國 立 交 通 大 學

電 機 與 控 制 工 程 學 系

碩 士 論 文

針對非特定語者語音辨識使用不同前處理技術之比較

A Comparison of Different Front-End Techniques for Speaker-Independent Speech Recognition

研 究 生：蕭 依 娜

指導教授：陳 永 平 教授

中 華 民 國　九十三年 六月

針對非特定語者語音辨識使用不同前處理技術之比較

A Comparison of Different Front-End Techniques for Speaker-Independent Speech Recognition

研 究 生：蕭依娜　　Student：Yi-Nuo Hsiao

指導教授：陳永平 教授　　Advisor：Professor Yon-Ping Chen

國 立 交 通 大 學
電 機 與 控 制 工 程 學 系

碩 士 論 文

A Thesis
Submitted to Department of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of Master

in

Electrical and Control Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

針對非特定語者語音辨識
使用不同前處理技術之比較

研究生： 蕭依娜　　指導教授：陳永平 教授

國立交通大學電機與控制工程學系

摘要

本論文針對非特定語者的系統，使用不同特徵粹取技術，透過以單音素為基礎之非特定語者的語音辨識系統以及以字元為基礎之非特定語者語音辨識系統的表現優劣來做為比較的依據。這些特徵粹取技術可以被分為以「語音產生方式」為主以及以「語音感知」為主兩類。第一類包含了線性預估編碼(LPC)、由線性預估編碼所衍生的倒頻譜係數(LPC-derived Cepstrum)以及反射係數(RC)。第二類則包含了梅爾倒頻譜係數(MFCC)以及感知線性預估(PLP)分析。由架構於非特定語者的實驗結果得知，由語音感知為主的第二類的辨識率較高於由語音產生方式為主的第一類，其中，梅爾倒頻譜係數 (MFCC) 在以單音為基礎下，辨識率為 78.3% ，以字元為基礎下，辨識率為 98.5%；感知線性預估 (PLP) 係數在以單音為基礎下，辨識率為 78.9% ，以字元為基礎下，辨識率為 98.5%。

A Comparison of Different Front-End Techniques for Speaker-Independent Speech Recognition

Student：Yi-Nuo Hsiao　　Advisor：Professor Yon-Ping Chen

Department of Electrical and Control Engineering
National Chiao Tung University

ABSTRACT

Several parametric representations of the speech signal are compared with regard to monophone-based recognition performance and syllable-based recognition performance of speaker-independent speech recognition system. The parametric representation, namely the feature extraction techniques, evaluated in this thesis can be divided into two groups: based on the speech production and based on the speech perception. The first group includes the Linear Predictive Coding (LPC), LPC-derived Cepstrum (LPCC) and Reflection coefficients (RC). The second group comprises the Mel-frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictive (PLP) analysis. From the experimental results, the speech perception group, including MFCC (78.3% for monophone-based and 98.5% for syllable-based) and PLP (78.9% for monophone-based and 98.5% for syllable-based), are superior to the features based on the speech production, including LPC, LPCC and RC, in the speaker-independent recognition experiments.

Acknowledgement

本論文能順利完成，首先感謝指導老師 陳永平教授這兩年來孜孜不倦的指導，除了課業的解惑外，亦著重學習態度、研究方法及語文能力上的培養，因此，在這些方面上亦讓我有相當的成長，謹向老師致上最高的謝意；此外，感謝 賓少煌學長於繁忙的工作之餘，仍抽空解答我研究上的疑惑以及給予建議，並不時給予我鼓勵，在此誠摯地表達感謝之意。最後，感謝口試委員 林進燈教授以及 林昇甫教授提供寶貴意見，使得本論文能臻於完整。

此外，還要感謝可變結構控制實驗室的克聰學長、豐裕學長、建峰學長、豐洲學長、培瑄、翰宏、智淵、世宏、倉鴻以及學弟們對我的照顧與陪伴，讓我在實驗室的研究生活充滿溫馨與快樂。另外，感謝室友宜錦及貞伶，在我最累的時候給我打氣。最後，感謝父母與妹妹給我生活上的照顧與精神上的支持。

謹以此篇論文獻給所有關心我、照顧我的人。

蕭依娜 2004.6.27

Contents

Index of Figures

Index of Tables

Chapter 1

Introduction

1.1 Motivation

Imaging that if we can control the equipments and tools in our surroundings through voice command, just like people in the sci-fi movies do, the world will be more convenient and fantastic. In many real-world applications, such as toys, cell phones, automatic ticket booking, goods ordering, etc and it can be foreseen that there will be more and more services provided in the form of speech in the future. The speaker-independent (SI) automatic speech recognition is the way to achieve the goal. Although the speaker-dependent automatic speech recognition system outperforms the speaker-independent automatic speech recognition system in the recognition rate, it is infeasible to collect large speech data of the user and then train the models in real applications, especially the popular commodities. Hence, the solution of providing services for general users is to build a speaker-independent (SI) automatic speech recognition system.

It has been shown that the selection of parametric representations significantly affects the recognition results in an isolated-word recognition system [16]. Therefore, this thesis focuses on the selection of the best parametric representation of speech data for speaker-independent automatic speech recognition. The parametric representation, namely the feature extraction techniques, evaluated in this thesis can be divided into two groups: based on the speech production and based on the speech perception. The first group includes the Linear Predictive Coding (LPC), LPC-derived Cepstrum (LPCC) and Reflection coefficients (RC). The second group comprises the

Mel-frequency Cepstral Coefficients (MFCC) and Perceptual Linear Predictive (PLP) analysis. In general, the speech signal is comprised of the context information and the speaker information. The objective of selecting the best features of speaker-independent automatic speech recognition is to eliminating the difference between speakers and enhancing the difference of phonetic characteristics. Therefore, in this thesis, two corpora are employed in the experiment to evaluate the performance of different features.

In recent years, Hidden Markov Model (HMM) has become the most powerful and popular speech model used in ASR due to its remarkable ability of characterizing the acoustic signals in a mathematically tractable way and better performance compared to other methods, such as Neural Network (NN), Dynamic Time Warping (DTW). The statistical model HMM plays an important role to model the speech signals especially for speech recognition system since the template method is no more feasible for large number of users and large vocabulary system. HMM is proceeded after the extracting the features from the speech signal where the features means MFCCs, LPCs, PLPs, etc. The Hidden Markov Model is employed to model the acoustic features in all the experiments in this thesis.

1.2 Overview

The chapter of thesis is organized as follows. In chapter 2, the front-end techniques of the speech recognition system will be introduced, including the feature extraction methods, such as LPC, MFCC and PLP, utilized in this thesis. The chapter 3 will show the concept of Hidden Markov Model and its training and recognition procedure. Then the experimental results and comparison of different features will be shown in chapter 4. The experimental conclusion will be given in the last chapter.

Chapter 2

Front-End Techniques of Speech Recognition System

In modern speech recognition systems, the front-end techniques mainly includes converting the analog signal to a digital form, extracting important signal characteristics such as energy or frequency response, and augmenting perceptual meanings of these characteristics, such as human production and hearing. The purpose of the front-end processing of the speech signal is to transform a speech waveform into a sequence of parameter blocks and to produce a compact and meaningful representation of the speech signal. Besides, the front-end techniques can also remove the redundancies of the speech and then reduce the computational complexity and storage in the training and recognition steps, thus the performance of recognition will improve through effective front-end techniques.

Independent of what the parameter kind extracted later is, there are four simple pre-processing steps, including constant bias removing, pre-emphasis, frame blocking, and windowing, which are applied prior to performing feature extraction. And these steps will be expressed and stated in the following four sections. In addition, three common feature extraction methods, Linear Prediction Coding (LPC) [2], Mel Frequency Cepstral Coefficient (MFCC) [3], and Perceptual Linear Predictive (PLP) Analysis [4], will be described in the last section of this chapter.

2.1 Constant bias Removing

The speech waveform probably has a nonzero mean, denoted as DC bias, due to the environments, the recording equipments, or the analogous-digital conversion. In

order to get better feature vectors, it is necessary to estimate the DC bias and then remove it. The DC bias value is estimated by

$$DC_{bias} = \frac{1}{N}\sum_{k=1}^{N} s(k) \tag{2-1}$$

where $s(k)$ is the speech signal possessing N samples. Then the signal after removing the DC bias, denoted by $s'(k)$, is given

$$s'(k) = s(k) - DC_{\text{bias}}, \qquad 1 \le k \le N \tag{2-2}$$

where N is the total samples of the speech signal. After the process of constant bias removing, the pre-emphasis filter is then applied to the speech signal $s'(k)$ which is stated in the next section.

2.2 Pre-emphasis

The purpose of pre-emphasis is to eliminate the effect of glottis while producing sound and to compensate the high-frequency parts depressed by the speech generation system. Typically, the pre-emphasis is fulfilled with a high-pass filter in a form as

$$P(z) = 1 - \mu z^{-1}, \qquad 0.9 \le \mu \le 1.0 \tag{2-3}$$

which increases the relative energy of the high-frequency spectrum and introduces a zero near μ. In order to cancel a pole near $z = 1$ due to the glottal effect, the value of μ is usually greater than 0.9 and it is set to be $\mu = 0.97$ in this paper. The pole and zero of the filter $P(z) = 1 - 0.97\, z^{-1}$ are 0 and 0.97 respectively. Furthermore, the frequency responses for the pre-emphasis filter with $\mu = 0.9$, 0.97, and 1 are given in Fig 2-1. The filter is intended to boost the signal spectrum 20dB per decade approximately [5]. Fig.2-2 shows the comparison of the speech signal before and after pre-emphasis.

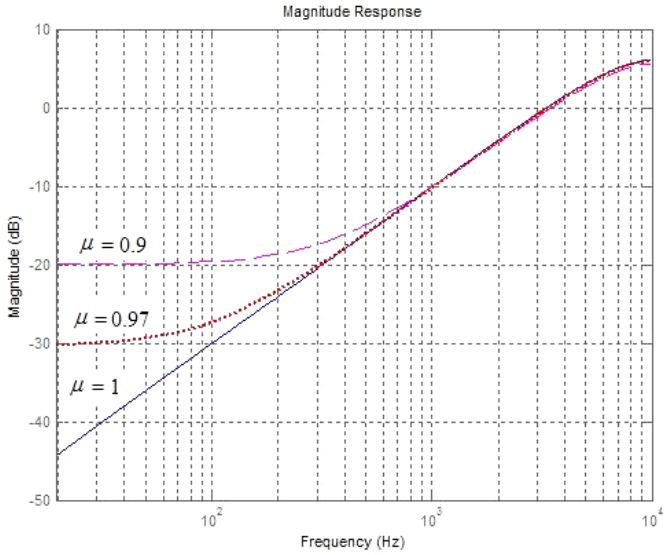


Fig.2- 1 Frequency Response of the pre-emphasis filter

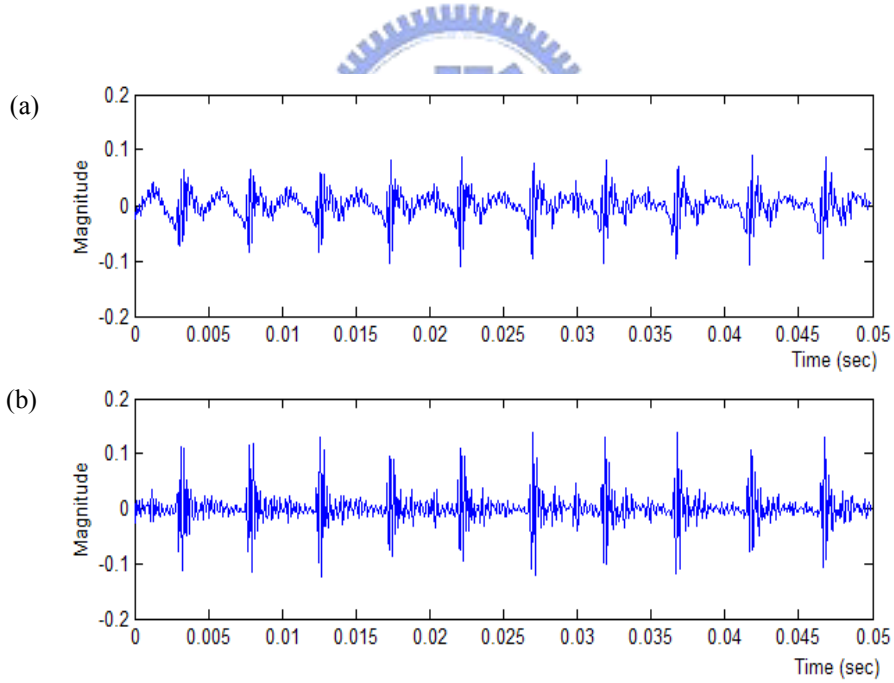


Fig.2- 2 Speech signal (a) before pre-emphasis and (b) after pre-emphasis

2.3 Frame Blocking

The objective of frame blocking is to decompose the speech signal into a series of overlapping frames. In general, the speech signal changes rapidly in time domain;

nevertheless, the spectrum changes slowly with time from the viewpoint of the frequency domain. Hence, it could be assumed that the spectrum of the speech signal is stationary in a short time, and then it is more reasonable to do spectrum analysis after blocking the speech signal into frames. There are two parameters should be concerned, that is frame duration and frame period, shown in Fig.2-3.

I. Frame duration

The frame duration is the length of time (in seconds), usually ranging between 10 ms ~ 30 ms, over which a set of parameters are valid. If the sampling frequency of the waveform is 16 kHz and the frame duration is 25 ms, there are 16 kHz × 25 ms = 400 samples in one frame. It is noted that the total number of samples in a frame is called the frame size.

II. Frame period

As shown in Fig.2-3, the frame period is often selected on purpose shorter than the frame duration to avoid the characteristics changing too rapidly between two successive frames. In other words, there is an overlap with time length equal to the difference of frame duration and frame period.

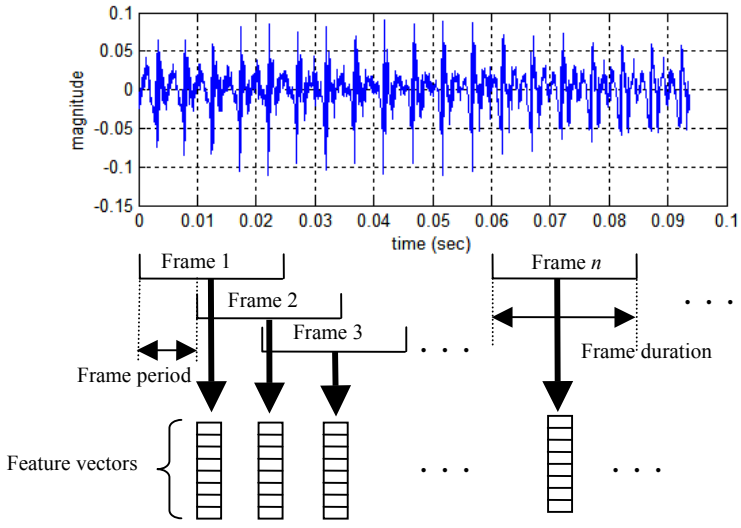


Fig.2- 3 Frame blocking

2.4 Windowing

After frame blocking, the process of windowing applies to each frame by multiplying a Hamming window, shown in Fig.2-4 for N=64, to minimize the spectrum distortion and discontinuities. Let the Hamming window be given as

$$w(n) = 0.54 - 0.46 \cdot cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \le n \le N-1 \tag{2-4}$$

where N is the window size, chosen the same as the frame size. Then the result of windowing process to m-th sample $s_m(n)$ can be obtained as

$$s_{mw}(n) = s_m(n)w(n), \qquad 0 \le n \le N-1 \tag{2-5}$$

Fig.2-5 shows an example of the time domain and frequency response for two successive frames, frame m and frame m+1, of the speech signal before and after multiplying by a Hamming window. From this figure, the spectrum of $s_{mw}(n)$ is smoother than the $s_m(n)$. It is noted that there is little variation between two consecutive frames in their frequency response.

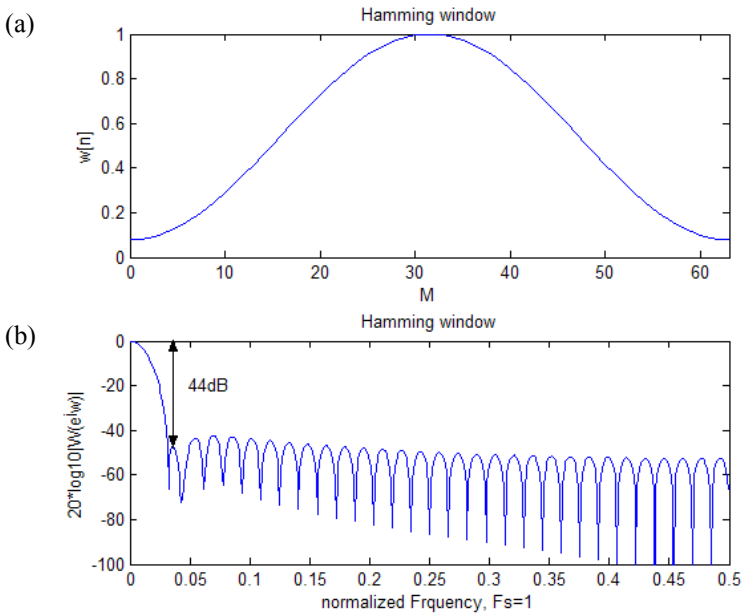


Fig.2- 4 Hamming window (a) in time domain and (b) frequency response

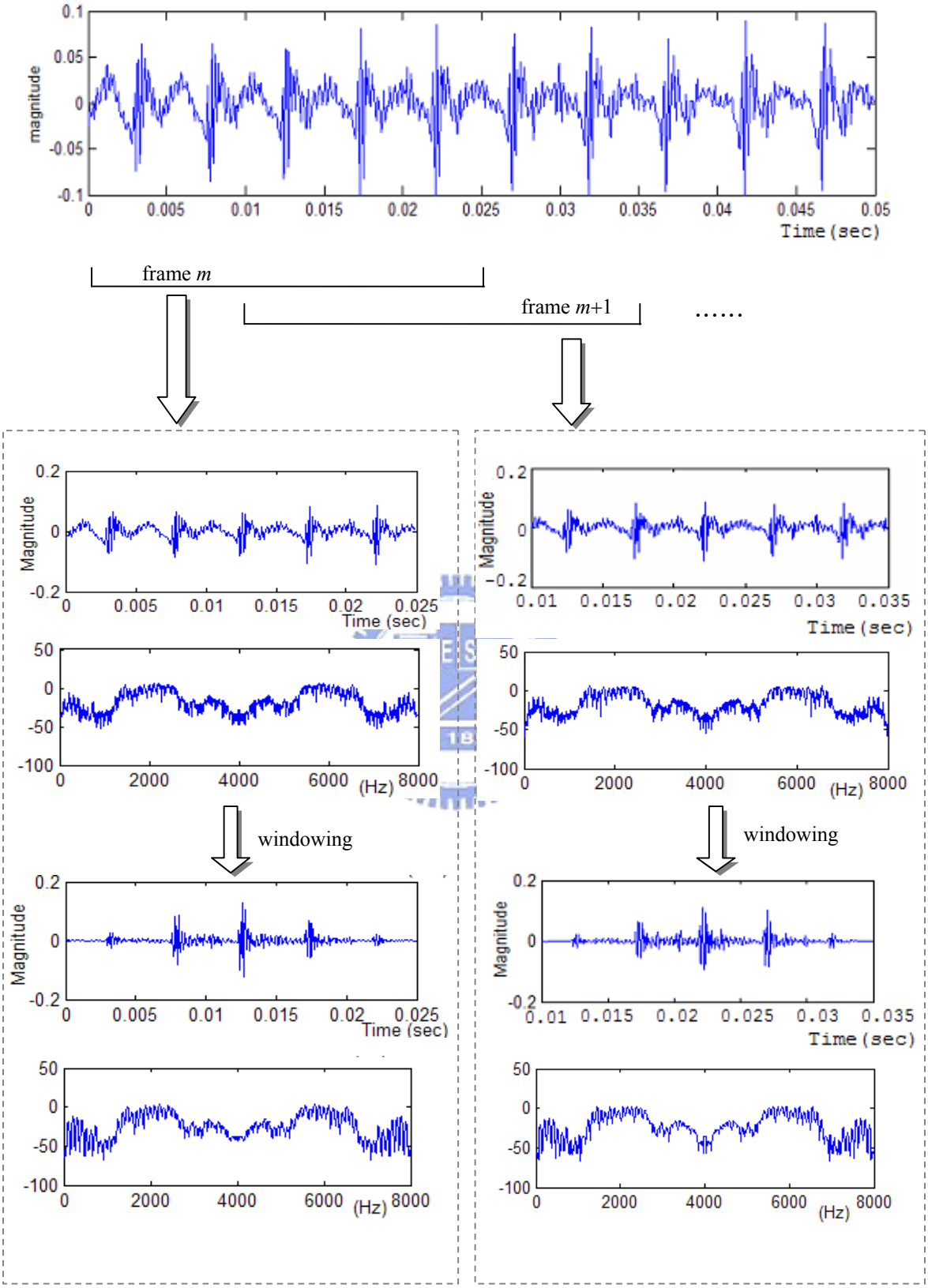


Fig.2- 5 Successive frames before and after windowing

2.5 Feature Extraction Methods

Feature extraction is the major part of front-end technique for the speech recognition system. The purpose of feature extraction is to convert the speech waveform to a series of feature vectors for further analysis and processing. Up to now, several feasible features have been developed and applied to the speech recognition, such as Linear Prediction Coding (LPC), Mel Frequency Cepstral Coefficient (MFCC), and Perceptual Linear Predictive (PLP) Analysis, etc. The following sections will present all the techniques.

2.5.1 Linear Prediction Coding (LPC)

For the past years, Linear Prediction Coding (LPC), also known as auto-regressive (AR) modeling, has been regarded as one of the most effective techniques for speech analysis. The basic principle of LPC states that the vocal tract transfer function can be modeled by an all-pole filter as

$$H(z)=\frac{S(z)}{GU(z)}=\frac{1}{1-\sum_{k=1}^{p}a_k z^{-k}}=\frac{1}{A(z)} \tag{2-6}$$

where $S(z)$ is the speech signal, $U(z)$ is the normalized excitation, G is the gain of the excitation, and p is the number of poles (or the order of LPC). As for the coefficients $\{a_1, a_2,\ldots,a_p\}$, they are controlled by the vocal tract characteristics of the sound being produced. It is noted that the vocal tract is a non-uniform acoustic tube which extends from the glottis to the lips and varies in shape as a function of time. Suppose that characteristic of vocal tract changes slowly with time, thus $\{a_k\}$ are assumed to be constant in a short time. The speech signal $s(n)$ can be viewed as the output of the all-pole filter $H(z)$, which is excited by acoustic sources, either impulse train with period P for voiced sound or random noise with a flat spectrum for unvoiced sound,

shown in Fig.2-6.

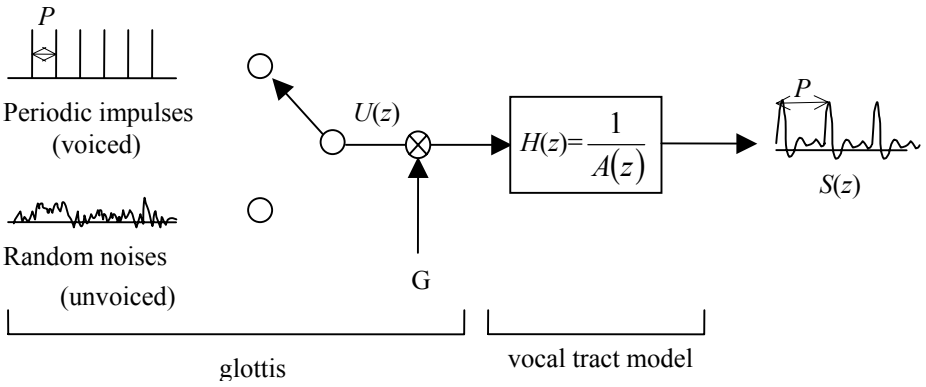


Fig.2- 6 Speech production model estimated based on LPC model

From (2-6), the relation between speech signal $s(n)$ and the scaled excitation $Gu(n)$ can be rewritten as

$$s(n)=\sum_{k=1}^{p} a_k s(n-k)+Gu(n) \tag{2-7}$$

where $\sum_{k=1}^{p} a_k s(n-k)$ is a linear combination of the past p speech samples. In general, the prediction value of the speech signal $s(n)$ is defined as

$$\hat{s}(n)=\sum_{k=1}^{p} a_k s(n-k) \tag{2-8}$$

and then the prediction error $e(n)$ could be found as

$$e(n)=s(n)-\hat{s}(n)=s(n)-\sum_{k=1}^{p} a_k s(n-k) \tag{2-9}$$

which is clearly equal to the scaled excitation $Gu(n)$ from (2-7). In other words, the prediction error reflects the effect caused by the scaled excitation $Gu(n)$.

To use the LPC is mainly to determine the coefficients $\{a_1, a_2,\ldots,a_p\}$ that minimizes the square of the prediction error. From (2-9), the mean-square error, called the short-term prediction error, is then defined as

$$E_n=\sum_{m=0}^{N-1+p} e_n^{\,2}(m)=\sum_{m=0}^{N-1+p}\left(s_n(m)-\sum_{k=1}^{p} a_k s_n(m-k)\right)^2 \tag{2-10}$$

where N is the number of samples in a frame. It is commented that the short-term

prediction error is equal to G^2 and the notation of $s_n(m)$ is defined as

$$s_n(m)=\begin{cases} s(m+n)w(m), & 0\le m\le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2-11}$$

which means $s_n(m)$ is zero outside the window $w(m)$. It can be imaged that In the range of $m=0$ to $m=p-1$ or in the range of $m=N$ to $m=N-1+p$, the windowed signals $s_n(m)$ are predicted as $\hat{s}_n(m)$ by previous p signals and some of the previous signals are equal to zero since $s_n(m)$ is zero when $m<0$ or $m>N-1$. Therefore, the prediction error $e_n(m)$ is sometimes large at the beginning ($m=0$ to $m=p-1$) or the end ($m=N$ to $m=N-1+p$) of the section ($m=0$ to $m=N-1+p$).

The minimum of the prediction error can be obtained by differentiating E_n with respect to each a_k and setting the result to zero as

$$\frac{\partial E_n}{\partial a_k}=0, \qquad k=1,2,...,p \tag{2-12}$$

and then E_n is replaced by (2-11), the above equation can be rewritten as

$$\sum_{m=0}^{N-1+p}\left(s_n(m)-\sum_{k=1}^{p}\hat{a}_k s_n(m-k)\right)s_n(m-i)=0, \qquad i=1,2,...,p \tag{2-13}$$

where i and k are two independent variables and $\hat{a}_k$ are the values of a_k for $k=1,2,\ldots,p$ that minimize E_n. From (2-13), we can further expand the equation as

$$\sum_{m=0}^{N-1+p}s_n(m)s_n(m-i)=\sum_{k=1}^{p}\hat{a}_k\sum_{m=0}^{N-1+p}s_n(m-k)s_n(m-i), \quad i=1,2,...,p \tag{2-14}$$

where the term $\sum_{m=0}^{N-1+p}s_n(m)s_n(m-i)$ and $\sum_{m=0}^{N-1+p}s_n(m-k)s_n(m-i)$ will be replaced by the autocorrelation function $r_n(i)$ and $r_n(i-k)$ respectively. The autocorrelation function is defined as

$$r_n(i-k)=\sum_{m=0}^{N-1+p}s_n(m-k)s_n(m-i), \qquad i=1,2,...,p \tag{2-15}$$

where $r_n(i-k)$ is equal to $r_n(k-i)$. Hence, it is equivalent to use $r_n(|i-k|)$ to replace the term $\sum_{m=0}^{N-1+p} s_n(m-k)s_n(m-i)$ in (2-16). By replacing (2-16) with autocorrelation function $r_n(i)$ and $r_n(i-k)$, we can obtain

$$\sum_{k=1}^{p} \hat{a}_k r_n(|i-k|) = r_n(i), \qquad i = 1,2,...,p \tag{2-16}$$

which matrix form is expressed as

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-2) & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-3) & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-4) & r_n(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_n(p-2) & r_n(p-3) & r_n(p-4) & \cdots & r_n(0) & r_n(1) \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(1) & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_{p-1} \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \vdots \\ r_n(p-1) \\ r_n(p) \end{bmatrix} \tag{2-17}$$

which is in the form of ***Rx*** = ***r*** where ***R*** is a Toeplitz matrix, that means the matrix has constant entries along its diagonal.

The Levinson-Durbin recursion is an efficient algorithm to deal with this kind of equation, where the matrix ***R*** is a Toeplitz matrix and furthermore it is symmetric. Hence the Levinson-Durbin recursion is then employed to solve (2-20), and the recursion can be divided into three steps, as

Step 1. Initialization

$$E(0) = r_n(0), \quad a(0,0) = 1$$

Step 2. Iteration (a_i^j is denoted as $a(i,j)$)

$$for \ \ i = 1 \ \text{to} \ p \quad \{$$

$$k(i) = \frac{r_n(i+1) - \sum_{j=1}^{i-1} a(j,i-1) r_n(i-j)}{E(i-1)}$$

$$a(i,i) = k(i)$$

$$for \quad j = 1 \ \text{to} \ i-1$$

$$a(j,i) = a(j,i-1) - k(i)a(i-j,i-1)$$

$$E(i) = \left(1 - k(i)^2\right)E(i-1)$$

}

Step 3. Final Solution

for $j = 1$ to p

$$a(j) = a(j,p)$$

where the $\hat{a}_j = a(j)$ for $j = 1, 2, \ldots, p$, and the coefficients $k(i)$ are called reflection coefficients whose value is bounded between 1 and -1. In general, the $r_n(i)$ is replaced by a normalized form as

$$r_{n\_\text{normailizd}}(i) = \frac{r_n(i)}{r_n(0)} \tag{2-18}$$

which will result in identical LPC coefficients (PARCOR) but the recursion will be more robust to the problem with arithmetic precision.

Another problem of LPC is to decide the order p. As p increases, more detailed properties of the speech spectrum will be reserved and the prediction errors will be lower relatively, but it should be notice when p is beyond some value that some irrelevant details will be involved. Therefore, the guideline for choosing the order p is given as

$$p = \begin{cases} F_s + (4 \text{ or } 5) & \text{voiced} \\ F_s & \text{unvoiced} \end{cases}, \tag{2-19}$$

where F_s is the sampling frequency of the speech in kHz [6]. For example, if the speech signal is sampled at 8 kHz, then the order p is can be chosen as 8~13. Another rule of thumb is to use one complex pole per kHz plus 2-4 poles [7], hence p is often chosen as 10 for the sampling frequency 8 kHz.

Historically, LPC is first used directly in the feature extraction process of the automatic speech recognition system. LPC is widely used because it is fast and simple. In addition, LPC is effective to compute the feature vectors by Levinson-Durbin recursion. It is noted that the unvoiced speech has higher error than the voiced speech since the LPC model is more accurate for voiced speech. However, the LPC analysis approximates power distribution equally well at all frequencies of the analysis band which is inconsistent with human hearing because the spectral resolution decreases with frequency beyond 800 Hz and hearing is also more sensitive in the middle frequency range of the audible spectrum.[11]

In order to make the LPC more robust, the cepstral processing, which is a kind of homomorphic transformation, is then employed to separate the source $e(n)$ from the all-pole filter $h(n)$. It is commented that the homomorphic transformation $\hat{x}(n) = D(x(n))$ is a transformation that converts a convolution

$$x(n) = e(n) * h(n) \tag{2-20}$$

into a sum

$$\hat{x}(n) = \hat{e}(n) + \hat{h}(n) \tag{2-21}$$

which is usually used for processing signals that have been combined by convolution. It is assumed that a value N can be found such that the cepstrum of the filter $\hat{h}(n) \approx 0$ for $n \geq N$ and the excitation of $\hat{e}(n) \approx 0$ for $n < N$. The lifter ("l-i-f-ter" is the inverse of the word "f-i-l-ter") $l(n)$ is used for approximately recovering $\hat{e}(n)$ and $\hat{h}(n)$ from $\hat{x}(n)$. Fig.2-7 shows how to recover $h(n)$ with $l(n)$ given by

$$l(n) = \begin{cases} 1 & |n| < N \\ 0 & |n| \geq N \end{cases} \tag{2-22}$$

and the operator D usually uses the logarithmic arithmetic and D^{-1} use inverse Z-transform. In the similar way, the $l(n)$ is given by

$$l(n)=\begin{cases}1 & |n|\geq N\\ 0 & |n|<N\end{cases} \tag{2-23}$$

which is utilized for recovering the signal $e(n)$ from $x(n)$.

In general, the complex cepstrum can be obtained directly from LPC coefficients by the formula expressed as

$$\hat{h}(n)=\begin{cases}0 & n<0\\ \ln G & n=0\\ a_n+\sum_{k=1}^{n-1}\left(\frac{k}{n}\right)\hat{h}(k)a_{n-k} & 0<n\leq p\\ \sum_{k=n-p}^{n-1}\left(\frac{k}{n}\right)\hat{h}(k)a_{n-k} & n>p\end{cases} \tag{2-24}$$

where $\hat{h}(n)$ is the desired LPC-derived cepstrum coefficients $c(n)$. It is noted that, while there are finite number of LPC coefficients, the number of cepstrum is infinite. Empirically, the number of cepstrum which is approximately equal to $1.5p$ is sufficient.

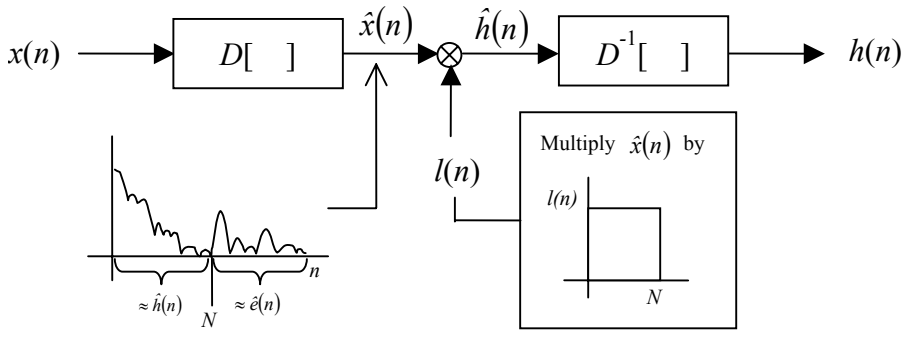


Fig.2- 7 Homomorphic filtering

2.5.2 Mel-Frequency Cepstral Coefficients (MFCC)

The Mel-Frequency Cepstral Coefficients (MFCC) is the most widely used feature extraction method for state-of-the-art speech recognition system. The conception of MFCC is to use nonlinear frequency scale, which approximates the behavior of the auditory system. The scheme of the MFCC processing is shown in Fig.2.8, and each step will be described below.

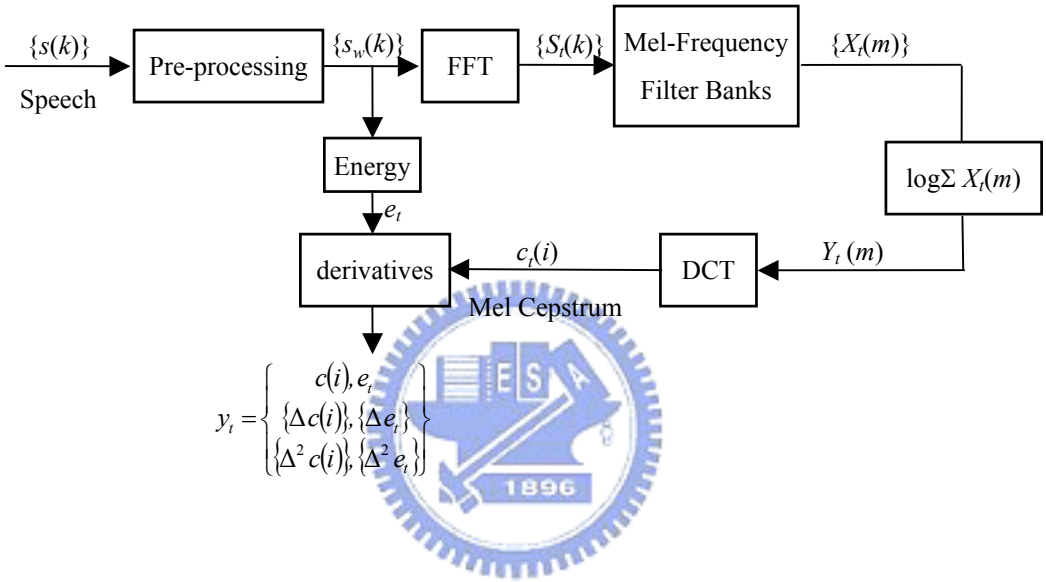


Fig.2- 8 Scheme of obtaining Mel-frequency Cepstral Coefficients

After the pre-processing steps discussed above, including constant bias removing, pre-emphasis, frame blocking, and windowing, are applied to the speech signal, the Discrete Fourier Transform (DFT) is then performed to obtain the spectrum where DFT is expressed as

$$S_t(k) = \sum_{i=0}^{N-1} s_w(i) e^{-j 2\pi ik/N}, \quad 0 \le k < N \tag{2-25}$$

where N is the size of DFT chosen the same as the window size. The Fast Fourier Transform (FFT) is often adopted to substitute for the DFT for more efficient computation. The Mel filter banks will be defined later after making a short introduction of the Mel scale.

The Mel scale, is obtained by Stevens and Volkman [8][9], is a perceptual scale motivated by nonlinear properties of human hearing and it attempts to mimic the human ear in terms of the manner that the frequencies are sensed and resolved. In the experiment, the reference frequency was selected as 1 kHz and equaled it with 1000 mels where a mel is defined as a psychoacoustic unit of measuring for the perceived pitch of a tone [10]. The subjects were asked to change the frequency until the pitch they perceived was twice the reference, 10 times, half, 1/10, etc. For instance, if the frequency they perceived is twice the reference, namely 2 kHz, while the actual frequency is 3.5 kHz, the frequency 3.5 kHz is mapping to the Mel frequency twice 1000 mels, that is, 2000 mels. The formulation of Mel scale is approximated by

$$B(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{2-26}$$

where $B(f)$ is a function for mapping the actual frequency to the Mel frequency, shown in Fig.2.9, and the Mel scale frequency is almost linear below 1 kHz and is logarithmic above. The Mel filter bank is then designed by placing M triangular filters non-uniformly along the frequency axis to simulate the band-pass filters of human ears, and the m-th triangular filter is expressed as

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{(k - f(m-1))}{(f(m) - f(m-1))} & f(m-1) \le k \le f(m) \\ \dfrac{(f(m+1) - k)}{(f(m+1) - f(m))} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases},$$

$$0 \le k < N, \quad 1 \le m \le M \tag{2-27}$$

which satisfies $\sum_{m=1}^{M} H_m(k) = 1$ and N is the size of the FFT. The boundary points $f(m)$ in the above equation can be calculated by

$$f(m)=\left(\frac{N}{F_s}\right)B^{-1}\left(B(f_l)+m\frac{B(f_h)-B(f_l)}{M+1}\right),\qquad 1\le m\le M \tag{2-28}$$

where f_l and f_h is the lowest and highest frequency (Hz) of the filter bank, F_s is the sampling frequency of the speech signal and the function $B(f)$ is the function to map the actual frequency to Mel frequency given in (2-24). The function $B^{-1}(b)$ is the inverse of the $B(f)$ given by

$$B^{-1}(b)=700\left(10^{b/2295}-1\right) \tag{2-29}$$

where b is the Mel frequency. It is noted that the boundary points $f(m)$ are uniformly spaced in the Mel scale. By replacing B and B^{-1} in (2-28) by (2-26) and (2-29), the equation can be rewritten as

$$f(m)=N\cdot\left[\frac{700+f_l}{F_s}\cdot\left(\frac{700+f_h}{700+f_l}\right)^{\frac{m}{M+1}}-\frac{700}{F_s}\right] \tag{2-30}$$

which can be used in programming. In general, M is equal to 20 for the speech signal with 8 kHz sampling frequency and 24 for 16 kHz sampling frequency. The Mel filter banks of the 8 kHz ($M=20$) and 16 kHz ($M=24$) are shown in Fig.2-10(a) and Fig.2-10(b) respectively. The region of spectrum below 1 kHz is processed by more filter banks since this region contains more information on the vocal tract such as the first formant. The nonlinear filter bank is employed to achieve both frequency and time resolution where the narrow band-pass filter at low frequencies enables harmonics to be detected and the longer band-pass filter at high frequencies allows for higher temporal resolution of bursts.

The Mel spectrum is derived by multiplied each FFT magnitude coefficient with the corresponding filter gain as

$$X_t(k) = |S_t(k)| H_m(k), \qquad 0 \le k < N-1 \tag{2-31}$$

and the results is accumulated and taken logarithm as

$$Y_t(m) = log \sum_{k=0}^{N-1} X_t(k), \qquad 0 \le m < M \tag{2-32}$$

which is robust to noise and spectral estimation errors. The reason of using the magnitude of $S_t(k)$ is that the information of phase is useless in speech recognition. The logarithm operation is utilized to reduce the component amplitudes at every frequency and to perform a dynamic compression in order to make the feature extraction less sensitive to variations in dynamics where the dynamics means the magnitude of the sound. Besides, the logarithm is applied to separate the excitation produced by the vocal tract and the filters that represents the vocal tract.

Since the log-magnitude spectrum $Y(m)$ is real and symmetric, the inverse Discrete Fourier Transform (IDFT) is reduced to the Discrete Cosine Transform (DCT) and applied to derive the Mel Frequency Cepstral Coefficients $c_t(i)$ as

$$c_t(i) = \sqrt{\frac{2}{M}} \sum_{m=1}^{M} Y(m) cos\left(\frac{i\pi}{M}\left(m - \frac{1}{2}\right)\right), \qquad i = 1, \cdots, L \tag{2-33}$$

where L is the number of cepstrum coefficients desired and $L \le M$. It is noted that the cepstrum is defined in the quefrency domain. The process of DCT successfully separates the excitation and the vocal tract, in other words, the low quefrencies, namely lower order of cepstrum, represents the slow changes of the envelope of the vocal tract and the high quefrencies, namely, higher order of cepstrum represents the periodic excitation. In general, 12 MFCCs (L=12) and the energy is adapted where the energy term computed by the log of the energy as

$$e_t = log \sum_{k=1}^{N} s_w(k)^2 \tag{2-34}$$

which often referred to as absolute MFCCs, and then the first and second-order

derivatives of these absolute coefficients are given

$$\Delta c_t(i) = \frac{\sum_{p=1}^{P} p\left(c_{t+p}(i) - c_{t-p}(i)\right)}{2\sum_{p=1}^{P} p^2}, \qquad i = 1, \cdots, L \tag{2-35}$$

and

$$\Delta^2 c_t(i) = \frac{\sum_{p=1}^{P} p\left(\Delta c_{t+p}(i) - \Delta c_{t-p}(i)\right)}{2\sum_{p=1}^{P} p^2}, \qquad i = 1, \cdots, L \tag{2-36}$$

which are useful to cancel the channel effect of the speech. In addition, the derivative operation is utilized to obtain the dynamic evolution of the speech signal, that is, the temporal information of the feature vector $c_t(i)$. If the value of P is too small, the dynamic evolution may not be caught; if the value P is too large, the derivatives have less meaning since two frames may describe different acoustic phenomena. In practice, the order of MFCC is often chosen as 39, including 12 MFCCs ($\{c(i)\}|_{i=1,2,\ldots,12}$), energy term ($e_t$) and their first-order derivatives ($\Delta\{c(i)\}|_{i=1,2,\ldots,12}$, $\Delta\{e_t\}$) and second-order derivatives ($\Delta^2\{c(i)\}|_{i=1,2,\ldots,12}$, $\Delta^2\{e_t\}$).

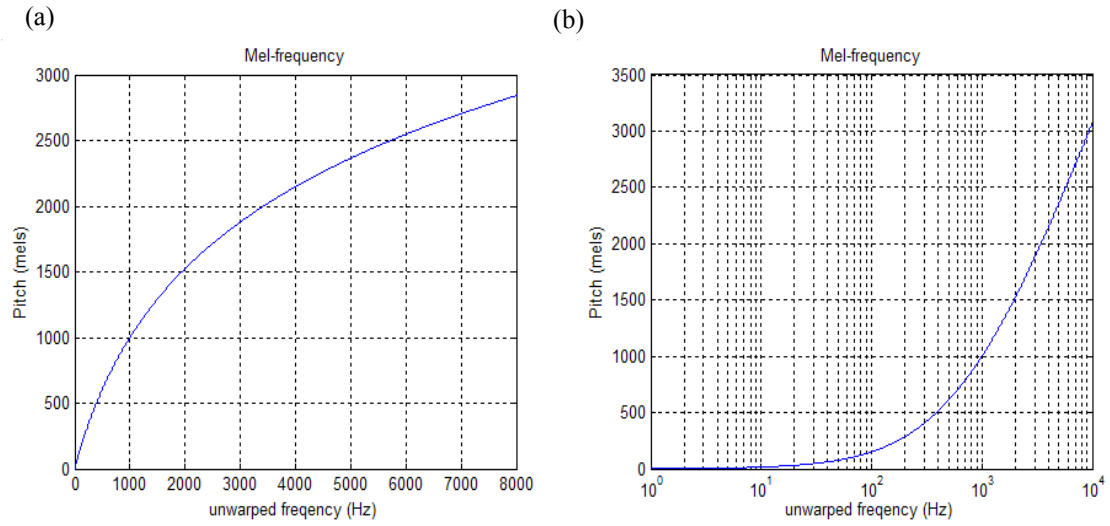


Fig.2- 9 Frequency Warping according to the Mel scale (a) linear frequency scale (b) logarithmic frequency scale

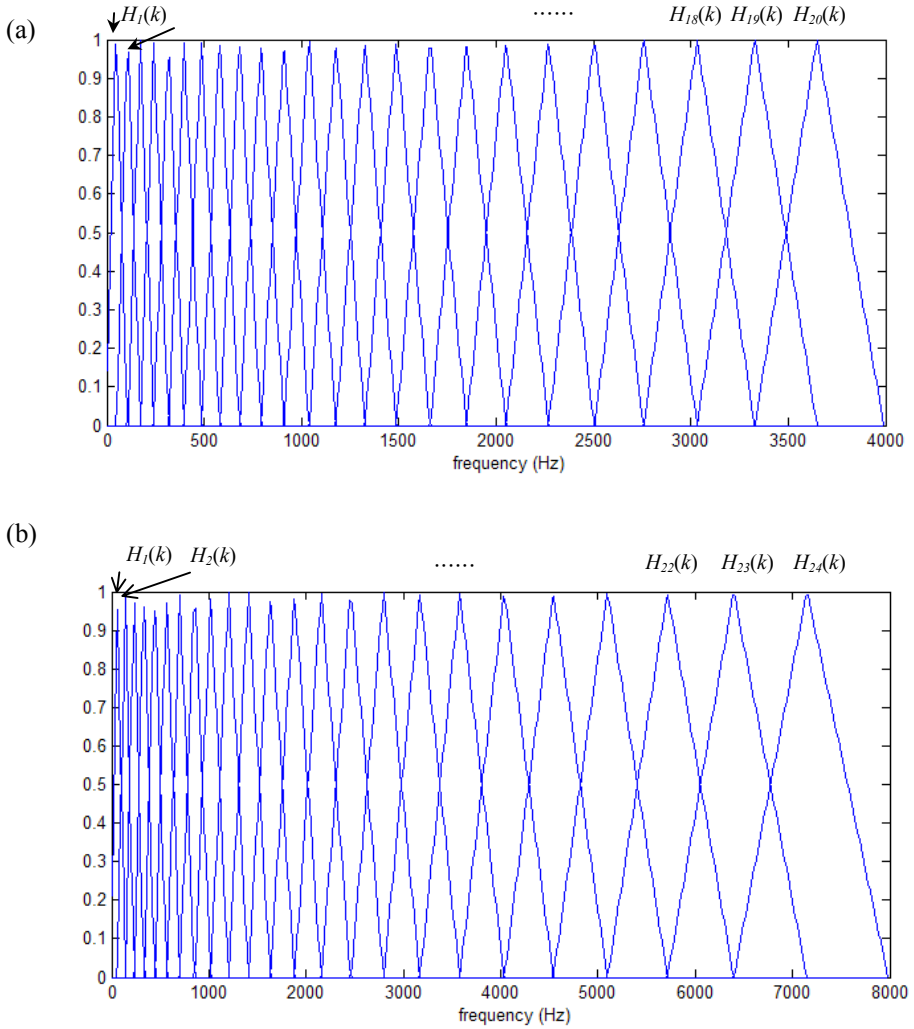


Fig.2-10 The Mel filter banks (a) F_s = 8 kHz and (b) F_s =16 kHz

2.5.3 Perceptual Linear Predictive (PLP) Analysis

The Perceptual Linear Predictive (PLP) analysis is first presented and examined by Hermansky in 1990 [4] for analyzing speech. This technique combines several engineering approximations of psychophysics of human hearing processes, including critical-band spectral resolution, the equal-loudness curve, and the intensity-loudness power law. As a result, the PLP analysis is more consistent with the human hearing. In addition, the PLP analysis is beneficial for speaker-independent speech recognition due to its computational efficiency and yielding a low-dimensional representation of

speech. The block diagram of the PLP method is shown in Fig.2.11, and each step will be described below. [12]

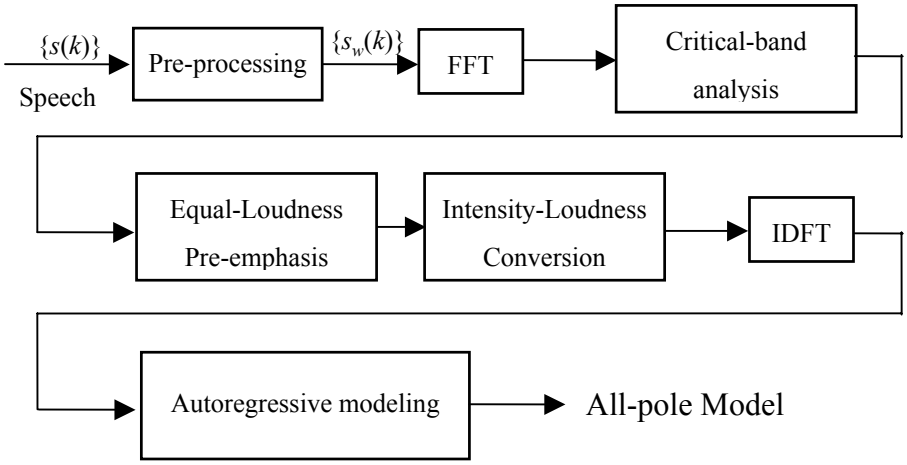


Fig.2-11 Scheme of obtaining Perceptual Linear Predictive coefficeints

Step I. Spectral analysis

The fast Fourier Transform (FFT) is first applied on the windowed speech segment ($s_w(k)$, for k=1,2,…N) into the frequency domain. The short- term power spectrum is expressed as

$$P(\omega) = [\mathrm{Re}(S_t(\omega))]^2 + [\mathrm{Im}(S_t(\omega))]^2 \tag{2-37}$$

where the real and imaginary components of the short-term speech spectrum are squared and added. There is an example in Fig.2-12 which shows the short-term speech signal and its power spectrum $P(\omega)$.

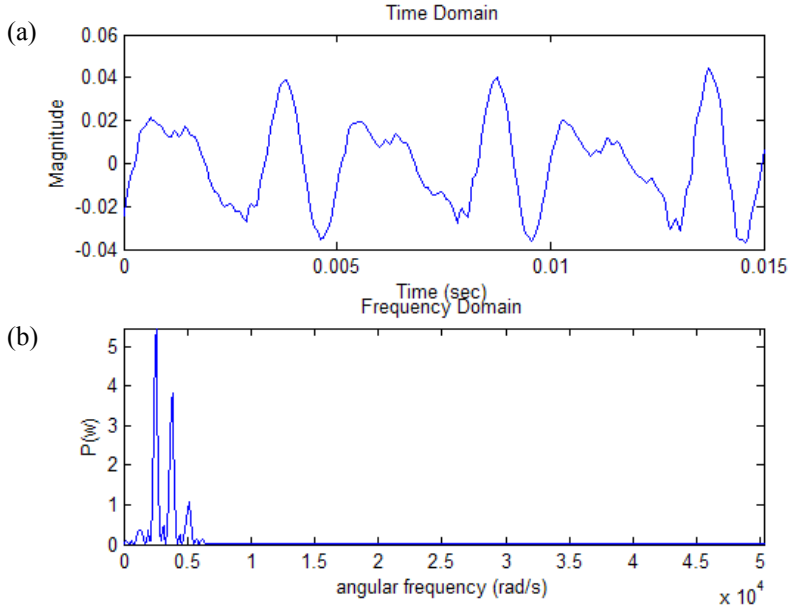


Fig.2-12 Short-term speech signal (a) in time domain and (b) power spectrum

Step II. Critical-band analysis

The power spectrum $P(\omega)$ is then warped along the frequency axis ω into the Bark scale frequency Ω as

$$\Omega(\omega) = 6\ln\left\{\frac{\omega}{1200\pi} + \sqrt{\left(\frac{\omega}{1200\pi}\right)^2 + 1}\right\} \tag{2-38}$$

where ω is the angular frequency in rad/sec, which is shown in Fig.2-13. The resulting power spectrum $P(\Omega)$ is then convoluted with the simulated critical-band masking curve $\Psi(\Omega)$ and get the critical-band power spectrum $\Theta(\Omega_i)$ as

$$\Theta(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} P(\Omega)\Psi(\Omega - \Omega_i), \quad i = 1,2,...,M \tag{2-39}$$

where M is number of Bark filter banks and the critical-band masking curve $\Psi(\Omega)$, shown in Fig.2-14, is given by,

$$\Psi(\Omega)=\begin{cases}0 & \text{for} \quad \Omega<-1.3\\ 10^{0.5(\Omega+0.5)} & \text{for} \quad -1.3\le\Omega\le-0.5\\ 1 & \text{for} \quad -0.5<\Omega<0.5\\ 10^{-1.0(\Omega-0.5)} & \text{for} \quad 0.5\le\Omega\le2.5\\ 0 & \text{for} \quad \Omega>2.5\end{cases} \tag{2-40}$$

where Ω is the Bark frequency just mentioned in (2-38). This step is similar to Mel filter banks processing of MFCC where the Mel filter banks are replaced by the analogous trapezoid Bark filter banks. The step between two banks is constant on the Bark scale, and the interval is chosen so that the filter banks must cover the whole analysis band. For example, 21 Bark filter banks, which cover from 0-Bark to 19.7-Bark in 0.985-Bark steps, are employed for analyzing speech signal of 16 kHz sampling frequency, shown in Fig.2-15. It is noted that 8 kHz is mapping to 19.687-Bark and the steps are usually chosen approximately 1-Bark. Fig.2-16 is the power spectrum after applying the Bark filter banks (M = 21) to the speech signal in Fig.2-12. The Bark filter banks and the Mel filter banks are both allocate more filters to the lower frequencies, where the hearing is more sensitive. Sometimes, the Bark filter banks are replaced with the Mel filter banks.

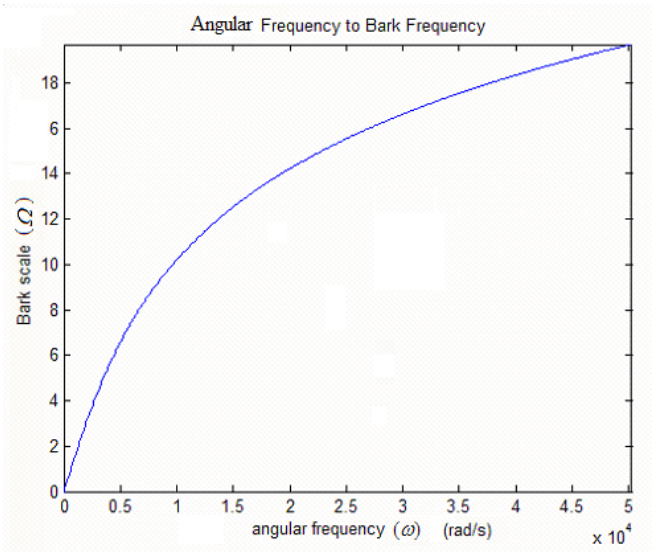


Fig.2-13 Frequency Warping according to the Bark scale

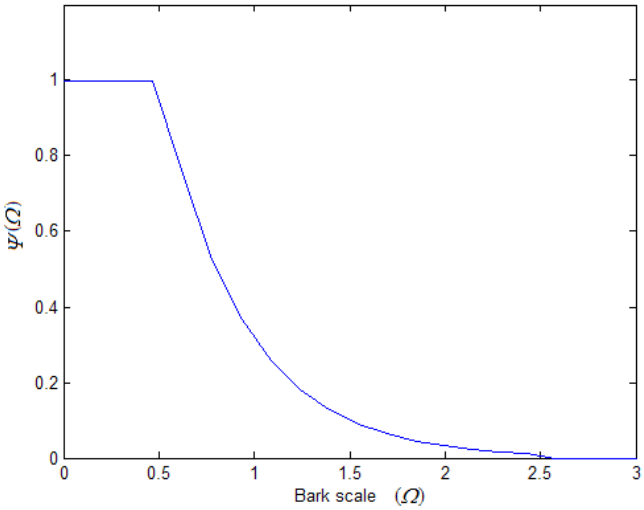


Fig.2-14 Critical-band curve

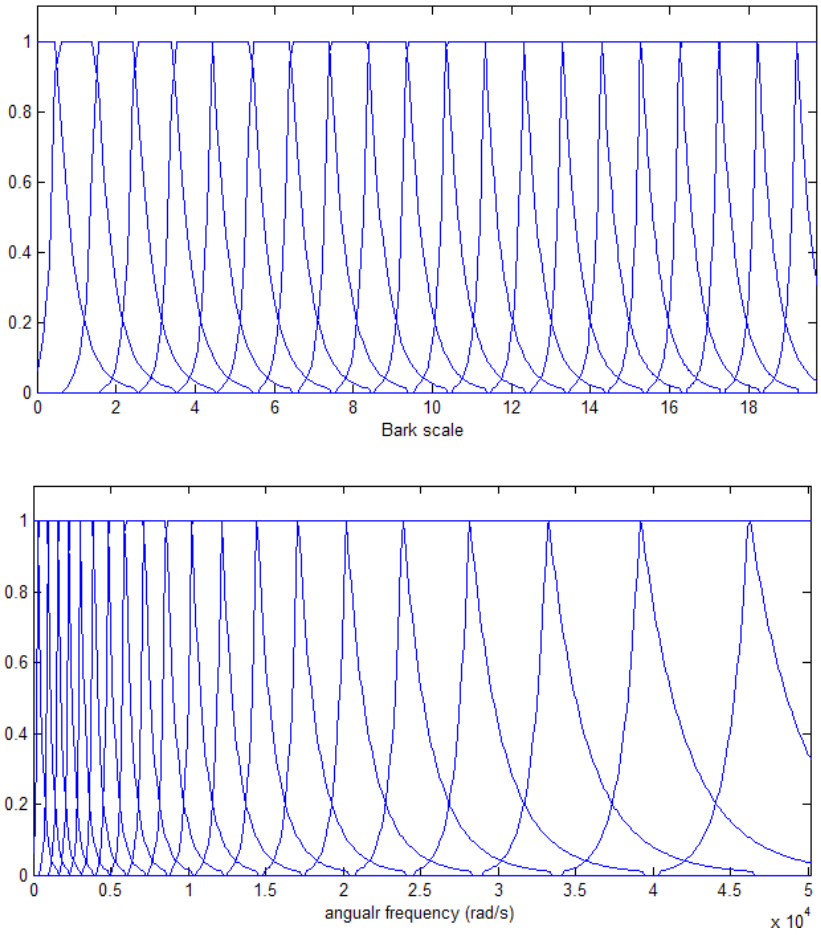


Fig.2-15 The Bark filter banks (a) in Bark scale (b) in angular frequency scale

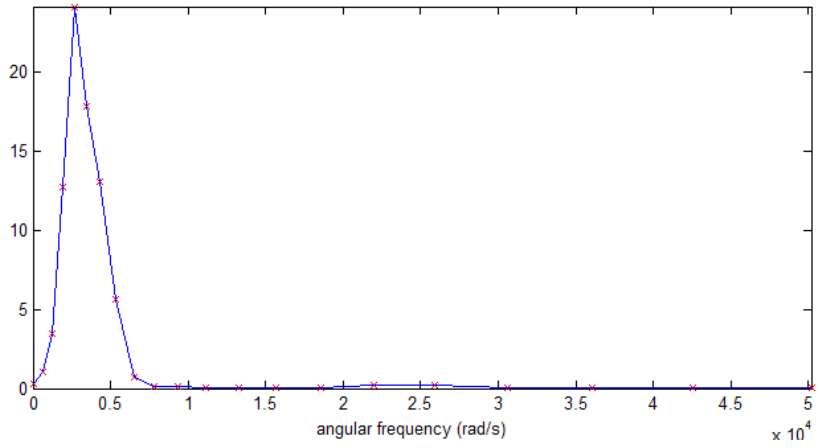


Fig.2-16 Critical-band power spectrum

Step III. Equal-loudness pre-emphasis

In order to compensate the unequal sensitivity of human hearing at different frequencies, the sampled power spectrum $\Theta(\Omega_i)$ obtained in the (2-39) is then pre-emphasis by the simulates equal loudness curve $E(\omega)$, expressed as

$$\Xi(\Omega_i) = E(\omega) \cdot \Theta(\Omega_i), \qquad i = 1,2,...,M \tag{2-41}$$

where the function $E(\omega)$ is given by

$$E(\omega) = \frac{(\omega^2 + 56.8\times10^6)\omega^4}{(\omega^2 + 6.3\times10^6)^2 \times (\omega^2 + 0.38\times10^9) \times (\omega^6 + 9.58\times10^{26})} \tag{2-42}$$

where E(ω) is a high pass filter. Then the value of the first and last samples are made equal to the values of their nearest neighbors, thus $\Xi(\Omega_i)$ begins and ends with two equal-valued samples. Fig.2-17 shows the power spectrum after equal-loudness pre-emphasis. From the Fig.2-17, the part of higher frequency in Fig.2-16 has been well compensated.

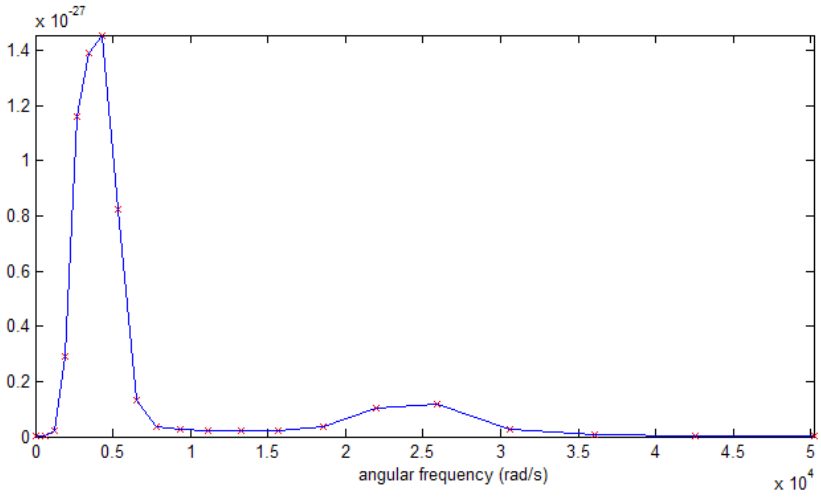


Fig.2-17 Equal loudness pre-emphasis

Step IV. Intensity-loudness power law

Since the nonlinear relation between intensity of the sound and its perceived loudness, the spectral compression is then utilized by using the power law of hearing given by

$$\Phi(\Omega_i) = \Xi(\Omega_i)^{0.33}, \qquad i = 1,2,\ldots,M \tag{2-43}$$

where a cubic root compensation of critical band energies is applied. This step can reduce the spectral-amplitude variation of the critical-band spectrum. It is noted that the log arithmetic is adopted in the process of MFCC.

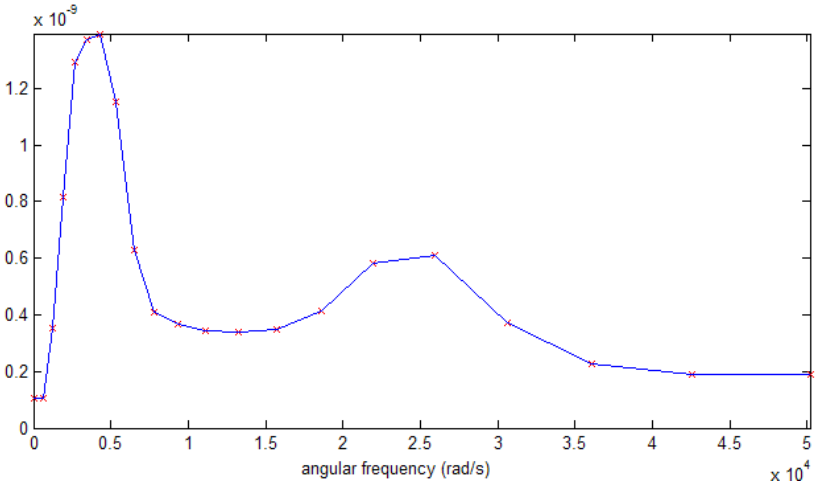


Fig.2-18 Intensity-loudness power law

Step V. Autoregressive modeling

The autocorrelation coefficients $r_s(n)$ are not computed in the time domain through (2-18) but is obtained as the inverse Fourier transform (IDFT) of the power spectrum $P(\omega)$ of the signal. The IDFT is better choice than the FFT here since only a few autocorrelation values are needed. If the order of the all pole model is equal to p, only the first p+1 autocorrelation values are used to solve the Yule-Walker equation. Then the standard Durbin-Levinson recursion is employed to compute the PLP coefficients.

Chapter 3

Speech Modeling and Recognition

During the past several years, Hidden Markov Model (HMM) [20][21][22] has become the most powerful and popular speech model used in ASR because of its wonderful ability of characterizing the speech signal in a mathematically tractable way and better performance comparing to other methods. The assumption of the HMM is that the data samples can be well characterized as a parametric random process, and the parameters of the stochastic process can be estimated in a precise and well-defined framework.

3.1 Introduction

In a typical HMM based ASR system, the HMM is proceeded after the feature extraction. The input of the HMM is the discrete time sequence of feature vectors, such as MFCCs, LPCs, etc. These feature vectors are customarily called observations, since these feature vectors represent the inforamtion observable from the incoming speech utterance. The observation sequence $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ is a set of the observations from time 1 to time *T*, where the time *t* is the frame index.

An Hidden Markov Model can be used to represnent a word (one, two, three, etc) , a syllable (“grand”, “fa”, “ther”, etc), a phone (/b/, /o/, /i/, etc), and so forth. The Hidden Markov Model is essentially structured by a state sequence $\boldsymbol{q}=\{q_1,q_2,\cdots,q_T\}$ where $q_t \in \{S_1,S_2,\cdots,S_N\}$, N is the total number of states and each state is generally associated with a multidimensional probability distribution. The states of HMM can

be viewed as collections of similar acoustical phenomena in an utterance. The total number of state N should be chosen well to represent these phenomena. In general, different number of state of HMM would lead to differnet recognition results [12].

For a particular state, an observation can be generated according to the associated probability distribution. This means that there is not a one-to-one correspondence between the observation and the state, and the state sequence cannot be determined unanimously by a given oberservation sequence. It is noticed that only the observation is visible, not the state. In other words, the model possesses hidden states and is named as the "Hidden" Markov Model.

3.2 Hidden Markov Model

Formally speaking, a Hidden Markov Model is defined as $\Lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, which includes the initial state distribution $\boldsymbol{\pi}$, state-transition probability distribution $\boldsymbol{A}$, and observation probability distribution $\boldsymbol{B}$. Each elements will be illustrated respectively as follows.

I. Initial state distribution $\boldsymbol{\pi}$

The initial state distribution is defined as $\pi = \{\pi_i\}$ in which

$$\pi_i = P(q_1 = S_i), \qquad 1 \le i \le N \tag{3-1}$$

where π_i is the probability that the initial state q_1 of the state sequence $\boldsymbol{q} = \{q_1, q_2, \cdots, q_T\}$ is S_i. Thus, the summation of the probability of all possible initial state is equal to 1, given as

$$\pi_1 + \pi_2 + \cdots + \pi_N = 1 \tag{3-2}$$

II. State-transition probability distibution $\boldsymbol{A}$

The state transition probability distribution $\boldsymbol{A}$ of an N-state HMM can be expressed as $\{a_{ij}\}$ or in the form of square matrix

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \tag{3-3}$$

with constant probability a_{ij}

$$a_{ij}=p\left(q_{t+1}=j \mid q_t=i\right), \qquad 1\le i,j\le N \tag{3-4}$$

representing the transition probability from state i at time t to state j at time $t+1$. Briefly, the transitions among the states are governed by a set of probabilities a_{ij}, called the transition probabilities, which are assumed not changing with time. It is noticed that the summation of all the probabilities from a particular state at time t to itself and the others at time $t+1$ should be equal to 1, i.e. the summation of all the entries in the i-th row is equal to 1, given as

$$a_{i1}+a_{i2}+\cdots+a_{iN}=1, \qquad i=1,2,...,N \tag{3-5}$$

For any state sequence $\boldsymbol{q}=\{q_1,q_2,\cdots,q_T\}$ where $\boldsymbol{q}_t\in\{S_1,S_2,\cdots,S_N\}$, the probability of $\boldsymbol{q}$ being generated by the HMM is

$$P(\boldsymbol{q}\mid \boldsymbol{A},\pi)=\pi_i\, a_{q_1q_2}a_{q_2q_3}\cdots a_{q_{T-1}q_T} \tag{3-6}$$

For example, the transition probability matrix of a three-state HMM can be expressed in the form as

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3-7}$$

where

$$a_{i1}+a_{i2}+a_{i3}=1, \qquad i=1,2,3 \tag{3-8}$$

for arbitrary time t. Fig.3-1 shows all the possible paths, labeled with transition probabilities between states, from time 1 to T. The structure without any constrain imposed on state transitions is called ergodic HMM. It is easy to find that the number of all possible paths $\left(N^2\right)^{T-1}$ (in this case $N=3$) would greatly increase as time increasing.

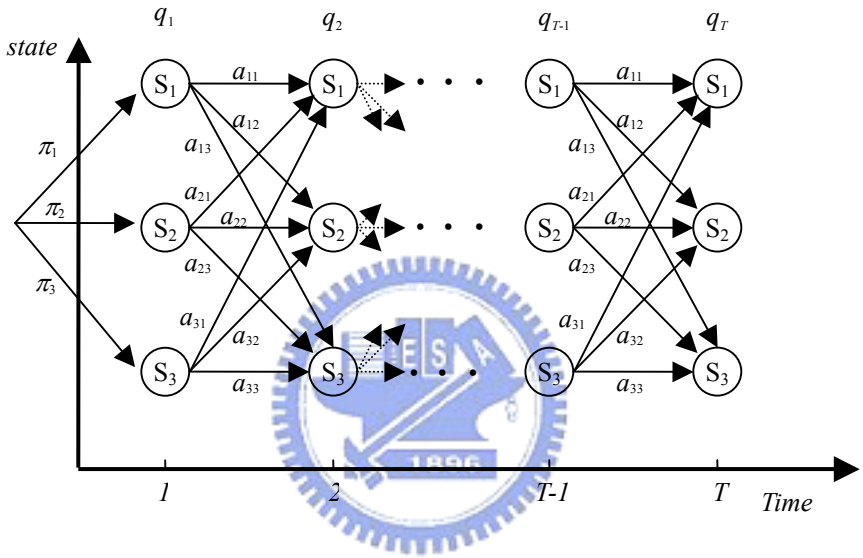


Fig.3-1 Three-state HMM

A left-to-right HMM (namely Bakis model) with the elements of the state-transition probability matrix

$$a_{ij}=0, \qquad \text{for} \quad j<i \tag{3-9}$$

is adopted in general cases to simplify the model and reduce the computation time. The main conception of a left-to-right HMM is that the speech signal varies with time from left to right, that is, the acoustic phenomena change sequentially and the first state must be S_1. There are two general types of left-to-right HMM, shown in Fig.3-2.

By using a three-state HMM as an example, the transition probability matrix $\boldsymbol{A}$ with left-to-right and one-skip constrain, shown in Fig.3-3, can be express as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \tag{3-10}$$

where $\boldsymbol{A}$ is an upper-triangular matrix with $a_{21} = a_{31} = a_{32} = 0$. Fig.3-4 shows all possible paths between states of a three-state left-to-right HMM from time 1 to time *T*. If no skip is allowed, the transition probability matrix $\boldsymbol{A}$ can be express as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \tag{3-11}$$

where the element a_{13} in (3-7) is replaced by zero. Similarly, Fig.3-5 shows all possible paths between states of a no-skip three-state HMM from time 1 to time *T*.

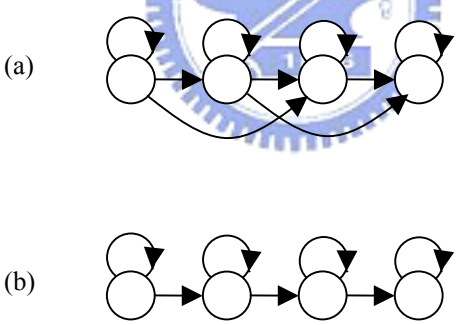


Fig.3-2 Four-state left-to-right HMM with (a) one skip and (b) no skip

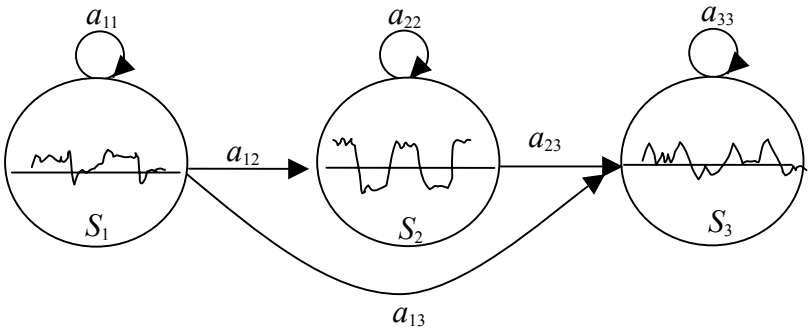


Fig.3-3 Typical left-to-right HMM with three states

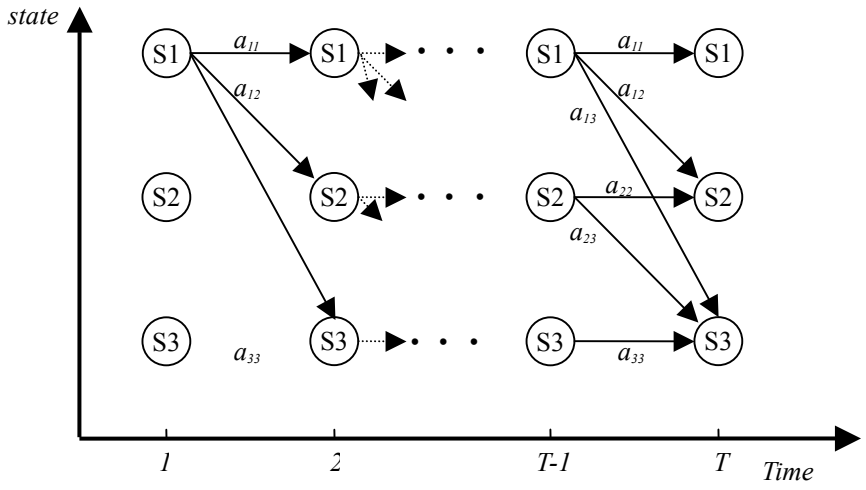


Fig.3-4 Three-state left-to-right HMM with one skip

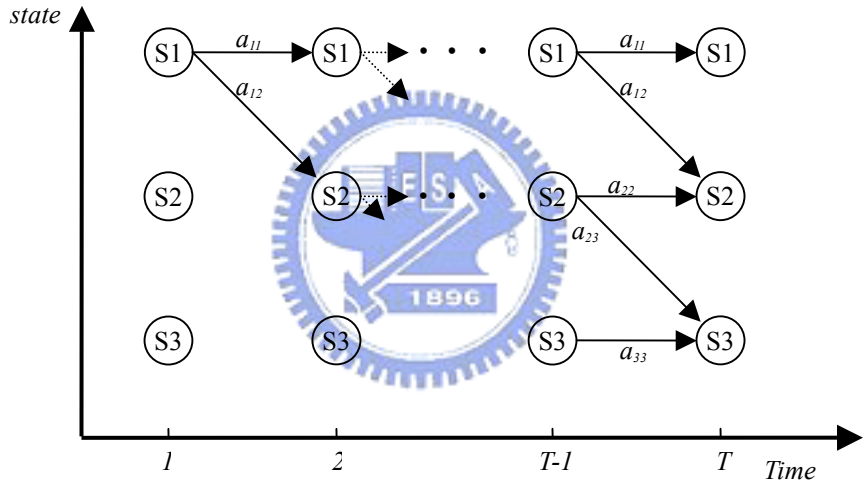


Fig.3-5 Three-state left-to-right HMM with no skip

III. Observation probability distribution ***B***

Since the state sequence $\boldsymbol{q}$ is not observable, each observation $\boldsymbol{o}_t$ can be envisioned as being produced with the system in state q_t. Assume that the production of $\boldsymbol{o}_t$ in each possible state S_i is stochastic, where i=1, 2,…, N, and is characterized by a set of observation probability functions $\boldsymbol{B} = \{b_j(\boldsymbol{o}_t)\}$ where

$$b_j(\boldsymbol{o}_t) = P(\boldsymbol{o}_t \mid q_t = S_j), \qquad j = 1,2,...,N \tag{3-12}$$

which discribes the probability of the observation $\boldsymbol{o}_t$ being produced with respect to state j. If the distribution of the observations are continuous and infinite, the finite mixture of Gaussian distributions, that is, a weighted sum of M Gaussian distributions is used, expressed as

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{M} w_{jm}\mathcal{N}(\boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}, \boldsymbol{o}_t)$$
$$= \sum_{m=1}^{M} w_{jk}\left[\frac{1}{(\sqrt{2\pi})^L |\boldsymbol{\Sigma}_{jm}|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm})^{\mathrm{T}} \boldsymbol{\Sigma}_{jm}^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm})\right)\right] \quad (3\text{-}13)$$

where $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ indicates the mean vector and the covariance matrix of the m-th mixture component in state S_j. The observations are assumed to be independent to each other, the covariance matrix can be reduced to a diagonal form $\boldsymbol{\Sigma}_{jm}$ as

$$\boldsymbol{\Sigma}_{jm} = \begin{bmatrix} \sigma_{jm}(1) & 0 & \cdots & 0 \\ 0 & \sigma_{jm}(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{jm}(L) \end{bmatrix} \quad (3\text{-}14)$$

or simplified as a vector with L-dimension as

$$\boldsymbol{\Sigma}_{jk} = \begin{bmatrix} \sigma_{jk}(1) & \sigma_{jk}(2) & \cdots & \sigma_{jk}(L) \end{bmatrix} \quad (3\text{-}15)$$

where L is the dimension of the observation $\boldsymbol{o}_t$. The mean vector can be expressed as

$$\boldsymbol{\mu}_{jm} = \begin{bmatrix} \mu_{jm}(1) & \mu_{jm}(2) & \cdots & \mu_{jm}(L) \end{bmatrix} \quad (3\text{-}16)$$

Then, the observation probability function $b_j(\boldsymbol{o}_t)$ can be written as

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{M} w_{jm}\left[\frac{1}{(2\pi)^{\frac{L}{2}}\left[\prod_{l=1}^{L}\sigma_{jm}(l)\right]^{\frac{1}{2}}} \prod_{l=1}^{L} exp\left(-\frac{(\boldsymbol{o}_t(l) - \mu_{jm}(l))^2}{2\sigma_{jm}(l)}\right)\right] \quad (3\text{-}17)$$

As for the weighting coefficient w_{jk}, it must satisfying

$$\sum_{m=1}^{M} w_{jm} = 1 \quad (3\text{-}18)$$

where w_{jk} is non-negative value.

Fig.3-6 shows that the probabilities of the observations sequence $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \boldsymbol{o}_4\}$ generated by state sequence $q = \{q_1, q_2, q_3, q_4\}$ are $b_{q_1}(\boldsymbol{o}_1)$, $b_{q_2}(\boldsymbol{o}_2)$, $b_{q_3}(\boldsymbol{o}_3)$, $b_{q_4}(\boldsymbol{o}_4)$, respectively.

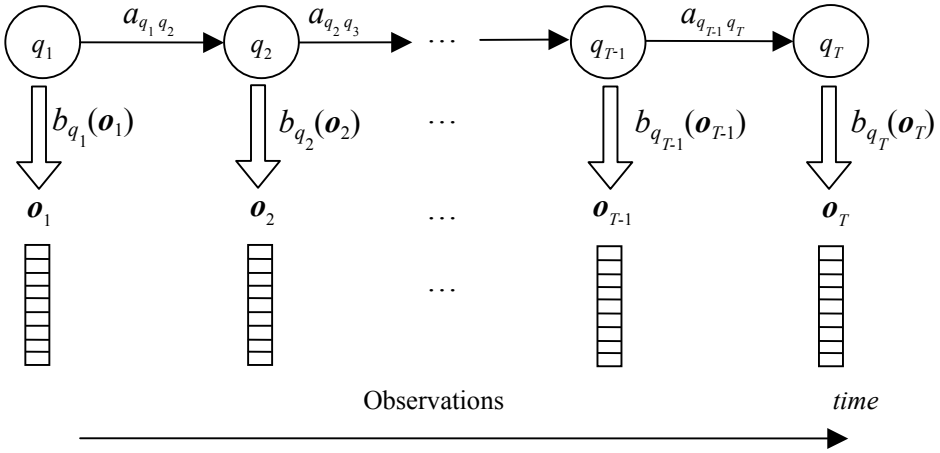


Fig.3-6 Scheme of probability of the observations

3.3 Training Procedure

Given a HMM $\boldsymbol{\Lambda}=\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ and a set of observations $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_T\}$, the purpose of training the HMMs is to adjust the model parameters so that the likelihood $\boldsymbol{P}(\boldsymbol{O}\,|\,\Lambda)$ is locally maximized by using iterative procedure. The modified k-means algorithm [19] and Viterbi algorithm are employed in the process of obtaing initial HMMs. The Baum-Welch algorithm (or called the forward-backward algorithm) is performed to train the HMMs. Before applying the training algorithm, prepareation work of the corpus and HMM is required prior to the trainging procedure as below

I. A set of speech data and their associated transcriptions should be prepared, and the speech data must be transformed to the a series of feature vectors (LPC, RC, LPCC, MFCC, PLP, etc).

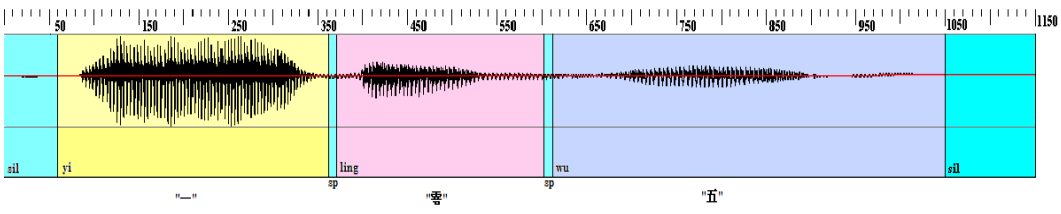
II. The number of states and the number of mixtures in a HMM must be determined, according to the degree of variations in the unit. In general, 3~5 states and 6~8 states are used for representing the English phone and Mardarin Chinese phone, respectively.

It is noted that the features are the the observations of the HMM, and these observations and the transcriptions are then utilized to train the HMMs.

The training procedure can be divided into two manners depending on whether the sub-word-level segment information, or called the boundary information, is available, that is labeled with boundary manually. If the segment information is available, such as Fig.3-7(a), the estimation of the HMM parameter would be easier and more precise; otherwise, training with no segment information would cost more computation time to re-align the boundary and re-estimate the HMM, in addition, the HMM often performs not as good as the one with well-segment information. The transcription and boundary condition should be saved in text files, such as the form in Fig.3-7(b)(c).

It is noted that if the speech doesn't have segment information, it is also necessary to get the transcription and save it before training. The block diagram of the training procedure is shown in Fig.3-8. The main difference between training the HMM with boundary information and training the HMM without boundary information is on the processing of creating the initialized HMM. Then, the following section will divided into two parts to present the details of creating the initialized HMM.

(a)



(b)

```
0 60 sil
60 360 yi
360 370 sp
370 600 ling
600 610 sp
620 1050 wu
1050 1150 sil
```

(c)

```
sil
yi
sp
ling
sp
wu
sil
```

Fig.3-7 (a) Speech labeled with the boundary and transcription save as text file (b) with and (c) without boundary information

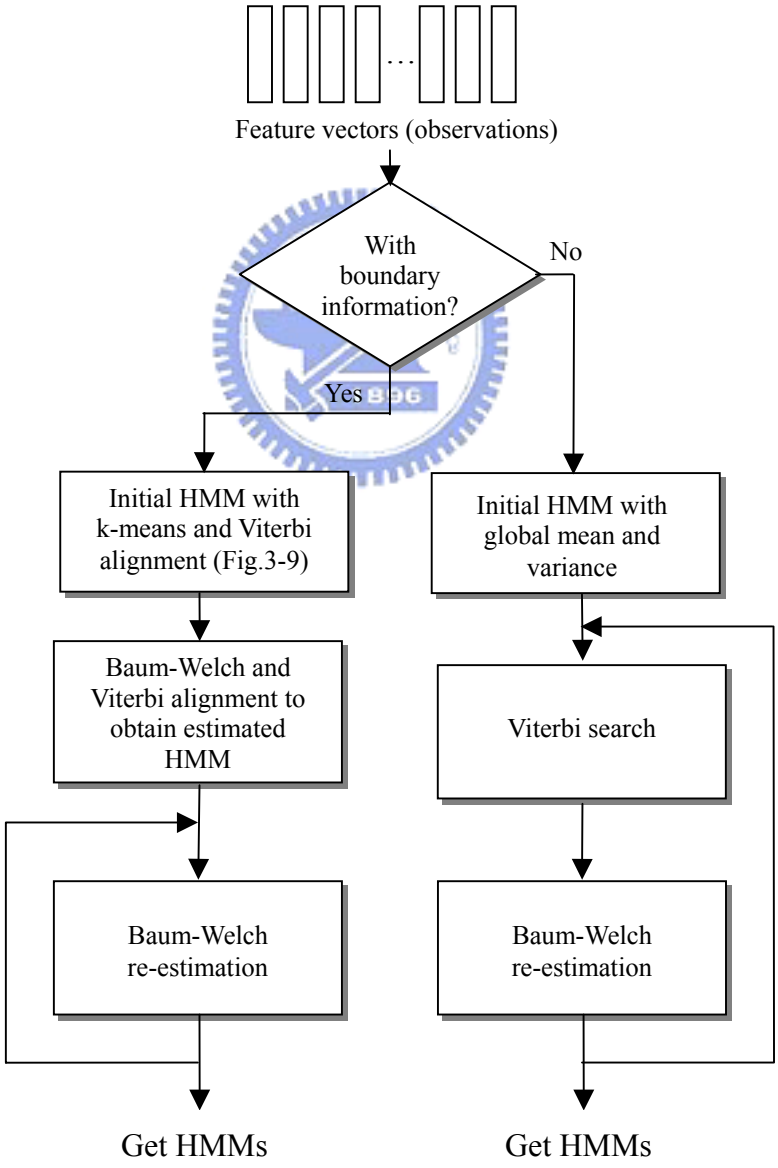


Fig.3-8 Training procedure of the HMM

I. Boundary information is available

The procedure of creating the initialized HMMs is shown in Fig.3-9, Fig 3-10. The modified k-means algorithm and the viterbi algorithm are utilized in training iteration. On the first iteration, the training data of a specific model are uniformly divided into N segments, where N is the number of states of HMM, and the successive segments are associated with successive states. Then, the HMM parameters π_i and a_{ij} can be estimated first by

$$\pi_j = \frac{\text{number of observations in state } j \text{ at time} = 1}{\text{number of observations at time} = 1} \tag{3-19}$$

$$a_{ij} = \frac{\text{number of transitions from state } i \text{ to state } j}{\text{number of transitions from state } i} \tag{3-20}$$

3.3.1 Midified k-means algorithm

For continuous-density HMM with M Gaussian mixtures per state, the modified k-means [13][14] are used for cluster the observations $\boldsymbol{O}$ into a set of M clusters which are associated to the number of mixtures in a state, shown in Fig.3-9. Let the i-th cluster of a m-cluster set at the k-th iteration denote as $\omega_{m,i}^{k}$ where $i = 1,2,\ldots,m$ and $k = 1,2,\ldots,k_{\max}$ with $k_{\max}$ being the maximum allowable iteration count. $Y(\omega)$ is the representive pattern for cluster ω. the number of clusters in the current iteration and i is the iteration counter in classification process. The modified k-means algorithm is given by

(i) Set m=1, k=1 and i=1; $\omega_{m,i}^{k} = \boldsymbol{O}$ and compute the mean $Y(\boldsymbol{O})$ of the entire training set $\boldsymbol{O}$.

(ii) Classify the vectors by minimum distance principle. Accumulate the total

intracluster distance for each cluster $\omega_{m,i}^{k}$ denoted as Δ_i^k. If none of the following conditions meet then back to (ii) and k=k+1.

a. $\omega_{m,i}^{k+1} = \omega_{m,i}^{k}$, for all i=1,2,…,m

b. k meets the preset maximum allowable number of iterations.

c. The change in the total accumulated distance is below the preset threshold Δ_{th}.

(iii) Record the mean and the covariance of the m-cluster,. If m is reached the number of mixtures M, then stop, else, go to (iv).

(iv) Split the mean of the cluster that has largest intracluster distance and m=m+1, reset k and go to (ii).

From the modified k-means, the observations are clustered into M groups where M is the number of mixtures in a state. The parameters can be estimated by

$$w_{jm} = \frac{\text{number of observations classified in cluster } m \text{ in state } j}{\text{number of observations classified in state } j} = \frac{N_{jm}}{N_j} \tag{3-21}$$

$$\mu_{jm} = \text{mean of the observations classifiedin cluster } m \text{ in state } j = \frac{1}{N_{jm}} \cdot \sum_{n=1}^{N_{jm}} \boldsymbol{o}_n \tag{3-22}$$

$$\boldsymbol{\Sigma}_{jm} = \text{covariance matrix of the observations classified in cluster } m \text{ in state } j$$

$$= \frac{1}{N_{jm}} \cdot \sum_{n=1}^{N_{jm}} \left(\boldsymbol{o}_n - \hat{\boldsymbol{\mu}}_{jm}\right)\left(\boldsymbol{o}_n - \hat{\boldsymbol{\mu}}_{jm}\right)^{\mathrm{T}} \tag{3-23}$$

where $\boldsymbol{o}_n$ $(1 \le n \le N_{jm})$ is the observations classified in cluster m in state j. Then the HMM parameters is all updated.

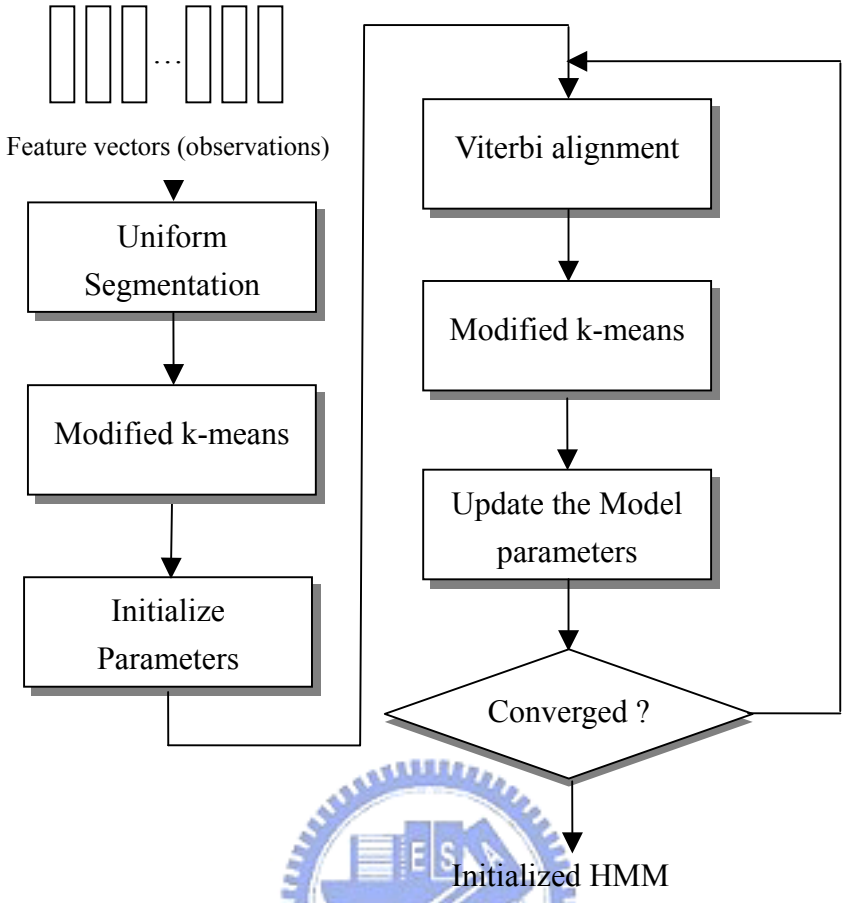


Fig.3-9 The block diagram of creating the initialized HMM

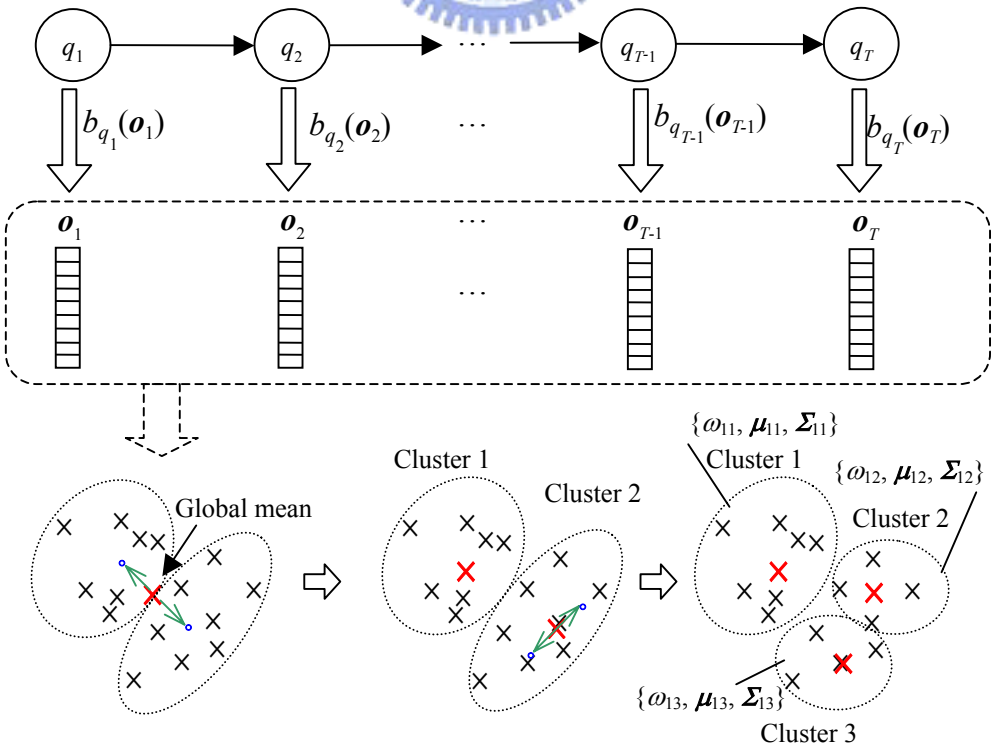


Fig.3-10 Modified k-means

3.3.2 Viterbi Search

Except for the first estimation of the HMM, the uniform segmentation is replaced by Viterbi alignment, viz Viterbi search, which is applied to find the optimal state sequence $\boldsymbol{q}=\{q_1, q_2,\ldots,q_T\}$ where model $\boldsymbol{\Lambda}$ and the observations sequences $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2,\cdots, \boldsymbol{o}_T\}$ are given. By the Viterbi alignment, each observation will be re-align to the state so that the new sate sequence $\boldsymbol{q}=\{q_1, q_2,\ldots,q_T\}$ maximizes the probability of generating the observation sequence $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2,\cdots, \boldsymbol{o}_T\}$.

By taking logarithm of the model parameters, the Viterbi algorithm [14] can be impletement with only N^2T additions and wihout any multiplications. Define $\delta_t(i)$ be the highest probability along the singal path at time t, expressed as

$$\delta_t(i)=\max_{\boldsymbol{q}=\{q_1,q_2,\ldots,q_{t-1}\}} \boldsymbol{P}\left(q_1,q_2,\ldots,q_{t-1},q_t=i,\boldsymbol{o}_1,\boldsymbol{o}_2,\ldots,\boldsymbol{o}_t \mid \boldsymbol{\Lambda}\right) \tag{3-24}$$

and by induction we can obtain

$$\delta_{t+1}(j)=\left[\max_i \delta_t(i)a_{ij}\right]b_j(\boldsymbol{o}_{t+1}) \tag{3-25}$$

which is shown in Fig.3-11.

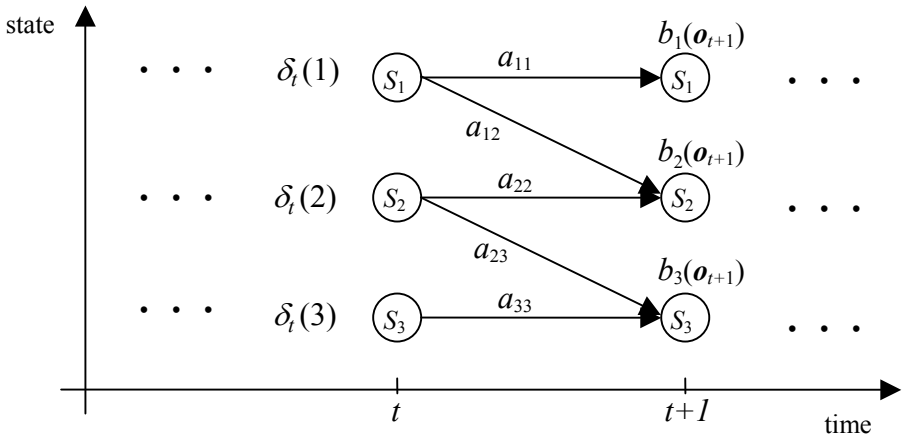


Fig.3-11 Maximization the probability of generating the observation sequence

The Viterbi algorithm is expressed as follows

(i) Preprocessing

$$\tilde{\pi}_i = \log(\pi_i), \qquad 1 \le i \le N \tag{3-26}$$

$$\tilde{b}_i(\boldsymbol{o}_t) = \log(b_i(\boldsymbol{o}_t)), \quad 1 \le i \le N, \ 1 \le t \le T \tag{3-27}$$

$$\tilde{a}_{ij}(\boldsymbol{o}_t) = \log(a_{ij}), \qquad 1 \le i \le N \tag{3-28}$$

(ii) Initialization

$$\tilde{\delta}_1(i) = \log(\delta_1(i)) = \tilde{\pi}_i + \tilde{b}_i(\boldsymbol{o}_1), \qquad 1 \le i \le N \tag{3-29}$$

$$\psi_1(i) = 0, \qquad 1 \le i \le N \tag{3-30}$$

where the array $\psi_i(j)$ is used for backtracking.

(iii) Recursion

$$\tilde{\delta}_t(j) = \log(\delta_t(j)) = \max_{1 \le i \le N}\left[\tilde{\delta}_t(j) + \tilde{a}_{ij}\right] + \tilde{b}_{ij}, \qquad 2 \le t \le T, \ 1 \le j \le N \tag{3-31}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N}\left[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}\right], \qquad 2 \le t \le T, \ 1 \le j \le N \tag{3-32}$$

(iv) Termination

$$\tilde{P}^* = \max_{1 \le i \le N}\left[\tilde{\delta}_T(i)\right] \tag{3-33}$$

$$q_T^* = \arg\max_{1 \le i \le N}\left[\tilde{\delta}_T(i)\right] \tag{3-34}$$

(v) Backtracking

$$q_t^* = \psi_{t+1}\left(q_{t+1}^*\right), \quad t = T-1, T-2, \ldots, 1 \tag{3-35}$$

From the above, the state sequence $\boldsymbol{q}$ which maximizes $\tilde{P}^*$ implies an alignment of observations with states.

The above procedures, viterbi alignment, modified k-means and parameter estimation, are applied until $\tilde{P}^*$ converges. After obtaining the initialized HMM, the Baum-Welch algorithm and the Viterbi search are then applied to get the first

estimation of the HMM. Finally, the Baum-Welch algorithm is performed repeatedly to reestimate the HMMs simultaneously. The Baum-Welch algorithm will be introduced later.

II. Boundary information is not available

In this case, all the HMMs are initialized to be identical and the mean and the variance of the all states are set to be eqaul to the global mean and variance. As for the initial state distribution $\boldsymbol{\pi}$ and state-transition probability distribution $\boldsymbol{A}$, there is no information to compute these parameters; hence, the parameters $\boldsymbol{\pi}$ and $\boldsymbol{A}$ should be set arbitrarily. From the above process, the initialized HMMs are then generated.

Afterwards, the processes for reestimating HMMs are resemble the reestimated processes for boundary information, that is using the Baum-Welch algorithm. After reestimating by Baum-Welch algorithm, the Viterbi search is also needed to re-align the boundaries of the sub-word. This step is different to the training procedure which already have boundary information. The next section will introduce the Baum-Welch algorithm employed in the HMM training processing.

3.3.3 Baum-Welch reestimation

The Baum-Welch algorithm, known as the forward-backward algorithm is the core of training HMM. Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_t, q_t = i \mid \boldsymbol{\Lambda}) \tag{3-36}$$

that means the probability of the state i at time t which having generating the observation sequence $\boldsymbol{o}_1$, $\boldsymbol{o}_2$,…, $\boldsymbol{o}_t$ given the model $\boldsymbol{\Lambda}$, shown in Fig.3-12. The forward

variable is obtained inductively by

Step 1. Initialization:

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \le i \le N \tag{3-37}$$

Step II. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{o}_{t+1}), \quad 1 \le j \le N, \ 1 \le t \le T-1 \tag{3-38}$$

In similar way, the backward variable is defined as

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, ..., \boldsymbol{o}_T \mid q_t = i, \boldsymbol{\Lambda}) \tag{3-39}$$

that represent the probability of the observation sequence from $t+1$ to the end given state i at time t and the model $\boldsymbol{\Lambda}$, shown in Fig.3-12. The backward variable is obtained inductively by

Step I. Initialization:

$$\beta_T(i) = 1, \quad 1 \le i \le N \tag{3-40}$$

Step II. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) b_j(\boldsymbol{o}_{t+1}) a_{ij}, \quad 1 \le i \le N, \ t = T-1, T-2, ..., 1 \tag{3-41}$$

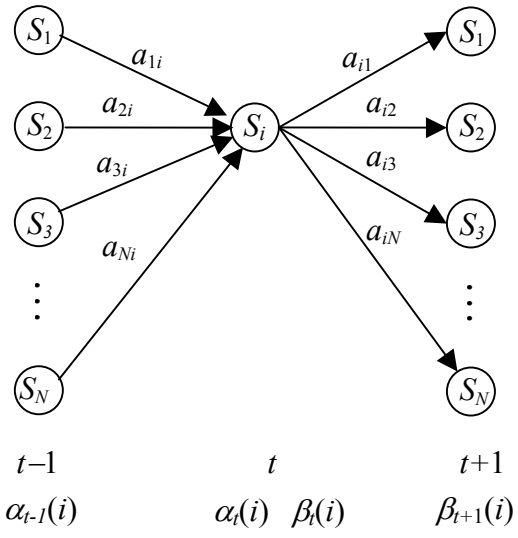


Fig.3-12 Forward variable and backward variable

Besides, three variables should be defined, that is $\xi_t(i,j)$ and the posteriori probability $\gamma_t(i)$ and $\gamma_t(i,j)$. The variable $\xi_t(i,j)$ is defined as

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid \boldsymbol{O}, \boldsymbol{\Lambda}) \tag{3-42}$$

which is the probability of being in state i at time t and state j at time t+1. The posteriori probability $\gamma_t(i)$ is expressed as

$$\gamma_t(i) = \boldsymbol{P}(q_t = S_i \mid \boldsymbol{O}, \boldsymbol{\Lambda}) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3-43}$$

which is the probability being in state i at time t. The variable $\gamma_t(i,j)$ is defined as

$$\gamma_t(i,k) = P(q_t = S_i, m_t = k \mid \boldsymbol{O}, \boldsymbol{\Lambda})$$

which represent the probability of being in state i at time t with the k-th mixture component accounting for $\boldsymbol{o}_t$.

The HMM parameter $\boldsymbol{A}$, π can be re-estimated by using the variables mentioned above as

$$\pi_i = \text{expected number of times in state } S_i \text{ at time } t = 1 = \gamma_1(i) \tag{3-44}$$

$$a_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i,j)} \tag{3-45}$$

$$w_{jk} = \frac{\text{expected number of times in state } S_j \text{ and mixture } k}{\text{expected number of times in state } S_j}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(i,k)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(i,k)} = \frac{\sum_{t=1}^{T} \gamma_t(i,k)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3-46}$$

$$\boldsymbol{\mu}_{jk} = \text{mean of the observations at state } S_j \text{ and mixture } k$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(i,k) \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{3-47}$$

$$\boldsymbol{\Sigma}_{jk} = \text{covariance matrix of the observations at state } S_j \text{ and mixture } k$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(i,k)(\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})(\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{3-48}$$

where

$$\xi_t(i,j) = \frac{P(\boldsymbol{q}_t = S_i, q_{t+1} = S_j, \boldsymbol{O} \mid \boldsymbol{\Lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\Lambda})} = \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{P(O \mid \boldsymbol{\Lambda})}$$
$$= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)} \tag{3-49}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{3-50}$$

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{s=1}^{N} \alpha_t(s)\beta_t(s)} \right] \left[\frac{w_{jk} b_{jk}(\boldsymbol{o}_t)}{\sum_{k=1}^{M} w_{jk} b_{jk}(\boldsymbol{o}_t)} \right] \tag{3-51}$$

From the statistical viewpoint of estimating HMM by Expectation-Maximization (EM) algorithm, the equations for estimating the parameters are the same as the equations derived from Baum-Welch algorithm. Besides, it has been shown that the likelihood function will converge to a critical point after iterations and the Baum-Welch algorithm leads to a local maximum only due to the complexity of the likelihood function.

3.4 Recognition Procedure

Given the HMMs and the observation sequence $\boldsymbol{O}=\{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_T\}$, the recognition stage is to compute the probability $P(\boldsymbol{O}|\boldsymbol{\Lambda})$ by using an efficient method, forward-backward procedure. This method has been introduced in the training stage. Recall the forward variable $\alpha_t(i)$ is defined as

$$\begin{aligned}\alpha_{t+1}(j) &= P(\boldsymbol{o}_1, \boldsymbol{o}_2, ...\boldsymbol{o}_t, q_t = S_i \mid \Lambda) \\ &= \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{o}_{t+1}), \qquad 1 \le i \le N\end{aligned} \tag{3-52}$$

and the backward variable $\beta_t(i)$

$$\begin{aligned}\beta_t(i) &= P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, ..., \boldsymbol{o}_T \mid q_t = i, \boldsymbol{\Lambda}) \\ &= \sum_{j=1}^{N} \beta_{t+1}(j) b_j(\boldsymbol{o}_{t+1}) a_{ij}, \quad 1 \le i \le N\end{aligned} \tag{3-53}$$

given the initial conditions

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \le i \le N \tag{3-54}$$

$$\beta_T(i) = 1, \qquad 1 \le i \le N \tag{3-55}$$

where N is the number of states. The probability of being in state i at time t is expressed as

$$P(\boldsymbol{O}, q_t = S_i \mid \boldsymbol{\Lambda}) = \alpha_t(i)\beta_t(i) \tag{3-56}$$

such as the total probability $P(\boldsymbol{O}|\boldsymbol{\Lambda})$ is then obtained by

$$P(\boldsymbol{O} \mid \boldsymbol{\Lambda}) = \sum_{i=1}^{N} P(\boldsymbol{O}, q_t = S_i \mid \boldsymbol{\Lambda}) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{3-57}$$

which is employed in the speech recognition stage.

Chapter 4

Experimental Results

Several speaker-independent recognition experiments are shown in this chapter. The effect and performance of different front-end techniques are discussed in the experimental results. The corpus will be described in section 4.1. The experiments are divided into two parts, including the monophone-based HMM and the syllable-based HMM. The experimental results will be shown in section 4.2, and 4.3, respectively.

4.1 Corpus

The corpora employed in this thesis are TCC-300 provided by the Associations of Computational Linguistics and Chinese Language Processing (ACLCLP) and the connected-digits database provided by the Speech Processing Lab of the Department Communication Engineering, NCTU. These corpora are introduced as below.

4.1.1 TCC-300

In the speaker-independent speech recognition experiments, the TCC-300 database from the Associations of Computational Linguistics and Chinese Language Processing (ACLCLP) was used for monophone-based HMM training. TCC-300 is a collection of microphone speech databases produced by National Taiwan University (NTU), National Chiao Tung University (NCTU) and National Cheng Kung University (NCKU). In this thesis, the training corpus uses the speech databases produced by National Chiao Tung University.

The speech signal is recording under the following conditions, listed in Table 4-1. The speech is saved in the MAT file format, which is a format for recording the

speech waveform in PCM format and, in addition, recording the condition of the environment and the speaker in detail by adding extra 4096 bytes file header into the PCM.

Table 4-1 The recording environment of the TCC-300 corpus produced by NCTU

File Format	MAT
Microphone	Computer headsets VR-2560 made by Taiwan Knowles
Sound card	Sound Blaster 16
Sampling rate	16 kHz
Sampling format	16 bits
Speaking style	read

The database provided by NCTU is comprised of paragraphs spoken by 100 speakers (50 males and 50 females). Each speaker read 10-12 paragraphs. The articles are selected from the balanced corpus of the Academia Sinica and each article contains several hundreds of words. These articles are then divided into several paragraphs and each paragraph includes no more than 231 words. Table 4-2 shows the statistics of the databases

Table 4-2 The statistics of the database TCC-300 (NCTU)

	Males	**Females**	**Total**
Amounts of speakers	50	50	100
Amounts of syllables	75059	73555	148614
Amounts of Files	622	616	1238
Time (hours)	5.98	5.78	11.76
Maximum words in a paragraph	229	131	-
Minimum words in a paragraph	41	11	-

4.1.2 Connected-digits corpus

This connected-digits corpus is provided by the Speech Processing Lab of the Department Communication Engineering, NCTU. All signals are stored in a format of PCM without file header. The recording format of the waveform files is listed in Table 4-2. The database consists of 3-11 connected digits, such as “011415726”, “79110”, “347”, etc, spoken by 100 speakers (50 males and 50 females). The statistics of the database is shown in Table 4-4.

Table 4-3 Recording environment of the connected-digits

Connected-digits format	
File Format	PCM
Sampling rate	16 kHz
Sampling format	16 bits

Table 4-4 Statistics of the connected-digits database

	Males	Females	Total
Amounts of speakers	50	50	100
Amounts of Files	500	499	999
Maximum digits in a file	3	3	-
Minimum words in a file	11	11	-

4.2 Monophone-based Experiment

The objective of this experiment is to evaluate the performance of different features based on monophone HMMs for speaker-independent speech recognition. The phonetic transcription SAMPA-T is employed in this thesis and then the monophone-based HMMs are then trained, which will states in the section 4.2.1 and 4.2.2, respectively. The experiment results will be shown in the last section.

4.2.1 SAMPA-T

SAMPA-T (Speech Assessment Method Phonetic Alphabet - Taiwan) developed by Dr. Chiu-yu Tseng, Research Fellow of Academia Sinica, are employed for transcribing the database with a machine readable phonetic transcription [23]. Table 4-5 and Table 4-6 are the comparison table of 21 consonants and 39 vowels of Chinese syllables between SAMPA-T, Chinese phonetic alphabet, and the type of pronunciations.

Table 4-5 The comparison table of 21 consonants of Chinese syllables between SAMPA-T and Chinese phonetic alphabets

Type	SAMPA	phonetic alphabet	Type	SAMPA	phonetic alphabet
plosive	b	ㄅ	affricates	dj	ㄐ
	p	ㄆ		tj	ㄑ
	d	ㄉ		dz`	ㄓ
	t	ㄊ		ts`	ㄔ
	g	ㄍ		dz	ㄗ
	k	ㄎ		ts	ㄘ
fricatives	f	ㄈ	nasals	m	ㄇ
	h	ㄏ		n	ㄋ
	s	ㄙ	liquid	l	ㄌ
	s`	ㄕ			
	sj	ㄒ			
	Z`	ㄖ			

Table 4-6 Comparison table of 39 vowels of Chinese syllables between SAMPA-T, and Chinese phonetic alphabets

SAMPA	phonetic alphabet	SAMPA	phonetic alphabet	SAMPA	phonetic alphabet
@n	ㄣ	aN	ㄤ	u@n	ㄨㄣ
i	ㄧ	@N	ㄥ	uai	ㄨㄞ
u	ㄨ	iE	ㄧㄝ	ua	ㄨㄚ
a	ㄚ	iai	ㄧㄞ	uaN	ㄨㄤ
o	ㄛ	iEn	ㄧㄢ	uei	ㄨㄟ
e	ㄝ	ia	ㄧㄚ	uo	ㄨㄛ
@	ㄜ	iaN	ㄧㄤ	y	ㄩ
@`	ㄦ	iau	ㄧㄠ	yE	ㄩㄝ
ai	ㄞ	in	ㄧㄣ	yEn	ㄩㄢ
ei	ㄟ	iN	ㄧㄥ	yn	ㄩㄣ
au	ㄠ	iou	ㄧㄡ	yoN	ㄩㄥ
ou	ㄡ	uan	ㄨㄢ	U	
an	ㄢ	oN	ㄨㄥ	U`	

p.s. U` is the null vowel for retroflexed vowels and U represents the null vowel for un-retroflexed vowels.

All the wave files should be corresponding to a transcription file. For example, a part of paragraph marked with Chinese phonetic alphabets and tones (1, 2,…, 5) are given in the database, shown in Table 4-7. Table 4-8 shows the transcriptions of the words in Table 4-7 marked with SAMPA-T. For monophone-based HMM training, the word-level transcriptions, such as shown in Table 4-8, should be further transferred to the phone-level transcriptions, shown in Table 4-9, where the tones are neglected. It is noted that the punctuation marks, such as comma and period, are replaced with the notation “sil” which means it is silent at this moment in time.

Table 4-7 A paragraph marked with Chinese phonetic alphabets

茶 味 有 苦 、 澀 、 嗆 、 薰 ，
ㄔㄚˊ ㄨㄟˋ ㄧㄡˊ ㄎㄨˇ 、 ㄙㄜˋ 、 ㄑㄧㄠˋ 、 ㄒㄩㄣ ，
由 其 中 才 能 品 味 出 茶 味 的 香 、 甘 、 生 津 ，
ㄧㄡˊ ㄑㄧˊ ㄓㄨㄥ ㄘㄞˊ ㄋㄥˊ ㄆㄧㄣˇ ㄨㄟˋ ㄔㄨ ㄔㄚˊ ㄨㄟˋ ㄉㄜ• ㄒㄧㄠ 、ㄍㄢ 、ㄕㄥ ㄐㄧㄣ ，
同 樣 的 ， 人 生 也 是 有 不 同 的 情 緒 ，
ㄊㄨㄥˊ ㄧㄤˋ ㄉㄜ• ， ㄖㄣˊ ㄕㄥ ㄧㄝˇ ㄕˋ ㄧㄡˇ ㄅㄨˋ ㄊㄨㄥˊ ㄉㄜ• ㄑㄧㄥˊ ㄒㄩˋ
起 起 落 落 ，
ㄑㄧˊ ㄑㄧˇ ㄌㄨㄛˋ ㄌㄨㄛˋ ，
不 也 是 由 痛 苦 中 才 能 真 正 體 會 快 樂 是 什 麼 嗎 ？
ㄅㄨˋ ㄧㄝˇ ㄕˋ ㄧㄡˊ ㄊㄨㄥˋ ㄎㄨˇ ㄓㄨㄥ ㄘㄞˊ ㄋㄥˊ ㄓㄣ ㄓㄥˋ ㄎㄨㄤˋ ㄏㄨㄟˋ ㄎㄨㄞˋ ㄌㄜˋ ㄕˋ ㄕㄜˊ ㄇㄜ• ㄇㄚ• ？

Table 4-8 Word-level transcriptions using SAMPA-T

ts`a2 uei4 iou2 ku3, s@4, tjiaN4, sjyn1,
iou2 tji2 dz`oN1 tsai2 n@N2 pin3 uei4 ts`u1 ts`a2 uei4 d@5 sjiaN1, gan1, s`@N1 djin1,
toN2 iaN4 d@5, Z`@n2 s`@N1 iE3 s`U`4 iou3 bu4 toN2 d@5 tjiN2 sjy4,
tji2 tji3 luo4 luo4,
bu4 iE3 s`U`4 iou2 toN4 ku3 dz`oN1 tsai2 n@N2 dz`@n1 dz`@N4 ti3 huei4 kuai4 l@4 s`U`4 s`@2 mo5 ma5?

Table 4-9 Phone-level transcriptions using SAMPA-T

ts` a uei iou ku sil s @ sp tj iaN sp sj yn sil
iou tj i dz` oN ts ai n @N p in uei ts` u ts` a u ei d @ sj iaN sil g an s` @N dj in sil
t oN iaN d @ sil Z` @n s` @N iE s` U` iou b u t oN d @ tj iN sj y sil
tj i tj i l uo l uo sil
b u iE s` U` iou t oN k u dz` oN ts ai n @N dz` @n dz` @N t i h uei k uai l @ s` U` s` @ mo ma sil

4.2.2 Monophone-based HMM used on TCC-300

From the phonetic transcription defined in SAMPA-T, there are 21 consonants and 39 vowels of Chinese dialects spoken in Taiwan. Hence, the total number of monophone-based HMM is equal to 62, including 21 consonants, 39 vowels, the silence model “sil”, and the short pause model “sp” where the “sp” denotes the short pause between two words. The number of states of the HMM is defined in Table 4-10

and the structure is shown in Fig.4-1. It is noted that the number of states here includes 2 null states, called entry and exit node, which cannot produce any observations, and the probabilities of staying in the null states is equal to zero. The entry and exist node make the HMMs much easier to connect together without changing parameters of the HMMs, for example, the word "樂" is a combination of the HMM "l" and the HMM "@", shown in Fig.4-2.

Besides, the shrot pause model "sp" used here is so called "tee-model" which has direct transition from entry to exist node. The silence model has extra transitions from states 2 to 4 and from states 4 to 2 in order to make the model more robust by allowing individual states to absorb the various impulsive noises in the training data. The backward skip allows this to happen without committing the model to transit to the following word.

Table 4-10 Definitions of HMM used in monophone-based experiment

Number of monophone-based HMMs	62	(60 monophones, "sp" and "sil")
Number of states of "sp"	3	(first and last state are null state)
Number of states of consonants (includes "sil")	5	(first and last state are null state)
Number of states of vowels	7	(first and last state are null state)
Number of Gaussian mixtures in a state	5	

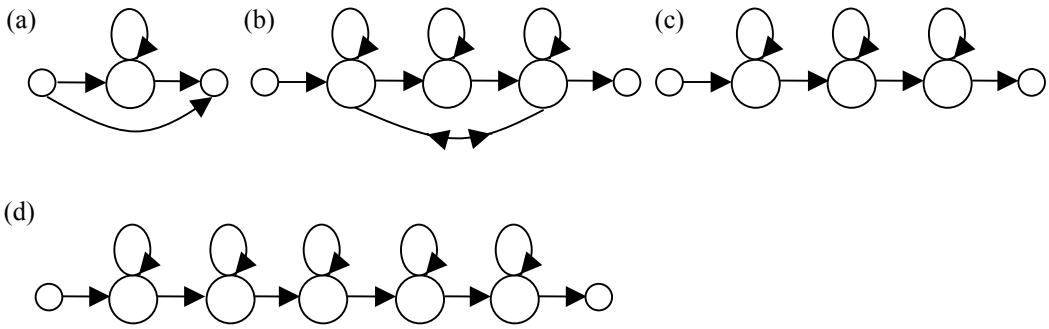


Fig.4-1 HMM structure of (a) sp, (b) sil, (c) consonants and (d) vowels

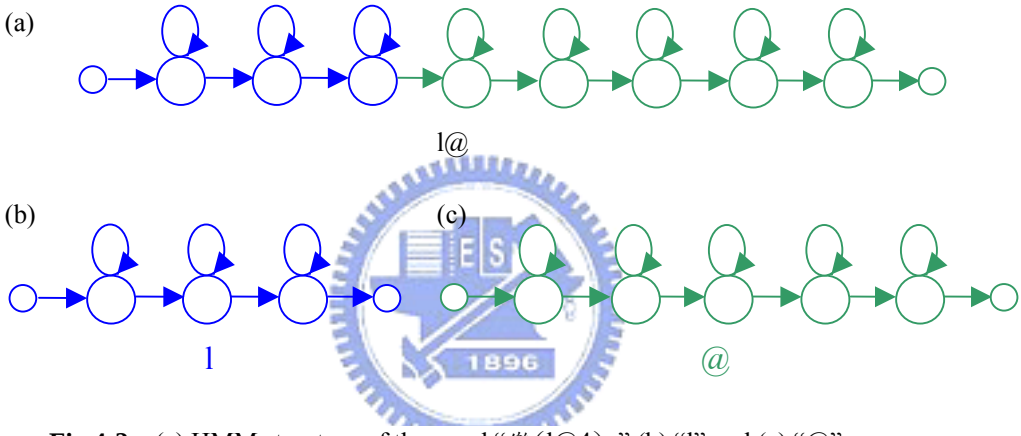


Fig.4-2 (a) HMM structure of the word "樂(l@4), " (b) "l" and (c) "@"

The training database is selected from the TCC-300, where eight folders (F_NEWG1–F_NEWG4 and M_NEWG1–M_NEWG4) produced by NCTU are employed to train the monophone-based HMMs. The training database comprises of 517 files spoken by 40 females and 515 files spoken by 40 males. All the MAT files should be converted to the wave format prior to training. The Hidden Markov Model Tool Kit (HTK) developed by Cambridge University Engineering Department (CUED) is employed in this thesis since it provides sophisticated facilities for speech research.

4.2.3 Experiments

The parameters of front-end processing are set as Table 4-11. The features adopted in the experiment are listed in Table 4-12. The flow chart of training the monophone-based HMMs is shown in Fig. 4-3. At the beginning, only the corpus and its corresponding Chinese phonetic alphabets are available. Hence, it is essential to transfer the Chinese phonetic alphabets to SAMPA-T before training. It is noted that there is no boundary information of the corpus. Here, six features selected in this thesis are based on LPC, MFCC, and PLP, which have been introduced in Chapter 2. In the process of training, there is no rule that how much times of doing the Baum-Welch re-estimation will get best model and consequently it is necessary to test and verify the recognition rate to find the best model.

Table 4-11 The parameters of front-end processing

Sampling frequency	16 kHz
Pre-emphasis filter	$1-0.97z^{-1}$
Hamming window	$w(n)=0.54-0.46\cdot cos\left(\frac{2\pi n}{N-1}\right), \quad 0\le n\le N-1$
Window size	400 samples (25ms)
Frame duration	25 ms
Frame period	10 ms

Table 4-12 Six different features adopted in this thesis

	Order	Number of filter banks	Energy	Δ	Δ^2
Linear Predictive Coefficients (LPC_39)	39	-	√	√	√
Linear Predictive Coefficients (LPC_38)	38	-	√	√	
Linear Predictive Reflection coefficients (RC)	39	-	√	√	√
LPC Cepstrum Coefficients (LPCC)	39	-	√	√	√
Mel-Frequency Cepstral Coefficients (MFCC)	39	26	√	√	√
Perceptual Linear Prediction Coefficients (PLP)	39	26	√	√	√

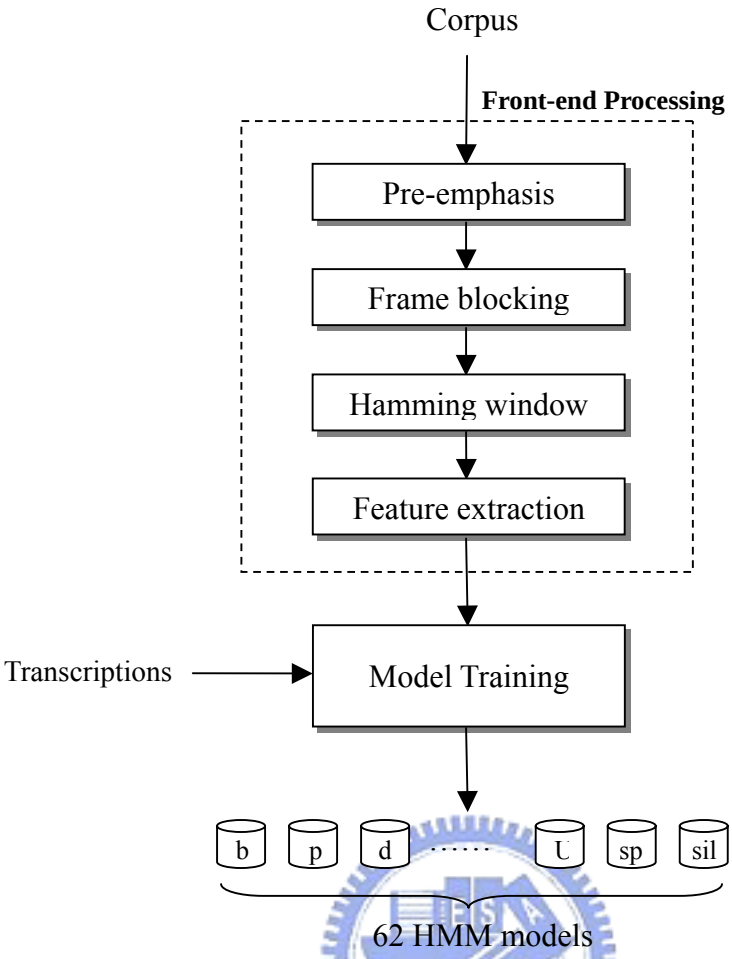
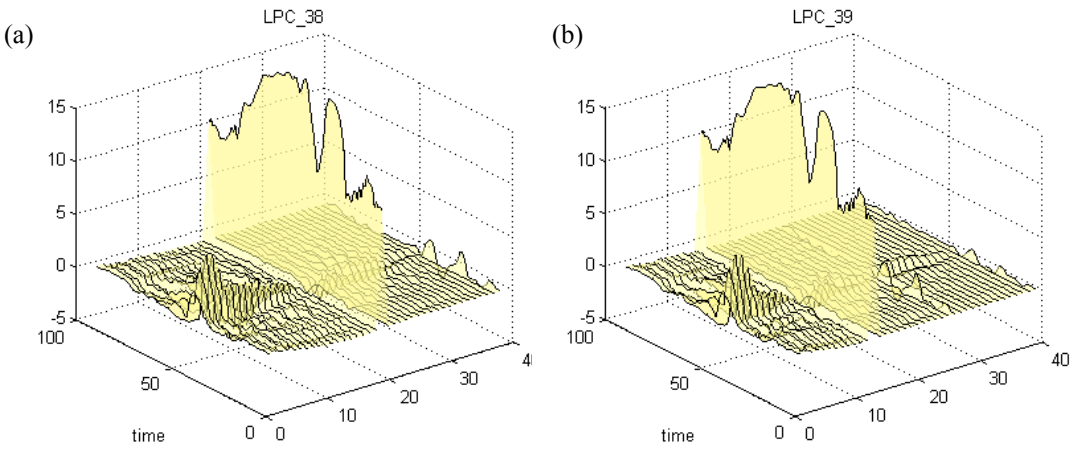


Fig.4-3 Flow chart of training the monophone-based HMMs

There is a 3-D view of six features performing on the word "不久" (bu4 djiou3) and the variations of the 39-dimensioned (or 38-dimensioned for LPC_38) vectors from frame 1 to frame 100 are shown in Fig. 4-4 where time denotes the frame order. The highest curve is at the 13-th element of the feature vectors (19-th element for LPC_38) since this element is the energy term.



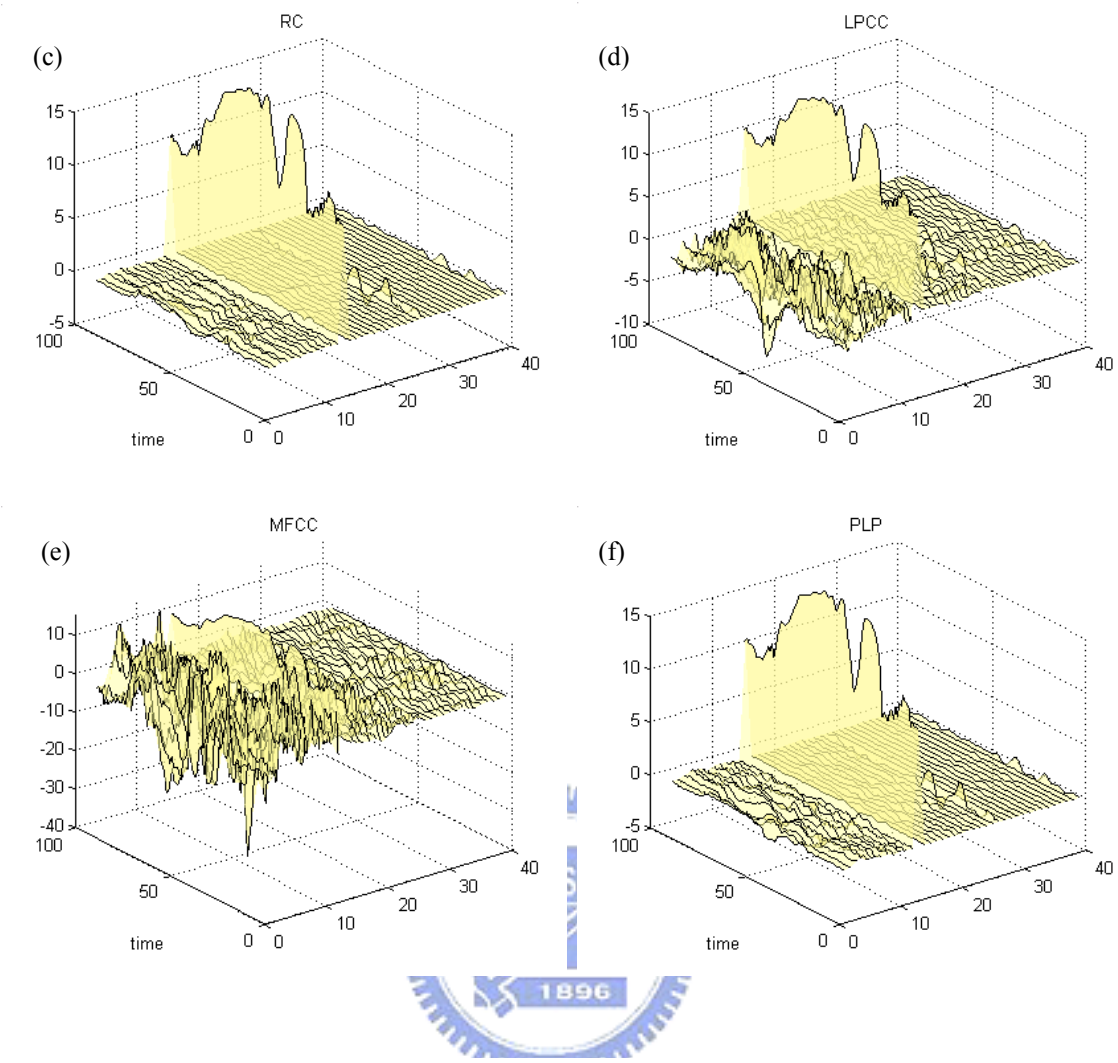


Fig.4-4 3-D view of the variations of the feature vectors (a) LPC-38 (b) LPC_39 (c) RC (d) LPCC (e) MFCC (f) PLP

The monophone-based HMMs are usually employed in Large Vocabulary Speech Recognition (LVSR). However, one of the factors which influence the recognition rate of the LVSR is the language model. Language model is a statistical model which attempts to capture the regularities of natural languages and improve the performance by estimating the probability distribution of various linguistic units, such as words, sentences, etc. If the recognition task is long paragraphs or articles, language model should be trained. Nevertheless, language model is to the key point in this thesis. Hence, the connected-digits corpus just mentioned in 4.1.2 is utilized for testing the monophone-based HMMs and the HMMs are trained by six different kinds

of feature extraction methods where the connected-digits needs only an uncomplicated grammar that the sentence are arbitrary permutation of digits.

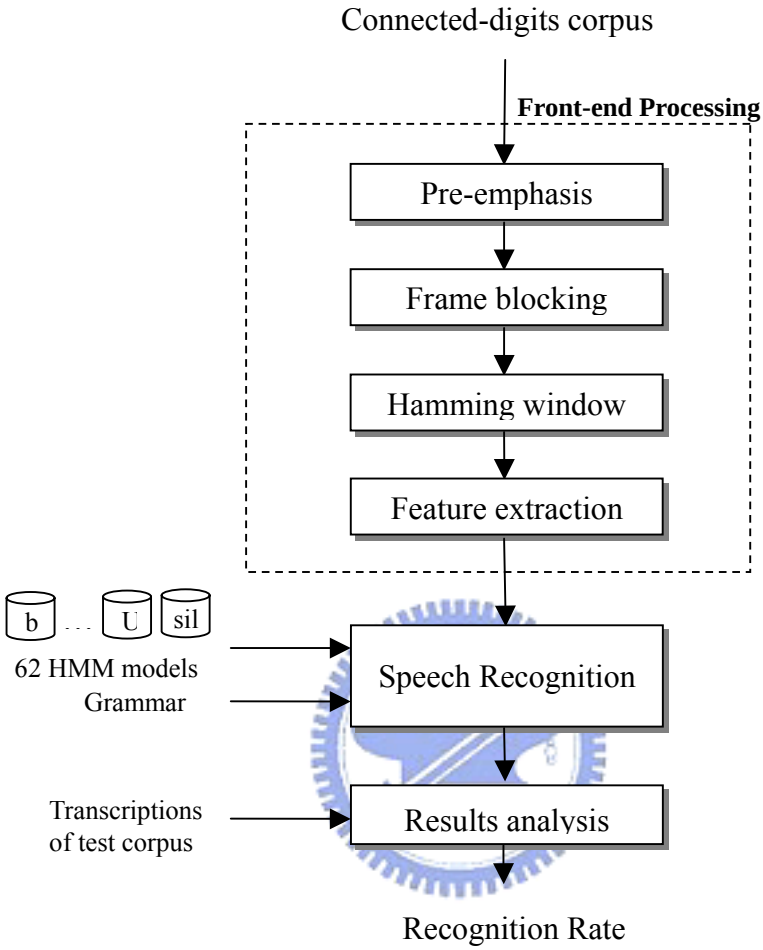


Fig.4-5 Flow chart of testing the performance of different features

The experimental results are shown in Table 4-12 and Fig. 4-6. The total number of digits, denoted by T, used in this experiment is 8432. There are three variables should be concerned in order to compute the recognition rate, that is, the number of insertions (I), the number of deletions (D) and the number of substitutions (S). For example, the output sentence of the recognition may be

i2 @`4 ba1 djiou3 sU4

while the actual sentence is

i2 ba1 liou3 sU4 san1

where "@`4" is an insertion error, "djiou3" is a substitution error and "san1" is an deletion error.

Based on the definition of mentioned the above, the performance of different features can be examined through two functions, the Correct (%) and the Accuracy (%). The Correct (%) is computed by

$$\text{Correct}(\%) = \frac{T - D - S}{T} \times 100 \tag{4-1}$$

and the Accuracy (%) is defined as

$$\text{Accuracy}(\%) = \frac{T - D - S - I}{T} \times 100 \tag{4-2}$$

which means that the Accuracy (%) concerns not only the deletion error and the substitution error but also the insertion error. Hence, the Accuracy (%) will lower than the percent of correct (%).

Table 4-13 Comparison of the Corr (%) and Acc (%) of different features

	LPC_38		**LPC_39**		**RC**		**LPCC**		**MFCC**		**PLP**	
Number of iterations	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc
1	73.7	40.1	69.4	38.3	82.0	46.0	87.3	67.1	89.6	67.2	90.0	67.9
2	75.2	47.5	72.8	47.2	83.7	54.3	88.9	70.8	91.2	72.2	91.6	73.7
3	76.4	51.3	74.8	49.4	84.0	56.1	89.2	71.6	92.0	74.2	92.2	75.6
4	77.8	54.1	75.9	50.6	83.7	54.5	89.6	71.5	92.4	74.4	92.9	76.5
5	78.7	55.9	76.4	52.3	83.6	54.0	89.6	71.8	92.8	75.7	93.2	77.3
6	79.7	56.6	76.8	53.9	83.6	54.2	89.6	71.7	92.9	76.3	93.3	78.0
7	80.1	57.3	76.9	54.2	83.6	54.3	89.5	71.9	93.0	77.1	93.4	78.6
8	80.6	58.2	77.1	54.5	83.7	54.4	89.5	71.9	93.0	77.4	93.4	78.7
9	81.0	59.0	77.0	54.4	83.9	54.7	89.7	72.1	93.1	77.7	93.4	78.7
10	81.1	59.1	77.1	54.3	83.9	55.0	89.6	72.1	93.1	78.1	93.4	78.8
11	81.2	59.2	77.1	54.3	84.1	55.4	89.6	72.1	93.2	78.2	93.3	78.9
12	81.2	59.2	76.8	54.0	84.2	56.0	89.6	72.2	93.2	78.3	93.2	78.9

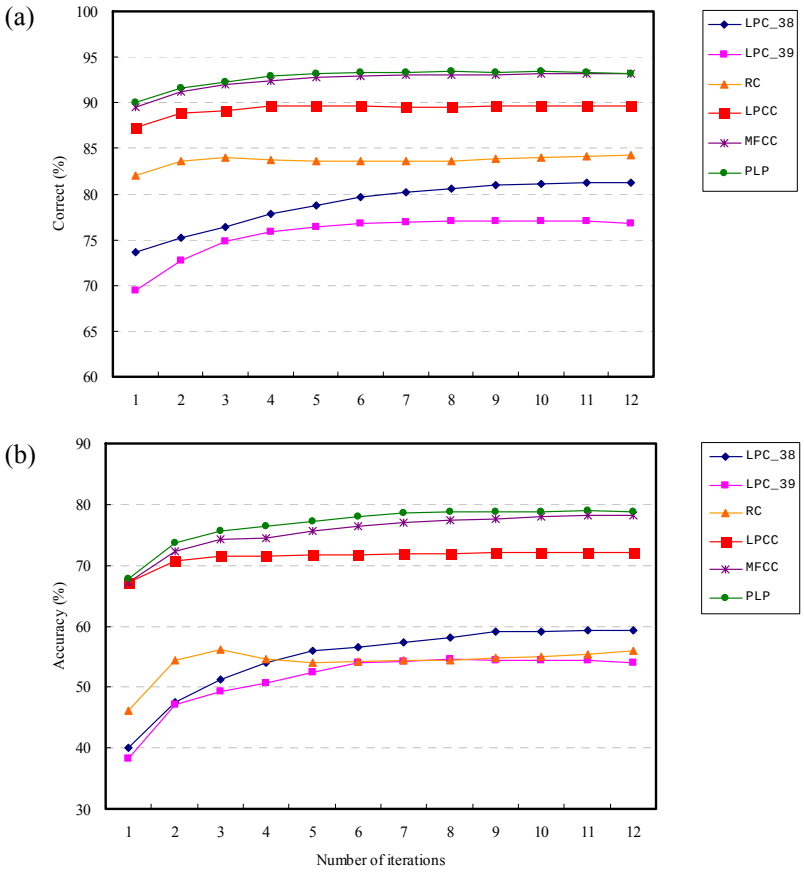


Fig.4-6 Comparison of the different features (a) Correct (%) (b) Accuracy (%)

From the Fig. 4-6(a), the percent of correct is the recognition rate without considering the insertion error and the performance of the connected-digits speaker-independent recognition based on monophone HMM is

$$\text{PLP} > \text{MFCC} > \text{LPCC} > \text{RC} > \text{LPC38} > \text{LPC39} \qquad (4\text{-}3)$$

where the PLP performs better than all the other features from iteration 1 to iteration 12 of the training. The performance of all the models almost saturates when coming up to iteration 12. The maximum percent of correct of PLP appears in iteration 8 of training and then decreases when more training iterations are performed. It shows that the PLP costs less time than others to reach a better model in the training stage. From

Fig. 4-6(b), it shows the recognition results when the insertion error is considered. The order of the performance is not the same as (4-3) especially for the RC. It infers that the RC tends to insert words between two words than other insertion methods.

Comparisons of the different features through average and the best Correct (%) and Accuracy (%) are shown in Fig. 4-7. The order of the performance from the good to the bad in this experiment is

$$\text{PLP} > \text{MFCC} > \text{LPCC} > \text{RC} > \text{LPC38} > \text{LPC39} \tag{4-4}$$

except for the max Accuracy (%) where RC is worse than LPC38, hence, the order of the best performance of the six features is

$$\text{PLP} > \text{MFCC} > \text{LPCC} > \text{LPC38} > \text{RC} > \text{LPC39} \tag{4-5}$$

where PLP still has the best performance than other features in monophone-based speaker-independent speech recognition.

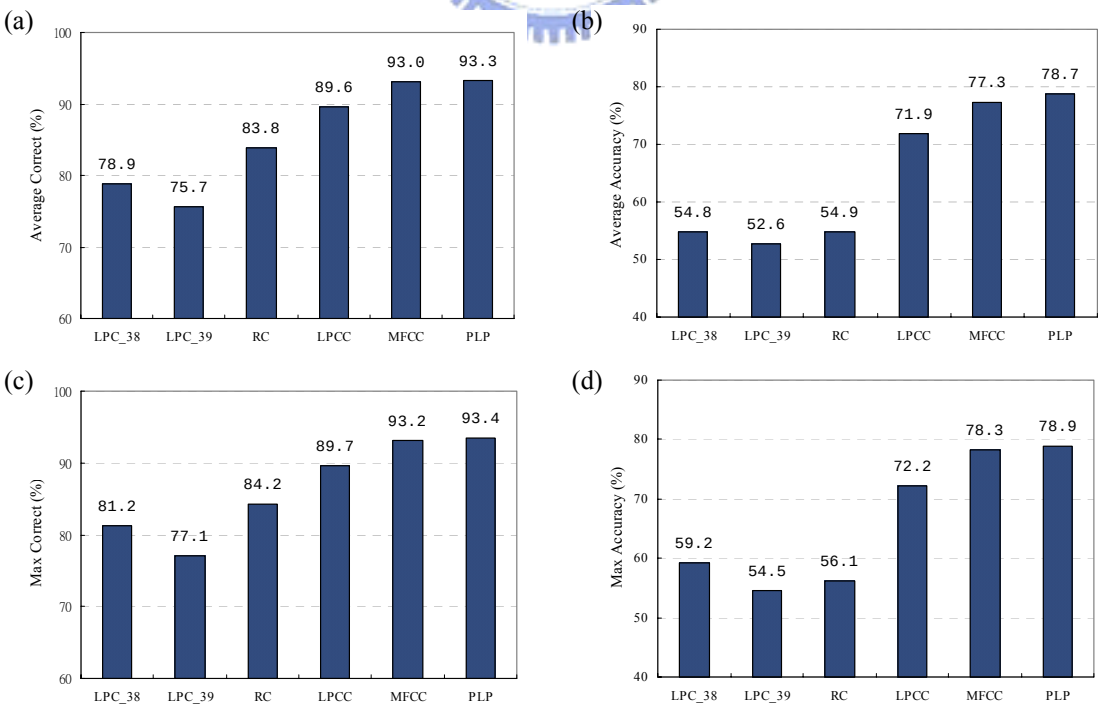


Fig.4-7 Monophone-based HMM experiment (a) Average Correct (%) (b) Average Accuracy (%) (c) Max Correct (%) (d) Max Accuracy (%)

4.3 Syllable-based Experiments

The purpose of this experiment is to examine the performance of different features while applying to the word-level HMM speaker-independent speech recognition. The word-level HMM is feasible when the recognition task is small; hence, the connected-digits corpus is employed to train the word-level HMM and then utilized to recognize the connected-digits sentences in this thesis.

4.3.1 Syllable-based HMM used on connected-digits corpus

The connected-digits sentence is composed of arbitrary combination of the digits - "0, 1, 2, 3, 4, 5, 6, 7, 8, or/and 9". Hence, the total number of word-level HMM needed is 12, including 10 digits, the silence model "sil", and short pause model "sp". The number of states of the HMM is defined in Table 4-14

The training database is selected from the connected-digits database mentioned in 4.1.2, where 800 files, where 400 files are spoken by 40 males and 400 files are spoken by 40 females, are selected for training the syllable-based HMMs. The other files of the corpus (199 files, 99 files spoken by 10 females and 100 spoken by 10 males) are adopted for testing the performance of the different features in syllable-based HMM speaker-independent speech recognition.

Table 4-14 Definition of HMM used in syllable-based experiment

Number of syllable-based HMMs	12	(10 digits, "sp" and "sil")
Number of states of HMM	8	(first and last state are null state)
Number of Gaussian mixtures in a state	4	

4.3.2 Experiments

The parameters of front-end processing are set the same as Table 4-11. The features adopted in the experiment listed in Table 4-15 which are the same as the parameters used in monophone-base experiments. The flow chart of training the syllable-based HMMs is shown in Fig. 4-8 where the digits “0, 1, 2, 3, 4, 5, 6, 7, 8, and 9” are denotes by “yi, er, san, si, wu, liu, qi, ba, jiu, and ling,” respectively. It is noted that the boundary information of the corpus is available. Therefore, the training procedure is different with the procedure of experiment in 4.2 which has no boundary information and the details of the difference between them have been introduced in section 3.3. In practice, the boundary information is beneficial for training HMM, that is, the HMM will be trained more precise with the boundary inforamtion. In addition, the number of HMMs is less than the HMMs used in previous experiment. The recognition results are supposed to be much higher than the results in 4.2.3.

Fig.4-9 shows the testing procedure of the sullable-based recognition procedure. Table 4-16 shows the experiment results of the performance of different features performing on the connected-digits speaker-independent recognition. The performance of the features is generally in the order from the good to the bad as

$$\text{PLP, MFCC} > \text{LPCC} > \text{RC} > \text{LPC\_38} > \text{LPC\_39} \qquad (4\text{-}6)$$

where the PLP and MFCC are resemble in maximum ACC(%), which is the major guide for judging the performance of the models.

Table 4-15 Six different features adopted in this thesis

	Order	Number of filter banks	Energy	Δ	Δ^2
Linear Predictive Coefficients (LPC_39)	39	-	√	√	√
Linear Predictive Coefficients (LPC_38)	38	-	√	√	
Linear Predictive Reflection coefficients (RC)	39	-	√	√	√
LPC Cepstrum Coefficients (LPCC)	39	-	√	√	√
Mel-Frequency Cepstral Coefficients (MFCC)	39	26	√	√	√
Perceptual Linear Prediction Coefficients (PLP)	39	26	√	√	√

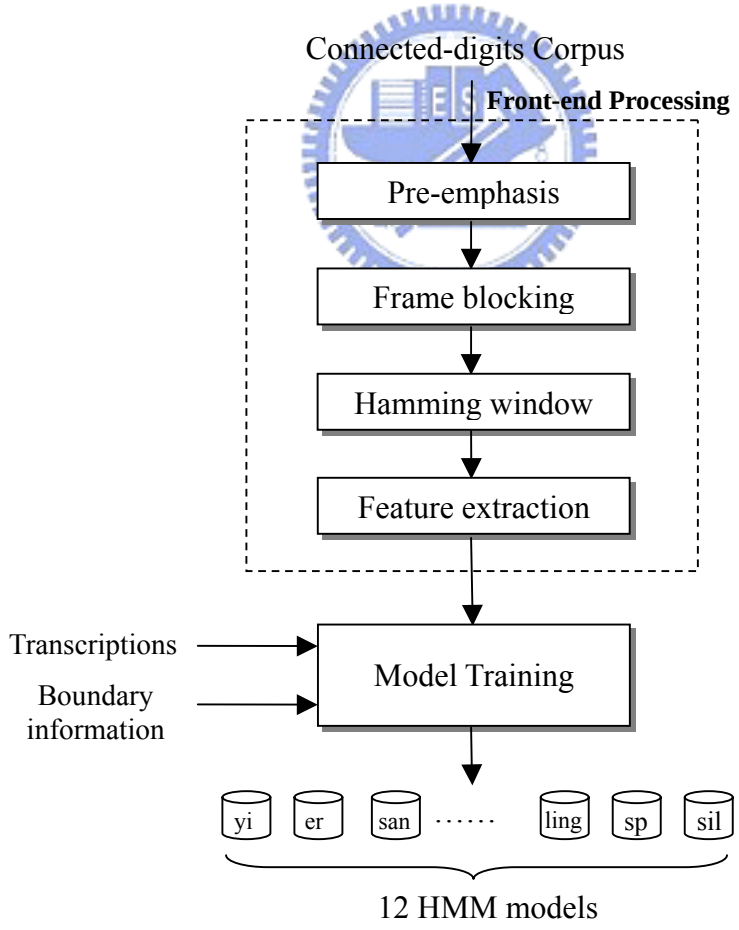


Fig.4-8 Flow chart of training the syllable-based HMMs

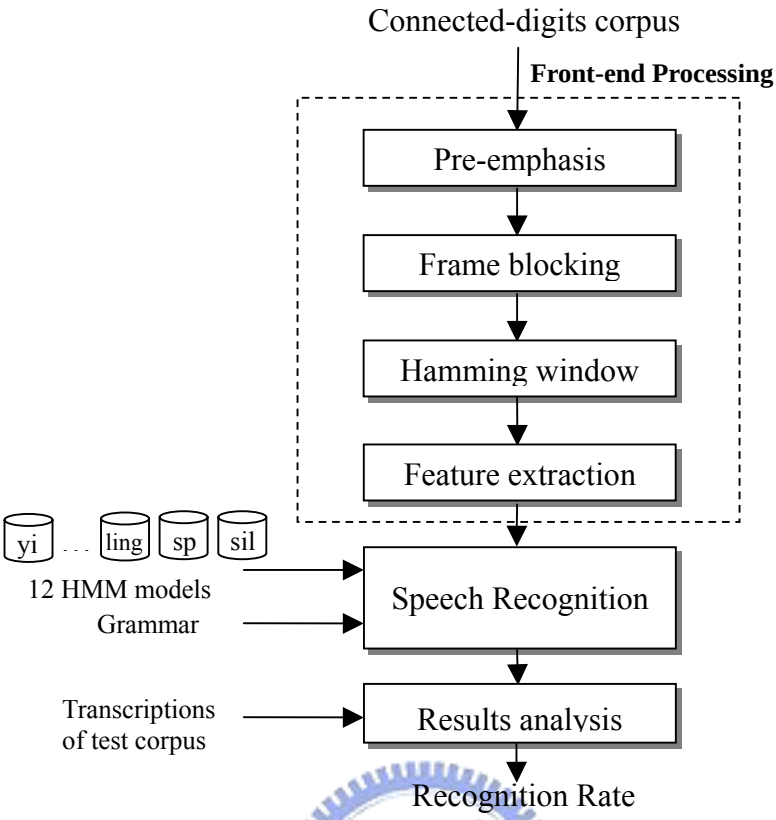


Fig.4-9 Flow chart of testing the syllable-based HMMs

Table 4-16 Comparison of the Corr (%) and Acc (%) of different features

	LPC_38		**LPC_39**		**RC**		**LPCC**		**MFCC**		**PLP**	
Number of iterations	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc
1	79.7	77.1	80.5	77.6	90.9	89.7	94.3	93.8	96.5	95.9	96.6	96.1
2	84.0	80.8	83.5	80.3	94.3	92.9	96.6	96.3	98.2	98.0	98.4	98.1
3	85.0	81.8	84.3	81.2	94.7	93.3	97.1	96.9	98.4	98.2	98.5	98.4
4	84.8	81.9	84.6	81.6	94.9	93.5	97.2	97.1	98.6	98.4	98.5	98.3
5	85.2	82.3	84.5	81.4	94.8	93.4	**97.3**	97.1	**98.7**	**98.5**	98.5	98.4
6	85.2	82.4	84.6	81.6	94.9	93.6	97.2	97.0	98.7	98.5	**98.6**	**98.5**
7	85.3	82.6	**84.7**	**81.7**	95.0	93.8	97.2	97.0	98.7	98.5	98.6	98.5
8	85.3	82.7	84.6	81.6	94.8	93.7	97.3	97.1	98.6	98.5	98.6	98.5
9	85.4	**82.9**	84.3	81.2	95.0	93.9	97.3	**97.2**	98.6	98.5	98.6	98.5
10	**85.6**	83.2	84.5	81.3	**95.1**	**93.9**	97.2	97.1	98.6	98.5	98.5	98.4

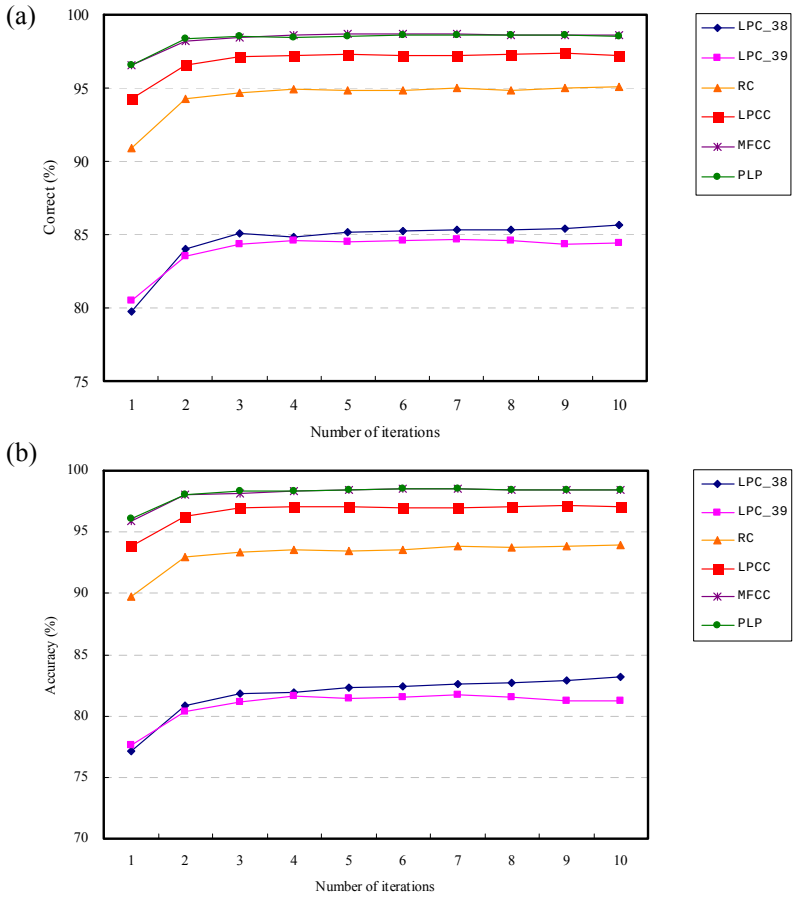


Fig.4-10 Comparison of the different features (a) Correct (%) (b) Accuracy (%)

Comparisons of the different features through average and the best Correct (%) and Accuracy (%) are shown in Fig. 4-11. In this case, the PLP and the MFCC are both a good choice of the connected-digits speaker-independent recognition due to their high recognition rate (Correct (%) and Accuracy (%)). The LPC_38 performs better than the LPC_39 since the sampling frequency of the speech is 16 kHz. From the guideline of selecting the order of filter p introduced in Chapter 2, the value of p should be chosen as 18-20 to represent the characteristic of the filter. Hence, the recognition rate of LPC_38 (p=18) is higher than the LPC_39. As for the reflection coefficients RC, the performance is much better than the performance of LPC_38. It can be inferred that the RC is more suitable for represent the speech signal of small-vocabulary speech recognition from the results of the two experiments.

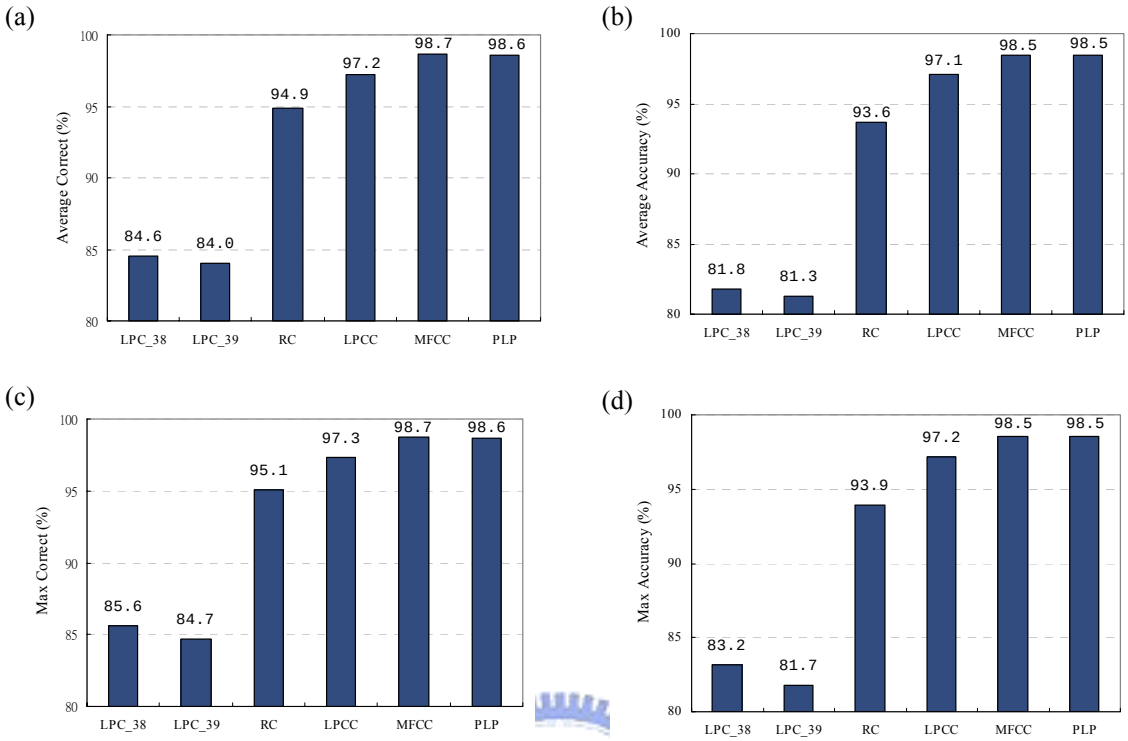


Fig.4-11 Syllable-based HMM experiment (a) Average Correct (%) (b) Average Accuracy (%) (c) Max Correct (%) (d) Max Accuracy (%)

Chapter 5

Conclusions

A short summary of different features is made as follows. The LPC states that the vocal tract transfer function can be modeled by an all-pole filter, and the number of coefficients is chosen to be sufficient to represent the vocal tract. The Reflection Coefficients (RC) model the reflection rate at each transition when the acoustic waves in the vocal tract are partially reflected at the transitions and interfere with waves approaching from the back. The LPC-derived Cepstral Coefficient (LPCC) is compact parametric representation of representing the spectrum of speech signals which can efficiently separate the excitation source from the all-pole filter. The conception of the Mel-Frequency Cepstral Coefficients (MFCC) is to use nonlinear frequency scale to approximate the behavior of the auditory system. The Perceptual Linear Predictive (PLP) analysis combines several engineering approximations of psychophysics of human hearing processes, including critical-band spectral resolution, the equal-loudness curve, and the intensity-loudness power law.

The results of the experiments can be explained from the essence of the features. In former studies of the feature extraction, such as LPC and RC, the idea is focuses on modeling the vocal tract of the human. However, the performance is not satisfied especially for multi-users system. Besides, the production model concerns only the vocal tract which varies from person to person. From the experimental results, it can be infer that the variation of the vocal tract between persons is larger than the variation of the ear between persons. From the viewpoint of the human being, the general communication is comprised of two types, speech generating and hearing. Intuitively, the objective of the speaker-independent speech recognition system is to

recognize the speech of different users. Hence, the key point is not who produced the speech but what the context was. In this case, the receiving side is more effective than the generation sides. Therefore, the features based on the speech perception, such as MFCC and PLP, are superior to the features based on the speech production, such as LPC, LPCC and RC, in the speaker-independent experiments.

In this thesis, the performance of different speech features for speaker-independent speech recognition system has been evaluated. It is noted that the PLP is not always better than MFCC because of the little difference between the recognition rates in the experiments in previous chapter, but it can be said that in most of cases the PLP and MFCC will perform better than LPC, RC and LPCC in speaker-independent speech recognition system. It can be concluded as follows. Firstly, features derived from FFT (MFCC, PLP) preserve more phonetic features than those derived form LPC spectrum (LPC, LPCC, RC). Secondly, the cepstrum parameters (LPCC) has higher recognition rate than LPC and RC. Thirdly, non-linear frequency analysis performs better than linear frequency analysis. Fourthly, LPC_38 has better performance than LPC_39. Fifthly, PLP provide highest discrimination of phonetics for monophone-based speaker-independent SR. In addition, there is a performance comparison table Table 5-1. From the table, the perceptual model is more effective than production model in speaker-independent Speech Recognition system.

Table 5- 1 Performance Comparison Table

Monophone-based experiment					
PLP	MFCC	LPCC	RC	LPC_38	LPC39
78.9%	78.3%	72.2%	56.0%	59.2%	54.5%
Word-based experiment					
PLP	MFCC	LPCC	RC	LPC_38	LPC39
98.5	98.5	97.2	93.9	81.7	83.2

Due to the limitation of the corpus and the difficulties of training with large amount of database, the experiments are not complete to show the statistics of various tasks, for example, the robust test of speech features for speaker-independent speech recognition system in different noisy environments is not fulfilled. In addition, the environments (echo, channel-effect, noise, etc) and the speakers (speed of speaking, gender, age, etc) will both affect the performance of the recognition system in practice. It is hard to start from the viewpoint of physiology to improve the features, thus to find a suitable feature for a particular task and adding new skills to eliminate the influence of environment and speakers are more feasible in the future.

References

[1] R.C. Rose and D. A. Reynolds, “Text-independent speaker identification using automatic acoustic segmentation,” *ICASSP, IEEE*, 1990.

[2] John Makhoul, “Linear Prediction: A Tutorial Review,” *Proceedings of the IEEE*, Vol. 63, No. 4, pp. 561–580, Apr. 1975.

[3] S. B. Davis and P. Mermelstein, “Comparison of Parametric Representation for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-28, pp357-366, 1980.

[4] Hynek Hermansky, “Perceptual Linear Predictive (PLP) Analysis of Speech,” *Acoustic Society of America*, Vol. 87, No. 4, pp.1738-1752, Apr. 1990.

[5] J.W. Picone**, “**Signal Modeling Techniques in Speech Recognition**,”** *Proceedings of the IEEE*, Vol. 81, No.9, pp.1215-1247, Sept. 1993.

[6] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, New York: Springer-Verlag, Berlin, first edition, 1976.

[7] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development,* Prentice Hall PTR, 25 April, 2001.

[8] Stevens, S.S, “On hearing by electrical stimulation,” *Journal of the Acoustical Society of America*, Vol. 8, pp.191-195.

[9] Stevens, S.S. and J. Volkman, “The Relation of Pitch to Frequency,” *Journal of Psychology*, Vol. 53, pp. 329, 1940.

[10] V. Mantha, R. Duncan, Yufeng Wu, Jie Zhao, A. Ganapathiraju and J. Picone,” Implementation and Analysis of Speech Recognition Front-End,” *Southeastcon '99. Proceedings. IEEE*, 25-28, pp.32-35, March 1999.

[11] Gabriel Costache, Inge Gavat, and Adrian Raileanu, “ Voice Command System,” *Politehnica University, Bucharest, International Workshop*, 16-18 May, 2002.

[12] Zhang Jie, Huang Zhitong, and Wang Xiaolan, “Selection and Analysis of HMM's State-number in Speech Recognition,” *Signal Processing Proceedings*, ICSP '98, Vol. 1, pp.641-pp.645, 1998.

[13] J. Wilpon, and L. Rabiner, ” A Modified K-means Clustering Algorithm for Use in Isolated Word Recognition,” *Acoustics, Speech, and Signal Processing* [see also *IEEE Transactions on Signal Processing*], IEEE Transactions on, Vol. 33 , Issue 3 , pp.587-pp.594, Jun. 1985.

[14] Lawrence Rabiner and B. H. Juang, *Fundamentals of Speech Procesing*, *Prentice Hall Co. Ltd.*, 1993.

[15] Peter Motlíček, “Feature Extraction in Speech Coding and Recognition,” Report of PhD research internship in ASP Group, OGI-OHSU, Portland, US, 2001/2002, pp.1-50, http://www.fit.vutbr.cz/~motlicek/publi/2002/rp.pdf.

[16] G. M. White and R. B. Neely, “Speech recognition experiments with linear prediction, bandpass filtering, and dynamic programming,” *IEEE Trans. Acoust., Speech, Signal Processing,*Vol. ASSP-24, pp. 183-188, Apr. 1976.

[17] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*, IEEE Press, 2000.

[18] Montri Karnjanadecha and Stephen A. Zahorian, “Signal Modeling for High Performance Robust Isolated Word Recognition,” *IEEE Trans. Speech and Audio Processing*, Vol. 9, No. 6, pp.647-654, Sept. 2001.

[19] J. G. Wilpon and L. R. Rabiner, ” A Modified K-Means Clustering Algorithm for Use in Isolated Word Recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processin*g, Vol. 33, No. 3, Jun. 1985.

[20] L. R. Rabiner, ” A tutorial on hidden Markov models and selected applications in

speech recognition," *Proceedings of the IEEE*, Vol. 77, Issue 2 , pp.257-pp,286, Feb. 1989.

[21] L. R. Rabiner, and B. H. Juang, "An Introduction to Hidden Markov Models ASSP Magazine," IEEE, Vol.3, Issue. 1 ,pp.4-pp.16, Jan 1986.

[22] 范世明, 高斯混合模型在語者辨識與國語語音辨認之應用, 國立交通大學碩士論文, 民國九十一年六月.

[23] C. Y. Tseng and F. C. Chou, "Machine Readable Phonetic Transcription System for Chinese Dialects Spoken in Taiwan," *The Journal of the Acoustical Society of Japan (E)*, Vol.20, No.3, pp.215-pp.223, May 1999.

[24] "Hidden Markov Model Toolkit," http://htk.eng.cam.ac.uk/.