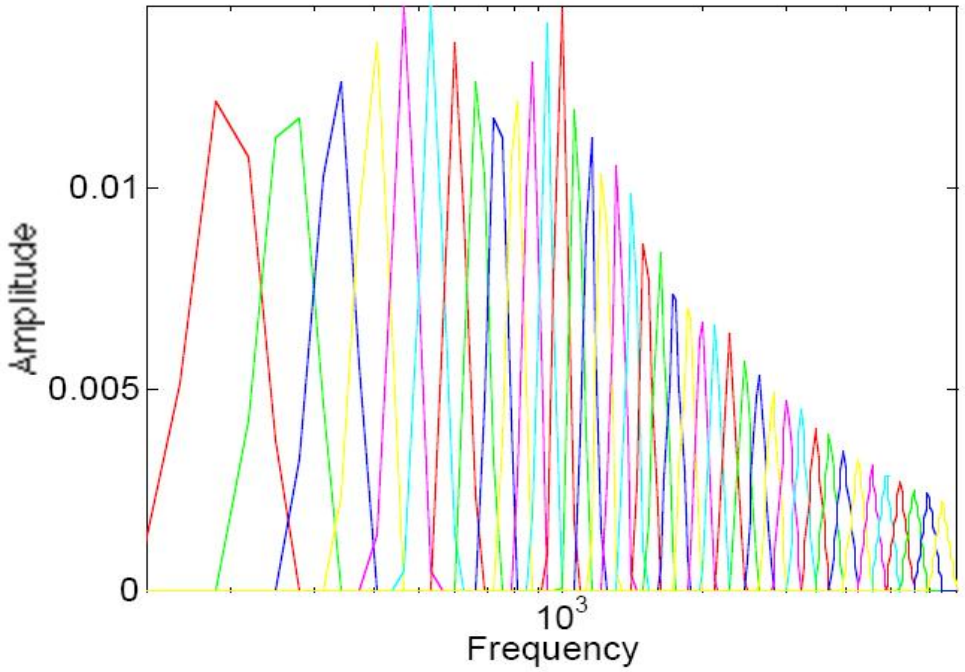


Figure 6.12 Frequency Response of the forty filters used in the experiment.

The features were extracted using the Speech Processing Toolbox written in Matlab for .wav files.[2]

6.7 Feature Vector Analysis

There were fixed number of frequency bands, 13 of them representing the 13 Cepstral Coefficients obtained from MFCC toolbox. The number of time bands were varying, since the hamming window was of fixed size and due to the varying lengths of the voice samples, the time bands were varying for each sample.

6.7.1 Normalization

Two different normalizations were performed on the obtained features and the results of both have been reported.

6.7.1.1 Normalization, 26 features

The entire wave file was normalized by time domain by taking the means and variances along each of the frequency bands. Since there were 13 frequency bands, this normalization resulted in 13 means and 13 variances per sample, leading to 26 features in all per sample.

6.7.1.2 Normalization, 91 Features

Each input wave file was divided into 7 parts as shown in Figure 6.10 and 13 bands were obtained for each of these 7 parts. Taking only the means across each of the 13 bands for all 7 parts results in 91 features per sample as shown in the equation 6.1

$$13 \text{ bands} * 7 \text{ parts} = 91 \text{ Features} \quad \textbf{(6.1)}$$

6.7.1.3 Normalization, 84 Features

The 1st Cepstra is removed from the analysis as it contains a powerful Energy Component. This results in 12 frequency bands and 7 parts according to phonemes for each file amounting to 84 features as shown in the equation 6.2

$$12 \text{ bands} * 7 \text{ parts} = 84 \text{ Features} \quad \textbf{(6.2)}$$

6.7.2 Distance Computation

Each of the 10 subjects in consideration gave 10 voice samples each. All of these 100 samples have been run through segmentation and MFCC extraction phase.

6.7.2.1 Nearest Neighbor

The 10 samples from each speaker were divided into 2 parts of 5 each. One part was used as training set to calculate the mean Eucledian distance for the subject. The other part of 5 was used to test the closeness of the Eucledian Distance of each sample to the mean. This experiment was performed on all of the normalizations mentioned in [5.7.1.1, 6.7.1.2 and 6.7.1.3]

6.7.2.3 Dichotmizer

Artificial Neural Network was used because it is equivalent to multivariate statistical Analysis. Samples of both classes are divided into groups of 225 in size. One pair set is used as training set and the other set is used as a validation set and the third is used as testing set.

Using the feature distance values, the Artificial Neural Network is trained using a *back-propagation algorithm*. All the MFCC features are generated from the voice samples as discussed earlier and the distance values between two voice samples *x* and *y* for each feature are fed into the Artificial Neural Network.

The Equation 6.4 shows the computes the distance obtained. The figure 6.13 shows the Artificial Neural Network used.

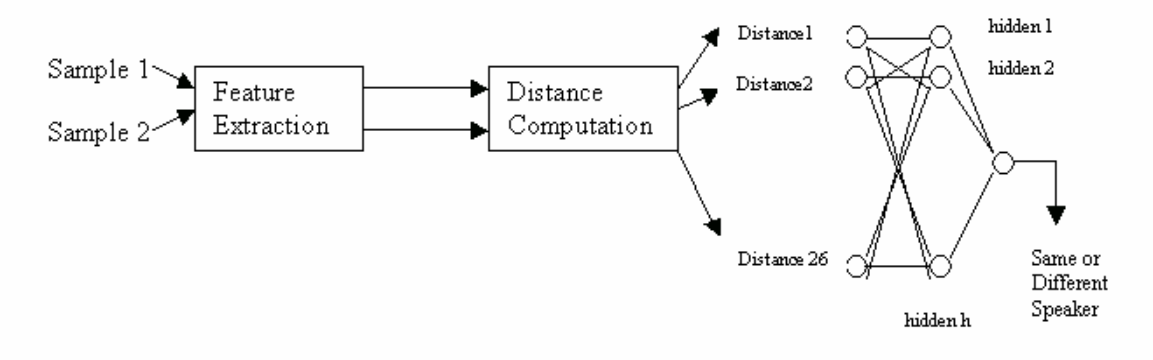


Figure 6.13 Artificial Neural Network used for 26 features

The absolute distance between features of a sample with all the other samples of the same subject (intra-class distance) and with all the other samples of other subjects (inter-class distance) was calculated

$$(10*9)/2 \quad = 45 \text{ intra class distance per subject}$$
$$45 * 10 = 450 \text{ intra class distances for 10 subjects} \qquad \textbf{(6.2)}$$

A total of 450 intra-class distances and 4500 inter-class distances were obtained. These were divided into groups of 225 each. So there were 2 groups of 225 intra-class distances, one for training and one for testing purposes and 20 groups of 225 inter-class distances.

6.8 Results

Two basics experiments (Nearest neighbor and Dichotmizer) were performed on different normalizations. The table 6.1 shows the results of Nearest Neighbor Experiment

Normalized Features	Error	Accuracy
26 Features	32%	68%
91 Features	25%	75%
84 Features	17%	83%

Table 6.1 Results of Nearest neighbor Experiment

Table 6.2 shows the results of the Dichotomizer ANN Experiment results

Normalized Features	Hidden Units	Type I Error	Type II Error	Accuracy
26 Features	10	3%	20%	77%
91 Features	32	0%	11%	89%
84 Features	28	0%	6%	94%

Table 6.2 Results of Dichotomizer Experiment

Four samples were removed from the data due to quality issues and the 84 Feature experiment was conducted for Dichotmizer ANN and the results obtained as shown in table 6.3. The reasons for removal included poor recording quality of samples and unduly long duration of the sample over other samples collected from the same subject.

Normalized Features	Type I Error	Type II Error	Accuracy
84 Features	0%	2%	98%

Table 6.3 Results of Dichotomizer Experiment with four bad quality samples removed

6.9 Conclusion

A methodology for establishing the discriminative power of biometric with respect to speech data has been described. A multiple category problem is viewed as a two-category problem by defining the distance and taking those values as positive and negative data.

This paradigm shift from *polychotomizer* to *dichotomizer* makes individuality problem a simple one. An experiment to show the individuality of voice by collecting samples from people was performed. Given two randomly selected voice samples, we can determine whether the same person spoke the two samples or not. A measure of confidence is associated with individuality. Using 26 feature distance values, we trained an Artificial neural network and obtained 77% overall correctness. The overall correctness reached a 98% when used with 84 features and removing the input with bad quality data. Recruiting a large number of subjects representative of the population is crucial in order to infer the results to the entire population. Collecting multiple biometric data from each subject is necessary to obtain the intra person distance data. Various signal-processing techniques are used to extract features from a given modality. Depending on feature measurement type, suitable distance measures are used to transform the feature space into feature distance space.

Chapter 7

This chapter consists of future work to be done and the list of references

Future Work

The results obtained from this experiment can be improved by performing various different measures on the extracted features. One of the different ways to perform a different distance measure is to divide the spectrograph into a fixed number of columns in the time domain and calculating the average of those features to obtain distance measures. The other distant measures could be to programmatically calculate the average of features according to the occurrence of the standard syllables, 9 that are present in the collected sample and run the training and testing procedures again.

We have also recorded 5 disguised samples from each subject and along with the means each subject used for each of the disguised sample. Experiments could be conducted to verify the speaker with the disguised as well as regular samples.

List of References

[1] B. S. Atal, "Automatic Recognition of Speakers from their Voices", *Proceedings of the IEEE*, vol 64, 1976, pp 460 – 475.

[2] M. Brookes, "Voicebox: Speech Processing Toolbox for Matlab", Imperial College, London, http://www.ee.ic.ac.uk. (last visited 04/2003)

[3] J.P. Campbell, "Speaker Recognition: A Tutorial", *Proc. of the IEEE*, vol. 85, no. 9, Sept 1997, pp. 1437-1462

[4] S. –H. Cha, "Establishing the Discriminative power of a Biometric, with Application to Handwriting Individuality", 2002.

[5] J. R. Deller, J. H. L. Hansen, J. G. Proakis, "Discrete-Time Processing of Speech Signals", Piscataway (N.J.), IEEE Press, 2000.

[6] Evgeny Karpov, University of Joensuu, “Real Time Speaker Identification”, 2003, Master’s Thesis,

[7] P. Fränti, T. Kaukoranta, O. Nevalainen, “On the Splitting Method for Vector Quantization Codebook Generation”, *Optical Engineering*, 36 (11), pp. 3043- 3051, November 1997.

[8] P. Fränti, J. Kivijärvi, “Randomized Local Search Algorithm for the Clustering Problem”, *Pattern Analysis and Applications*, 3 (4), 358-369, 2000.

[9] S. Furui, “Digital Speech Processing, Synthesis and Recognition”, New York, Marcel Dekker, 2001.

[10] H. Gish and M. Schmidt, “Text Independent Speaker Identification”, *IEEE Signal Processing Magazine*, Vol. 11, No. 4, 1994, pp. 18-32.

[11] X. Huang, A. Acero and H.-W. Hon, “Spoken language processing”, Upper Saddle River, New Jersey, Prentice Hall PTR, 2001.

[12] T. Kinnunen, T. Kilpeläinen, P. Fränti, “Comparison of Clustering Algorithms in Speaker Identification”, *Proc. IASTED Int. Conf. Signal Processing and Communications (SPC 2000)*, pp. 222-227, Marbella, Spain, 2000.

[13] T. Kinnunen, I. Kärkkäinen, “Class-Discriminative Weighted Distortion Measure for VQ-based Speaker Identification”, *Springer-Verlag Berlin Heidelberg 2002, Volume 2396, pp 681*-688.

[14] J.W. Koolwaaij, “Automatic Speaker Verification in Telephony: a probabilistic approach”, 2000, http://himalaya.lab.telin.nl/~koolwaaij/research/Pub/koolwaaij.2000.4.shtml (last visited 12/2003)

[15] Mahendran Sriganesh and Kandasamy sugumaran “Speaker Verification in Forensic Applications”, 1995

[16] J. M.Naik, “Speaker Verification: A Tutorial”, *IEEE Communications Magazine*, January 1990, pp.42-48.

[17] S. Ong, S. Sridharan, Cheng-Hong Yang, Miles Moody, “Comparison of Four Distance Measures for Long Time Text-Independent Speaker Identification”, *ISSPA*, 1996, pp. 369-372

[18] S. Pankanti, S. Prabhakar, and A. K. Jain, “On the individuality of finger prints,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1010-1025, 2002.

[19] J. G. Proakis and D. G. Manolakis, “Digital Signal Processing, Principles, Algorithms, and Applications”, New York, Macmillan Publishing Company, 1992.

[20] L. Rabiner and B.-H. Juang, “Fundamentals of Speech Recognition”, Englewood Cliffs (N.J.), Prentice Hall Signal Processing Series, 1993.

[21] D. A. Reynolds, “An Overview of Automatic Speaker Recognition Technology”, *ICASSP* 2002, pp 4072-4075.

[22] D. A. Reynolds, “Experimental Evaluation of Features for Robust Speaker Identification”, *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No 4, October 1994, pp. 639-643.

[23] L. Rigazio, P. Nguyen, D. Kryze, J.-C. Junqua, “Separating Speaker and Environment Variabilities for Improved Recognition in Non-Stationary Conditions”, *Eurospeech 2001 – Scandinavia*.

[24 S. W. Smith, “The scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Publishing, 1999

[25] F. K. Soong, A. E. Rosenberg, L. R. Rabiner and B. H. Juang, “A Vector Quantization Approach to the Speaker Recognition”, *AT&T Technical Journal*, Vol. 66, pp. 14-26, Mar/Apr 1987.

[26] S. N. Srihari, S. –H. Cha, H. Arora and S. Lee, “Individuality of handwriting,” *Journal of forensic sciences*, vol. 47, no. 4, pp. 856-872, 2002

[27] S. Theodoridis, K. Koutroumbas, “Pattern recognition”, San Diego, Academic Press, 1999

[28] R. Vergin, D. O’Shaughnessy, “Pre-Emphasis and Speech Recognition”, *Electrical and Computer Engineering*, 1995. Canadian Conference, Volume: 2, pp. 1062-1065.

[29] Wave Editor – an advanced WAV file editor and recorder. http://www.code-it.com/downloads.htm (last visited 11/2003)