The Pennsylvania State University

The Graduate School

SPEECH SILENT/UNVOICED/VOICED CLASSIFICATION WITH HIDDEN MARKOV MODELS

A Thesis in

Electrical Engineering

by

Peng Lee

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2008

The thesis of Peng Lee was reviewed and approved* by the following:

Robert M. Nickel
Adjunct Professor of Electrical Engineering
Thesis Advisor

Ken Jenkins
Professor of Electrical Engineering
Head of the Department and Member of Committee

Kultegin Aydin
Professor of Electrical Engineering
Graduate Program Coordinator

*Signatures are on file in the Graduate School

Abstract

Robust silent/unvoiced/voiced (SUV) classification in noise is still considered a difficult speech processing problem. Most mechanisms proposed in the past perform well in either noise-free or with only mild additive noise environments. Their performance, however, degrades dramatically in low SNR situations. We propose an MFCC codebook based mechanism with a hidden Markov model to address the problem for a *dedicated speaker scenario* in a low SNR environment. Our experiments show that we can achieve a 90% correct classification accuracy in 5 dB SNR with stationary (white) noise and a 78% classification accuracy in 5 dB SNR with nonstationary (babble) noise. Our results compare favorably with the performance of a GMM based multi-features classifier and a state-of-the-art SUV classifier based on the discrete wavelet transform (DWT).

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere thanks to my advisor Dr. Robert M. Nickel for his continuous guidance and support and for giving me an opportunity to work in the area of speech processing. My sincere thanks to Dr. W. Kenneth Jenkins for finding time to review my work and for the feedback provided. I am also very grateful to all the professors under whom I took graduate courses at Penn State. I would also like to thank my friend and lab-mate Xiaoqiang Xiao for all the valuable discussions and inputs. Finally, I would like to thank my parents for their unconditional love and support throughout my graduate studies.

Chapter 1

Introduction

SUV speech classification is used to classify speech signals into three different classes: silent, unvoiced and voiced. SUV speech classification plays an important role as a pre-processing stage in numerous applications to detect the presence of each class of speech. For example, in voice over IP and mobile telephony applications SUV speech classification can be used to reduce the bandwidth usage and network traffic by transmitting audio packets only if speech is detected. It can be used to improve the performance of speech recognition applications, speaker recognition applications and speaker localization applications because the algorithm will be applied only on parts of the audio that are identified as speech. A prominent source localization application where SUV speech classification can be used as a pre-processing stage is video conferencing applications that use source localization algorithms to steer a video camera in the direction of the signal source. In such an application SUV speech classification is useful as a pre-processing stage that ensures that source localization is performed and the video camera is steered in the direction of the audio source, only when speech is detected. SUV speech classification is also used in applications as a pre-processing step that detects noise so that it can be removed by algorithms like spectral subtraction and side-lobe cancellation. For example, SUV speech classification is used in hearing aids to reduce noise. In the source localization application mentioned earlier SUV speech classification is also used to detect noise so that it can be reduced.

1.1 Voiced, Unvoiced and Silent Speech

A general linear discrete time model for speech production is shown in Figure 1.1. This model is called *terminal analog model* [2], meaning that the signals and systems involved in the model are only superficially analogous to the real physical system.

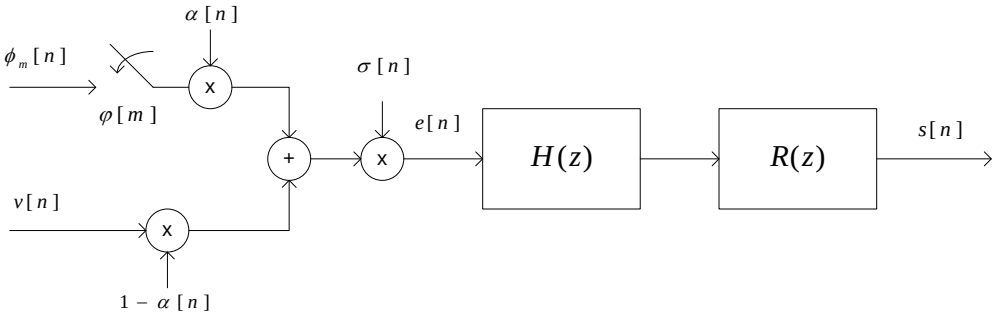

Figure 1.1. A general discrete-time model of speech production

1.1.1 Discrete Time Filter Modeling of Speech Production

In discrete time speech processing, it is fairly convenient to model speech as a digital filter $h[n]$ with excitation $e[n]$ which is able to generate three fundamental excitation types: silent, unvoiced and voiced. In voiced type, $e[n]$ is described as a sequence of elementary glottal pulses $\Phi_m[n]$. The voiced mode excitation is described as an impulse train with *epoch* period $\varphi[m]$. In unvoiced type the excitation is generally realized with a *white* random noise process $v[n]$. The voiced and unvoiced excitation types of speech excitation can be illustrated as one simple model [1]:

$$e[n] = \sigma[n] \cdot \left[(1 - \alpha[n]) \cdot v[n] + \alpha[n] \cdot \sum_{m=-\infty}^{\infty} \Phi_m[n - \varphi[m]] \right]. \tag{1.1}$$

The voiced-unvoiced portion control coefficient $\alpha[n]$ decides the *voicing level* of the excitation. For $\alpha[n] = 1$, we have the fully voiced excitation and for $\alpha[n] = 0$ we have the purely unvoiced excitation. When $0 < \alpha[n] < 1$, that permits the modeling of *mixed*

excitation. The overall strength of the excitation is controlled by $\sigma[n]$. With $\sigma[n] = 0$ we obtain speech silence.

To complete the whole speech production model, there still is a very important part: the realization of acoustic filtering (vocal-tract model). With the given speech production model the acoustical filtering effect can be realized as a linear difference equation. A generally used mathematical description that accounts for filtering can be described as [1]:

$$s[n] = e[n] - \sum_{i=1}^{M} a_i[n]\, s[n-i]\,. \tag{1.2}$$

For the characteristics of speech and simplicity, the so called *autoregressive model* (1.2) has been shown to have sufficient closeness to the vocal-tract model of speech production.

In the z-transform domain, we have the following approximation of the vocal-tract filtering and the rdiation model for speech production [2]:

$$S(z) = E(z) \cdot H(z) \cdot R(z)\,. \tag{1.3}$$

where $E(z)$, $H(z)$ and R(z) are the Z-transforms of excitation, acoustical filtering and the acoustic radiation. This relationship can be shown in Figure 1.1 as well.

1.2 Purpose of SUV Speech Classification

SUV speech classification is a pattern recognition problem in which features (characteristics) of the speech signal are used to classify speech frames into three different classes. The features are used as inputs to the SUV speech classification system that uses a classification algorithm to determine the class of the segment. For SUV speech classification the speech signal is segmented into small frames of fixed length, the values of the features are calculated for each frame and passed as input to the classification algorithm. Typically frames of length 20 to 40 ms are used [2], [3]. The features and types of

classes for classification depend on the requirements of the application. For example, if an application needs to distinguish speech from other types of signals as required in VoIP applications then the audio signal can be classified into two classes namely speech and non-speech. Some applications might need to further classify speech into voiced and unvoiced speech. For example, this is required in source localization applications where the source localization algorithm is applied only on voiced sections of speech. Voiced speech is typically modeled as a deterministic waveform, while unvoiced speech is generally a stochastic waveform. If the source localization application requires noise cancellation for recording the speech then the speech signal is classified into three categories, namely noise (speech silent), voiced speech and unvoiced speech.

The features of the speech signal that can be used for SUV classification include [4]:

- The short term energy of the signal is the sum of the squares of the amplitude of the signal samples in a frame. Short term energy can be used to distinguish between voiced and unvoiced parts of speech.

- Zero crossing measurements give the number of times a signal has changed its sign within the frame. Different types of speech signals have different zero crossing measurements. Generally it is expected that voiced speech will have lower zero crossing measurements when compared to unvoiced signals.

- The ratio of low-frequency band energy to high frequency band energy can be used as a feature for classification.

- Voiced speech signals are usually low frequency signals and unvoiced speech signals are high frequency signals. It is expected that voiced speech will have high energy in the low frequency band (<2 kHz) and unvoiced signals will have high energy in the high frequency band (either 2 to 4 kHz or <4 kHz) signals.

- Higher order statistics (HOS) [5] like skewness and kurtosis of the LPC residual error can also be used as features for classification. Gaussian noise has HOS that

is different compared to the speech signal. Hence these HOS features can be used to distinguish speech signals from Gaussian noise.

Most of the vowels and a portion of consonants are typically treated as voiced speech (deterministic waveforms). A portion of the consonants are treated as unvoiced speech for its noise-like (stochastic) characteristic. However, there is no a strong relationship between voiced/unvoiced speech and phonemics. The classification of voiced/unvoiced speech is based on the presence of vocal folds vibration. For research on SUV speech classification, people usually measure the vocal folds vibration by recording the *Electroglottogram* (EGG) signals. EGG will be introduce in the following section to detect the presence of voice activity.

A major challenge in designing a SUV speech classification system is identifying the features that can be used to effectively classify the speech signal into different classes. A good feature set will ensure that there is no overlap between the classes in the multi-dimensional feature space. Statistical feature selection methods can be used to select an optimal feature subset from a given feature set. These feature selection methods can be used to remove features that do not make a significant contribution toward the speech classification.

1.3 Electroglottograph (EGG) Signal

Electroglottograph (EGG) is a system which provides information on the closure of vocal folds by measuring the electrical resistance between two electrodes placed around the neck. Even if the signal provides only an approximate value of the glottal surface, it provides a very good information about the period of vocal folds vibration, since we are at the source of the phonation (no influence of the vocal tract) and because there is no turbulent noise. In general, an EGG signal is composed of a high-frequency component which is relative to the vocal folds vibration (voice) and a low-frequency

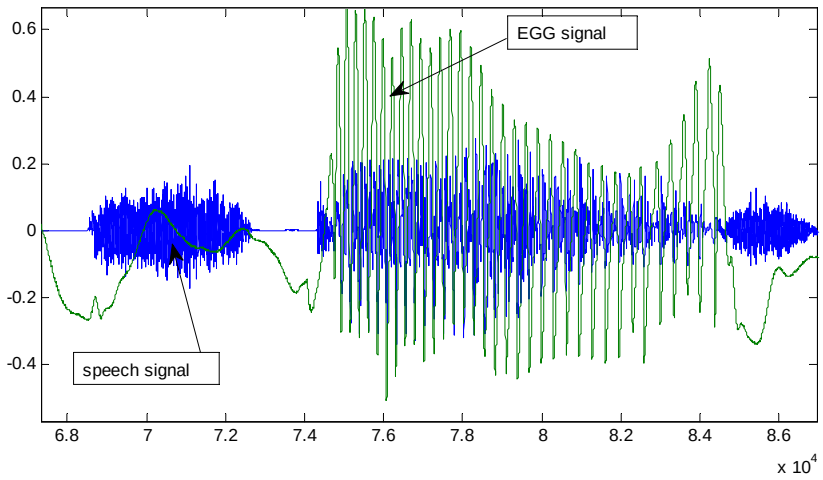

Figure 1.2. The Egg signal and speech signal can be shown in this figure.

component which is relative to slow movement of the larynx. For a good analysis of the EGG signal as an indications for voiced signals we must bandpass filter the EGG signal.

1.4 Current Development of SUV Speech Classification

Research of SUV speech classification began to prosper when Bishnu and Rabiner utilized the zero-crossing rate, speech energy, correlation, first LPC coefficients and the energy in prediction error to train a Gaussian SUV speech classifier [6]. The speech segment is assigned to a particular class based on a minimum distance rule obtained under the assumption that the measured parameters are distributed according to the multidimensional Gaussian probability density function. Then in 1977 Rabiner and Sambur proposed a LPC distance measure based SUV speech classification mechanism [7]. An average speech spectrum is obtained from the training set for each of those three classes of speech. The SUV speech is classified according to the average speech spectrum of each class.

In 1980, Un and Lee proposed a delta-modulation based SUV speech classification algorithm [8]. Classification is made depending upon the counting bit alternations of the

bit stream from linear delta modulation (LDM) of the speech signal and zero crossing rate of a bandpass filtered output of the decoded LDM signal. The classification decision is made by two predefined thresholds. In the same year, Cox and Timothy suggested a nonparmetric rank-order statistics based algorithm to deal with the problem of SUV speech classification in practical condition [9]. In 1987 Bruno *et al.* proposed a probabilistic Bayesian approach based [10] method. This approach applied the maximum posteriori probability criterion with an adaptive updating probability density function estimated in the training phase.

In 1989, Childers and Hahn *et al.* tried to process the Electroglottogram (EGG) signal as an indication of voiced, unvoiced, mixed (a combination of voiced and unvoiced) and silent speech [11]. In this method, they empirically set up a threshold for the measurement of the level crossing rate (LCR) and the energy of the EGG signal (DEGG) to determine the speech class. Qi and Hunt in 1993 proposed a SUV speech classification algorithm. They employed a hybrid cepstral coefficients and waveform features to train a feed-forward neural network classifier to achieve the discrimination. They can be considered as pioneers who successfully applied an artificial neural network to the problem of SUV speech classification [12].

There were less major publications concerning the research topic of SUV speech classification between 1993 and 2002 until 2003. Lobo and Loizou proposed a new algorithm developed for voiced-unvoiced speech discrimination in noise by using short segments (3.2ms) of speech modeled as a sum of basis functions from a Gabor dictionary [13]. In each iteration, a Gabor atom is fitted (using the matching pursuit algorithm) to the residual obtained by subtracting the best-fit Gabor atom from the previous residual. A Radial Basis function neural network is trained on the reduced feature vector set to discriminate between voiced and unvoiced and speech silence segments.

In 2005 Umapathy and Krishnan proposed a joint time-frequency approach for classifying pathological voices [14]. The speech signals were decomposed using an adaptive time-frequency transform algorithm, and features were extracted from the decomposi-

tion parameters and analyzed using statistical pattern classification techniques. The decisions of the segmentation are made by a linear discriminant based classifier. This paper is a more recent voiced-speech-detection related journal publication in the IEEE transactions. After the voiced detection mechanism, Tuan and Kubin proposed a discrete wavelet transform (DWT)-based phonetic classification algorithm that employs a neural network in 2005 [15]. The propsed mechanism can classify speech into more detail phonetic groups such as transient, voiced vowel, voiced consonant and unvoiced consonant categories in noise-free conditions. Then they proposed a modified scheme in 2006 which is a DWT based low-complexity and efficient classification method that is able to discriminate speech under noise interference [16]. This paper can be compared favorably with the past development of SUV classifier, and we are going to compare their performance with ours of proposed mechanism.

1.5 Research Objectives

A large number of the published journal and conference papers listed in the previous section tends to focus on the performance of speech SUV classification or voice activity detection in noise free condition. There are few algorithms able to obtain outstanding classification rates below an SNR of 10 dB. For the reason above, we decided to seek an algorithm which provides an improved classification rate under severe noise corruption.

With a large number of literature reviewing we notice that the waveform features such as short-term spectro-temporal autocorrelation, zero crossing rate, spectrum tilt, pre-emphasized energy ratio, low-band to full band energy ratio, and short-term energy are extensively used in various combinations and algorithms in published papers and ITU series standards. That means those waveform features can really separate the speech signal well in silent, voiced and unvoiced sections. At the beginning of our research, we found that the SUV classes of speech signal are well-separated this the multidimensional feature space. Then we decide to apply one technique of principal component analysis

called *Fisher* linear discriminant [17] to detect presence of voiced speech by linearly projecting the selected features onto a vector. With the optimal threshold obtained in the training phase we can simply detect the presence of voiced speech signal. However, Fisher linear discriminant based voiced speech detection can only classify speech signals into two categories: voiced and unvoiced/silent. That restriction limits the applications of the speech classification with only two classes of speech labeling. Then we tried to extend our research to silent/unvoiced/voiced three-way classification. We applied a *Gaussian* mixture density to model these three class of speech in the training part and used a *Gaussian mixture model* technique [17] to decide the most probable class. Details of this mechanism will be presented in chapter 2.

The classification accuracy of the GMM based multi-feature classifier will degrade much with increasing levels of noise energy ,especially when the SNR is below 10dB. We planned to design a more noise-robust speech classifier which is inspired by inventory-based speech enhancement technique with hidden Markov models, which is widely used in speech recognition technology. With a lot of effort put into designing the mechanism and performing the experiments we found that the SUV classification accuracy compares very favorably to the state-of-the-art discrete-time wavelet based SUV speech classifier and GMM based multi-feature classifier.

1.6 Thesis Organization

Chapter 1 will give readers an introduction to the fundamentals of speech production, the purpose of SUV speech classification, past and current SUV speech classification and related developments published in the $IEEE$ transactions journal and major conferences such as $ICASSP$ and $Interspeech$. At the end of this chapter we enumerate our inspiration of the proposed mechanisms from a two-way voice speech detection based on Fisher linear discriminant, then a three-way SUV speech classification with Gaussian mixture models, and finally the mel-frequency cepstrum coefficients code book based SUV speech

classification with hidden Markov model.

We are going to present the waveform features first in chapter 2 including short-term spectro-temporal autocorrelation, speech peakiness, zero crossing rate, spectrum tilt, pre-emphasized energy ratio, low-band to full-band ratio and short-term energy. Then we will give a full derivation of the proposed mechanisms of Fisher linear discriminant and the Gaussian mixture models.

We found that mel-frequency cepstrum coefficients (MFCC) are extensively used in speech recognition research. For that reason we borrow the concepts of phoneme recognition, then transfer phoneme classification into a broader concept of classification, namely the three way silent, unvoice and voiced classification. The computation and derivation of the feature vector MFCC, codebook design and clustering algorithm will be discussed in the chapter 3.

Chapter 4 will present the modified version of the proposed MFCC codebook search based SUV classification with aid of a hidden Markov model. The content will include the introduction of codebook design under noise, transition/emission statistics, training of hidden Markov models and the block diagram of MFCC Inventory-based SUV speech Classification with hidden Markov model. Chapter 5 will evaluate the performance of a number of proposed schemes and compare them with a state-of-the-art DWT based mechanism.

Chapter 2

SUV Classification based on Waveform Features and Pattern Recognition Techniques

A block diagram for a general model of speech feature extraction and classification is shown in Figure 2.1. Speech type classification can be done in two parallel processes of training and testing, which can be divided into several stages. In the training part we can provide a reference for feature comparison, feature extraction, and class modeling. While in the testing part, with the decision making rule we can discriminate speech into several categories corresponding to the clustering in the training part.

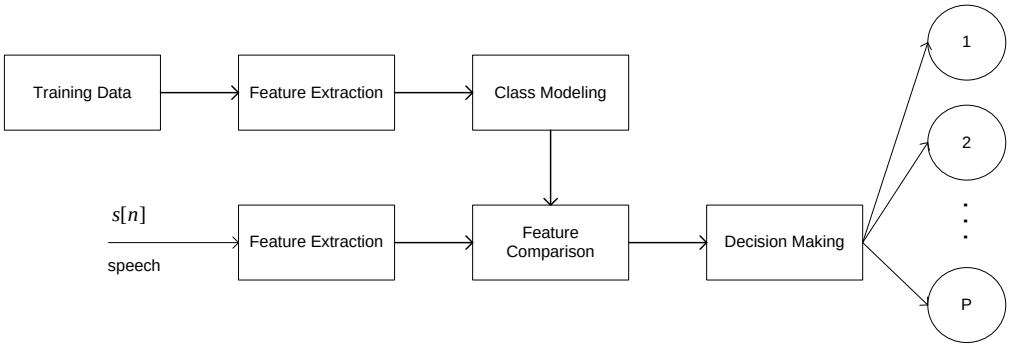

Figure 2.1. A block diagram that illustrates the general process for feature extraction, training, and pattern comparison with decision making [1].

2.1 Features Extraction

Waveform Features are extracted from a short-term analysis with a sliding window $w[n]$ (*Hanning* window). The window size is defined with 20ms with 50% overlaping, which means speech frames are shifted by 10ms. The overlapping of speech frames increases the computational load but lowers the impact of the variability of speech.

2.1.1 Short-Term Spectro-Temporal Autocorrelation

The the measurement of periodicity in each frame is estimated by the spectro-temporal autocorrelation function, the following expression can describe the mathematical definition of this analysis [4]:

$$R_n[\tau] = \beta \cdot R_T[\tau] + (1-\beta) \cdot R_S[\tau]. \tag{2.1}$$

If we let $s_t[k]$ and $S_f(\omega)$ correspond to the speech signal in time and frequency domains respectively, then R_T (the temporal autocorrelation) and R_S (the spectral autocorrelation) are defined by:

$$R_T[\tau] = \frac{\sum_{k=0}^{N-\tau-1}[s_t[k]\, s_t[k+\tau]]}{\sqrt{\sum_{n=0}^{K-\tau-1} s_t^2[k] \sum_{k=0}^{K-\tau-1} s_t^2[k+\tau]}}. \tag{2.2}$$

$$R_S[\tau] = \frac{\int_0^{\pi-\omega_r} S_f(\omega)\, S_f(\omega+\omega_r)}{\sqrt{\int_0^{\pi-\omega_r} S_f^2(\omega) \int_0^{\pi-\omega_r} S_f^2(\omega+\omega_r)}}. \tag{2.3}$$

Where β is set 0.5 in the realization, this measurement can determine the degree of periodicity of each frame. The higher the degree of periodicity the higher the possibility of a voiced frame. $R_n[\tau]$ is the temporal autocorrelation measurement of each frame, the *maximum* value of the measurement in each frame will be treated as the degree of

periodicity $PROD[n]$ (2.4), in shich n denotes the frame index.

$$PROD\,[n] = max\,\{R_n[\tau]\}\,. \tag{2.4}$$

2.1.2 Peakiness of Speech

Periodic or voiced speech contains regular pulses which do not appear in unvoiced speech. This feature is described as peakiness of speech and it can be used to identify voiced speech when it has relatively high value. In order to enhance the peakiness, the linear prediction residual $e\,[n]$ can be used to compute its value [4].

$$PEAK\,[n] = \frac{\sqrt[\xi]{\sum_{m=-\infty}^{\infty} w\,[m-n+M]\cdot|\hat{e}\,[n]|^{\xi}}}{\sum_{m=-\infty}^{\infty} w\,[m-n+M]\cdot|\hat{e}\,[n]|}. \tag{2.5}$$

The so called residual signal $\hat{e}\,[n]$ can be estimated from equation (1.2):

$$\hat{e}\,[n] = s\,[n] + \sum_{i=1}^{M} \tilde{a}_i\,[n]\,s\,[n-1]\,. \tag{2.6}$$

The peakiness measure applies the nonlinearity defined by the exponent $\xi > 1$ (generally $\xi = 2$) to emphasize the relative weight of peaks. $w\,[n]$ introduces the pre-defined $Hanning$ window.

2.1.3 Zero Crossing Rate

Unvoiced speech has the characteristics of a noise process, which means that the number of times the signal crosses the zero line is significantly higher than with the voiced part of speech, which has a much slower zero-crossing rate. The zero-crossing rate also depends on the pitch of the signal if the frame is voiced. For example, the zero-crossing rate of

voiced female speech is higher than that of voiced male speech [4].

$$STZC\left[n\right]=\frac{1}{2}\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\times\left|\text{sign}\left(s\left[m\right]\right)-\text{sign}\left(s\left[m-1\right]\right)\right|. \tag{2.7}$$

The notation $\text{sign}\left(x\right)$ is defined as:

$$\text{sign}\left(x\right)=\begin{cases}+1 & \text{,for } x\geq 0\\ -1 & \text{,for } x<0\end{cases} \tag{2.8}$$

The zero crossing rate counts the number of transitions from positive samples to negative ones withing the range of the sliding window $w\left[n\right]$.

2.1.4 Spectral Tilt

Voiced speech has higher energy in low frequencies and unvoiced speech usually has higher energy in high frequencies resulting in opposite spectral tilts. The spectral tilt can be represented by the first order normalized autocorrelation of the first reflection coefficient [4].

$$TILT\left[n\right]=\frac{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot s\left[m\right]\cdot s\left[m-1\right]}{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot s^{2}\left[m\right]}. \tag{2.9}$$

This is a very reliable parameter especially for plosive detection and to avoid individual spikes in low-level signals. Its ability to indicate unvoiced and voiced sounds in general is also very accurate.

2.1.5 Pre-Emphasized Energy Ratio

Voiced and Unvoiced speech can be discriminated via the normalized pre-emphasized energy ratio [4].

$$PEER\left[n\right]=\frac{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot\left\{s\left[m\right]-s\left[m-1\right]\right\}}{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot\left|s\left[m\right]\right|}. \tag{2.10}$$

The variance of the difference between adjacent samples is usually much lower in voiced regions than in unvoiced regions. The first-order correlation of voiced samples is around 0.85 but that of unvoiced samples is nearly zero, which is a clear indication of the voiced-unvoiced discriminatory characteristic of this parameter.

2.1.6 Low-Band to Full Band Energy Ratio

Voiced speech usually has higher low-frequency energy than unvoiced speech. Therefore the energy ratio of the first 1 kHz to the full-band energy can give a good indication whether the speech is voiced. When voiced, the energy ratio is close to one and when unvoiced, since the low-band energy is significantly smaller, the ratio will be less than one.The low-band to full-band energy ratio is derived as followed [4]:

$$LBFB\left[n\right]=\frac{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot s_{LP}^{2}\left[m\right]}{\sum_{m=-\infty}^{\infty}w\left[m-n+M\right]\cdot s^{2}\left[m\right]}. \tag{2.11}$$

Where $s_{LP}\left[n\right]$ is the low-pass filtered speech signal cut off at 1 kHz.

2.1.7 Short-Term Energy Measure

Because the voiced speech tends to have larger short-term energy than the unvoiced speech, we can employ this parameter as a judge criterion to classify speech signal between voiced and unvoiced classes. We can obtain very high classification accuracy

especially in high SNR environments, *i.e.* with mild noise interference [4].

$$STEM\,[n] = \sum_{m=-\infty}^{\infty} w\,[m-n+M] \cdot s^2\,[m]\,. \tag{2.12}$$

Once all the features have been computed, we can combine all the features in a *feature vector* $x\,[n]$ (2.13) for the speech classification mechanisms which are going to be introduced in the following sections.

$$x\,[n] = [PROD\,[n]\ \ PEAK\,[n]\ \ STZC\,[n]\ \ TILT\,[n]\ \ PEER\,[n]\ \ LBFB\,[n]\ \ STEM\,[n]]. \tag{2.13}$$

2.2 Pattern Comparison and Speech Recognition

At the beginning, we tried to simplify the problem of SUV speech classification by reducing the class number from three to only two classes: (*i*) voiced sections and (*ii*) a combination of unvoiced and silent sections. For a two class classification, a single linear discriminant function called *Fisher linear discriminant* can be used with sufficient training to find the suitable decision rule. To locate the voiced part of speech is very useful in speech activity detection which can be used in wireless mobile communication, pitch estimation and many other applications mentioned in chapter 1. Some of these kinds of techniques are standardized in an ITU series [18]. Our research goal is to separate speech into silent/unvoiced/voiced part. In general, *Gaussian mixture models* are a widely used and to better represent the statistical characteristics of a feature space for each class. Therefore, GMM is adopted in the research to model the statistics for the feature characteristics of each class.

2.2.1 Fisher Linear Discriminant

In general, Fisher linear discriminant determines a weight vector w which projects the feature sets onto a line. The projected feature sets on this particular line can be well separated in the sense of minimizing the variance around each class centroid and maximization distance of centroids from the two classes. Figure 2.2 graphically introduce this general idea.

We begin by considering the problem of projecting data from d dimensions onto a line. Of course, even if the samples formed well-separated, compact clusters in d-space, projection onto an arbitrary line will usually produce a overlapping mixture of samples from all of the classes and thus produce poor recognition performance. However, by moving the line around, we might be able to find an orientation for which the projected samples are well separated. This is the goal of classical discriminant analysis.

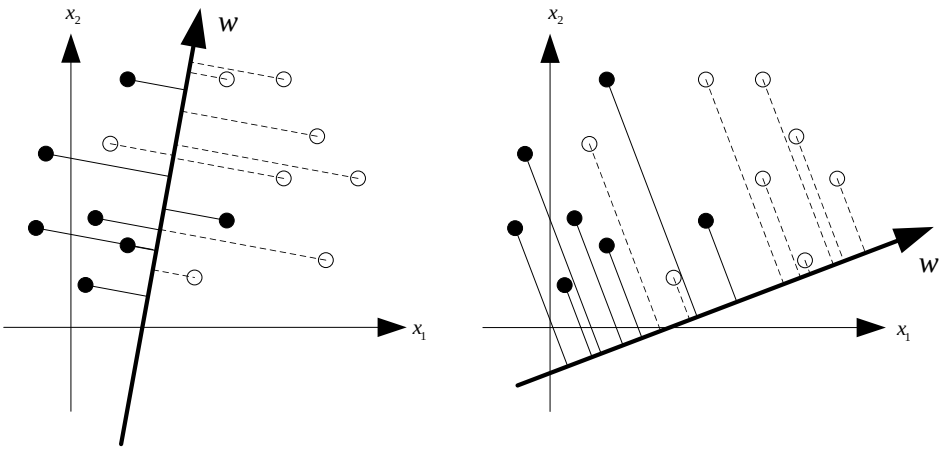

Figure 2.2. Projection of the same set of samples onto different lines in the direction marked w. The figure on the right shows superior separation between the black and white circles

By definition, Fisher linear discriminant employs the linear function $\mathbf{w}^t\mathbf{x}$ for which the criterion function [17]

$$\mathbf{J}(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{2.14}$$

is maximized (and independent of $|\mathbf{w}|$). $\tilde{m}_i$ and $\tilde{s}_i$ represent the sample mean of the projected points and scatter of the projected samples labeled by C_i. While the $\mathbf{w}$ maximizing $\mathbf{J}(\cdot)$ leads to the best separation between the two projected sets (in the sense just described), after finding the optimal $\mathbf{w}$ an optimal threshold T can be determined through the training data based on the least error of the decision making.

One can show that in, we can rewrite the cost function $\mathbf{J}$ as [17]:

$$\mathbf{J}(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_W \mathbf{w}}. \tag{2.15}$$

Where $\mathbf{S}_W$ is the $within-class\ scatter\ matrix$, $\mathbf{S}_B$ the $between-class\ scatter\ matrix$ and $\mathbf{m}_i$ is the sample mean of class C_i:

$$\mathbf{S}_W = \sum_{\mathbf{x}\in C_1} (\mathbf{x}-\mathbf{m}_1)(\mathbf{x}-\mathbf{m}_2)^t + \sum_{\mathbf{x}\in C_2} (\mathbf{x}-\mathbf{m}_1)(\mathbf{x}-\mathbf{m}_2)^t. \tag{2.16}$$

and

$$\mathbf{S}_B = (\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^t. \tag{2.17}$$

Equation (2.15) is well known in mathematical physics as the generalized Rayleigh quotient. Equation (2.18) can optimize the cost function in (2.15) [17]

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1-\mathbf{m}_2). \tag{2.18}$$

Thus, we have obtained $\mathbf{w}$ for Fisher linear discriminant: The linear function yielding the maximum ratio of between-class scatter to within-class scatter. The classification has been converted from a multi-dimensional problem to a more manageable one-dimensional one. This mapping is many to one, and in theory it cannot possibly reduce the minimum achievable error rate if we have a very large training set. In general one is willing to sacrifice some of the theoretically attainable performance for the advantage of working in one-dimension. All that remains is to find the threshold, that is, the point along the one dimensional subspace taht separates the projected point sets.

2.2.2 Gaussian Mixture Model

One of the disadvantages of Fisher linear discriminant is that the decision boundary between two classes are restricted to hyperplane in the feature space. We try to further extend our experiment from two-class to a three-class SUV classification mechanism. Therefore a more complicated technique that permits the definition of more flexible decision boundaries is given by *Gaussian mixture model* (GMM) [3] [17] method.

First, separation of the whole data set into several subsets with certain rules is needed. Each subset is mathematically represented by a mixture of Gaussian distributions. Suppose there are k clusters in a subset, each cluster (component) is a Gaussian distribution parameterized by μ_k and Σ_k which correspond to the mean and the covariance matrix of the multidimensional Gaussian distribution f_k. The density of component k is [17]:

$$f_k(x) = \frac{1}{\left(2\pi^{d/2}\left|\Sigma_k\right|^{1/2}\right)} exp\left(-\frac{1}{2}(x-\mu_k)^t \Sigma_k (x-\mu_k)^t\right). \tag{2.19}$$

The prior probability (weight) of component k is a_k. The mixture density can be described as follow, where c is the label of the subset,

$$g_c(x) = \sum_{k=1}^{K} a_k f_k(x). \tag{2.20}$$

The expectation maximization (EM) algorithm (The EM algorithm is summerized in Appendix A.)is typically used to estimate the model parameters from a training set.

2.3 Proposed SUV Scheme and Implementation

Figure 2.3 shows the implementation of the proposed Gaussian mixture model based SUV speech classification. First, it is necessary to divide the training set of speech into three silent, unvoiced and voiced subsets. Then we can find the GMM for each class with the EM algorithm. By comparing the value of $f_v(x[n])$, $f_u(x[n])$ and $f_s(x[n])$ the speech class can be determined with the maximum likelihood of the mixture Gaussian,

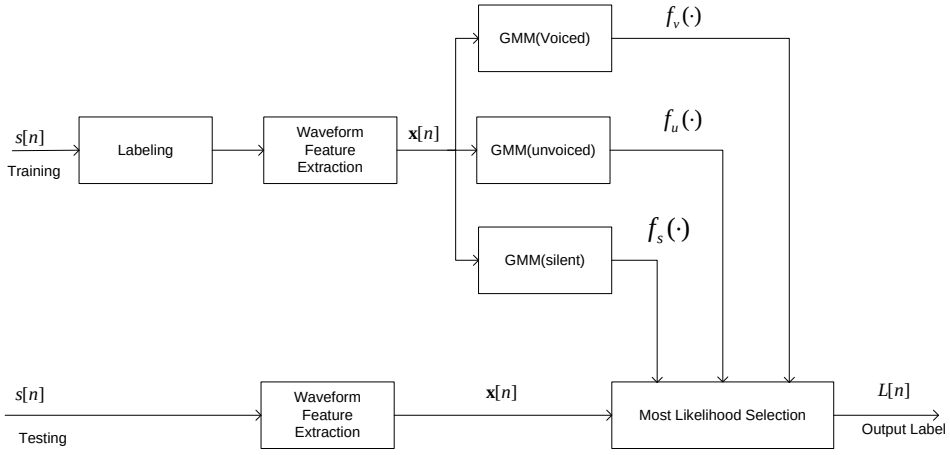

Figure 2.3. This diagram shows the process of GMM based SUV classification. $L[n]$ is the output label for the testing speech.

The mechanism can as well work under noise with training set of data in the same noise level, the performance analysis of GMM based SUV speech classification will be discussed in chapter 5, and will be compared with DWT based, MFCC codebook based and HMM based MFCC codebook inventory search SUV speech classification mechanisms.

Chapter 3

Mel-Frequency Cepstrum Coefficients Codebook Search

After introducing the linear discriminant based and GMM based speech classification techniques, it is our next goal to improve the classification accuracy at low SNR conditions. Through literature research from papers in the area of speech recognition, we found that some cepstrum coefficients are widely used in the speech recognition research community. There are different way to compute cepstrum coefficients. Particularly important in speech signal processing $mel\text{-}frequency\ cepstrum\ coefficients$. They are very effective features with two important characteristics: (i) they focus on signal properties that are highly useful for a given detection/classification task and (ii) they reduce the overall amount of data by discarding irrelevant information.

3.1 Mel-Frequency Cepstrum Coefficients

An analysis of the human ear auditory system shows several observations through experiment: (i) hearing is favorably described in a 2-dimensional time-frequency domain, (ii) human ears tend to be somewhat phase-insensitive, (iii) the perception of sound intensity is not linear, and (iv) the perception of frequency is not linear.

To calculate the mel-frequency cepstrum coefficients, first, it is necessary to compute

the short-term power spectrum $S[n,k]$:

$$S\left[n,k\right]=\left|\sum_{m=-\infty}^{\infty}w\left[m-n\right]\cdot s[m]\cdot e^{-j2\pi km/R}\right|,\text{for all } n \text{ and } k\text{=0, 1, ..., } R\text{-1.} \quad (3.1)$$

The short-term power spectrum is the magnitude squared of the discrete Fourier transform of windowed sections of the speech signal $s[n]$. The window is chosen as a *Hanning* window (3.2) in this research and 20 ms window size is selected with 10ms overlapping.

$$w\left[n\right]=0.5\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right). \quad (3.2)$$

The observation that frequency perception is nonlinear can be modeled by *frequency warping*. A widely used warping function that resulted from averaging a large number of subjective frequency perception experiments is the mel-frequency mapping [1]:

$$\text{mel}\left(f\right)=\begin{cases}2595.0376\cdot\log_{10}\left(1+\frac{f}{700}\right) & \text{for } x>1000\\ f & \text{for } x\leq 1000\end{cases} \quad (3.3)$$

The mel-function assigns a linear mapping to frequencies below 1 kHz and a logarithmic mapping to frequencies above 1 kHz. Other than that, we still need to define a set of edge and center frequencies f_m for the bands that are linearly aligned on the mel-scale, this nonlinear alignment on the frequency scale can be shown as [1]:

$$f_m=\text{mel}^{-1}(\frac{m}{Q+1}\cdot\text{mel}(\frac{F_s}{2})),\text{ for m=0, 1, ..., Q+1.} \quad (3.4)$$

We define a set of Q overlapping weighting functions $H_m(f)$ for $m=1...Q$. Figure 3.1 will demonstrate a set of 24 triangular shaped weighting function between 0 and 4 kHz. For the discrete frequency variable $k=R\cdot\frac{f}{F_s}$ from equation (3.1) the triangular filters

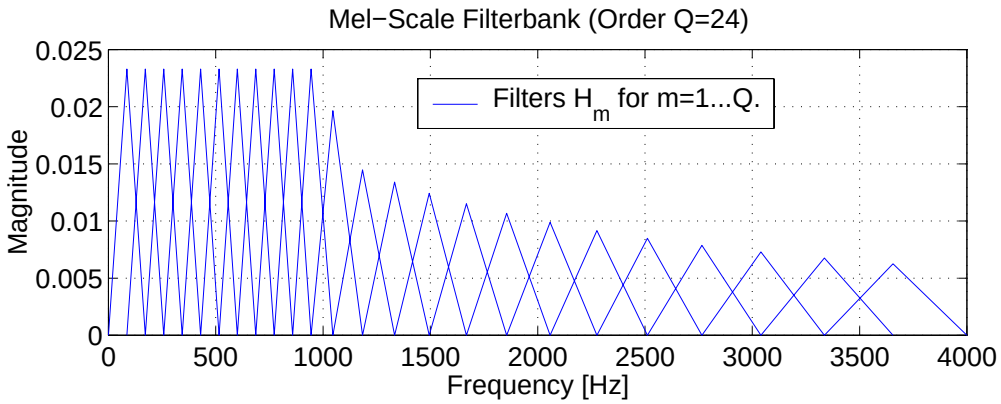

Figure 3.1. The mel-scale filterbank for 24 filters between 0 and 4kHz. The filters have a constant bandwidth for center frequencies up to 1 kHz and an increasing bandwidth above 1 kHz. The bandwidth increase is motivated by psychoacoustic effects.

are represented by [1]:

$$H_m[k] = \begin{cases} \frac{2(F_s k/R - f_{m-1})}{(f_{m+1} - f_{m-1})(f_m - f_{m-1})} & \text{for } f_{m-1} \leq Fs \cdot k/R \leq f_m \\ \frac{2(F_{m+1} - F_s k/R)}{(f_{m+1} - f_{m-1})(f_{m+1} - f_m)} & \text{for } f_{m-1} \leq Fs \cdot k/R \leq f_m \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

For the reason of simplicity, we choose the number of available spectral samples R to be even. We can define the *natural log mel − averaged power spectrum* $S[n, m]$ [1]:

$$\tilde{S}[n, m] = \ln \left(\sum_{k=0}^{R/2} S[n, k] \cdot H_m[k] \right), \text{ for all } n \text{ and } m = 1 \ldots Q. \tag{3.6}$$

The natural logarithm in equation (3.6) is inspired by the human ear's non-linear sound intensity perception. A logarithmic mapping more closely represents a natural perception of loudness.

The mel-frequency cepstrum coefficients (MFCCs) can be computed with [1]:

$$c[n, k] = \sum_{m=1}^{Q} \widetilde{S}[n, m] \cdot \cos \left(k \left(m - \frac{1}{2} \right) \frac{\pi}{Q} \right), \text{ for all } n \text{ and } k = 0 \ldots Q. \tag{3.7}$$

The MFCCs are computed through $\tilde{S}[n, m]$ via a discrete cosine transform (DCT). The DCT provides a decorrelation of the coefficients, which is advantageous for classification issues.

Cepstral analysis has widely received a lot of attention in the speech processing research area. The most important advantage of cepstral techniques are their inherent ability to separate the excitation signal from from the spectral envelope of the acoustical filtering.

3.2 Vector Quantization

First we denote the MFCCs of the speech signal as $\mathbf{c}_l, l = 1 \ldots L$, where each vector is a p-dimensional vector. Vector quantization is widely used in speech processing since it can significantly reduce the information rate of a vector representation. Furthermore, computational complexity will also be reduced in comparison with directly processing the raw (uncoded) speech data. For example, in our research the speech data is sampled as 16kHz and stored with 16-bits speech amplitudes. The information rate of the raw speech signal requires 256 kbps for storage in an uncompressed format. In our case the spectral analysis dimension p is chosen as 13 and the speech frame length is 20 ms with 10 ms overlapping (which means there are 100 vectors in one second). That result is 13×16×100 bps, or 20.8 kbps which is a 12.3-to-1 reduction over the uncompressed signal. Such compression in storage rate is impressive. Based on the concept of ultimately needing only a single MFCC for each basic speech unit, it may be possible to further reduce the raw MFCC vectors to those drawn from a small, finite number of *unique* MFCC vectors (i.e. phonemes), each corresponding to one of the basic speech units. The concept of building a *codebook* of distinct analysis vectors is motivated from the research of speech recognition with a basic set of phonemes. Based on the reasoning above, we need to build a codebook with 58 MFCC codewords (we defer the discussion of code word number selection to the next subsection). Then to represent an arbitrary

MFCC vector all we need is the index of the codebook vector that best matches the input vector. Assuming a rate of 100 spectral vectors per second, we see that a total bit rate of about 600 bps is required to represent the MFCC vectors of a speech signal. This rate is about $1/27^{th}$ the rate required by the continuous MFCC vectors. Hence the VQ representation is potentially a very efficient representation of spectral information in the speech signal.

Before discussing the concepts involved in designing and implementing a practical VQ system, There are advantages of this type of representation:

- The storage of spectral analysis is reduced.
- Based on the VQ quantization, the MFCC vector similarity computation is reduced to a table lookup similarities between pairs of codebook vectors
- By associating a *phonetic label* with each codebook vector, the process of choosing a best codebook vector to represent a given spectral vector becomes equivalent to assigning a phonetic label to each spectral frame of speech.

The disadvantage of the use of VQ codebook to represent MFCC analysis of speech:

- Since there is only a finite number of code book vectors, the process of choosing the best representation of a given MFCC vector inherently is equivalent to quantizing the vector and leads to a certain degree of quantization error.
- The larger codebook we choose the more storage is required for the codebook entries and the more computational cost we incur.

3.3 Implementation of MFCC Codebook design

1. A large set of MFCC vectors, $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_L$ forms the training set. The training set is used to create the optimal set of codebook vectors for representing the MFCC vector variability observed in the training set. We assume the MFCCs of the three classes silent, unvoiced and voiced are well-separated with each other. We perform a k-means algorithm on the voiced and the unvoiced part of the data with codebook sizes 39 and 9. The codebook of 39 and 9 are based on the number of phonemes defined in the voiced and unvoiced speech. The definition can be found in the standard phonemes of American English into broad sound classes. The codebook size 10 for silent frames can be treated as a minimum representation of silence/noise characteristics.

2. It is necessary to compute a measure of similarity, or distance between a pair of an incoming test MFCC vector and a certain codeword vector in the codebook. We denote the MFCC vector similarity by $d(\mathbf{c}_i, \mathbf{c}_j)$ between $\mathbf{c}_i$ an $\mathbf{c}_j$. The discussion of MFCC vector similarity measurement will be presented in the following subsection.

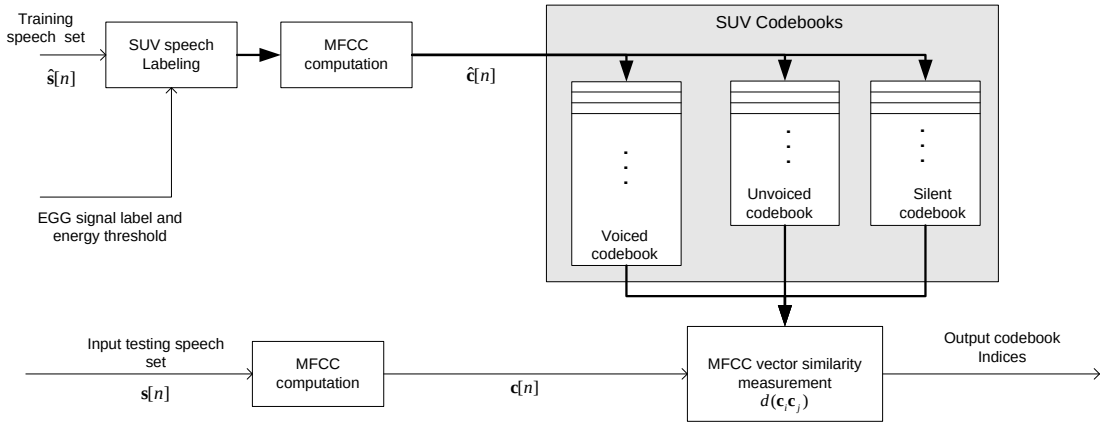

Figure 3.2. This figure shows the whole idea of MFCC codebook search based SUV classification in clean conditions.

3.3.1 MFCC Vector Classification Procedure

The classification procedure for arbitrary incoming test MFCC vectors and a certain codeword requires basically a full search through the three silent, unvoiced and voiced codebooks to find the best natch. We define the codebook vectors of the 58-vector codebook as $\mathbf{y}_m$, $1 \leq m \leq 58$, and we denote the MFCC vector to be classified as $\mathbf{c}$, then the index, m^*, of the best code book entry is

$$m^* = \arg \min_{1 \leq m \leq 58} d\left(\mathbf{c}, \mathbf{y}_m\right). \tag{3.8}$$

After obtaining the optimal codebook index m^* we can refer m^* to the class that the codeword belongs to. We define that codewords 1-39 belong to the voiced class, 40-48 belong to the unvoiced class and 49-58 belong to the silent/noise class. For example, if we have $m^* = 42$, then the vector will be classified as unvoiced.

In clean conditions, we can define a distance measurement via the Euclidean distance, which simply can be represented as

$$d\left(\mathbf{c}, \mathbf{y}_m\right) = \sum_{k=1}^{13} \left(c\left[k\right] - y_m\left[k\right]\right)^2, \text{ where } k = 1 \ldots 13 \text{ denotes the dimension index.} \tag{3.9}$$

In general, Euclidean distance measurements can achieve great classification accuracy. While in the conditions under noise the pursuit of a robust cepstral vector similarity measurement can be more effective if we employ an analytical understanding of effects of noise upon MFCC vectors. Mansour and Juang [19] reported that the additive white noise causes the cepstral vector norm to shrink but leaves the cepstral vector orientation more or less intact. The vector norm shrinkage is harmful when people calculating the similarity between a pair of spectral vectors when Euclidean distance is used. Since the norm shrinkage was found to be a function of the noise level, the vector norm itself can be used to be the nonuniform weighting for each speech frame in the accumulative distance during dynamic sequence comparison. In [19], the following expression of cepstral

projection measure was suggested as a good choice for noise conditions. We can directly apply this expression to our MFCC vector [19],

$$d\left(\mathbf{c}, \mathbf{y}_m\right) = |\mathbf{c}| \left(1 - \frac{\mathbf{y}_m^t \mathbf{c}}{|\mathbf{y}_m| \, |\mathbf{c}|}\right). \tag{3.10}$$

We can rebuild the codebooks under noise conditions. We are mapping each codeword (in clean conditions) with the corresponding MFCC computed under noise interference. We then find the new cluster centroids under noise. The centroids $\mathbf{y}_m^*$ can be defined as the new codeword under noise:

$$\mathbf{y}_m^* = \frac{1}{K} \sum_{k \in M} \left(\mathbf{c}_m^* \left[k\right]\right). \tag{3.11}$$

Where K denotes the number of MFCC vectors belonging to codeword index m and k is the vector index of of codeword m.

3.3.2 K-means Algorithm

The K-means algorithm is essentially a model free clustering algorithm which can partition a pool of data into K clusters (prototypes). With pre-defined K centroids as input, the classification of a query point x is made to the class of the closest prototype [17]. First we assume there are M prototypes denoted by $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$. Each training sample is assigned to one of the prototypes. Denote the assignment function by $L(\cdot)$. Then $L\left(\mathbf{c}\left[n\right]\right) = m$ indicates the n^{th} training vector that belong to prototype m. The goal of the K-means algorithm is to minimize the total mean squared error between the training vectors and their corresponding clusters, that is, the trace of the training data set within cluster covariance matrix.

$$\arg\min_{Y,L} \sum_{n=1}^{N} \left\| \mathbf{c}[n] - \mathbf{y}_{L(\mathbf{c}[n])} \right\|^2. \tag{3.12}$$

Which can be denoted by

$$G\left(\mathbf{Y}, L\right) = \sum_{n=1}^{N} \left\| \mathbf{c}[n] - \mathbf{y}_{L(\mathbf{c}[n])} \right\|^2 . \tag{3.13}$$

The K-means algorithm tries to find the cluster centroid (prototype) which is surrounded by training vectors. The cluster centroid can provide the role of a compact representation for the training vector. Once the Y is fixed the optimal assignment function $L\left(\cdot\right)$ needs to follow the $nearest - neighbors$ rule, which can be shown as:

$$L\left(\mathbf{c}\left[n\right]\right) = arg \min_{m \in \{1,2,\ldots,m\}} \left\| \mathbf{c}\left[n\right] - \mathbf{y}_m \right\| . \tag{3.14}$$

If $L\left(\cdot\right)$ is fixed the prototype $\mathbf{y}_m$ should be the average (centroid) of all the training vectors assigned to the m_{th} prototype:

$$\mathbf{y}_m = \frac{1}{N_m} \sum_{n:L(\mathbf{c}[n])=m} \mathbf{c}\left[n\right] . \tag{3.15}$$

Where N_m is the number of samples assigned to prototype m. The K-means algorithm can be fully describe by the following two steps:

- With equation (3.14) we can optimize the assigned training vector to their nearest cluster centroid by Euclidean distance.
- With equation (3.15) we can directly compute the updated new cluster centroid by the training vector assigned to it.

Generally speaking, the algorithm may converge with a fast pace. We can set up a threshold test for the termination criterion.

Chapter 4

MFCC Codebook Search With Hidden Markov Model

MFCC codebook based SUV classification can have an accurate classification rate in a noise-free environment. However, we found that the classification accuracy may degrade very quickly with increasing noise levels. We turned our research focus toward *hidden Markov models* (HMM) and showed that with the *transition* and *emission* statistics (probability) enumerated from the HMM the classification rate will significantly improve, which means that frame-by-frame relationships provide sufficient information as a reference for the MFCC codebook search.

4.1 Discrete-time Hidden Markov Model

The term *hidden Markov model* is an extension of a Markov model. In a Markov model the state at any time t can be denoted by $\omega(t)$. A particular sequence of length T can be denoted by ω=$\{\omega_1, \omega_2, \ldots, \omega_T\}$. The basic production of a Markov model is formed by *state transition probabilities*:

$$P\left(\omega_j\left(t+1\right) \mid \omega_i\left(t\right)\right) = a_{ij}. \tag{4.1}$$

The coefficient a_{ij} represents time-independent probability of having the state ω_j at time step $t+1$ given that we had the previous state ω_i at time step t. Generally speaking, this mathematical description can be treated as a first-order discrete time Markov model since the probability at $t+1$ depends only on the state at t.

With the basic assumption of a first-order Markov model above we extend it to a hidden Markov Model, we assume that at every step t the system is in a state $\omega(t)$ but now we also assume that it emits a visible symbol $v(t)$. Originally, a more sophisticated Markov model allowed for the emission of continuous functions (e.g., spectra). We will restrict the model to the case where a discrete symbol is emitted. Again, we define a sequence of such visible states as $\mathbf{V} = \{v(1), v(2), \ldots, v(T)\}$.

For any state $\omega(t)$ we have a probability of emitting a particular visible state $v_k(t)$:

$$P\left(v_k\left(t\right) \mid \omega_j\left(t\right)\right) = b_{jk}. \tag{4.2}$$

We only have access to the visible states and the ω_j are unobservable. This full description is called a hidden Markov model.

4.1.1 Computation and Analysis of Hidden Markov Models

Once we have the state transition probability in equation (4.1) and the state emission probability in equation (4.2), these two probability descriptions obey the normalization condition:

$$\sum_j a_{ij} = 1 \quad \text{for all } i. \tag{4.3}$$

and

$$\sum_k b_{jk} = 1 \quad \text{for all } j. \tag{4.4}$$

Where the limits on the summation are over all hidden states and all visible symbols respectively.

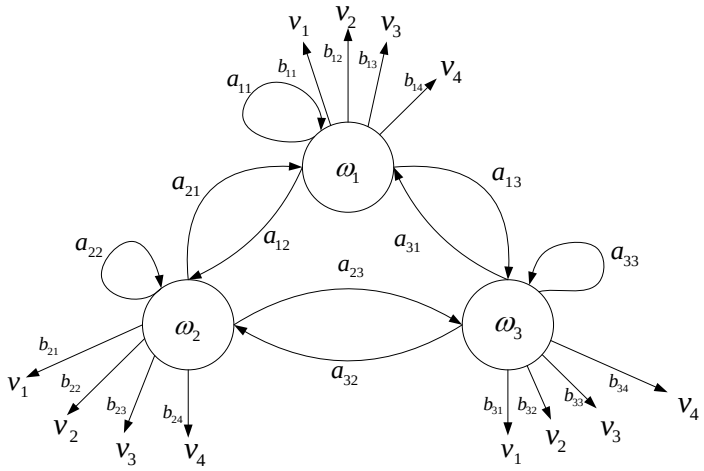

Figure 4.1. This figure illustrates the general idea of a hidden Markov model with transition probability a_{ij} and emission probability b_{jk}. The visible state are noted by v_k

The probability of the observation sequence $\mathbf{V}$ can be described by the following equation

$$P(\mathbf{V}) = \sum_{r=1}^{r_{max}} P(\mathbf{V} \mid \omega_r) P(\omega_r), \tag{4.5}$$

where each r represents a sequence $\omega_r = \{\omega_1, \omega_2, \ldots, \omega_T\}$ of length T. In a general case, for the total number of codewords (hidden states) M, there will be $r_{max} = M^T$ possible terms in the sum of equation (4.5), corresponding to all possible sequences of length T.

Since the hidden Markov model we are dealing with is the first-order one, we can rewrite the transition probability $P(\omega_r)$ in equation (4.5) as:

$$P(\omega_r) = \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1)). \tag{4.6}$$

Equation (4.6) represents a product of the a_{ij}. We can also rewrite emission probability $P(\mathbf{V} \mid \omega_r)$ in equation (4.5) by:

$$P(\mathbf{V} \mid \omega_r) = \sum_{t=1}^{T} P(v(t) \mid \omega(t)), \tag{4.7}$$

that is, a product of b_{jk}. Now, by putting equation (4.6) and equation (4.7) together we

can have equation (4.5) become by the following:

$$P(\mathbf{V}) = \sum_{r=1}^{r_{max}} \prod_{t=1}^{T} P(v(t) \mid \omega(t)) P(\omega(t) \mid \omega(t-1)). \tag{4.8}$$

For a reason of simplicity, we can calculate $P(\mathbf{V})$ recursively. Since each term $P(v(t) \mid \omega(t))$ consists of only $v(t)$, $\omega(t)$ and $\omega(t-1)$. The following $\alpha_j(t)$ can describe this recursive idea,

$$\alpha_j(t) = \begin{cases} 0 & t = 0 \text{ and } j \neq \text{initial state} \\ 1 & t = 0 \text{ and } j = \text{initial state} \\ \left[\sum_i \alpha_i(t-1) a_{ij}\right] b_{jk} v(t) & \text{otherwise,} \end{cases} \tag{4.9}$$

where the term $b_{jk}v(t)$ means the transition probability b_{jk} selected by the visible state emitted at time t. As a result, α_j represents the probability that the HMM is in hidden state ω_j at step t having generated the first t elements of $\mathbf{V}$.

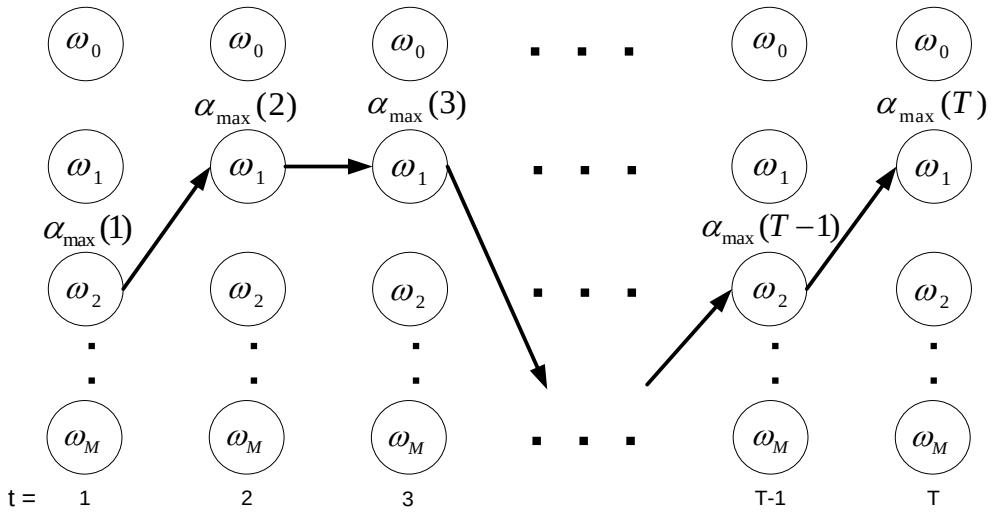

Figure 4.2. This figure illustrates the HMM decoding process.

4.1.2 Decoding of the Hidden Markov Model

Given an observed states sequence $\mathbf{V}$ with length T, the decoding process is to find the most probable sequence for the hidden states. However, finding the optimal sequence may cause an exhaustive searching with computation complexity $O(M^T T)$ which is prohibitive in practice. We consider a simpler decoding algorithm with the idea of *dynamic programming*:

■ **Algorithm (Dynamic Programming HMM Decoding)** [17]

1 **begin initialize** Path $\leftarrow$, $t \leftarrow 0$

2 **for** $t \leftarrow t+1$

3 $j \leftarrow j+1$

4 **for** $j \leftarrow j+1$

5 $\alpha_j(t) \leftarrow b_{jk} v(t) \sum_{i=1}^{M} \alpha_i(t-1)\, a_{ij}$

6 **until** $j = M$

7 $j' \leftarrow \arg\max_j \alpha_j(t)$

8 *Append* $\omega_{j'}$ *to* Path

9 **until** $t = T$

10 **return** Path

11 **end**

The dynamic programming based HMM decoding process that calculates the total probabilities holds the computational complexity $O\left(M^2 T\right)$. This complexity is significantly reduced from the exhaustive full path search which is equivalent to $O\left(M^T T\right)$. Figure 4.2 shows a black arrow that represents the path which connects the hidden states with the highest value of α_j at each time step t.

4.2 Implementation of the MFCC Codebook Search with an HMM

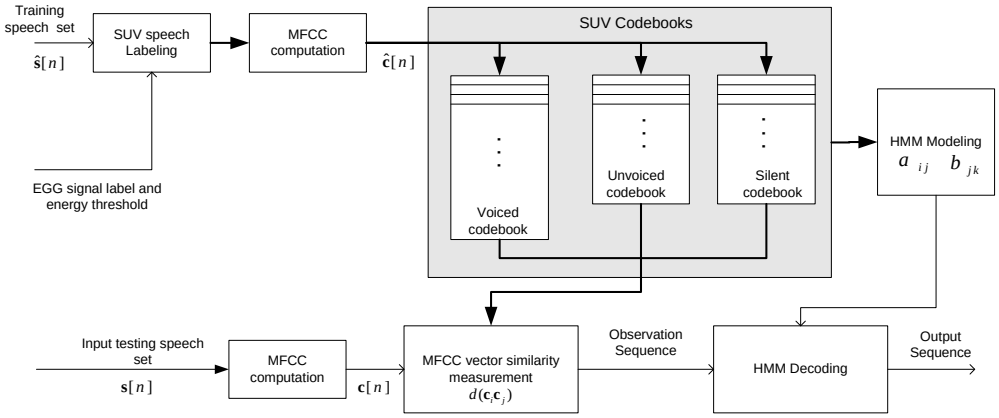

Figure 4.3. This figure is extended with HMM modeling and HMM decoding from Figure 3.2

The idea of MFCC codebook search based SUV speech classification with a hidden Markov model is illustrated in Figure 4.3. The mechanism of the MFCC codebook search will give us an observation sequence. The estimation of the most probable state sequence can be decoded by the algorithm shown in the previous subsection. With the estimated state sequence we can refer the number of the codeword index at each time step to determine which class that speech frame belongs to corresponding to the time step. The classification accuracy will be discussed in the next chapter and will show favor against a state-of-the-art SUV speech classification based on the discrete wavelet transform.

Chapter 5

Performance Evaluation

Our SUV speech classification experiment is designed for a single dedicated speaker with approximately 25 minutes of training set of data which is used to design the MFCC codebooks and to build the hidden Markov model. There are four SUV speech classification mechanisms going to be compared including GMM based, MFCC codebook search based, MFCC codebook search with HMM and finally the novel discrete wavelet transform based technique. Performance will be discussed by correctness measurement P_s, P_u, P_v and P_t which correspond to silent, unvoiced, voiced speech classes and total correctness respectively. The speech are data re-sampled from 22.5 kHz to 16 kHz and artificially corrupted with additive white and babble noise over the SNR range of 30, 20, 10 and 5 dB.

5.1 Discrete-Time Wavelet Based Speech SUV Classification

Recently, a wavelet decomposition method is used to classify speech frames into silent, unvoiced and voiced classes. This classifier utilize a single multidimensional feature which is extracted from the Teager energy operator of the wavelet coefficients. The feature is enhanced and compared with quantile-based adaptive thresholds to determine

the potential speech classes [15].

A discrete-time signal $s[n]$ can be represented as:

$$s\left[n\right]=\sum_{l}\sum_{i}\left\langle \varphi_{l,i},s\right\rangle \widetilde{\varphi}_{l,i}\left[n\right]. \tag{5.1}$$

The discrete-time wavelet basis function $\widetilde{\varphi}_{l,i}\left[n\right]$ is constructed from iterated filters. Based on the approach above, the discrete-time signal $s[n]$ can be decomposed into the sum of approximation with E detail representations at E resolution stages:

$$s\left[n\right]=\sum_{i=-\infty}^{\infty}S^{(E)}\left[2i\right]\cdot g_{0}^{(E)}\left[n-2^{(E)}i\right]+\sum_{e=1}^{E}\sum_{i=-\infty}^{\infty}S^{(e)}\left[2i+1\right]g_{1}^{(e)}\left[n-2^{(e)}i\right]. \tag{5.2}$$

where

$$S^{(E)}\left[2i\right]=\left\langle h_{0}^{(E)}\left[2^{E}i-n\right],s\left[n\right]\right\rangle , \tag{5.3}$$

and

$$S^{(e)}\left[2i+1\right]=\left\langle h_{1}^{(e)}\left[2^{e}i-n\right],s\left[n\right]\right\rangle , \tag{5.4}$$

are the approximation coefficients (low-frequency part) and the detail coefficients (high-frequency part), respectively, at the output of the iterated filter bank with E stages. $g_{0}^{e}\left[n\right]$ is an equivalent filter obtained through e stages of low-pass synthesis filters $g_{(0)}\left[n\right]$. Then $W_{e,t}\left[i\right]$ can be defined as the sequence of all wavelet coefficients which are derived form wavelet decomposition at the scale e,

5.1.1 Teager Energy Operator

The *Teager energy operator* is an efficient nonlinear operator for many speech processing applications. This nonlinear operator can enhance the discriminability of speech

signals from noise. The Teager energy operator expands the difference between the approximation subband and detail subband. This improvement is very useful for unvoiced frames dominated by the Teager energy operator as:

$$T_{e,t}[i] = W_{e,t}^2[i] - W_{e,t}[i-1]\, W_{e,t}[i+1]. \tag{5.5}$$

5.1.2 Sigmoidal Delta Feature

The power of the voiced frames is mostly contained in the approximation subband and much less in the detail subband, and vice versa the for the unvoiced frames. For silent frames a relatively equal power distribution occurs. With the characteristics observed above a delta parameter which is the power difference between approximation subband and detail subband can be computed as:

$$D[t] = \frac{1}{N_a}\sum_{i=1}^{N_a} T_{e,t}^2[i] - \frac{1}{N_d}\sum_{1}^{N_d} T_{e,t}^2[i]. \tag{5.6}$$

Parameters $N_a = \frac{N}{2^m}$ and $N_d = N - N_a$ are the length of approximation and detail parts, respectively. N is the number of samples in one speech frame.

In general, the voiced and unvoiced frames give very high values of D with positive and negative signs, respectively. To balance the impact of the large range of values of D, a sigmoidal function is applied to $D[t]$ as:

$$D_s[t] = \frac{2}{1 + e^{2D[t]}} - 1 \tag{5.7}$$

In order to make a final decision of SUV classification, this DWT method introduces a quantile based adaptive threshold related to the noise level. The delta values $D[t]$ are sorted in ascending order over a buffer of one second length with one frame shifting, then the threshold T_q is determined by the q^{th} quantile. The quantile $q = 0.3$ is selected experimentally over the range $q = 0, \ldots, 1$.

The classification decision of S/U/V is determined by the delta parameter $D_s[t]$ of each input speech frame which is calculated and compared with the determined threshold by the following rule:

$$D_s[t] = \begin{cases} V, & \text{if } D_s[t] > T_q \\ U, & \text{if } D_s[t] < T_q \\ S, & \text{otherwise.} \end{cases} \tag{5.8}$$

5.2 Experimental Results

In our experiments we employed approximately 25 minutes of training data from a dedicated speaker to design the MFCC codebooks and to estimate the hidden Markov model parameters. The employed training data was the `BDL` subset of the CMU ARCTIC database from the Language Technologies Institute at Carnegie Mellon University[1]. The `BDL` subset stems from a US English male speaker. It contains 1132 phonetically balanced English utterances, most of which are between one and four seconds long. The data was appropriately low-pass filtered and downsampled to a processing sampling rate of 16 kHz. We split the `BDL` subset into training and testing parts with 631 and 500 utterance respectively.

Additive noise was taken from the NOISEX database at the Institute for Perception--TNO, The Netherlands Speech Research Unit, RSRE, UK[2]. For our experiments we used white noise and babble noise. The noise was added to the training and testing data at signal-to-noise ratios (SNR) of 30 dB, 20 dB, 10 dB and 5 dB.

In our experiments we compared four SUV speech classification mechanisms. These include the state-of-the-art DWT based method, the GMM based method, a plain MFCC codebook search technique, and the proposed HMM based method. The plain MFCC codebook search technique illustrates, hence, the performance of our proposed method

[1]The corpus is available at ⟨http://www.festvox.org/cmu_arctic⟩.
[2]The noise is available at ⟨http://spib.rice.edu/spib/select_noise.html⟩.

without the "error-correcting" capabilities of the hidden Markov model.

The performance of the four procedures was assessed by counting the relative number of correct SUV classification decisions. We separately analyzed the percentage of correct decisions for silent frames P_s, for unvoiced frames P_u, and for voiced frames P_v. The percentage of total correct decisions is reported as P_c.

$$\begin{aligned} P_v &= P\left(\text{frames classified voiced} \mid \text{frames labeled voiced}\right). \\ P_u &= P\left(\text{frames classified unvoiced} \mid \text{frames labeled unvoiced}\right). \\ P_s &= P\left(\text{frames classified silent} \mid \text{frames labeled silent}\right). \\ P_c &= P\left(\text{correct classified over all classes}\right). \end{aligned} \tag{5.9}$$

On the tables of experiment results the highest classification rate over these four mechanisms in each row are highlighted with bold face. We can observe that our proposed method can always have the highest P_c in comparison with the other three mechanisms in both white and babble noise environments. We can also notice a fact in the columns of methods MFCC and HMM that hidden Markov model can always help plain MFCC codebook based method to correct the SUV classification dramatically in the low SNR conditions. It proves our assumption that the information of transition and emission statistics we retrieved from the training speech contains fairly important factors.

Another fact worth discussing is that our proposed HMM based method has very reliable classification accuracy constantly over P_v. It implies that the voiced part of training speech contains most useful transition and emission statistics for training a hidden Markov model. Besides, the volume of the voiced speech section in our training set dominates the whole training database by 70% strong. Higher classification accuracy in P_v may lead to higher classification accuracy in P_c.

The experiment results in P_c from GMM based mechanism are comparable with ours and sometimes are more favorable comparing to DWT based method. However, P_v in our HMM based mechanism are always higher than that in GMM based method. As a matter of fact, the importance of voiced part detection can never be neglected in speech

pre-processing stage.

Compared with the voiced speech, the unvoiced speech usually has lower segmental SNR, which makes the classification of unvoiced speech detection much more difficult. However, our proposed HMM based method can show favor over the most recent proposed DWT-based SUV classification on each SNR level in both white noise and babble noise environment, especially, for unvoiced speech detection. The results verify that the transition statistics in our proposed HMM based method greatly improve the detection of unvoiced speech.

In summary, our proposed HMM based mechanism can have a certain degree of superiority comparing to the other three methods in those aspects listed above. As a result, it can be treated as a novel and robust mechanism in the SUV speech classification research community.

Table 5.1. Classification rate in clean condition

Class	DWT	GMM	MFCC	HMM
P_v	0.9735	0.9396	0.9745	N/A
P_u	0.5602	0.7688	0.7742	N/A
P_s	0.9981	0.9590	0.9347	N/A
P_c	0.9156	0.9110	0.9315	N/A

Table 5.2. Classification rate in 30dB white noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9678	0.8982	0.8970	0.9814
P_u	0.5013	0.8592	0.5186	0.8151
P_s	0.9998	0.9601	0.5625	0.7116
P_c	0.9032	0.9003	0.7768	0.9100

Table 5.3. Classification rate in 20dB white noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9567	0.9175	0.8588	0.9782
P_u	0.4266	0.7672	0.5386	0.8314
P_s	0.9999	0.9614	0.5989	0.7216
P_c	0.8846	0.8964	0.7605	0.9124

Table 5.4. Classification rate in 10dB white noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9425	0.9232	0.8385	0.9739
P_u	0.3517	0.7567	0.4952	0.8310
P_s	0.9931	0.9658	0.5799	0.6777
P_c	0.8626	0.8989	0.7362	0.9029

Table 5.5. Classification rate in 5dB white noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9588	0.9215	0.8315	0.9700
P_u	0.4946	0.6082	0.4614	0.8263
P_s	0.6837	0.7271	0.5993	0.6543
P_c	0.8380	0.8344	0.7282	0.8959

Table 5.6. Classification rate in 30dB babble noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9692	0.9080	0.8471	0.9744
P_u	0.5045	0.8053	0.6235	0.7574
P_s	0.9984	0.9314	0.4496	0.7831
P_c	0.9044	0.8926	0.7459	0.9055

Table 5.7. Classification rate in 20dB babble noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9739	0.9054	0.7251	0.9588
P_u	0.4190	0.7980	0.6133	0.7269
P_s	0.7918	0.9126	0.4845	0.7641
P_c	0.8565	0.8867	0.6682	0.8867

Table 5.8. Classification rate in 10dB babble noise

Class	DWT	GMM	MFCC	HMM
P_v	0.8919	0.9038	0.6173	0.9464
P_u	0.0613	0.7620	0.4547	0.5143
P_s	0.9133	0.9518	0.5074	0.6958
P_c	0.7704	0.8848	0.5708	0.8289

Table 5.9. Classification rate in 5dB babble noise

Class	DWT	GMM	MFCC	HMM
P_v	0.9474	0.7026	0.5948	0.9130
P_u	0.0219	0.5807	0.3537	0.4439
P_s	0.4564	0.9457	0.4489	0.6249
P_c	0.7172	0.7167	0.5283	0.7830

5.3 Conclusion

We proposed a novel method for speech SUV classification. This new idea is based on a codebook-driven speech processing scheme combined with the statistical analysis of the relationship between codewords. The required statistical descriptions were obtained from noise enrollment and from speaker enrollment in clean conditions. We have shown that our proposed speech classification scheme compares favorably to other classification approaches.

Appendix A

Expectation Maximization

A.1 Derivation of EM Algorithm

Parameters estimated at the p_{th} iteration are marked by a superscript (p).

1. Initialize parameters

2. E-step: Compute the posterior probabilities for all $i = 1, \ldots, \ldots,$ n, $k = 1, \ldots, K$

$$p_{i,k} = \frac{a_k^{(p)} f_k(x)}{\sum_{k=1}^{K} a_k^{(p)} f_k(x)}. \tag{A.1}$$

3. M-step:

$$a_k^{(p+1)} = \frac{\sum_{n=1}^{i} p_{i,k}}{n}. \tag{A.2}$$

$$\mu_k^{(p+1)} = \frac{\sum_{i=1}^{n} p_{i,k} \cdot x_i}{\sum_{i=1}^{n} p_{i,k}}. \tag{A.3}$$

$$\Sigma_k^{(p+1)} = \frac{\sum_{i=1}^{n} p_{i,k} \left(x_i - \mu_k^{(p+1)}\right) \left(x_i - \mu_k^{(p+1)}\right)^t}{\sum_{i=1}^{n} p_{i,k}}. \tag{A.4}$$

4. Repeat step 2 and 3 until converge.

For mixtures of other distributions, the EM algorithm is very similar. The E-step involves computing the posterior probabilities. The M-step always involve parameter optimization. For our initialization, we adopt a $k-means\ algorithm$ which is discussed

in chapter 3. By applying the k-means algorithm, we can initialize μ_k and Σ_k using all the samples classified to cluster k and initialize a_k by the proportion of data assigned to cluster k by the k-means algorithm. In practice, we may want to reduce model complexity by putting constraints on the parameters. For instance, we assume equal priors and identical covariance matrices for all the components [17].

Bibliography

[1] Nickel, R. M. (2006) "Automatic Speech Character Identification," *IEEE Circuits and Systems Magazine*, **6**(4), pp. 8–29.

[2] Deller, J. R., J. H. L. Hansen, and J. G. Proakis (1993) *Discrete-Time Processing of Speech Signals*, Mcmillan.

[3] Quatieri, T. F. (2002) *Discrete-Time Speech Signal Processing: Principles and Practice*, Pearson Education.

[4] Kondoz, A. M. (2004) *Digital Speech Coding for Low Bit Rate Communication Systems*, Wiley-Interscience.

[5] Li, K., M. N. S. Swamy, and M. O. Ahmad (2005) "An Improved Voice Activity Detection Using Higher Order Statistics," *IEEE Transactions on Speech and Audio Processing*, **13**(5), pp. 965–974.

[6] Atal, B. S. and L. R. Rabiner (1976) "A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-24**(3).

[7] Rabiner, L. R. and M. R. Sambur (1977) "Application of an LPC Distance Measure to the Voiced-Unvoiced-Silence Detection Problem," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-25**(4).

[8] Un, C. K. and H. H. Lee (1980) "Voiced/Unvoiced/Silence Discrimination of Speech by Delta Modulation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-28**(4).

[9] Cox, B. V. and L. M. Timothy (1980) "Nonparametric Rank-Order Statistics Applied to Robust Voiced-Unvoiced-Silence Classification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-28**(5).

[10] Bruno, G., M. Di Benedetto, A. Gilio, and P. Mandarini (1987) "A Bayesian-Adaptive Decision Method for the V/UV/S Classification of Segments of a Speech Signal," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-35**(4).

[11] Childers, D. G., M. Hahn, and J. N. Larar (1989) "Silent and Voiced, Unvoiced and Mixed Excitation Classification of Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **37**(11).

[12] Qi, Y. and B. R. Hunt (1993) "Voiced-Unvoiced-Silence Classifications of Speech Using Hybrid Features and a Network Classifier," *IEEE Transactions on Speech and Audio Processing*, **1**(2).

[13] Lobo, A. P. and P. C. Loizou "Voiced/Unvoiced Speech Discrimination in Noise Using Gabor Atomic Decomposition," in *ICASSP 2003*, vol. 1, pp. I–820– I–823.

[14] Umapathy, K., S. Krishnan, V. Parsa, and D. G. Jamieson (2005) "Discrimination of Pathological Voices Using a Time-Frequency Approach," *IEEE Transactions on Biomedical Engineering*, **52**(3).

[15] Tuan, P. V. and G. Kubin "DWT-Based Phonetic Groups Classification Using Neural Networks," in *ICASSP 2005*, vol. 1, pp. 401–404.

[16] ——— "Low-Complexity and Efficient Classification of Voiced/Unvoiced/Silence for Noisy Environments," in *Interspeech 2006*.

[17] Duda, R. O., P. E. Hart, and D. G. Stork (2002) *Pattern Classification*, Wiley-Interscience.

[18] (1996), "ITU-T G.729 Annex A," International Telecommunication Union.

[19] Rabiner, L. R. and B.-H. Juang (1993) *Fundamentals of Speech Recognition*, Prentice Hall.