AUXILARY RECOGNITION METHODS FOR HMM MODELLING UNDER INSUFFICIENT TRAINING DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY
BY
GÖKÇEN MALCI

93381

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR DEGREE OF
MASTER OF SCIENCE
IN THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2000

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur Öztürk

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Fatih Canatan

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Zafer Ünver

Supervisor

Examining Committee Members

Prof. Dr. Zafer Ünver

Prof. Dr. Mübeccel Demirekler M·Demirekler

Assoc. Prof. Dr. Tolga Çiloğlu

Assoc. Prof. Dr. Buyurman Baykal

Dr. Nedim Karaca

ABSTRACT

AUXILARY RECOGNITION METHODS FOR HMM MODELLING
UNDER INSUFFICIENT TRAINING DATA

Malcı, Gökçen
M.S., The Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Zafer Ünver
January 2000, 75 pages

The study in this thesis can be considered within the general framework of speaker dependent isolated word recognition using Hidden Markov Model (HMM). The HMM structure used is left-to-right and has five states. Each frame in the speech is represented by a feature vector of subband cepstral coefficients. The specific matter of concern is to improve the recognition performance when the training data is insufficient for obtaining reliable model parameters. Some auxiliary methods to be used in conjunction with HMM evaluations are proposed to achieve this purpose. As auxiliary methods; feature elimination, variance modification, incorporation of state duration probability, state distribution pattern with hybrid HMM-DTW (Dynamic Time Warping) and weighting the state probability contribution by genetic algorithm are used.

In feature elimination, those features that are identified as unimportant in discriminating different words are eliminated.

In variance modification, very small covariance values which are caused by insufficient training data are enlarged to prevent its destructive effect on the recognition performance.

In incorporating state duration probability, state duration associated with each state is modelled as a probability density function.

In the next method, state distribution pattern is combined with Dynamic Time Warping (DTW), another recognition procedure.

As the last method, probability contribution of each state is multiplied by a set of optimal coefficients to give relative importance to some states.

Keywords: Speaker dependent, isolated word recognition, HMM, subband cepstral, insufficient training data, feature elimination, state distribution.

ÖZ

EĞİTİM VERİLERİNİN YETERSİZLİĞİ DURUMUNDA SAKLI MARKOV MODELLEMEYE DAYANAN YARDIMCI TANIMA YÖNTEMLERİ

Malcı, Gökçen
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Zafer Ünver
Ocak 2000, 75 sayfa

Bu tezdeki çalışma, kişiye bağımlı Saklı Markov Model (SMM)'i kullanan yalıtılmış kelime tanıma işlevinin genel taslağı içinde ele alınabilir. Kullanılan SMM yapısı soldan sağa devinimli ve beş durumludur. Sesteki herbir çerçeve altband kepstral katsayılardan oluşan bir öznitelik vektörüyle eşlenir. Özel olarak ilgilenilen nokta ise, güvenilir model değişkenlerini elde etmek için kullanılan eğitim verisinin yetersiz olması durumunda tanıma başarısının artırılmasıdır. Bu amaca ulaşma doğrultusunda, SMM bulgularına eklenecek yardımcı tanıma metodları üzerinde durulmuştur. Yardımcı metodlar olarak; öznitelik azaltımı, varyans düzenlemesi, durum süresinin olasılık olarak katkısı, durum dağılım örüntüsünün karma SMM-DZS (Dinamik Zaman Katlanması) ile birleşimi ve durum olasılıklarının katkısının genetik algoritma ile ağırlıklandırılması kullanılmıştır.

Öznitelik azaltımında, kelimelerinin ayrımında önemsiz gorülen öznitelikler kaldırılır.

Varyans düzenlenmesinde, yetersiz eğitim datası sonucu oluşan çok küçük varyans değerlerinin sisteme olumsuz etkisini gidermek için geneişletilirler.

Durum zaman olasılığının katılmasında, her bir duruma ait durum zamanları bir olaslılık dağılım foknsiyonu ile modellenir.

Bir sonraki metodda, durum dağılım örgüsü bir başka tanıma yöntemi olan Dinamik Zaman Sarımı (DZS) ile birleştirilir.

En son ele alınan metodda, bazı durumları göreceli olarak önemlendirmek için, her bir durumun olasılığa katkısının genetik algoritmayla bulunan bir takım katsayılarla ağırlıklandırlması kullanıldı.

Anahtar kelimeler: Kişiye bağımlı, yalıtılmış kelime tanıma, SMM, altband kepstral, yetersiz eğitim datası, öznitelik azaltımı, durum dağılımı.

ACKNOWLEDGMENTS

I would like to thank my adviser, Prof. Dr. Zafer Ünver for his help. I would like to sincerely thank Prof. Dr. Mübeccel Demirekler and Assoc. Dr. Tolga Çiloğlu for their valuable insight and guidance throughout my thesis.

Thanks to all musicians making the world worth living.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Isolated word recognition is the task of correctly recognizing a word utterance belonging to a specific library.

1.1 Automatic Speech Recognition Techniques

Isolated word recognition and continuous speech recognition are the two main branches of automatic speech recognition. In isolated recognition, input to the system is a single speech unit; generally an isolated word. The aim is to find the best matching word in the vocabulary of the system to the input utterance. In continuous recognition, input is a sequence of words and the system utilizes its own vocabulary to form a sequence of words that best matches the input.

The subject of this thesis is isolated word recognition. Isolated word recognition systems can be classified according to the speaker dependency and the vocabulary size. In speaker dependent systems, the system has the best performance with the user who trains the system whereas speaker independent systems are optimized to have the similiar performance with different users. If the recognizer works on a few dozen of words, it is regarded as a small vocabulary system. A large vocabulary system works on several hundred or more speech units. In this thesis, the vocabulary consists of 60 words which can regarded as medium size.

Many techniques have been proposed for the solution of the automatic speech recognition problem. We can classify these techniques as [1]:

1. Model based or pattern recognition approaches,
2. Acoustic-phonetic approaches,
3. Knowledge based approaches.

In model based approaches, a model of some kind is used to represent each word. Usually these models or pattern structures are obtained by modelling human speech production. As in most pattern recognition systems, the method has two steps: training and testing. In the training part, the system is trained through different tokens of each word and a specific model or pattern structure is assigned to each word. In the testing part of the technique, a test word to be recognized is converted into the specific pattern structure and the best matching model is found. Model based approaches are the most commonly used ones in recognition due to the facts: they are simple to use, to understand and more robust and less sensitive to different speech vocabularies, users, feature sets and above all it has been proven to have high performance [2]. Dynamic Time Warping (DTW) and Hidden Markov Model (HMM) are the well-known approaches in this area and the latter is used in this study.

Acoustic-phonetic approach is based on the idea that there are finite, distinct phonetic units in the spoken language and they can be characterized from the speech signal properties [3]. Based on this idea, speech is first segmented into discrete regions with specific phonetic properties and one or two phonetic labels are assigned to each segment according to the signal properties of the segment. Later, a valid word where the string of these labels might have been produced is searched from the vocabulary and the best matching one is chosen as the recognized word.

Knowledge based approaches aim to imitate the high level reasoning process of human brain. They are the combination of the acoustic-phonetic approaches and the pattern comparison approaches. Every possible source of knowledge is brought together and the decision is made based on the rules specific to each knowledge source [4]:

- acoustic knowledge; obtained from spectral or signal properties of speech units,
- lexical knowledge; understanding the pattern of acoustic features where the word might have come from,
- syntactic knowledge; analyzing the grammatical correctness of the word combinations,
- semantic knowledge; analyzing the sense produced by the word combinations.
- pragmatic knowledge; understanding the meaning of the sentence in an exact way. With the cues obtained from the input, the decision is made based on the rules of each knowledge source. However, this approach has gained limited success due to the complexity of the system and the difficulty in modelling the human knowledge processing in the form of rules.

1.2 A Simplified Overview of the System and the Outline of the Thesis

In this thesis, HMM pattern recognition technique has been implemented. The overall process can be outlined as:

- Preprocessing and feature extraction,
- Training and modelling each word,
- Testing and recognizing the word.

A simplified block diagram is shown in Figure 1.1.

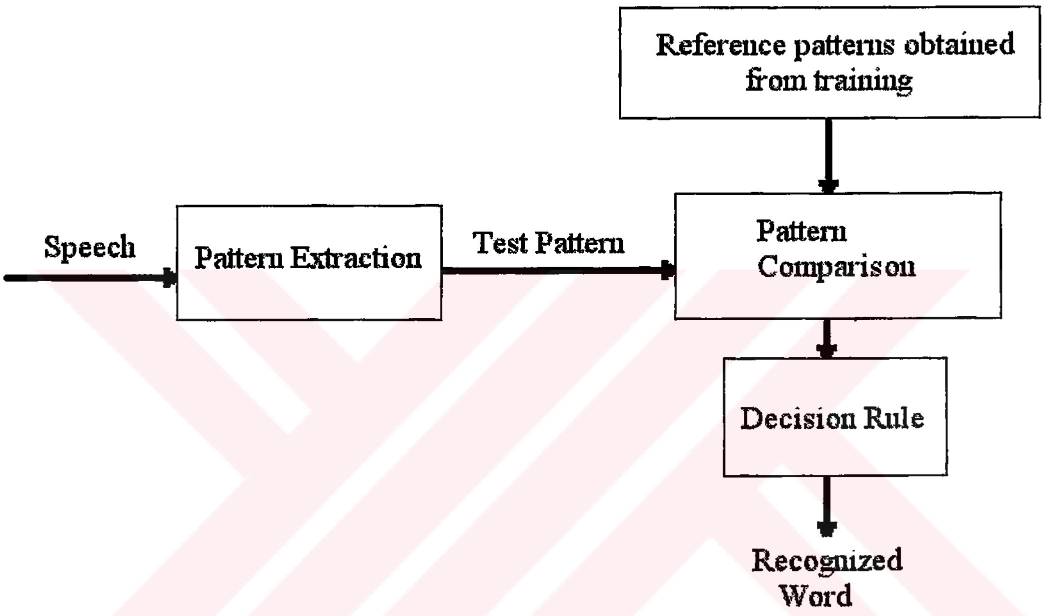

Figure 1.1 General diagram of the model based recognition system.

Speech data is assumed to be statistically stationary for 10-20 msec. Segments of this length are called *frames*. Speech can be thought of as the concatenation of these frames. In preprocessing, frames are spectrally shaped and then the spectral information of each frame is stored in a feature vector. Therefore, a speech pattern is obtained as a string of these feature vectors. The preprocessing stage and how these feature vectors are obtained will be discussed in detail in Chapter 2.

During training, model parameters of each word are obtained to ensure that the model is capable of representing the word for different utterances. This is achieved by utilizing a number of different utterances. In this way the variations pertaining to the different styles of saying the same word are taken into consideration. The way of obtaining the model parameters is discussed in detail in Chapter 3.

In the testing part, the word utterance to be recognized is preprocessed and a pattern (sequence of feature vectors) representing it is obtained. Later, this pattern is compared to those of the words in the vocabulary via HMM formalism and the most likely model comes out as the recognized word.

For the system implemented in this thesis, the training number is kept small to avoid annoying, long training sessions. However, this brings out the problem that model parameters obtained for each word are unreliable. This unreliability associated with model parameters affects the recognition performance of the system in a negative manner. Therefore, some auxiliary recognition methods must be used to increase the performance of the system. These methods and their results are explained in detail in Chapter 4.

Chapter 5 will give the conclusions, suggestions about the isolated word recognition.

CHAPTER 2

PREPROCESSING

Preprocessing and pattern extraction is the first stage in isolated word recognition (ISR) system. In this stage, speech signal, which is described by a large number of parameters, namely by its samples, is converted into a sequence of feature vectors with less number of parameters. This kind of representation is useful in the sense that it increases robustness against noise, variations in speech equipments and variations in speaker [5].

Preprocessing mainly consist of four operations, namely endpoint detection, spectral shaping, spectral analysis and parametric transform. A typical preprocessing system is shown in Figure 2.1.

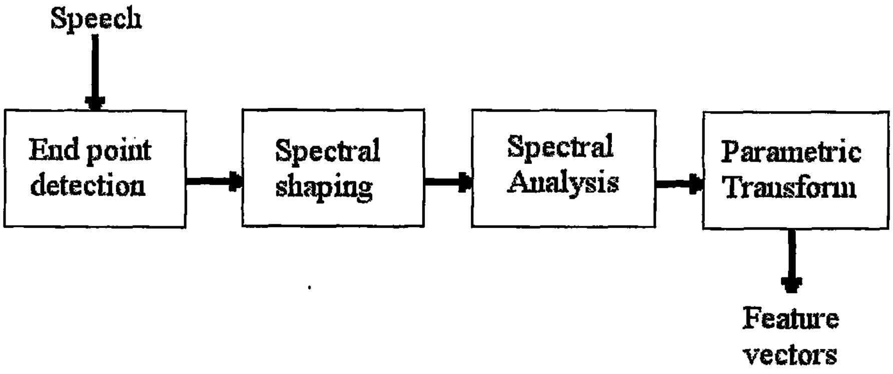

Figure 2.1 A typical preprocessing system.

2.1 Speech Production and Representation

Speech sounds are created by vibrations in human vocal tract. Figure 2.2 shows a typical speech waveform. Horizontal line in the center of the figure shows the resting atmospheric pressure. In the waveform, any reading below the line means that the pressure is lower than the resting atmospheric pressure at that time [6].

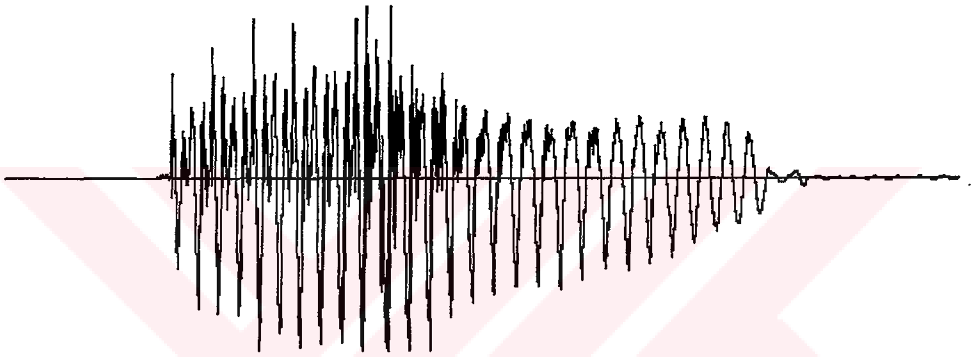

Figure 2.2 A typical speech waveform.

When someone speaks to microphone, these changes in pressure are converted to proportional variations in electric voltage. Then computers convert these analog variations to digital sound waveforms. Two operations are done through this process: sampling and quantization. In this thesis, 8 kHz sampling rate is used and quantization uses 16 bits giving 65536 quantization levels, which is commonly used in speech applications.

Speech sounds can be classified in different ways. One of them is through the state of the vocal cords. So, there are three states: *silence* meaning there is no speech; *voiced*, where vocal cords are vibrating and the speech produced is quasi periodic in nature; *unvoiced* where vocal cords are not vibrating and the speech produced is random in nature. This classification leads to a simplified mathematical model of the human speech production system, which consists of a time varying filter excited by a source, switching between quasi-periodic and noisy states as shown in Figure 2.3.

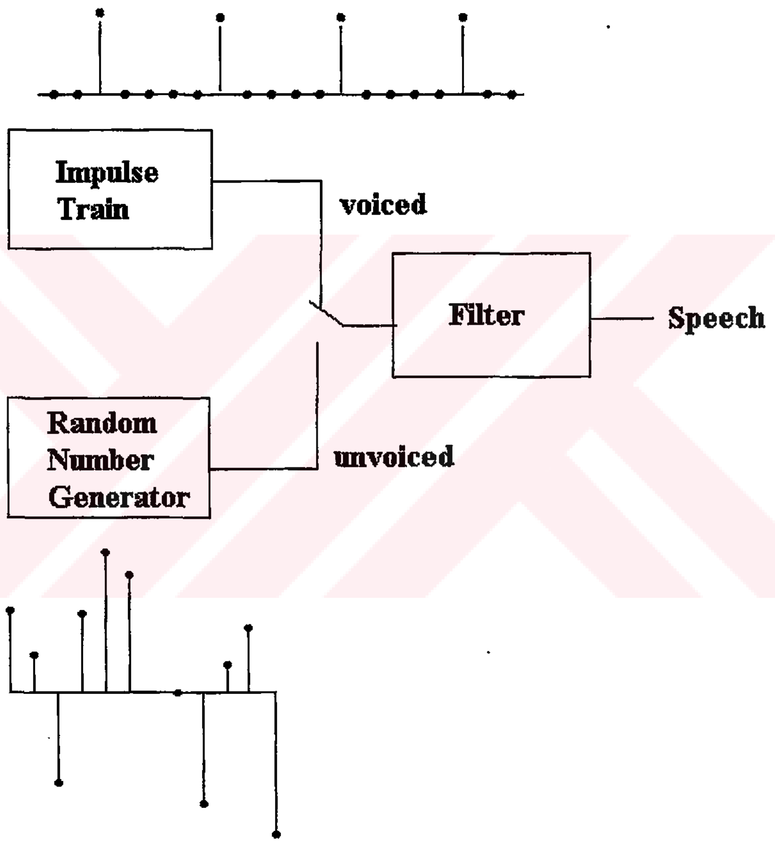

Figure 2.3 The model of human speech production system.

For voiced sounds, the source is a periodic impulse train with periods called the pitch period. For unvoiced sounds, the source is a random noise. One simple version of this model assumes that the filter is a linear, all pole, finite order system which gives rise to many preprocessing techniques [7].

Most of the time, we cannot segment the speech into silence, voiced and unvoiced regions clearly. Another way of representing the information associated with sounds is the speech spectrogram where the speech intensity in frequency bands over time is represented. Spectral analysis is performed on 10-15 msec sections of the speech waveform with advancing in 10 msec. This is in fact the two dimensional representation of the time-series display of the waveform where the gray level values represent the spectral intensity.

In Figure 2.4, a time-series display of the frequency domain function, i.e., absolute values of FFT's of the windowed frames are given. The associated spectrogram shown in Figure 2.5

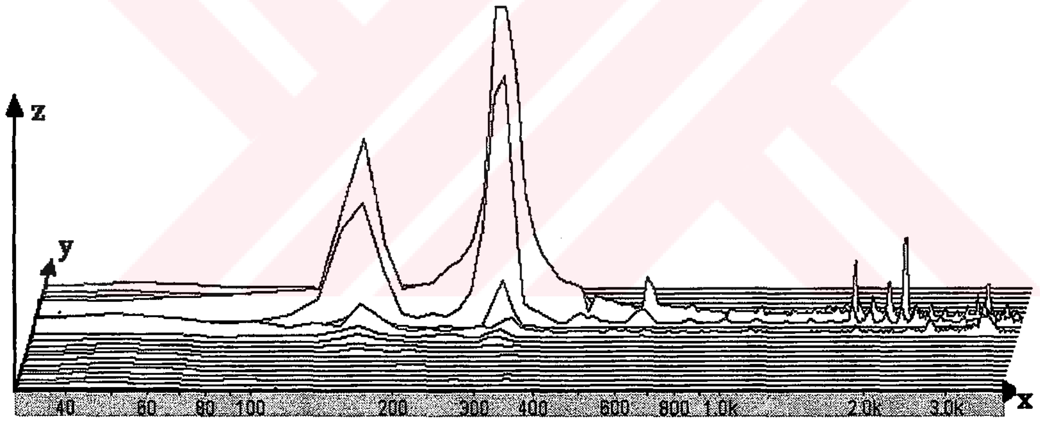

Figure 2.4 Time series display of magnitude spectrum.

In Figure 2.4, the x-axis shows frequency in logarithmic scale, the y-axis is the time running and the z-axis shows the magnitude of the Fourier transform of the window. The spectrogram is the projection of this figure into two dimensions with gray levels showing the magnitude.

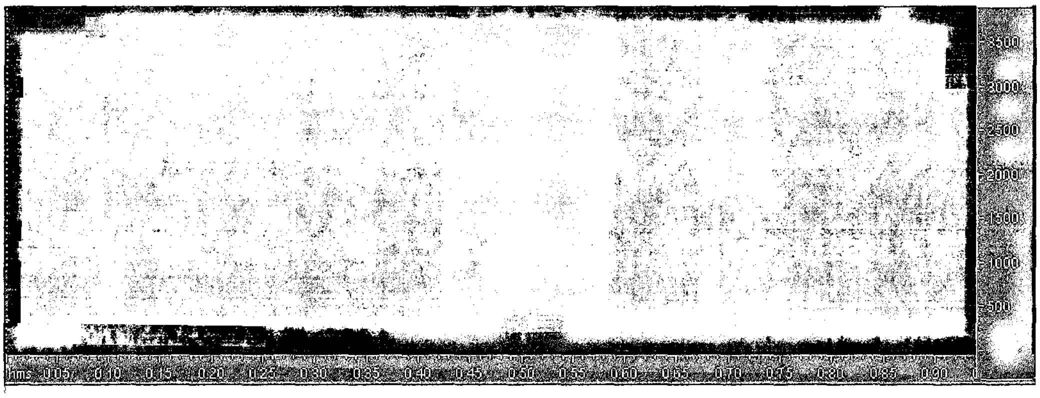

Figure 2.5 A typical spectrogram.

2.2 Endpoint Detection

The first step in preprocessing stage is the detection of the speech waveform from the background silence and locating the beginning and endpoints of the waveform. Fortunately, the recordings of the words are made in the laboratory environment where the background noise was an easy one to overcome. The problem of locating the beginning and end of a speech utterance in an acoustic background of silence is important in many areas of speech processing. In particular, the problem of word recognition is inherently based on the assumption that one can correctly locate the region of the speech utterance [8]. For the endpoint detection problem, four algorithms are used and the best working one is taken as the endpoint detector.

2.2.1 Energy and Zero Crossing Rate Algorithm

Zero crossing rate (ZCR) and energy of the speech signal are two most widely used measures for locating the endpoints of the utterance. The energy is the basic measure used to distinguish between speech or background silence. The ZCR provides a rough spectral measure and very roughly shows the major energy concentration regions in frequency and helps in discriminating voiced speech from unvoiced fricatives, stop bursts and silence [12].

The endpoint algorithm should be simple, efficient to process, reliable to locate significant acoustic events, and capable of being applied to varying background noises.

There are several problems associated with the endpoint detection task. These include: weak fricatives (/f, th, h/) at the beginning or end of the utterance, weak plosive bursts (/p, t, k/), final nasals, voiced fricatives at the end of the words which become devoiced, trailing off of certain voiced sounds [9]. The algorithm proposed below is simple and capable of catching most acoustical activities. The main outline of the algorithm is described below.

- First, the speech waveform is filtered with a highpass filter of cutoff frequency near 100 Hz to suppress 60 Hz hum.

- Second, the speech is divided into 10 msec duration of statistically stationary segments called **frames**. The frames are not obtained by simply dividing the speech into sharp distinct regions of 10 msec. As can be observed in Figure 2.6, for each frame, an overlapping smoothing window is used to corporate temporal correlation. In this thesis, the frame length T_f is selected as 10 msec and the window length T_w is 15 msec.

- Third, two parameters are computed for each windowed frame: the energy and the zero crossing rate of the frame. Let $x_i[.]$ be the windowed signal belonging to the *frame i*. Then, the speech energy belonging to *frame i* is defined as:

$$E(i) = \sum_{n=-\infty}^{n=\infty} \left| x_i^2[n] \right| \qquad (2.1)$$

Zero crossing rate is defined as the number of zero level crossings per frame.

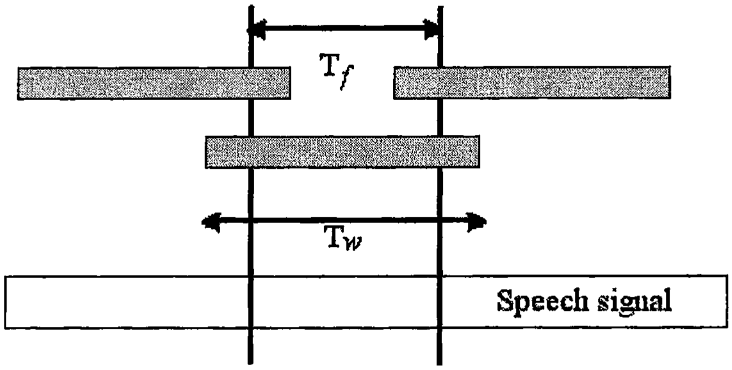

Figure 2.6 A frame and its associated window.

- Fourth; three threshold values, namely *energy lower threshold* (TL), *energy upper threshold* (TU) and *zerocrossing threshold* (ZCT) are found. A typical energy and zero crossing curves can be observed in Figure 2.7. To find the first two threshold values, the energy level function $E(i)$, is computed and then the minimum and the maximum of the energy function are found. TL and TU are computed according to the following rule [10]:

  $I1=0.03*(max-min)+min$

  $I2=4*min$

  $TL=min\ (I1,I2)$

  $TU=5*TL$

The ZCT is computed by finding the zero crossing rate of silence, i.e. the background noise: First 10 frames of the speech are assumed to be silence and the average zero crossing rate μzc and the standard deviation σzc are computed. Then below formula is used to compute ZCT:

$$ZCT = min\,(25,\ \mu zc + 2\sigma zc)$$

where 25 is a constant assumed to be the highest ZCT value for indication of speech.

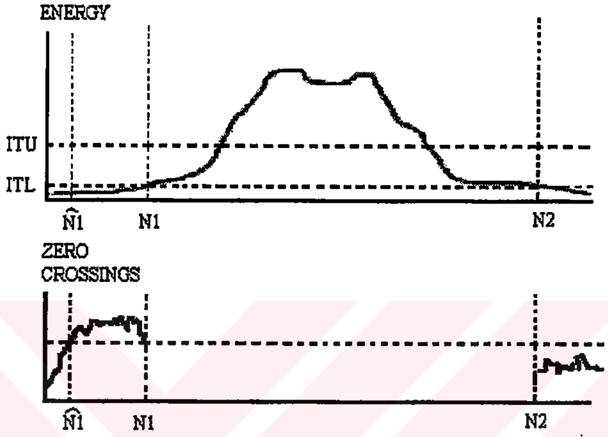

Figure 2.7 A typical energy and zero crossing level curves.

Zero crossing rate function, *ZC(i)*, shows the zero crossing rate of *frame i.*

- In the last step, the algorithm locates endpoints considering first, the energy function and then modifying them by zero crossing rate function. The algorithm first searches the point where the energy exceeds ITL and then exceeds ITU without falling below ITL. This point is *N1* as shown in Figure 2.7. In the same manner, the speech utterance is searched from backwards to locate the endpoint initial guess namely, *N2*. Then the interval from *N1* to *N1*-25 is searched; if the zero crossing threshold is exceeded by at least three consecutive points following *N1*, the beginning point is updated as this point. Such an update is done in Figure 2.7, and beginning point is shifted to $\hat{N}1$. If this is not the case, then the beginning point is set as N1. The same idea applies to the determination of the endpoint. This algorithm is simple and uses the energy and zero crossing rate measures for decision.

2.2.2 Energy Pulse Detection Method

Energy Pulse Detection algorithm is based on log energy measure assuming that the speech mainly lies on the high-energy region. The energy function has the property that during silence it fluctuates around 0-dB level, and during speech it is considerably larger [11]. Therefore, the energy alone can be used for the endpoint detection. Differing from the previous algorithm, the definition of energy is given as in Equation 2.2. Let $x_i[.]$ denote the speech signal of *frame i*. The energy of *frame i* is defined as:

$$E_i = 10\log_{10} \sum_{n=-\infty}^{n=\infty} \left(\left| x_i^2[n] \right| \right) - Q \qquad (2.2)$$

where Q is the term for the background noise characteristics. Q is found through the procedure given below:

First, the minimum energy frame is found as:

$$E_{\min} = \min\{E_i\} \qquad (2.3)$$

Then, a histogram is made using the energy values which are less then $E_{\min} + 10\,dB$. Three consecutive values of the histogram are averaged and Q is chosen as the energy level that corresponds to the peak of the histogram. After obtaining the modified energy function for each frame, the previous method of locating the beginning and endpoint of the utterance is used.

2.2.3 Teager's Energy Based Methods

The classical measures for energy definitions are either the summation of squared samples or summation of absolute valued samples which can be expressed as:

$$E_i = \frac{1}{W}\sum_{n=1}^{W} |x_i[n] \tag{2.4}$$

$$E_i = \left[\frac{1}{W}\sum_{n=1}^{W} x_i^2[n]\right]^{\frac{1}{2}} \tag{2.5}$$

where $x_i[n]$ is the n^{th} sample of the i^{th} frame.

In modelling speech production, Teager developed a new algorithm for computing the energy of the signal; this algorithm is presented as *Teager's Energy Algorithm* which is described as [13]:

$$E_i = \frac{1}{W}\sum_{n=1}^{W} x_i^2[n] - x_i[n+1]x_i[n-1]. \tag{2.6}$$

For a single tone, if $x[n] = A\cos(\Omega n + \Phi)$ where A is amplitude, Ω is the frequency of the oscillation, and Φ is the initial phase, this energy measure is capable of responding rapidly to changes both in A and Ω as can be seen in the formula below [14].

$$E = x^2[n] - x[n+1]x[n-1] = A^2\sin^2(\Omega) \approx A^2\Omega^2.$$

Note that the classical energy is proportional to A^2 only.

In our endpoint calculations, two forms of Teager's energy are used for each frame: one of them is obtained as above on sample based information called the *sample based Teager energy* as described by Equation 2.6, the other one is obtained through the insight gained by Teager energy. In the second method, first the power spectrum is calculated, then it is weighted by the square of the frequency, and finally the square root of the sum of the weighted spectrum is found as Teager energy. The formulation is given below.

Let $x_i[.]$ denote the i^{th} frame, and $X_i(k)$ denote the N point DFT of $x_i[n]$ at frequencies $\frac{2\pi k}{N}$. Then energy of the frame is defined as:

$$E_i = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}\left|X_i(k)\right|^2 k^2} \qquad (2.7)$$

This average energy is used to represent the energy of one frame called the *frame based Teager energy*. After representing each frame with either sample based or frame based Teager energy measure, the algorithm finds the upper and lower energy thresholds and detects the beginning and end of the utterance as described in the first method.

2.2.4 Discussion on Endpoint Algorithms

To test the performance of the endpoint algorithms, each one is used as an endpointer in HMM based word recognition system. HMMs will be explained later. The recognition results for one of the test datasets worked on are given in Table 2.1.

Based on the overall performance of the system, the *frame based teager energy method* based endpoint detector is used in implementing the isolated word recognition system. In the remaining part of this chapter, feature extraction techniques will be discussed.

Table 2-1 Performance of endpoint detection methods.

ENDPOINT DETECTION ALGORITHM	RECOGNITION PERCENTAGE
Energy and zero crossing rate method	%96.67
Energy Pulse Detection method	%96.67
Sample Based Teager Energy Method	%94.33
Frame Based Teager Energy Method	%98.8

2.3 Spectral Shaping

Before making any calculations on speech data, it is filtered through a preemphasis filter with transfer function given below.

$$H(z) = 1 - az^{-1} \tag{2.8}$$

where *a* is a number close to 1. In this thesis, *a* is taken as 0.9375. The Preemphasis filtering has two purposes.

First, it is used to suppress the effects due to the non-uniform frequency responses of the equipments used in A/D conversion and even in signal acquisition channels. Usually those frequency characteristics are approximated by lowpass filters, so to compensate their effect, a high pass nature preemphasis filter is used.

Second, preemphasis filtering flattens the signal spectrally and makes it less susceptible to finite precision effects later in signal processing [15].

2.4 Spectral Analysis of Speech Data

2.4.1 Windowing

The first step in spectral analysis procedure is to divide the speech into sequence of *frames*. As explained previously, the frame concept arises from the fact that the speech varies slowly in time. Therefore, the speech signal is assumed statistically stationary for 10-15 msecs duration segments. The windowing procedure with overlapping is used to corporate the temporal correlation between the frames. Usually, Hamming window is used for this purpose. The shape of the window and its duration affect the feature vector components. In this thesis, Hamming window of 1.5 times the frame length is used. The frame length is chosen as 10 msec which corresponds to 80 samples in 8kHz sampling rate, so the window length is 120 samples.

2.4.2 Feature Extraction Methods

For speech signals, there are a few different ways of extracting and representing information of a frame. These can be classified in three main groups: digital filter banks, Fourier Transform based methods, and Linear Prediction based methods [16]. Figure 2.8 is a schematic representation of the available techniques. In this thesis, the digital filter bank method with a tree structure implementation is used. All the methods are shortly explained below with an emphasis on the method used.

2.4.2.1 Digital Filter Banks

Digital filter banks are used to resemble human ear reception system. In Figure 2.9, we can see the simplified human auditory system. In the process of hearing a sound, the speech wave vibrates the tympanic membrane, and this vibration is transmitted by mechanical means to the cochlea [17]. The cochlea is filled with a liquid and contains the basilar membrane. The vibrations at the entrance of cochlea create standing waves in the liquid. These waves cause the basilar membrane to vibrate. It is believed that the basilar membrane has different resonance characteristics at different distances along it. As a result of this, at a given place on the basilar membrane vibrations of only a particular range of frequency are absorbed. This phenomenon resembles the action of filter banks. The physiological studies have shown that the human perception of frequency content of sounds does not follow a linear scale. This led to the definition of *perceptiually meaningful frequency*. There are two popular scales called *mel* and *bark* which are defined as [18]:

$$mel = 2595\log(1 + f/700)$$
$$bark = 13\arctan(0.76f/1000) + 3.5\arctan\left(f^2/7500^2\right)$$

Another observation in physiological experiments is that, frequencies of a complex sound within a certain bandwidth cannot be individually identified. But if one of the components of this sound falls outside of this bandwidth, then it can be individually identified. This bandwidth is known as the *critical bandwidth* [19]. The critical bandwidths expressed in mel scale are nearly constant.

The above discussion about human auditory system leads to the conclusion that in designing the recognition system, we either change frequency into perceptually meaningful ones through the mel (or bark) scales or take the critical bandwidth concept into consideration and treat frequencies in the critical bandwidth in the same manner. In this thesis, the critical bandwidth concept is taken into account and a tree structure filter bank is designed as will be explained below.

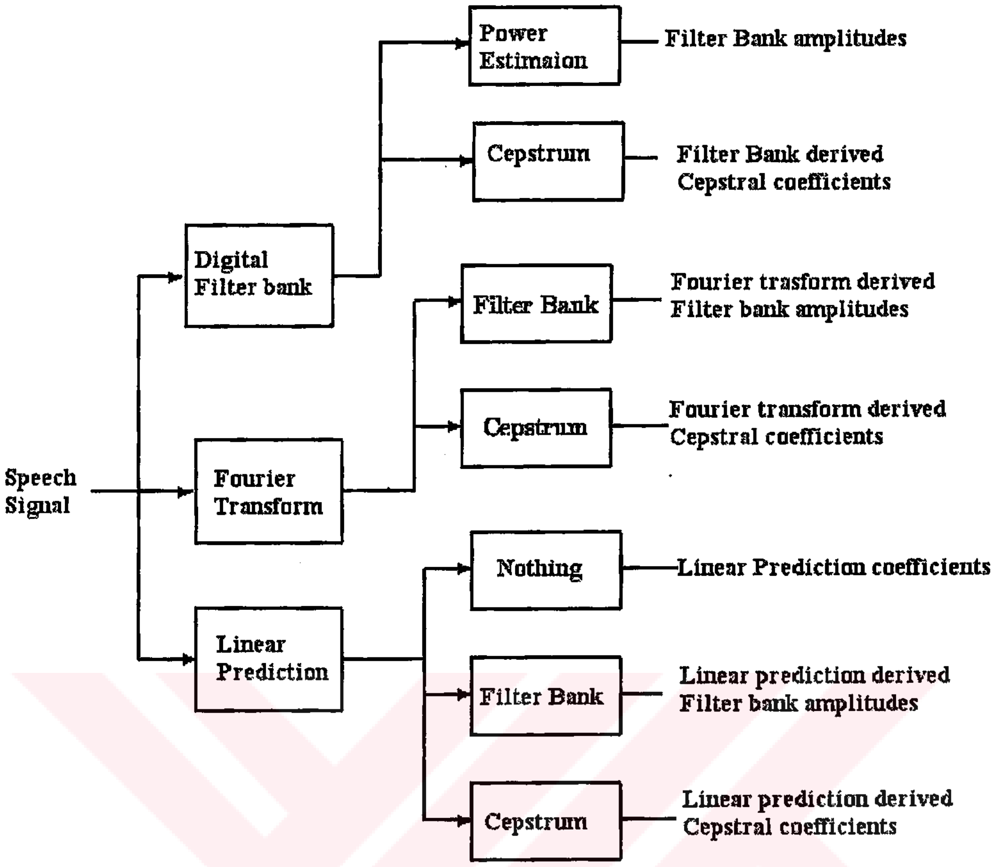

Figure 2.8 A general overview of feature extraction methods.

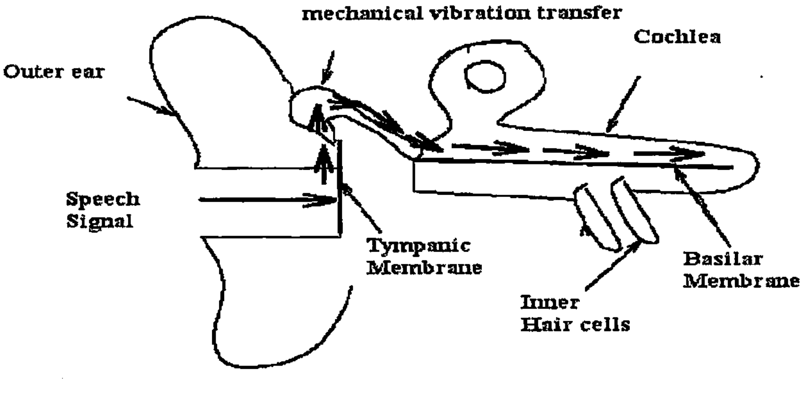

Figure 2.9 Simplified human auditory system.

a) Filter Bank Derived Cepstrum Coefficients

To simulate the human auditory system, the filter bank tree structure is used as shown in Figure 2.13. The main building blocks of the filter bank tree are a lowpass and a complementary highpass filter followed by a downsampler by 2 as shown in Figure 2.10.

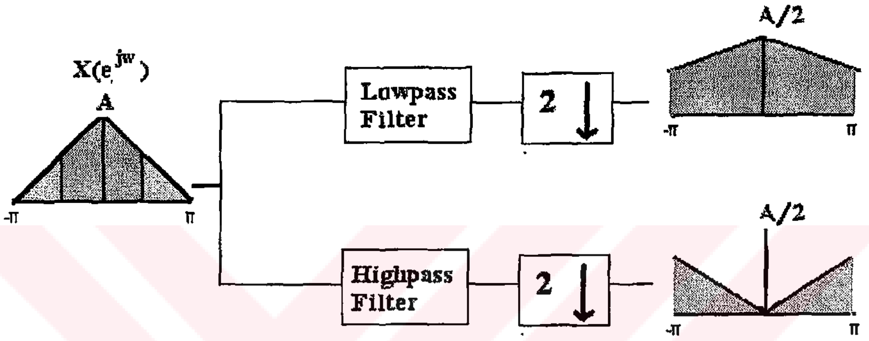

Figure 2.10 Main building block of filter bank tree structure.

There can be many choices for the lowpass filter. One of the possible choices for the lowpass filter is the 7^{th} order Lagrange filter with the transfer function given below [20]:

$$H_L(z) = \frac{-1}{32}z^3 + \frac{9}{32}z^1 + \frac{1}{2} + \frac{9}{32}z^{-1} + \frac{-1}{32}z^{-3} \qquad (2.9)$$

The corresponding high pass filter is:

$$H_H(z) = 1 - H_L(z) = \frac{1}{32}z^3 - \frac{9}{32}z^1 + \frac{1}{2} - \frac{9}{32}z^{-1} + \frac{1}{32}z^{-3} \qquad (2.10)$$

The magnitude spectra of these filters in dB scale are shown in Figure 2.11. One important thing about the design of the filter bank tree is that; the process of highpass filtering and downsampling by 2 inverts the frequency spectrum. So while further decomposing the highpass subband signals, the lowpass and high pass filters must be switched in order to get the desired frequency decomposition structure.

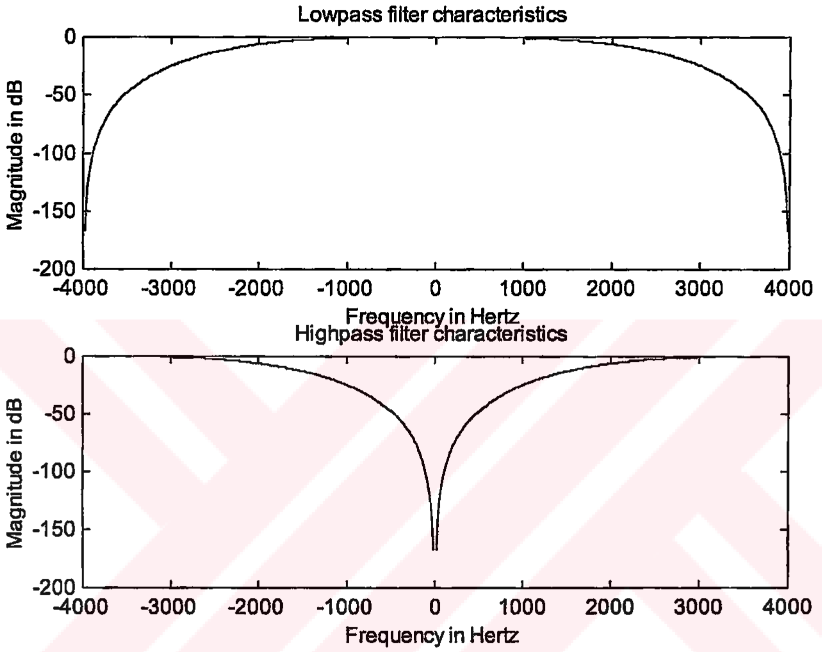

Figure 2.11 Lowpass and highpass filter characteristics.

As a practical rule, if the number of high pass filters used in a branch is even, then the lowpass and highpass filters are used without switching. However, if the number of highpass filters used in a branch is odd, then the lowpass and highpass filters must be switched to get the desired frequency decomposition.

The overall tree structure implemented is given in Figure 2.13. In this figure, the output of each filter branch, which is a kind of energy measure, is indicated as s_i.

This three structure partitions the frequency scale as shown in Figure 2.12, which is similar to the human ear reception structure.

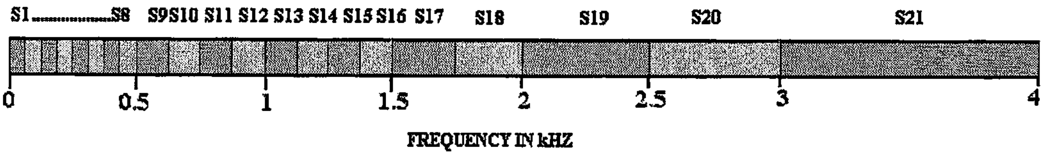
S1.................S8 S9S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S20 S21
0
0.5
1
1.5
2
2.5
3
4
FREQUENCY IN kHZ

Figure 2.12 The frequency partitioning of the filter bank tree.

Each frame is represented by a vector whose components are calculated as below:

$$\hat{S}_i = \frac{1}{W_i}\sum_{n=0}^{W_i} \left|s_i[n]\right|. \tag{2.11}$$

In this expression W_i is the length of the i^{th} filter branch output sample sequence, $s_i[.]$. Note that $\hat{S}_i$ can be considered as the power of the signal in the i^{th} branch. Applying cosine transformation gives

$$c(n) = \frac{1}{21}\sum_{i=0}^{21} \log\left(\hat{S}_i\right)\cos\left(\frac{2\pi(n+1)(i-0.5)}{21}\right) \tag{2.12}$$

As a result of this discussion, to represent each frame 12 dimensional feature vectors are used where each dimension component is found through Equation 2.13.

$$F_n^m = \frac{1}{21}\sum_{i=0}^{20} \log\left(\hat{S}_i^m\right)\cos\left(\frac{2\pi(n+1)(i-0.5)}{21}\right) \tag{2.13}$$

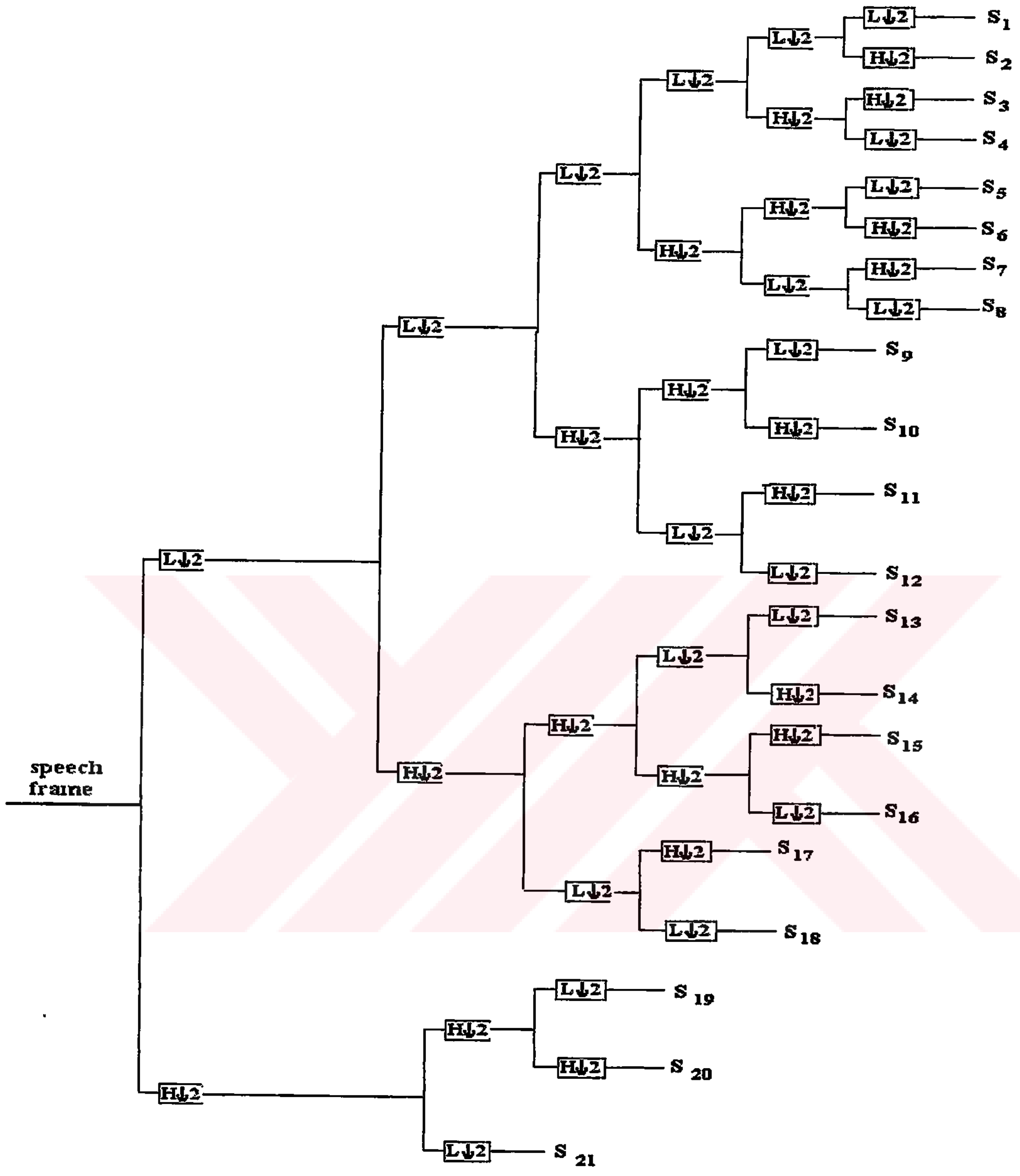

Figure 2.13 The filter bank tree implemented.

In Equation 2.13, m represents the frame number and n shows the index of the corresponding element of the feature vector, $1 \le n \le 12$, and $\hat{S}_i^m$ is the power value of i^{th} filter branch output of the m^{th} frame.

b) **Filter Bank Amplitudes**

This formulation is quite similar to the Filter Bank Derived Cepstral Coefficients method and is different only at the last step. Each element of the feature vector is computed as below.

Let $\hat{s}_i(n)$ denote the n^{th} sample of i^{th} filter output of the m^{th} speech frame, where $0 \le n \le N$. By calculating the power of the signal at the output of each filter through the Equation 2.14, another representation is obtained

$$\hat{s}_i = \sqrt{\sum_{n=0}^{N-1} \hat{s}_i^2(n)} \tag{2.14}$$

Again we form feature vectors but this time using a different style.

2.4.2.2 Fourier Transform Techniques

Fourier transform based techniques uses the Fourier Transform of the signal as a first step. After performing the Fourier analysis, there is a variety of ways of representing the information. Generally the acoustic frequency scale (mel or bark) is used in these operations. We will concentrate on the two most important techniques, namely the Fourier transform based filter bank amplitudes and the Fourier transform based cepstral coefficients.

a) **Fourier Transform Based Filter Bank Amplitudes**

The first step of the technique is to compute the DFT of the signal using,

$$\hat{S}(k)=\sum_{n=0}^{N-1}\hat{s}(n)\exp\left(-\frac{2\pi jkn}{K}\right) \qquad (2.15)$$

In this expression, K shows the total number of points over which DFT is taken. The resulting spectrum $\hat{S}(k)$ is filtered using the critical band filter bank. This operation is carried out in the frequency domain and given by,

$$\hat{S}_i(k)=H_i(k)\left|\hat{S}(k)\right|^2 \qquad (2.16)$$

where $H_i(k)$ is the corresponding weight factor at frequencies $\frac{2\pi k}{K}$ of the i^{th} critical band filter. (It is also possible to use | . | instead of $(.)^2$ operation.) Finally a single parameter is obtained by averaging the outputs of filter bank tree.

$$\hat{S}_i=\frac{1}{K}\sum_{k=0}^{K-1}\hat{S}_i(k) \qquad (2.17)$$

This averaging procedure enhances the robustness against spurious fluctuations of the spectra, in addition to reducing the number of parameters.

b) Fourier Transform Derived Cepstral Coefficients

Let $\hat{S}_i$ represent the energy measure value obtained for the i^{th} filter for the m^{th} frame and P show the number of filters. Equation 2.18 is used to obtain the Fourier Transform Cepstral Coefficient based feature vectors:

$$c(n)=\frac{1}{P}\sum_{i=1}^{P-1}\log\left(\hat{S}_{im}\right)\cos\left(\frac{2\pi(i+0.5)n}{P}\right). \qquad (2.18)$$

Usually, n is taken in the range $0 \le n \le P-1$.

2.4.2.3 Linear Prediction Coefficients

Derivation of Linear Prediction (LP) coefficients, or LPC analysis is based on the simplified human speech production model, where the speech is assumed to be the output of an all pole filter excited by white noise or periodic pulses. Thereafter, the power spectral density of speech, $\Gamma_{ss}(f)$ is obtained as:

$$\Gamma_{ss}(f) = |H(f)|^2 \sigma_{nn}^2 \tag{2.19}$$

where σ_{nn}^2 is the excitation noise variance and

$$H(z) = \frac{1}{A(z)} = \frac{1}{a_0 + a_1 z^{-1} + ... + a_L z^{-L}} \tag{2.20}$$

is the transfer function of the all pole filter of vocal tract. It can be proved that the linear prediction operation defined by,

$$\tilde{s}(n) = \sum_{i=1}^{L} \hat{a}_i s(n-i) \tag{2.21}$$

does exactly the opposite procedure and the prediction error can be written as:

$$\sigma_{ee}^2 = \Gamma_{ss}(f) |\hat{H}(f)|^2 \tag{2.22}$$

where $\hat{H}(z) = \hat{a}_0 + \hat{a}_1 z^{-1} + ... + \hat{a}_L z^{-L}$ and σ_{ee}^2 is the variance of the prediction error. The parameters that minimize the prediction error are $a_i = \hat{a}_i$, $0 \le i \le L$, which yield $\sigma_{ee}^2 = \sigma_{nn}^2$. This relationship makes it possible to evaluate parameters, a_i and σ_{nn}^2 of the speech production model through Linear Prediction. The LP model parameters are directly related to the covariance of the speech signal by the so-called *Yule-Walker* equations. These equations can be solved efficiently by *Levinson-Durbin* algorithm [21].

a) LPC Derived Filter Bank Amplitudes

This method is quite similar to the Fourier transform derived filter bank amplitudes method. In Equation 2.16, $|S(f)|^2$ is replaced by the estimate of $\Gamma_{ss}(f)$, which is more robust than the Fourier transform based estimate. Then we follow the same procedure outlined in part *a* of 2.4.2.2. Since the method is computationally heavy, alternative algorithms are proposed.

b) LPC Derived Cepstral Coefficients

As the name implies, the LPC derived cepstral coefficients are obtained by calculating the cepstrum representation of the sequence of LPC coefficients. More precisely, cepstrum coefficients $c(n)$ are given by:

$$c(n) = -a_n - \sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)c(n-j)a_j \qquad 2 \le n \le N_c \qquad (2.23)$$

$c(1) = -a_1$ and $c(0)$ is removed from the recursion due to its less reliability. N_c shows the number of cepstral coefficients to be taken.

The relationship is derived on the linear frequency scale. We can either use this relationship or incorporate a nonlinear frequency scale and redefine the recursion relationship.

2.5 Parameter Transforms

In this stage, a final modification is done on the feature vectors obtained through the methods described previously. Generally *M* dimensional feature vectors are used to represent each frame information. (*M*=12 in this thesis.) The intention of parameter transform is to further increase the desirable properties of the parameters such as robustness and characterization of temporal behaviour [22]. Some operations involved in parameter transformation are *differentiation*, *weighting* and *averaging*.

A parameter obtained by differentiating a measurement is called a *delta parameter*. Even though a delta parameter represents the temporal behaviour in a better way, it can be very sensitive to noise, because differentiation tends to amplify noise.

Another parameter transform *averaging* can reduce the noise, but it can smooth out the useful temporal information. Weighting can be thought as a generalized process of both averaging and differentiation.

In this thesis, *differentiation* is used as a parametric transform. The intention of differentiation is to add first or higher order derivatives to the feature vector, so that the dynamical behaviour is represented better. There are many ways of calculating the approximate derivatives numerically. The procedures vary from very simple schemes such as taking the differences to fairly sophisticated ones. In this thesis, differentiation is implemented by the formula below:

$$\frac{d}{dt} f_i = G * \sum_{k=-K}^{K} k f_{i+k} \tag{2.24}$$

where f_i denotes the 12 dimensional feature vector representing i^{th} frame. In this thesis, K is taken as 2; and G as 0.335 giving rise to:

$$\frac{d}{dt} f_i = 2.G.f_{i+2} + G.f_{i+1} - G.f_{i-1} - 2.G.f_{i-2} \ . \tag{2.25}$$

This derivative vector is appended to the 12 dimensional initial feature vector and a final 24 dimensional feature vector is obtained in compact form as : $\bar{f}_i = \begin{pmatrix} f_i \\ \dfrac{d}{dt} f_i \end{pmatrix}$.

As a result, speech is transformed into a pattern of 24 dimensional feature vector sequence.

2.6 Conclusion

In this chapter, preprocessing techniques for isolated word recognition system are discussed. Preprocessing answers the question of how a pattern is assigned to each word to be recognized. Due to the human speech production system, speech can be assumed statistically stationary for 10-15 msec duration segments called frames, so a reasonable method of obtaining a pattern for speech is dividing the waveform into frames and representing each frame by its properties. Mostly these properties are related to the spectral characteristics of the frame.

There are some steps that must be taken to achieve this goal, namely, endpoint detection, spectral shaping, signal analysis techniques for each frame and parameter transformation. These topics are covered in detail in this chapter.

CHAPTER 3

PATTERN MODELLING

In this chapter, building models for the words to be recognized will be discussed. This procedure is also known as *training*. The problem associated with word recognition comes from the fact that different acoustic renditions of the speech are seldom realized at the same speaking rate [23]. So, the model should hold the property that it can represent different tokens of the same word regardless of the speaker rate and duration variations. Obviously, there is a strict need to overcome this problem before a decision is made. Two methods have been commonly used to overcome this problem: dynamic time warping (DTW) and Hidden Markov Model (HMM).

3.1 Dynamic Time Warping (DTW)

The name dynamic time warping comes from the fact that the technique warps two speech patterns through the dynamic programming technique. The speech pattern is a sequence of feature vectors and the way of obtaining it is obtained is discussed in detail in Chapter 2. To obtain the model for $word_i$, the i^{th} word of vocabulary, the recognizing system should be *trained* through different tokens of the same word so that it learns the properties specific to the $word_i$ irrespective of the duration and speaking rate variations.

The system is trained for the $word_i$ through N different tokens of the same word. N is either taken as 5 or 3 in this thesis. DTW in a way finds an average optimal sequence (model) to represent the $word_i$ which is smallest in *distance* to other utterances of the same word.

3.1.1 DTW Structure in Detail

Consider two speech patterns X and Y associated with the different tokens of the same word. Let $(x_1 x_2 x_3 ......x_{T_x})$ and $(y_1 y_2 y_3 ......y_{Ty})$ represent feature sequences of X and Y. The notation, $i_x = 1...T_x$, and i_y =1... T_y is used to represent time indices or frame numbers of X and Y patterns, respectively. Usually $T_x \neq T_y$. The dissimilarity of X and Y is based on spectral distortion measures between the frames. The distance between the i_x^{th} frame of X and the i_y^{th} frame of Y is denoted as $d(i_x, i_y)$. There can be many choices for the dissimilarity function and in [24] there is a long discussion about this issue. The Euclidean measure is used in this thesis.

The DTW procedure involves a time warping function ϕ which relates indices of two speech patterns, i_x and i_y, as:

$$i_y = \phi_y(i_x) \qquad i_x=1,\ldots T_x. \tag{3.1}$$

A time warping example is given in Figure 3.1.

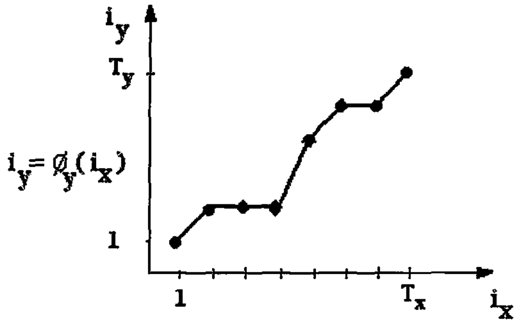

Figure 3.1 An example of a time warping function.

Note that i_y is a monotonically increasing function starting at 1 and ending at T_y. This function can be considered as a path in two dimensional space as shown in Figure 3.2. Note that even for a small size problem quite a large number of paths are possible.

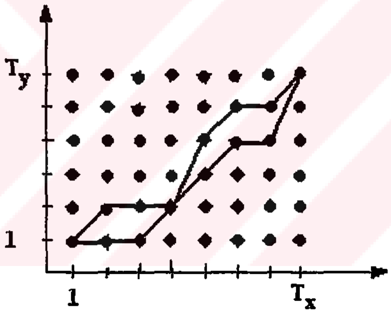

Figure 3.2 Time warping paths.

Thus the aim of DTW is to find an optimal time warping function or an optimal path such that the dissimilarity of X and Y patterns which is defined in Equation 3.2 is minumum:

$$d_\phi(X,Y) = \frac{\sum_{k=1}^{T_X} d(\phi_x(k),\phi_y(k))m(k)}{M_\phi} \tag{3.2}$$

where $m(k)$ is a nonnegative weighting coefficient, and M_ϕ is a path normalizing constant defined as the sum of $m(k)$ over all k:

$$M_\phi = \sum_{k=1}^{T_x} m(k) \tag{3.3}$$

3.1.2 DTW Algorithm

DTW algorithm is based on the dynamic programming concept for solving sequential decision problems stated by Bellman [25] as:

The optimal path has the property that, whatever the initial state and decision are, the remaining decisions starting from time k must constitute an optimal path with regard to the state at time k.

To illustrate this idea, consider Figure 3.2. The aim is to find the best path that minimizes the distance between a given sequence of T_y number of test feature vectors and T_x number of reference feature vectors. Assume that, there is a nonnegative cost $\beta(i, j)$ associated with every point (i, j) that represents the difference between i^{th} reference vector and j^{th} test vector. The problem is to find the total minimum distance and the corresponding sequence of moves going from point $(1,1)$ to (T_x, T_y). Let $\varphi_k(1, j)$ denote the minimum cost of going from point $(1,1)$ to (i, j) in k steps. Then the minimum cost $\varphi_{k+1}(i+1, l)$ we are looking for is achieved through the point $(i+1, l)$ with the following property:

$$\varphi_{k+1}(i+1, l) = \min_{l}[\varphi_k(i, j) + \beta(i+1, l)] \tag{3.4}$$

Generalizing this idea for obtaining the minimum cost and sequence of moves from $(1,1)$ to (i, j), we have:

$$\varphi(i, j) = \min_{k}[\varphi(i, k) + \gamma(k, j)] \tag{3.5}$$

where $\varphi(i, j)$ is the minimum cost of reaching to (i, j) from $(1,1)$ and $\gamma(k, j)$ is the minimum cost going from (k, j) to (i, j). Equation 3.5 implies that *any partial, consecutive sequence of moves of the optimal sequence from* $(1,1)$ *to* (i, j) *must also be optimal, and that any intermediate point must be the optimal point linking the optimal partial sequences before and after that point* [28]. This is the key idea of the DTW algorithm and the Viterbi Algorithm of the HMM structure that will be described in Section 3.2.2.

The DTW problem may have some constraints on the paths. Figure 3.3 shows some of them. All figures in Figure 3.3 indicate possible one step transitions. For example in Figure 3.3.a, the feasible points that may come after the point (i, j) are $(i+1, j)$, $(i, j+1)$ and $(i+1, j+1)$. The other figures can be interpreted similarly. There may be a lot of different type local constraints, and in Figure 3.3 only a few of them are given.

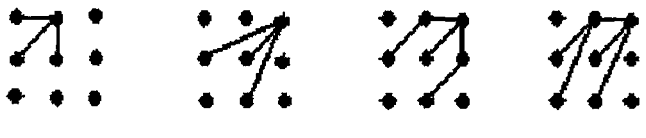

Figure 3.3 Possible one step transitions.

Usually two more constraints are defined for local moves with paremeters $Q_{\max}$ and $Q_{\min}$ showing the maximum and minimum expansions in time warping which are defined as:

$$Q_{\max} = \frac{\text{maximum } \Delta y \text{ the constraint allows}}{\Delta x \text{ for that } \Delta y} \tag{3.6}$$

$$Q_{\min} = \frac{\text{minimum } \Delta y \text{ that the constraint allows}}{\Delta x \text{ for that } \Delta y} \tag{3.7}$$

For example, $Q_{\max} = 2$ and $Q_{\min} = 0.5$ for Figure 3.3.b. These parameters put also a global constraint on the time durations of the X and Y patterns as:

$$Q_{\min}T_x \leq T_y \leq Q_{\max}T_x \tag{3.8}$$

So, the DTW algorithm works only for T_x and T_y values which satisfys the above constraint. These parameters also specify the feasiable region where DTW is applicable. In Figure 3.4 the feasible region for $Q_{\max} = 2$ and $Q_{\min} = 0.5$ is shown.

Another concept associated with the local constraint graph is path weighting. Each possible path has a weight $m(k)$ which is considered in dissimilarity measure defined in Equation 3.2.

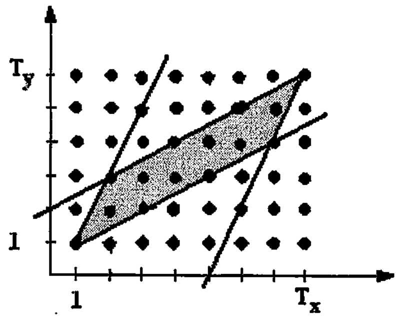

Figure 3.4 The feasible region.

After choosing the local constraint graph, the distance measure between two frames and a weighting schema, the DTW algorithm is applied as follows:

- Step 1: Compute $d(1,1)$.

- Step 2: Increment x coordinate i_x and for all $i_y = 1..T_y$ do the following:

  Check if (i_x, i_y) is in the feasible region; if it is in the feasible region, calculate the mininum distance to reach (i_x, i_y). Store the distance value and the path direction that has led to this decision in an array. If it is not in the feasible region, increment i_Y.

- Step 3: Increment i_x by one and go to Step 2 if $i_x < T_x$ .Note that for calculating the minimum distance, the previously stored mininum values are used.

The distance at $(i_x, i_y) = (T_x, T_y)$ divided by M_ϕ gives the distortion between the two patterns X and Y. Note that, the optimal previous point is also indicated for each intermediate point. Going backwards in these stored optimal points, the optimal path can be obtained and M_ϕ is the summation of all incremental costs along the optimal path.

3.1.3 Building the Model

In the previous discussion, the DTW procedure to find the optimal path that gives the minimum distance between X and Y patterns is explained. Here we will explain how to build the reference model. Let X_1 ... X_N be the speech patterns associated with N training utterances (or tokens) of the same word, say the i^{th} word of the vocabulary. First, the optimal path between X_1 and X_2, i.e. the time warping function, is found.

Then a new pattern X' with duration T_{X_1} is formed by avareging the feature vectors taking the optimal time warping function into account. Second, X' and X_3 is time warped and a new optimal pattern sequence is found. This procedure goes on until all training tokens are time warped. Finally, a last pattern obtained is used as a model pattern to represent the word *i*.

3.2 HMM Modelling

HMM uses the assumption that there are finite, distinctive phonetic units in the spoken language called phonemes, and they can be characterized by their spectrum.

This idea helps to model the word independent of speaker rate and duration variations. Consider the Turkish word " bir " /b-iy-r/ in terms of its phonemes. One can associate each phoneme with a state with possibilities of going to another state or staying at the same state as shown in Figure 3.5. So, long duration "bir"s can be handled with long waitings at the states and the short ones with rapid state transitions. This idea led to the design of HMM with the underlying assumption that the speech signal can be well characterized as a parametric random process, and the parameters of the stochastic process can be determined in a precise, well defined manner [26].

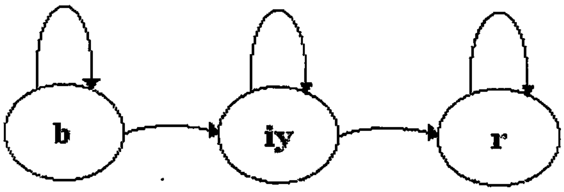

Figure 3.5 State representation of /b-iy-r/.

3.2.1 HMM Parameters

A Markov Model is a finite state machine which goes from state i to state j at every time unit with some probability, a_{ij}. In Figure 3.6, an example of a four state Markov Model with transition probabilities is given. Transition probabilities associated with Figure 3.6 is also expressed below in matrix form. Each entry of the matrix, p_{ij}, gives the transition probability of going from state i to state j.

$$P = \begin{bmatrix} 0.5 & 0.4 & 0 & 0.1 \\ 0 & 0.8 & 0.2 & 0 \\ 0 & 0.3 & 0.3 & 0.4 \\ 0.6 & 0 & 0.1 & 0.3 \end{bmatrix}$$

Figure 3.7 shows two different Markov Model structures. In Figure 3.7.a, any transition is possible for each state, while in Figure 3.7.b, the next state's index can not be less than the present state's index. The Markov Model given in Figure 3.7.a is called ergodic, while 3.7.b is left to right.

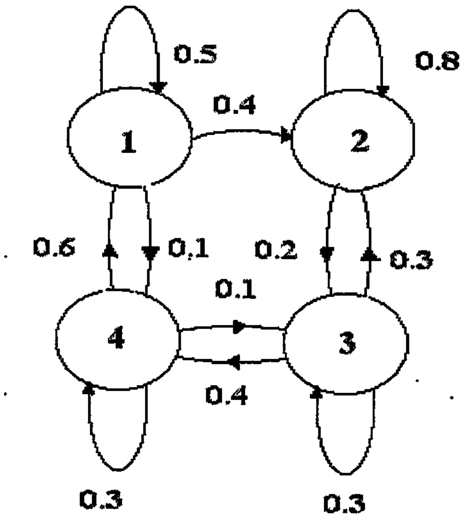

Figure 3.6 An example of a four state Markov Model.

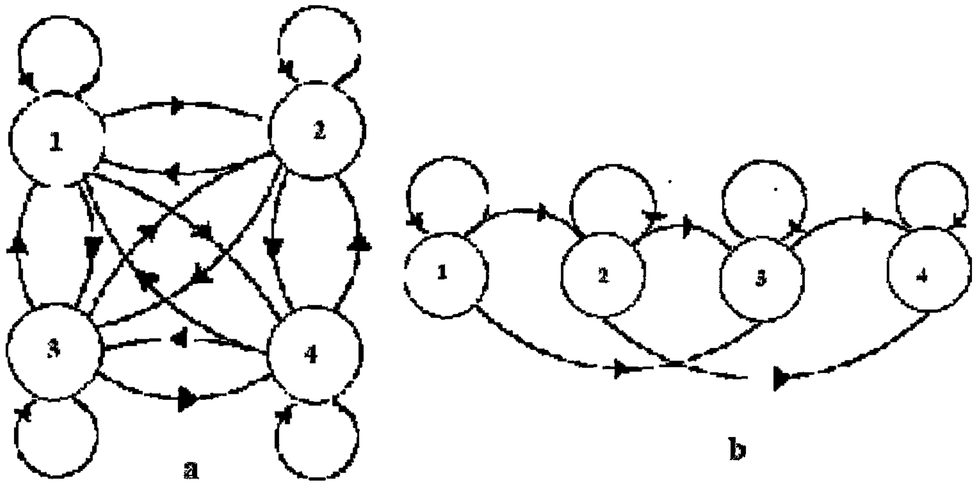

Figure 3.7 Two Markov Model examples.

A Hidden Markov Model is a stochastic process characterized by the following parameters:

- Number of states, N.
- Transition probabilities.
- Initial state distribution.
- State dependent observation probability density function.

For the following discussions, let q_t be a random variable that represents the state at time *t* and o_t be the random variable that represents the speech vector produced at time *t*. The initial state distrubution gives information about the initial state and is expressed as $\pi = \{\pi_i\}$ where $\pi_i = \Pr\{q_1 = i\}$, and i is the state number. $1 \le i \le N$. Starting from a state *i* at *t*=1, the next state to be visited is governed by the transition probabilities a_{ij} or, in general, if i is the state at time t and j is the state at time $t+1$, $a_{ij} = \Pr\{q_{t+1} = j \mid q_t = i\}$. All transition probabilities can be expressed as a matrix $A = \lfloor a_{ij} \rfloor$.

At each state visited, HMM produces an output vector denoted by o_t in a probabilistic manner. The likelihood that o_t is generated at state j is given by $b_j(o_t)$ where $b_j(.)$ is the probability density function associated with state j.

To model signals whose properties change over time in a successive manner like speech, left to right Markov Model is the most appopriate one to use. It is assumed that the sequence of observed feature vectors (speech pattern) corresponding to each word is generated by a Markov Model. In Figure 3.8., an example of an otput sequence generated by Markov Model is given. In this figure, o_t is the feature vector generated at time t. $O = (o_1 \ldots o_8)$ is the output sequence, $Q = (11233344)$ is the state sequence. Note that for this example,

$$P(O,Q \mid M) = b_1(o_1)a_{11}b_1(o_2)a_{12}b_2(o_3)a_{23}b_3(o_4)a_{33}b_3(o_5)a_{33}b_3(o_6)a_{34}b_4(o_7)a_{34}b_4(o_8)$$

where M is the model consisting of (π, A, b).

In practice only the observation sequence is known and the state sequence is unknown or *hidden*. This is why it is called the *Hidden Markov Model*, so the problem is to find the best matching state sequence and the maximum likelihood value.

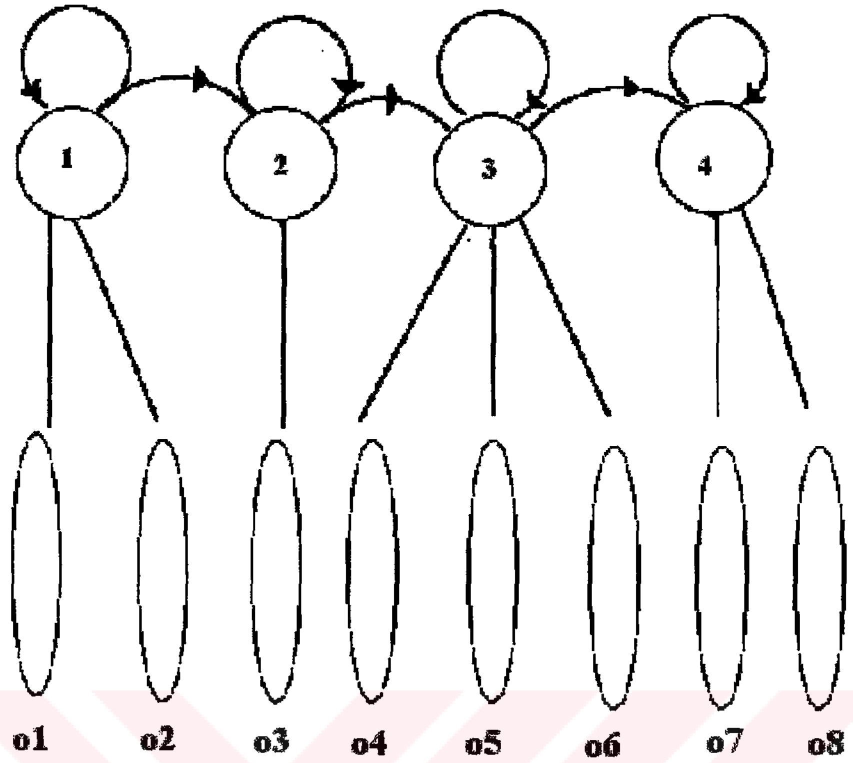

Figure 3.8 An example of an output sequence generated by the Markov Model.

As stated perviously, HMM is characterized by

- Number of states, N.
- Transition probabilities, $A = \lfloor a_{ij} \rfloor_{NxN}$.
- Observation vector probability distribution, $b_j(.)$ j = 1,....N.

In this thesis, a left-to-right HMM model shown in Figure 3.8 is used. Number of states is taken as 5. The probability of staying at the same state is taken as 0.8 so the probability of transition to the next state becomes 0.2, i.e. $a_{ii} = 0.8, a_{i(i+1)} = 0.2$.

A single Gaussian is used as the observation probabilty density function for each state. So, the likelihhod of observing the output feature vector at time t in state j is given by Equation 3.9.

$$b_j(o_t) = \frac{1}{\sqrt{2\pi|\Sigma_j|^n}} \exp\left(\frac{-(o_t - \mu_j)\Sigma_j^{-1}(o_t - \mu_j)^T}{2}\right) \tag{3.9}$$

where μ_j and Σ_j are the mean vector and the covariance matrix of state j, respectively; $|\Sigma_j|$ and Σ_j^{-1} are the determinant and the inverse of matrix Σ_j.

3.2.2 Viterbi Algorithm

Viterbi Algorithm, the name of dynamic programming in speech literature, will be used both to train the word models, i.e. to find the mean vector and covariance matrix parameters of each state and also during testing to find the optimal state sequence.

- **Training**

Let X_1^i ... X_K^i be the speech patterns (observation vector sequences) associated with K training utterances (or tokens) of the same word, the i^{th} word of the vocabulary.

First, each sequence is equally divided into N parts and the initial mean and covariance estimates for each state are calculated as:

$$\mu_j^i = \frac{\sum_{m=1}^{m=K}\sum_{n=1}^{n=N_j^m} o_n^m}{\sum_{m=1}^{m=K} N_j^m} \qquad \Sigma_j^i = diag\left(\frac{\sum_{m=1}^{m=K}\sum_{n=1}^{n=N_j^m} (o_n^m - \mu_j^i)(o_n^m - \mu_j^i)^T}{\sum_{m=1}^{m=K} N_j^m - 1}\right) \tag{3.10}$$

where μ_j^i and Σ_j^i are the mean vector and the covariance matrix of the j^{th} state of the i^{th} word, respectively; *diag(X)* represents a diagonal matrix with diagonal elements equal to the entries of the matrix X; o_n^m is the n^{th} vector of the m^{th} pattern; N_j^m is the number of frames of the m^{th} pattern belonging to the state j.

Next with these initial estimates the optimal state sequence distribution is found for each utterance by the Viterbi Algorithm. Then, with these new state distrubitions, μ_j^i and Σ_j^i are updated through Equation 3.10. This operation is carried out until state distrubitions found by Viterbi Algorithm converge to some point.

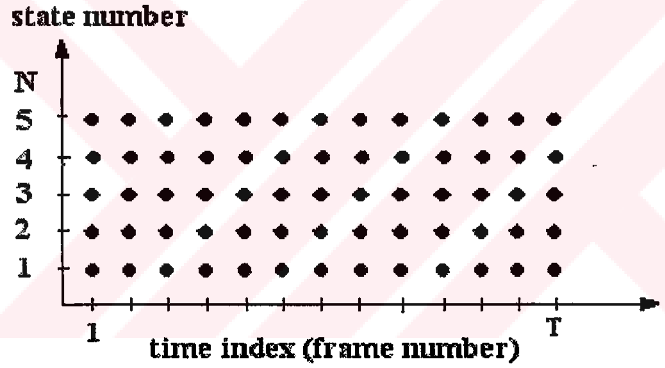

Figure 3.9 State-time space.

Given the mean and covariance matrix estimates for the state and the observation vector sequence $O = (o_1 ... o_T)$, the optimal state sequence calculation is done by the Viterbi Algorithm. Figure 3.9 shows the state and time space. The Viterbi algorithm determines the optimum path which is most probable for the given data. The path must start at $(1,1)$ and end at $(T,5)$. This path gives the state distribution of the observation sequence. Since the left to right HMM model with one transition is used, the transition strategy schema is as shown in Figure 3.10. The feasible points are shown in the grey region, which are imposed by the HMM structure. So, our calculations will include only the points in the feasiable region. For each pioint in the feasible region, two numbers will be stored in an array. One of them will show the maximum likelihood of being up to that point, the other number will be 0 or 1 indicating the previous point in the optimal path. 0 denotes the upper branch of the local transition schema, and 1 denotes the lower branch of the local transition schema.

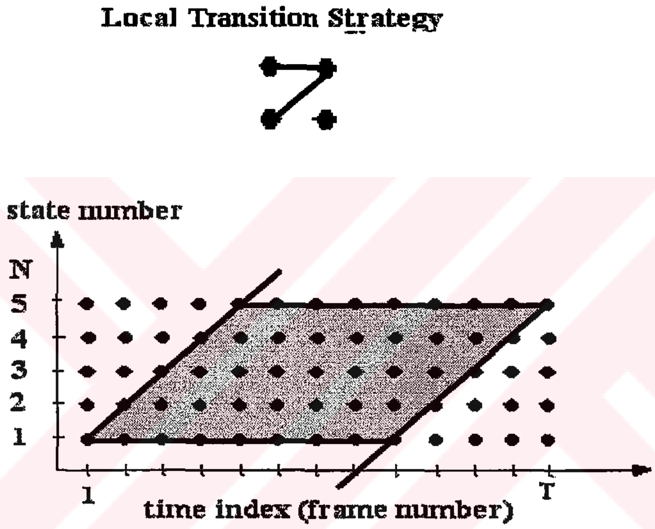

Figure 3.10 Transition moving starategy and the associated feasible region.

Let j=1..5 denote the state number, and i=1..T denote time index. Then (i, j) is a feasible point in Figure 3.10. Let f_{ij} denote the maximum likelihood of being up to that point and g_{ij} indicate the previous point that path has come from.

The Viterbi Algorithm is as follows:

- First, compute f_{ij} and g_{ij} for i=1... $(T$-N+1$)$, j=1 with $g_{ij} = 0$ and for i=1.N , j=1..N with $g_{ij} = 1$. These points have one option to be visited.

- Second, for the second row j=2 for *all* i in that row; compare $f_{(i-1)(j-1)}*0.2$ with $f_{(i-1)j}$ *0.8. If the former is larger, meaning that transition to next state is more probable, then $g_{ij}=1$ and $f_{ij}=b_{ij}*f_{(i-1)(j-1)}*0.2$; else $g_{ij}=1$ and $f_{ij}=b_{ij}*f_{(i-1)j}*0.8$.

b_{ij} is the probability of observing i^{th} output vector at state j given by Equation 3.9.

- Third, go to second step and do this procedure for all j up to N. (5 in our case).

- At $(i,j)=(T,N)$, f_{ij} will have the probability that the most likely path has and

starting from g_{TN} and going backwards in the directions g_{ij} indicate, the optimal state distrubition is obtained.

3.2.3 Building the Model

Let X_1 ... X_N be the speech patterns (observation vector sequences) associated with N training utterances (or tokens) of the same word, say the i^{th} word of the vocabulary. As indicated, first an initial estimate is made on the mean vector and covariance matrix of the states of HMM by assuming that the observation vector sequences are distributed equally to all states. Then by the Viterbi Algorithm, the new state distributions are obtained for each utterance, and the mean and covariance values are updated. This procedure is applied until the state distributions imposed by the Viterbi Algorithm converges. The method described above corresponds to maximum likelihood estimation of the parameters, i.e. mean vectors and covariance matrices.

CHAPTER 4

MODIFICATIONS FOR PERFORMANCE IMPROVEMENT

4.1 General Overview of the System

In this chapter, the recognition procedure and the discriminative techniques are discussed. The aim is correctly recognizing an unknown utterance from a vocabulary containing *M* words under insufficient training data case. In this thesis, we have worked on 60-word vocabulary, which can be found in the Appendix.

To model the i th *word*, $1 \leq i \leq M$, in the vocabulary, a HMM model denoted by M^i is constructed. In this thesis, left-to-right 5-state HMM model with transition probabilities of staying at the same state 0.8 and going to the next state 0.2 is used, moreover, continuous Gaussian probability density function with a diagonal covariance matrix is used as state dependent observation vector distribution. How a word is modelled is discussed previously in Chapter 3 in a detailed way.

For a test utterance to be recognized, the processing shown in Figure 4.1 is applied:

- First, an observation or feature sequence $O = (o_1 o_2 ....o_T)$ is generated through the feature extraction mechanism explained in Chapter 2.

- Second, this sequence is passed through all word models, and $P(O \mid M^i)$, the probability that the observation sequence O is to be produced by model M^i is computed by the Viterbi Algorithm.

- Third, the decision logic sets the recognized word as the word which yields the largest probability.

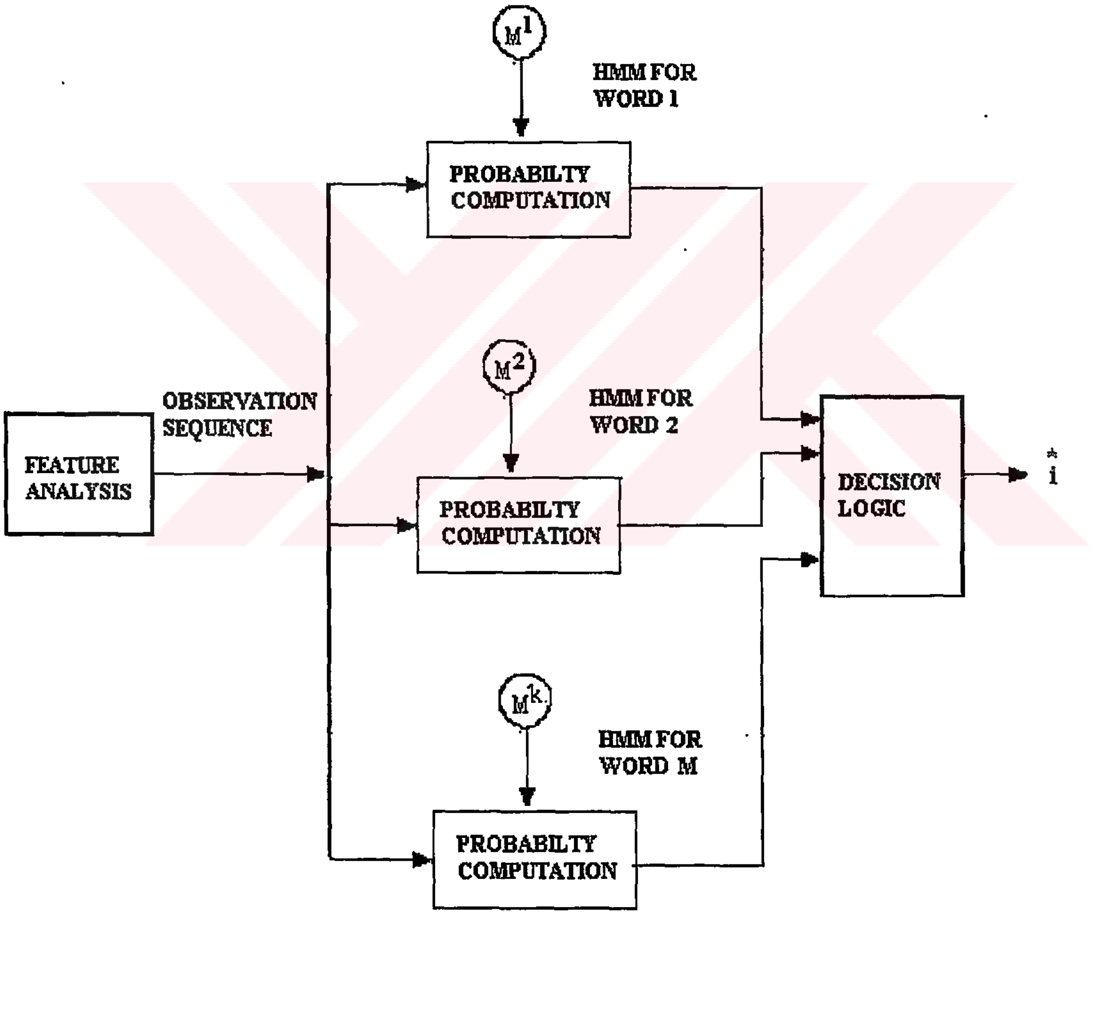

Figure 4.1 Raw HMM recognition system.

In this thesis, to model each word, five or less test utterances are used. This insufficient data brings out an important problem, namely the model parameters are unreliable. This unreliability directly affects the system performance in a negative manner. Therefore, there is a need for post processing techniques to improve the recognition performance.

Furthermore, as the size of vocabulary increases, the performance will surely degrade. This is another motivation for the study on auxiliary modification techniques. The techniques which are worked on can be grouped under three categories:

i) Adjusting Model Parameters.

ii) Feature Vector Reduction Techniques.

iii) Information Fusion Techniques.

These techniques and their results will be discussed in the subsequent parts of this chapter.

4.1.1. Results of the Raw HMM Structure

Raw HMM structure is the one shown in Figure 4.1. Three data sets are used to obtain the performance scores for this system. One of them is used for training the system to model the word parameters. Five different utterances per each word are recorded to be used in training. Two different training modes, 5 or 3 utterances per word, are used during the tests. Number of training utterances is kept small to avoid long training sessions. The other two data sets are used for testing. These are indicated as dataset1 and dataset2 in the tables showing recognition performances for different techniques. Dataset1 consists of 10 test utterances for each word and is recorded one day after the recording of the training data. Dataset2 consists of 10 test utterances for each word and is obtained one month later than the recording of the training data. Since the vocabulary consists of 60 words, dataset1 and dataset2 each have a total of 600 utterances. The performance scores of this raw HMM structure is given in Table 4.1.

Table 4-1 Performance scores of raw HMM structure.

NUMBER OF UTTERANCES OF EACH WORD USED IN TRAINING	5	3
Dataset1	%98.5	%96.3
Dataset2	%97.1	%93.6

Table 4-2 Percentage that the correct word is in the candidate set containing one, two or three most likely words that HMM generated for testing dataset1

DATASET1	FIRST	FIRST & SECOND	FIRST, SECOND & THIRD
#of utterances in training: 5	%98.5	%99.3	%99.3
#of utterances in training: 3	%96.3	%98.0	%98.5

An observation associated with our experiments is that, in most of the incorrect decisions, the correct word model has the second or third largest likelihood value. Based on this observation, three candidates that are most likely to HMM evaluation undergo additional evaluation, and the final decision is obtained after this evaluation. The results in Table 4.2 and Table 4.3 show the percentage that the correct word is in the set of candidates where this set contains the first, the first two, and the first three best words.

Table 4-3 Percentage that the correct word is in the candidate set containing one, two or three most likely words that HMM generated for testing dataset2.

DATASET2	FIRST	FIRST & SECOND	FIRST, SECOND & THIRD
#of utterances in training: 5	%97.1	%98.8	%99.6
#of utterances in training: 3	%93.6	%96.1	%97.1

4.2 Adjusting the Model Parameters

Since the decision is based on the HMM models, a logical way of improving the recognition performance may be the modification of unreliable model parameters. The model parameters are:

- Number of states.
- Transition probabilities.
- Mean vectors and diagonal covariance matrices of states.

i. Number of States

In Figure 4.2, the relationship between number of states and recognition performance in terms of error percentage can be inspected. The graph is obtained from dataset2 where 5-utterance per word is used in training.

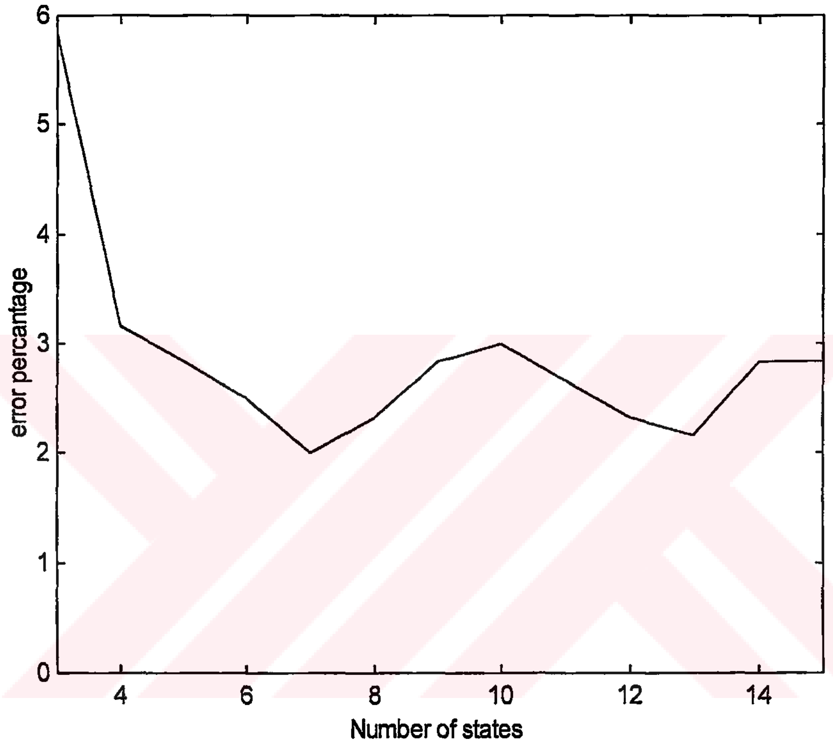

Figure 4.2 Relation between number of states and error percentage.

As seen, the error rate achieves a local minimum at *N*=7. However, the error is somehow insensitive to number of states when this number is larger than 6. Also, it is an important fact that this minimum might be specific to the vocabulary. So, under these considerations *N*=*5* is taken during all studies.

ii. Transition Probabilities

The probability of staying at the same state is taken as 0.8 and the probability of visiting the next state is taken as 0.2 for all the modifications done in this chapter. It is observed from the previous studies that changing this parameter does not affect the performance seriously.

iii. Mean Vector and Covariance Matrix

In this part, only the covariance matrix is modified. The small covariance values with unreliable means may yield very small incorrect likelihood values even at points close to the mean. To overcome this problem, the small covariance values are enlarged [27]. A hard limiter is implemented on the values of the diagonal covariance matrix. If a_{ii} denotes the i^{th} element of a covariance matrix of any state with a value less than δ_1, or greater than δ_2, a_{ii} is set to these thresholds. δ_1 and δ_2 are found experimentally. Results of this modification are given in Table 4.4.

Table 4-4 Results of the hard-limiting covariance values.

NUMBER OF UTTERANCES PERWORD USED IN TRAINING	5	3
Dataset1	%99.3	%98.5
Dataset2	%98.5	%96.6

4.3 Feature Reduction Techniques

The feature reduction technique is a feature modification method, which can be expressed, in a general form as:

$$\hat{x} = Vx \ . \tag{4.1}$$

where x is the 24 dimensional feature vector, V is the $mx24$ transformation matrix and $\hat{x}$ is the modified feature vector with m elements. In this study, $m < 24$ and V consists of rows with one 1 as the nonzero entry leading to removing the components of the feture vector that are not discriminative. The *decision logic* used in this section is that: if the probability difference between the two most probable outputs of HMM structure is less than the *threshold*, apply feature reduction techniques to make the decision; else rely on the decision according to the HMM evaluation.

4.3.1 Gauss Curves Method

Suppose the i^{th} word of vocabulary is confused with the j^{th} word meaning that the test utterances of the i^{th} word are recognized by the system as the j^{th} word or vice versa. To discriminate between these words, the method outlined below is used:

Take the models of i^{th} and j^{th} words as indicated in Figure 4.3. In this figure, μ_k^i and Σ_k^i are the mean vector and the diagonal covariance matrix of the k^{th} state of i^{th} word. Diagonality of covariance matrix implies that the probability density function corresponding to each state can be considered as a multiplication of probability density functions of each element of the feature vector.

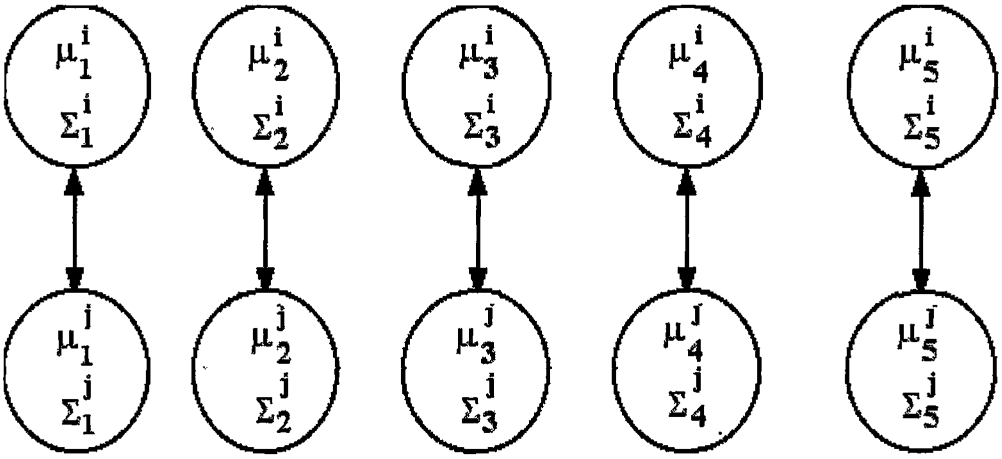

Figure 4.3 Schematic diagram that shows the comparison of word *i* and word *j*.

For each state of the two words, i and j, 24 scalar Gaussian density function pairs can be formed with means and variances from the corresponding component of the mean vector and the diagonal covariance matrix, respectively. For each component, the area shown in Figure 4.4, $S=A+B$ is computed and used as a confusion measure. This area indicates the total probability of error:

$$\Pr(error\, in\, total) = \Pr(error|\, j \text{ is the actual word}) + \Pr(error|\, i \text{ is the actual word}) \qquad (4.2)$$

The components that yield large S values are thought to be the possible source of recognition error; so these components are removed for that state and HMM evaluation is repeated accordingly.

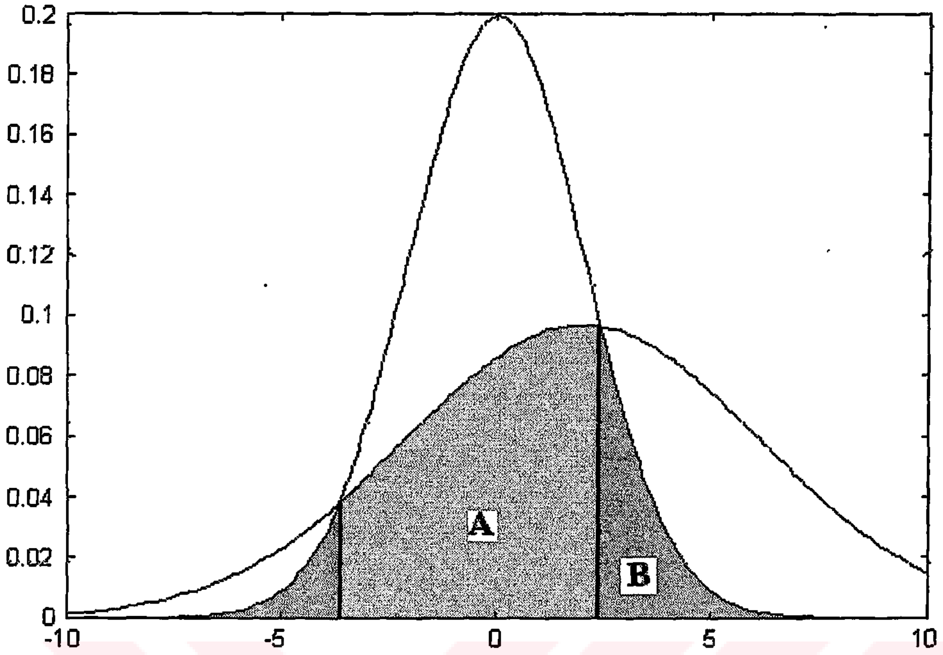

Figure 4.4 A one dimensional gaussian pair.

The results of this approach are given in Table 4.5 and Table 4.6.

Table 4-5 Results of component elimination by the Gauss curve method (5 number of utterances used in training)

NUMBER OF DIMENSION	DATASET1	DATASET2
24	%99.3	%98.5
22	%99.0	%97.3
20	%99.0	%97.6
16	%98.5	%96.5
8	%98.5	%96.5

Table 4-6 Result of component elimination by the Gauss curve method (3 number of utterances are used in training).

NUMBER OF DIMENSION	DATASET1	DATASET2
24	%98.5	%96.6
22	%98.3	%96.5
20	%98.0	%96.2
16	%98.0	%95.8
8	%97.6	%95.6

4.3.2. Bhattcharyya Measure Method

In this method, the components to be removed are determined according to the distance measure called the *Bhattcharyya measure* which is defined as [28]:

$$d = (\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \left(\frac{\frac{|\Sigma_1| + |\Sigma_2|}{2}}{\sqrt{|\Sigma_1||\Sigma_2|}} \right) \tag{4.3}$$

In this definition, μ_1 and μ_2 are the mean vectors and Σ_1 and Σ_2 are the covariance matrices for that state of the i^{th} and j^{th} confusing words belonging to the state for which the dimensions to be reduced for. Note that, if $\mu_1 = \mu_2$ and $\Sigma_1 = \Sigma_2$, i.e. the models are identical, then *d*=0 and also, if $\mu_1 = \mu_2$ then *d* is computed by the second term which takes the covariance difference into account.

For each state, *d* is computed and then each component's contribution to the distance measure is found. The least contributing components are thought to be the possible source of recognition error, and they are removed for that state and HMM evaluation is repeated accordingly. The results of this reduction for different sizes are tabulated in Table 4.7 and Table 4.8.

Table 4-7 Results of component elimination by the Bhattcharyya curve method (5 number of utterances used in training).

NUMBER OF DIMENSION	DATASET1	DATASET2
24	%99.3	%98.5
22	%99.0	%97.5
20	%99.0	%97.3
16	%99.0	%96.8
8	%98.8	%97.0

4.4 Information Fusion Techniques

There can be many ways of incorporating the information to make more reliable decisions. Three methods which are studied in this thesis are:

i. Incorporating State Duration Probability.

ii. State Distribution Pattern with Hybrid HMM-DTW.

iii. Weighting State Probability Contributions Through a Genetic Algorithm.

4.4.1 Incorporating State Duration Probability

The raw HMM structure relies on the "*select maximum*" principle meaning that the test utterance observation sequence is passed through all the models and the model index producing the maximum likelihood is chosen as the recognized word index. The test utterance has a state distribution pattern for each model showing how the frames are distributed to each state. This information is not used in the raw HMM structure, and this method is a way of incorporating this information.

Table 4-8 Results of component elimination by the Bhattcharyya curve method (3 number of utterances used in training)

NUMBER OF DIMENSION	DATASET1	DATASET2
24	%98.5	%96.6
22	%98.0	%97.1
20	%98.3	%96.3
16	%97.8	%95.8
8	%97.6	%95.8

The test observation sequence is passed through all models and the likelihood value of the i^{th} model, $P(M^i \mid O)$ is obtained. Define d_j^i as number of frames assigned to the j^{th} state of the i^{th} model by HMM $j = 1..5$ and in compact form in can be represented by a vector $d^i = (d_1^i \; d_2^i \; d_3^i \; d_4^i \; d_5^i)$. State duration function for the j^{th} state, $p_j(x)$, is modelled with gamma distribution with parameters α_j, λ_j as given below:

$$p_j(x) = \begin{cases} \dfrac{\lambda_j e^{-\lambda_j x} (\lambda_j x)^{\alpha_j - 1}}{\Gamma(\alpha_j)}, & x \geq 0 \\ 0 \quad , & x < 0 \end{cases} \quad \text{and} \quad \Gamma(\alpha_j) = \int_0^\infty e^{-y} y^{\alpha - 1} dy. \tag{4.4}$$

The α_j, λ_j parameters are found previously in the training phase of the system. As a result, the likelihood values are modified as:

$$P'(M^i \mid O) = P(M^i \mid O) + \sum_{j=1}^{5} p_j(d_j^i). \tag{4.5}$$

The recognized word is chosen as the model having the maximum modified likelihood value. The results of this approach are given below in Table 4.9.

Table 4-9 Results of incorporating state duration probability.

NUMBER OF UTTERANCES PERWORD USED IN TRAINING	5	3
Dataset1	%99.3	%98.6
Dataset2	%98.5	%96.5

4.4.2 State Distribution Pattern with Hybrid HMM-DTW

In this method, state duration information, the likelihood value given by HMM and the distance measure given by DTW are combined to make a final decision. It has been observed that HMM evaluation may point out a model that has an odd *strange* state distribution such as accumulation of frames in one state. Most of the time such distributions correspond to a wrong decision. Therefore, the information about state distribution pattern must be taken into account to make decision.

Let c_1, c_2 and c_3 be the indices of the three candidate words that are most likely according to the HMM evaluation. Consider d^{i1}, d^{i2} and d^{i3} vectors defined before associated with these models. An "odd state distribution" measure used in this thesis for word *i* is the number of d_j^i's that are equal to one. So the procedure is to choose the best model according to this criterion, i.e. eliminate the one with highest odd state distribution measure. Considering the state distribution patterns, the number of states which are 1 frame distributed is counted for each candidate. Then the final decision is made between two models having the least odd state distribution measure. The test utterance to be recognized is passed through the HMM and DTW models of these chosen two candidates and the probability of decision is found for each model as:

$$P_{decision} = \delta_1 P_{HMM} + \frac{\delta_2}{D_{DTW}} \tag{4.6}$$

where δ_1, δ_2 are weighting factors determined experimentally, P_{HMM} is the likelihood value generated by HMM and D_{DTW} is the distance given by the DTW models. The decided word is chosen as the one having the larger decision probability. The results of this approach are given in Table 4.10.

Table 4-10 Results of the hybrid DTW-HMM approach.

NUMBER OF UTTERANCES PERWORD USED IN TRAINING	5	3
Dataset1	%99.1	%98.8
Dataset2	%99.3	%98.0

4.4.3 Weighting State Probability Contributions through Genetic Algorithm

In this method, a Genetic Algorithm is used to discriminate between two models by weighting the contribution of each state to the overall likelihood. A detailed discussion on genetic algorithms can be found in [29]. As in the previous section, let c_1 and c_2 be the indices of the two most likely models. The likelihood produced by the raw HMM structure is the product of probability contributions of each states as:

$$P\left(M^{c_1} \mid O\right) = P_1 P_2 P_3 P_4 P_5 \tag{4.7}$$

In this method, each model likelihood value is modified by weighting the probability contributions by some coefficients as:

$$P'\left(M^{c_1} \mid O\right) = w_1 P_1 w_2 P_2 w_3 P_3 w_4 P_4 w_5 P_5 . \;\; w_i \; 1 \le i \le 5 \text{ and } 0 \le w_i \le 1 . \tag{4.8}$$

The aim is to find the best coefficients to help the discrimination of c_1 and c_2. In the implementation of the genetic algorithm, the population size is taken as 25 with mutation rate 0.08 and crossover rate 0.7. Each member of population is a 5 dimensional coefficient array, $(w_1w_2w_3w_4w_5)$, where each coefficient is represented by 10 bits. So, each population member is a 50-bit array.

First, an initial population is created with 24 randomly generated coefficient arrays and the last member of population is taken as $w_i\ i = 1..5$, (11111) coefficient array to take the unmodified case into account.

Second, for each member of population, the fitness function value is evaluated. The fitness function is calculated as:

$$f(w_1, w_2, w_3, w_4, w_5) = \left(\sum_{\substack{for\ all \\ utterances\ of \\ \text{c1 and c2}}} \frac{1}{1 + \exp\left(-\alpha\left(P - \hat{P}\right)\right)} \right).K \text{ and} \tag{4.9}$$

where

$$\begin{aligned} &K = 1\,if\ all\left(P - \hat{P}\right) > 0 \\ &else\,K = 0 \end{aligned}. \tag{4.10}$$

In this calculation, P is the likelihood value obtained after passing the training utterance from its own model after modification of coefficient array and $\hat{P}$ is the likelihood value obtained after passing the training utterance from the other model. The aim is to enlarge the difference between P and $\hat{P}$ so that the models can be discriminated better than the raw HMM structure. K puts a constraint on the fitness function such that all utterances are recognized truly.

Third, the members having the best fitness value are taken as parents and the children obtained are replaced with the worst members of the previous population.

Fourth, the second and the third steps are repeated 10 times to obtain the optimum coefficient array $w_i, i = 1..5$,$(w_1 w_2 w_3 w_4 w_5)$. The likelihood values of most likely models, c_1 and c_2, are updated with these optimum coefficients and the decision is made in favour of the one that yields the larger value. The results of this approach are given in Table 4.11.

Table 4-11 Results of weighting by genetic algorithm.

NUMBER OF UTTERANCES PERWORD USED IN TRAINING	5	3
Dataset1	%99.2	%98.5
Dataset2	%98.0	%95.6

4.5 Discussion on Results

The methods described can be grouped under three categories:

- Adjusting Model Parameters

Basically, we have dealt with hard-limiting covariance matrix values only. It had a significant effect on the performance of the system. Also it is a computationally easy method.

- Feature Vector Reduction Techniques

In this part, the feature reduction with Gauss curve and Bhattcharyya distance measure are studied on. As the dimensions are reduced, the performance has not increased.

- Information Gathering Techniques

Incorporating the state distribution in terms of probability has increased the performance slightly. The best result is obtained with the State Distribution Pattern with Hybrid HMM-DTW approach. Although, the genetic algorithm approach increases performance, it is computationally heavy and does not increase as much as hard-limiting approach does.

CHAPTER 5

CONCLUSION

In this thesis, we have worked on the isolated word recognition, a sub area of the automatic speech recognition with model based approach. As a model, the left to right HMM with 5 states and transition probabilities $(0.8,0.2)$ is used. The system consists of three parts:

- Preprocessing,
- Training to obtain model parameters,
- Testing and recognition.

In the preprocessing stage, the information about how a speech is produced and perceived by ear is incorporated into the design of the recognition system. The speech is divided into frames and each frame is represented by a feature vector. As a feature vector, subband based cepstral coefficients are used to simulate the human ear reception system. Different types of feature vectors are discussed in Chapter 2.

During training, each word in the vocabulary is modelled by a set of parameters. However, in this thesis, the number of utterances to model each word is small; as a result the model parameters are unreliable. Therefore, auxiliary methods for discrimination are essential. The recognition system consists of three parts:

- Input observation sequence,
- Model parameters,
- Decision logic.

For input vector modifications, the feature vector reduction methods are used. As feature reduction methods, the Gauss curve and Bhatcharrya measure methods are used. The reduction of dimensions through these methods did not yield positive solution. They are based on the assumption that the dimensions are independent of each other. This assumption might be misleading, or the methods proposed are not the right reduction mechanisms. Focus is made on a reduction method, because it also decreases the computational load of the system during testing. As a future work, a general transformation like $\hat{x} = Vx$ might be used where V is the transformation matrix is. Also, a nonlinear transformation such as $\hat{x} = f(x)$ related with model parameters may help the solution to the problem. However, they might be computationally heavier than the feature reduction methods.

To modify model parameters, the covariance matrix values are hard limited. Again this is an effective and computationally attractive solution. But for large vocabularies, probably it will not help to the solution of the recognition improvement problem. As a future work, other kinds of operations, related with model parameters in some way can be used to modify the covariance matrices.

In decision logic part, various sources of information either from the HMM model, like the state distribution, or from other sources like the DTW models are incorporated to make the decision. In the state duration probability method, the gamma distribution function is used to model the state duration probability. It had a slight improvement on the results but not as much as a variance limiting one.

It is observed that the odd state distribution patterns usually do not correspond to the correct models. This insight is combined with another recognition procedure DTW (Dynamic Time Warping) on to the base HMM evaluations. This hybrid approach has proven good success. Weighting state probability contributions through the genetic algorithm is a costly solution and has not brought success as the hybrid approach of HMM and DTW. It has the intention of putting weight on the discriminative states. As an alternative approach, using phonetic structures to locate regions of dissimilarity can help the discrimination process.

REFERENCES

[1] L. Rabiner, B.H. Juang, "Fundamentals of Speech Recognition", Englewood Cliffs, NJ, Prentice Hall, 1993.

[2] J.L. Flanagan, C.H. Coker, L.R. Rabiner, R.W. Schafer, and N. Umeda, "Synthetic Voices for Computes", IEEE Spectrum, 7 (10):22-45, October 1970.

[3] J.L. Flanagan, "Speech Analysis,Synthesis, and Perception", 2nd ed., Springer Verlag, New York, 1972.

[4] V.R. Lesser, R.D. Fennel, L.D. Erman, and D.R. Reddy, "Organization of the Hearsay-II Speech Understanding System", IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-23(1):11-23, 1975.

[5] N.D.Warakagoda, "A Hybrid ANN-HMM ASR System with NN based Adaptive Preprocessing", Msc.Thesis, June 5,1999

[6] Spectogram Reading home page, "http://cslu.cse.ogi.edu/tuturdemos/SpectogramReading/waveform.html.".

[7] G. Fant, "Speech Sounds and Features", M.I.T. Press, 1973.

[8] M. Savaji, "A robust algorithm for endpointing of speech", Speech Communication, 8:45-60, 1989.

[9] Ney, H. "An optimization algorithm for determining the endpoints of uttereances", ICASSP, pg 720-723, 1989.

[10] L.Rabiner, M.Sambur, "An algorithm for determining the endpoints of utterances", Bell Syst.Tech.J., 54(2):297-315, 1975.

[11] L.F.Lamel, L.R.Rabiner, A.E.Rosenberg, and J.G.Wilpon, "An improved endpoint detector for isolated word recognition", IEEE Trans, ASSP, ASSP-29:777-785, Aug.1981.

[12] G.S.Ying,C.D.Mitchell,L.H.Jamieson,"Endpoint detection of isoalted utterances based on a modified Teager Energy Measurement", ICASSP-2,pp:732-736, 1993.

[13] H.M.Teager, "Some observations on oral air flow during phonation", IEEE Trans, ASSP, ASSP-28:599-601, Oct.1980.

[14] J.F. Kaiser, "On a simple algorithm to calculate 'energy' of signal", Proc. IEEE ICASSP-90, pp.381-384, Apr. 1990.

[15] J.Jungua, J.Haton, "Robustness In Automatic Speech Recognition", Kluwer Academic Publishers, 1996.

[16] G.M.White and R.B. Neely, "Speech Recognition Experiments with Linear Prediction , Bandpass Filtering, and Dynamic Programming", IEEE Trans. ASSP, ASSP-24(2): 183-188, 1976.

[17] O.Ghitza, "Robustness againist noise: The role of timing-synchrony measurement", ICASSP, pg:1995-2001, 1998.

[18] W.Ainsworth, "Mechanisms of Speech Recognition", Pergamon Press.

[19] W.Munson, H.Fletcher, "Relation between loudness and masking", J.Acoust. Soc.Am., 34:1865-1875.

[20] S.M.Phoong, C.W.Kim,P.Vaidyanathan, and R.Ansari, "A new class of two channel biortogonal filter banks and wavelet bases",IEEE Trans.on Signal Processing, pp.694-665,1995.

[21] J.D.Markel and A.H.Gray, "Linear Prediction of Speech",Springer Verlag,1976.

[22] S.Furui,"Speaker Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum",IEEE Trans.ASSP, ASSP-34 (1):52-59, February 1986.

[23] S.Levinson,"Continously variable duration hidden Markov models for automatic speech recognition", Computer Speech and Language, 1(1):29-45,1986.

[24] R.M.Gray,A.Buzo,A.H.Gray and Y.Matsumata, "Distortion Measures for Speech Recognition", IEEE Trans.ASSP,ASSP-28 (4): 367-376, August 1980.

[25] R.E.Bellman, "Dynamic Programming", Princeton University Press, Princeton, New Jersey, USA, 1957.

[26] J.D.Ferguson, "Hidden Markov Analysis: An introduction" in Hidden Markov Models for Speech, Institute of Defense Analyses, Princeton, NJ, 1980.

[27] S.E.Levinson, L.R.Rabiner, and M.M.Sondhi," An introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", Bell System Tech.J., 62 (4): 1035-1074, April 1983.

[28] K.Fukunaga,"Introduction to Stattistical Pattern Recognition",Academic Press Publishers, 2nd ed.,1990.

[29] D.E.Goldberg, "Genetic Algorithms in Search,Optimization and Machine Learning", Addison-Wesley,1989.

APPENDIX

This is the list of vocabulary that is used during my studies:

0 aç

1 büyüt

2 çık

3 evet

4 geri

5 git

6 gizle

7 haber

8 hayır

9 ileri

10 in

11 iptal

12 kapat

13 küçült

14 posta

15 sağ

16 seç

17 sol

18 sonraki

19 tamam

20 tazele

21 tekrar

22 vazgeç

23 yazdır

24 yeni

25 ahmet

26 anne

27 baba

28 doktor

29 itfaiye

30 market

31 nilgun

32 okul

33 polis

34 şirket

35 köroğlu

36 kuzuoğlu

37 kuzucuoğlu

38 karakaş

39 karakaya

40 gürkaya

41 gürkaynak

42 altıntaş

43 altınel

44 kiremitçi

45 kirazcı

46 öz

47 özalp

48 özkan

49 şimşek

50 adana

51 ankara

52 çanakkale

53 çankırı

54 izmir

55 nevşehir

56 kırşehir

57 lüleburgaz

58 çankaya

59 eskişehir