8-2015

Comparison GMM and SVM Classifier for Automatic Speaker Verification

Shamama Afnan
Clemson University, safnan@g.clemson.edu

COMPARISON GMM AND SVM CLASSIFIER FOR AUTOMATIC SPEAKER VERIFICATION

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Shamama Afnan
August 2015

Accepted by:
Dr. John N. Gowdy, Committee Chair
Dr. Robert Schalkoff
Dr. Carl Baum

Abstract

The objective of this thesis is to develop automatic text-independent speaker verification systems using unconstrained telephone conversational speech. We began by performing a Gaussian Mixture Model Likelihood ratio verification task in speaker independent system as described by MIT Lincoln Lab. We next introduced a speaker dependent verification system based on speaker dependent thresholds.

We then implemented the same system applying Support Vector Machine. In SVM, we used polynomial kernels and radial basis function kernels and compared the performance. For training and testing the system, we used low-level spectral features.

Finally, we provided a performance assessment of these systems using the National Institute of Standards and technology (NIST) speaker recognition evaluation 2008 telephone corpora.

Dedication

I dedicate this work to my parents in recognition of their love and inspiration. I would also like to dedicate to my husband who always provides me endless support, patience and encouragement. And, also to my sister, brother-in-law and nephew who have always been a great source of motivation and inspiration.

Acknowledgments

It is my pleasure to express my sincere gratitude and appreciation to my advisor Dr. John N. Gowdy for his constant support, guidance and encouragement throughout the graduate studies at Clemson University. His mentorship was a rewarding experience for me, which I will treasure my whole life.

I would also like to extend my special appreciation and thanks to Dr. Robert J. Schalkoff and Dr. Carl Baum for being in my advisory committee.

I am very grateful to Poorna and Sanjay for participating in many useful discussions at the early stages of my work. Special thanks to my husband Amir Ahmed Asif for always standing beside me, giving necessary feedback, calming my fear and inspiring me at all stages of my work. I would like to convey thanks to my friends for their timely help, support, humor, and encouragement. Finally, I would like to express my indebtedness and gratitude to my family, who are responsible for all the achievements in my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speaker recognition is the process of automatically recognizing a person based on the information included in his/her speech. Speaker recognition consists of two fundamental tasks: speaker identification and speaker verification. Speaker identification is the task of identifying who is speaking from a set of known speakers. Speaker verification is the task of determining whether a person is the claimed speaker or not.

The most common application of speaker verification is to control access to information, services, and computer accounts. It can be used to reset passwords and replace PINs because speech is something that cannot be forgotten, lost or stolen. Speech is an important feature of a person which can be identified from a distance, over telephone also.

Speech can be considered as a strong biometric signature because of the following two reasons: first, speech is a natural signal to produce, and second, the telephone system provides a ubiquitous, familiar network to obtain and deliver the speech signal. For telephone based applications, there is no need for special signal transducers or networks to be installed at application access points since a

cell phone gives one access almost anywhere. Even for non-telephone applications, sound cards and microphones are low-cost and readily available [1].

The applications of speaker recognition technology are continually growing. Some areas are listed where speaker recognition technology is currently used for are listed below:

- Controlling access to computer networks or websites.
- Automated password reset services.
- Telephone banking, remote electronic and mobile purchases.
- Home-parole and prison-call monitoring.
- Voice mail browsing and intelligent answering machines.
- Annotating recorded meetings or video with speaker labels for quick indexing.
- Storing and retrieving personal settings based on user verification for multi-user sites or devices [2].

Therefore, to ensure a secured method of authenticating speakers, speaker verification with proper statistical, analytical and signal processing techniques is very important.

Chapter 2

Background

Speaker recognition consists of two fundamental tasks: (1) speaker identification, (2) speaker verification. Speaker recognition can be a closed-set or an open-set task. In a closed-set recognition, the unknown voice must come from a fixed set of known speakers. However, in an open-set task, imposters are not known to the system [2], [3]. Here, we will discuss speaker verification.

2.1 Speaker Verification

A speaker verification system consists of two distinct phases: a training phase and a test phase. Figure 2.1 represents the structure of a speaker verification system [3]. The first step consists in extracting parameters from the speech signal to obtain a representation suitable for statistical modeling. The second step consists in obtaining a statistical model based on the parameters. For testing, features from the test sample are compared to one or more of the speaker models to verify the test samples. The entries of the system are a claimed identity (Target Speaker) and the speech samples pronounced by unknown speakers (Imposters).

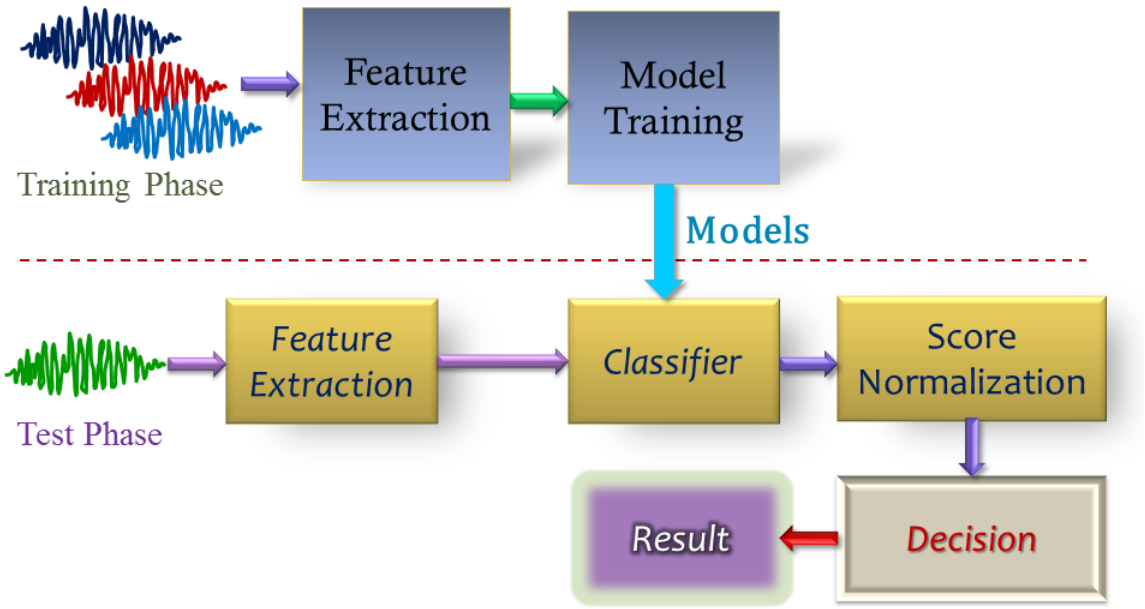

Figure 2.1: The structure of a speaker verification system [3].

2.2 Types of Speaker Verification

The speech used for these tasks can be either text dependent or text-independent. In a text-dependent application, the recognition system has prior knowledge of the text to be spoken and it is expected that the user will cooperatively speak this text. The prior knowledge of the text can improve the performance of a recognition system. In a text-independent application, there is no prior knowledge by the system of the text to be spoken. Text independent verification is more difficult to implement than text-dependent task. However, text-dependent speaker verification is currently the most commercially viable and useful technology for the two basic tasks [2], [3].

2.3 Feature Extraction

Feature extraction is one of the most fundamental parts in any speaker verification system. The features extracted from the speech signal convey information about the speaker's identity. The information in speech signals can be found at different time spans and rates. Features that capture information about a person's vocal tract information through the frequency spectrum of speech will operate using short time spans ($\sim 20 - 30$ms). Prosodic information such as a person's average pitch inflection per sentence, is a feature derived from a longer time span ($\sim 1 - 2$s). Moreover, for the aperiodic features like phonemes or words, the time spans and rates are variable [3].

Speaker verification features capture speaker dependent characteristics from different levels. Low level features extract information about acoustic characteristics related to vocal production, such as frequency spectrum or short term pitch estimates [4]. On the other hand, features based on higher level information such as idiolect and pronunciation, require the output of phone or word recognition system.

2.3.1 Low Level Features

Low level features are used in most speaker recognition and verification systems. These are some form of spectral based information. Low level features such as spectra, consist of short time span, fixed rate analysis of continuous phenomenon (figure 2.2). In short-term analysis, typically 20 ms windows shifted by 10 ms, are used to compute a sequence of magnitude spectra using either LPC or FFT analysis. Most commonly the magnitude spectra are then converted to cepstral features after passing through a mel frequency filterbank. Next, mel-scale

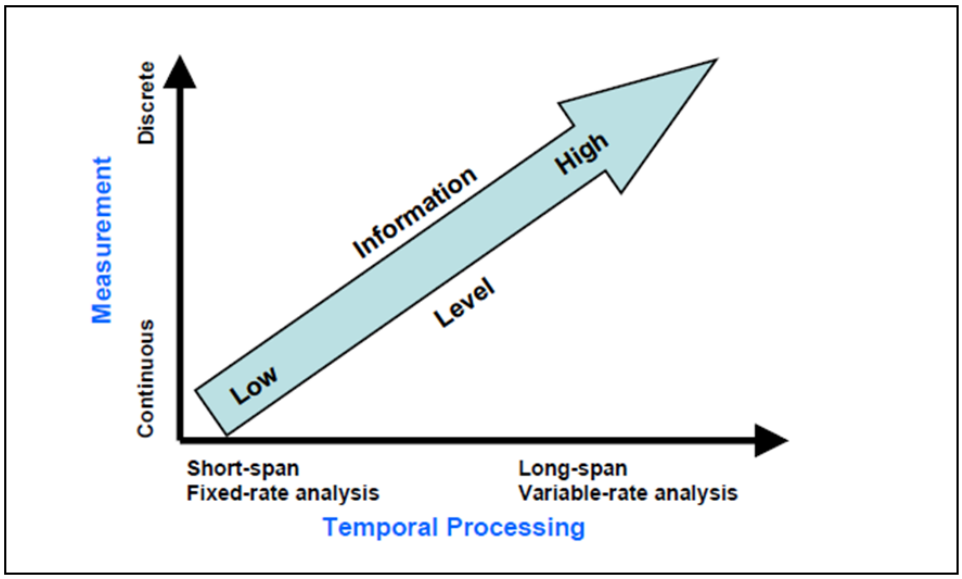

Figure 2.2: Some form of spectral based features is used in most speaker verification systems [3].

cepstral feature vectors are extracted from the speech frames and time-differential (delta) cepstra are appended. The mel-scale cepstrum is the discrete cosine transform of the log spectral energies of the speech segment. Features based on short term spectral analysis are called low level features [2], [5], [6], [7].

2.3.1.1 Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs are the most common features in automatic speaker recognition systems [8], [9]. MFCC processing uses a fixed window of ~ 20 millisecond. MFCCs are represented by a real valued N-dimensional vector, where N is typically 12. The coefficients are a parameterization of spectrum which contains information of speaker's physical characteristics [3].

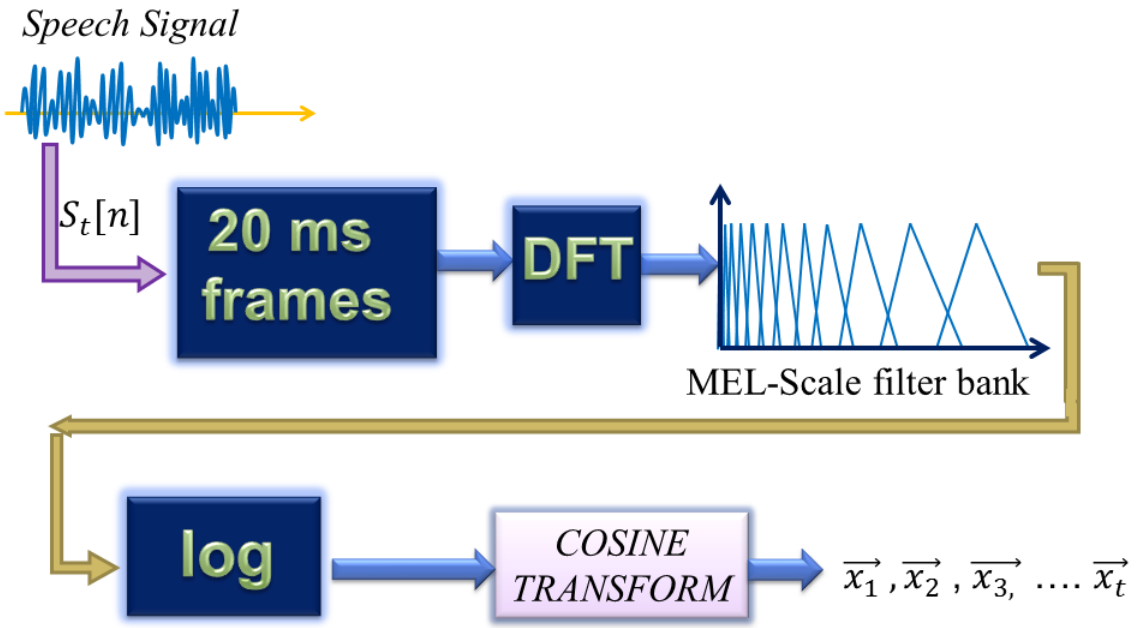

Figure 2.3: Mel Scale Cepstal Feature Analysis [10].

Steps of extracting MFCC

Figure 2.3 shows a block diagram of extracting the MFCCs. In [11], the following steps are mentioned on how to extract the MFCCs:

1. Frame the signal into small (20-40 ms) frames. 25ms is commonly used as frame length. The next steps are applied to every frame and one set of 12 MFCC coefficients is extracted for each frame [11].

2. For each frame calculate the periodogram estimate of the power spectrum. The periodogram is an estimate of the spectral density of a signal. The periodogram estimate identifies which frequencies are present in the frame [11].

3. Apply the mel filterbank to the power spectra and sum the energy in each filter. A mel is a unit of measure of perceived pitch or frequency of a tone [4]. The spectral energies are calculated over logarithmically spaced filters (mel-filters)

with increasing bandwidth. The formula for converting from linear frequency to the Mel Scale is

$$M(f) = 1125 \ln(1 + \frac{f}{700}) \tag{2.1}$$

4. Take the logarithm of all filterbank energies. This is also motivated by human auditory system. We don't identify pitch sensitivity on a linear scale. Generally to double the perceived volume of a sound, we need to put 8 times as much energy into it. This compression operation makes our features match more closely what humans actually hear. The logarithm allows us to use cepstral mean subtraction, which is a channel normalization technique [11].

5. Take the DCT of the log filterbank energies. Because our filterbanks are all overlapping, the filterbank energies are quite correlated with each other. The DCT decorrelates the energies which means diagonal covariance matrices can be used to model the features in a classifier [11].

6. Keep DCT coefficients 2-13, discard coefficient 1 and coefficient 14 and up. The resulting features (12 coefficients for each frame) are called Mel Frequency Cepstral Coefficients [11]. After coefficient 13, the coefficients become so small that the values are in the same level as computational noise level.

2.3.1.2 Linear Prediction Cepstral Coefficients (LPCCs)

LPCCs are often used in automatic speaker recognition systems. However, using LPCCs are more challenging for the speaker recognition system in a noisy environment [9], [12]. Like MFCC, LPCC processing uses a fixed window of ~ 20 millisecond.

Linear Prediction

The all pole Linear Prediction models a signal s_n by a linear combination of its past values and a scaled present input [13], [14].

$$s_n = -\sum_{k=1}^{m} a_k s_{n-k} + \Theta u_n \tag{2.2}$$

where s_n is the present output, m is the prediction order, a_k are the model parameters called the predictor coefficients, s_{n-k} are past outputs, Θ is a gain scaling factor, and u_n is the present input. Since the input u_n is generally ignored in the application, the LP approximation $\hat{s}_n$, depends only on past output samples.

$$\hat{s}_n = -\sum_{k=1}^{m} a_k s_{n-k} \tag{2.3}$$

Then, the prediction error e_n is the difference between the actual signal s_n and the predicted signal $\hat{s}_n$.

$$e_n = s_n - \hat{s}_n = s_n + \sum_{k=1}^{m} a_k s_{n-k} = \Theta u_n \tag{2.4}$$

Therefore, any signal can be represented by linear predictor parameters and the corresponding LP error.

$$s_n = -\sum_{k=1}^{m} a_k s_{n-k} + e_n \tag{2.5}$$

From (2.2), the linear prediction transfer function can be written as

$$H(z) = \frac{S(z)}{U(z)} \tag{2.6}$$

where

$$H(z) = \frac{\Theta}{1 + \sum_{k=1}^{m} a_k z^{-k}} = \frac{\Theta}{A(z)} \tag{2.7}$$

Here, A(z) is the m-th order inverse filter. We can calculate the linear prediction coefficients of all-pole models by using the autocorrelation method [14].

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_{m-1} \\ R_1 & R_0 & \cdots & R_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m,1} & a_{m,2} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{bmatrix} \tag{2.8}$$

In [4], we can find the following equations for calculating Linear Prediction Cepstral Coefficients.

$$\gamma_{\hat{\theta}} = \begin{cases} c_{\hat{\theta}}(0;m) = \log \hat{\mathbf{\Theta}}_0(m), & n \equiv 0 \\ 2c_{\hat{\theta}}(n;m), & n > 0. \\ 0, & 0 \end{cases} \tag{2.9}$$

The recursion for converting LP to CC parameters is

$$\gamma_{\hat{\theta}} = \begin{cases} \log \Theta_0(m) & n \equiv 0 \\ \hat{a}(n;m) + \sum_{k=1}^{n-1} \frac{k}{n} \gamma_{\hat{\theta}} \hat{a}(n-k,m), & n > 0. \end{cases} \tag{2.10}$$

2.3.2 High Level Features

As shown in Figure 2.2, the high level features, such as phones or words, consist of longer time span and involve variable rate analysis of discrete events. For high level feature extraction, input speech is converted into a series of tokens, $T = \{t_i\}$. The tokens are time-ordered discrete symbols such as words, phones, and pitch gestures. Modeling of token streams is usually accomplished by computing probabilities of n-grams of token context [15]. Words, phones and pitch gestures are some example of token types. Figure 2.3 depicts some common features used in automatic speaker recognition systems indicating their location in the feature attribute space.

2.3.2.1 Word and Phone Tokenization

Word and phone features are being used in recent speaker recognition systems. Since it depends on the duration of the word phone units, the analysis window is variable [16], [17], [18]. As the count of words and phone is an integer, this feature type can be considered as discrete. Word and phone models represent the pronunciation differences of speakers. These are considered as high level information [19].

2.3.2.2 Prosodic Statistics

Prosodic statistics capture the idiolect of individual speakers. As the prosodic statistical measures are of continuous values, the feature type is continuous. These features are based on measurements of energy, duration and pitch derived over large speech segments [20].

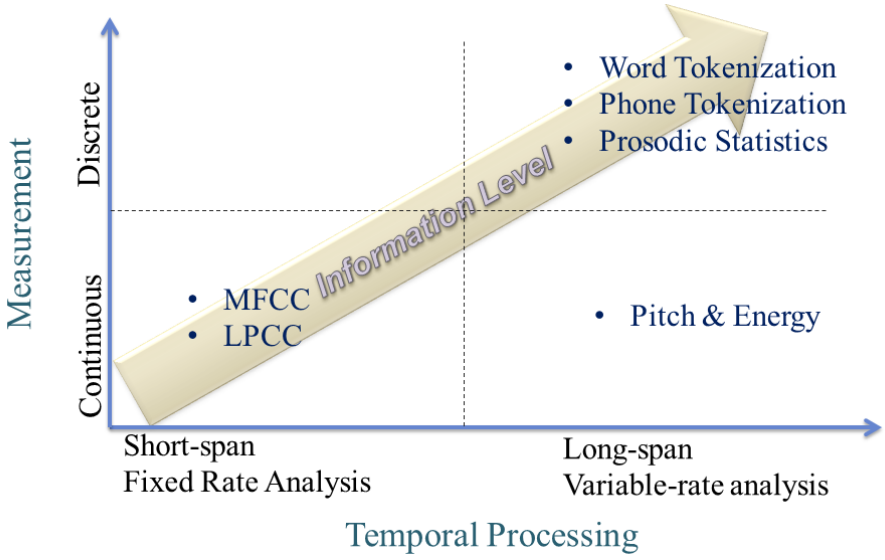

Figure 2.4: Approximate location of common feature in the feature attribute state [3].

2.3.2.3 Pitch and Energy

When these features are combined with short phrases, the spanning of analysis window will be variable depending on the duration of the short phrase. The target is to measure pitch and energy gestures by modeling the joint slope dynamics of pitch and energy [21].

2.4 Speaker Modeling

Desirable attributes of a speaker model are: (1) based on a theoretical understanding of mathematical model behavior; (2) generalizable to new data so that the model does not over fit the enrollment data and can match new data; and (3) inexpensive in both size and computation. There are many modeling techniques used in speaker verification systems which have some or all of these attributes. The

selection of modeling is largely dependent on the type of speech to be used, the expected performance, the ease of training and computational considerations [2].

2.5 Imposter Modeling

There are two approaches used for representing the imposter model in the likelihood ratio test. The first approach is known as likelihood sets. It uses a collection of other speaker models to compute the imposter match score. The second approach is known as universal background modeling. It uses a single speaker-independent model trained on speech from a large number of speakers to represent speaker-independent speech. This approach also allows the use of Maximum A-Posteriori (MAP) training to adapt the claimant model from the background model, which can increase performance and decrease computation and model storage requirements [2].

2.6 Classifiers

The two fundamental tasks of Speaker Recognition are 1) Speaker Identification and 2) Speaker Verification. The speaker identification task is closed set recognition, where all the speakers are known. Figure 2.4 shows a general structure of a speaker identification system [3].

The speaker verification task is a binary decision of whether the unknown speaker is the same as the claimed speaker or not. This is an open set task and uses general imposter models. The general structure of the speaker verification system is presented in Figure 2.5 [3].

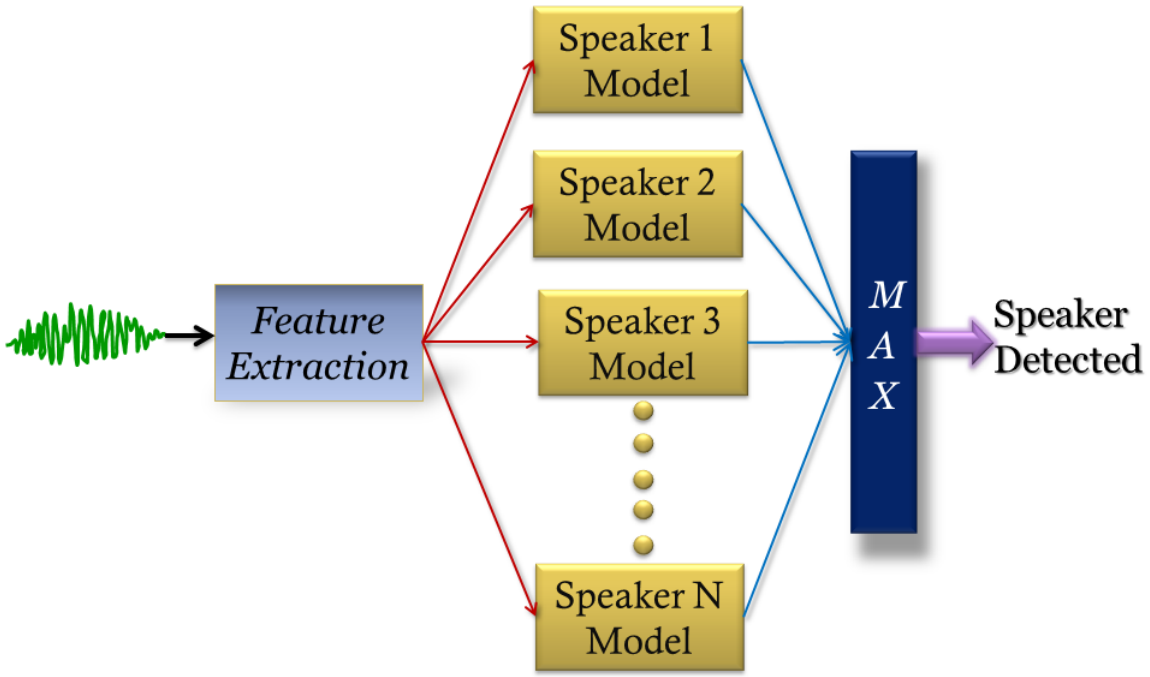

Figure 2.5: General classifier structure for speaker identification system [3].

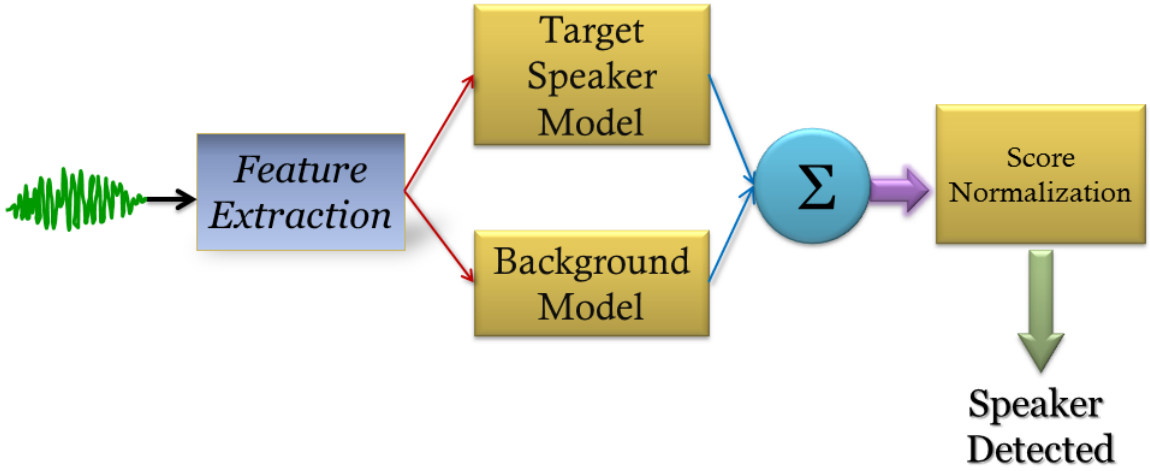

Figure 2.6: General classifier structure for speaker verification system [3].

2.6.1 Gaussian Mixture Model (GMM)

The Gaussian Mixture Model can be used in speaker verification applications. A GMM is used as a probabilistic model for text-independent verification [22], [23], [10]. An extension of GMM-based systems to speaker verification was described and evaluated on several speech corpora in [24], [25]. Later, GMM-based systems have been applied to the annual NIST Speaker Recognition Evaluations (SRE). At MIT Lincoln Laboratory, a GMM-based system was developed by employing Bayesian adaptation of speaker models from a universal background model and handset-based score normalization [26]. The system is referred to as the Gaussian Mixture Model-Universal Background Model (GMM-UBM) speaker verification system. Here, we will discuss the basics of speaker verification and the likelihood ratio detector approach.

2.6.1.1 Likelihood Ratio Detector

Consider a segment of speech, Y, and a hypothesized speaker, S. The task of speaker detection is to determine if Y was spoken by S. Here, we assume that Y contains speech from only one speaker [27].

The single-speaker detection task can be restated as a basic hypothesis test between

λ_{hyp} : Speech segment Y from speaker S

$\lambda_{\overline{hyp}}$: Speech segment Y is not from speaker S.

The verification test to decide between these two hypotheses is a likelihood ratio test given by

$$\Lambda(Y) = \frac{p(Y|\lambda_{hyp})}{p(Y|\lambda_{\overline{hyp}})} \begin{cases} \geq \theta & \text{Accept Hypothesis } \lambda_{hyp} \\ \leq \theta & \text{Reject Hypothesis } \lambda_{hyp} \end{cases}$$

where, $p(Y|\lambda)$, is the probability density function and θ is the decision threshold.

2.6.1.2 GMM-UBM Verification System

Consider the set of feature vectors$X = \{x_1, x_2, ..., x_T\}$, where x_T is a feature vector of discrete time $t \in [1, 2, ...T]$. For a D-dimensional feature vector, x, the mixture density used for the likelihood function can be written as:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i p_i(x) \tag{2.11}$$

where $p_i(x)$ is the individual Gaussian density function,

$$p_i(x) = \frac{1}{(2\pi)^{n/2}|\sum_i|^{1/2}} \times e^{-\frac{1}{2}\{(x-\mu_i)'\sum_i^{-1}(x-\mu_i)\}} \tag{2.12}$$

The parameters of the model are [27]:
the mixture weight, w_i , where $\sum_{i=1}^{M} w_i = 1$;
the N-dimensional mean vector, μ_i ;
and, the N by N dimensional covariance matrix, $\sum_i$.

In [27], only diagonal covariance matrices are used. This is done for three reasons. First, the density modeling of an M-th order full covariance GMM can be achieved equally well using a larger order diagonal covariance GMM. Second, diagonal-matrix GMMs are more computationally efficient than full covariance GMMs for training, since repeated inversions of a $D \times D$ matrix are not required.

Third, diagonal matrix GMMs outperform full matrix GMMs.

In the GMM-UBM system, we use a single, speaker-independent background model to represent $p(X|\lambda_{\overline{hyp}})$. The UBM is a large GMM trained to represent the speaker-independent distribution of features. To obtain the final model, we should train the UBM using expectation maximization (EM) algorithm [28].

2.6.1.3 Log-Likelihood Ratio Computation

The log-likelihood ratio for a test sequence of feature vectors X is computed as [27]:

$$\Lambda(X) = logp(X|\lambda_{hyp}) - logp(X|\lambda_{ubm}). \tag{2.13}$$

The hypothesized speaker model is adapted from the UBM which yields a faster scoring method. This fast scoring approach is based on two effects. The first is that when a large GMM is evaluated for a feature vector, only a few of the mixtures contribute significantly to the likelihood value. This is because the GMM represents a distribution over a large space but a single vector will be near only a few components of the GMM. Thus, likelihood values can be approximated very well using only best scoring mixture components.

2.6.2 Support Vector Machine (SVM)

SVMs map inputs into a high dimensional space and then separate classes with a hyperplane. The kernel is a critical aspect of using SVMs in the high dimensional mapping. The sequence kernel is based upon generalized linear discriminants. This strategy has several important properties. First, the kernel uses an explicit expansion into SVM feature space and has low computational complexity. Second, the SVM builds upon a simpler mean-squared error classifier to produce

a more accurate system. Finally, the system is competitive and complimentary to other approaches, such as Gaussian mixture models [29].

An SVM is a discriminative classifier which models the boundary between a speaker and a set of impostors. This approach contrasts to traditional methods for speaker verification which separately model the probability distributions of the speaker and the general population. By exploring SVM methods, it is possible to benchmark the performance of new classification methods for speaker verification, to gain more understanding of the speaker verification problem, and to observe if SVMs provide complementary information to traditional GMM approaches [29].

An SVM is a two-class classifier constructed from sums of a kernel function $K(.,.)$, [30]

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x_i}) + d \tag{2.14}$$

where the t_i are the ideal outputs, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$ [31]. The vectors x_i are support vectors. The ideal outputs are either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1, respectively. The kernel function is formed as:

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{b}(\mathbf{x})^t \mathbf{b}(\mathbf{y}) \tag{2.15}$$

where $b(x)$ is a mapping from the input space to a possibly infinite dimensional space [15]. Since the SVM is a two-class classifier, a target model is trained for speaker verification. The set of known non-targets are used as the remaining class. For speaker verification, the target speaker's utterances are labeled as class 0. A background speaker set (class 1) is also constructed that consists of example im-

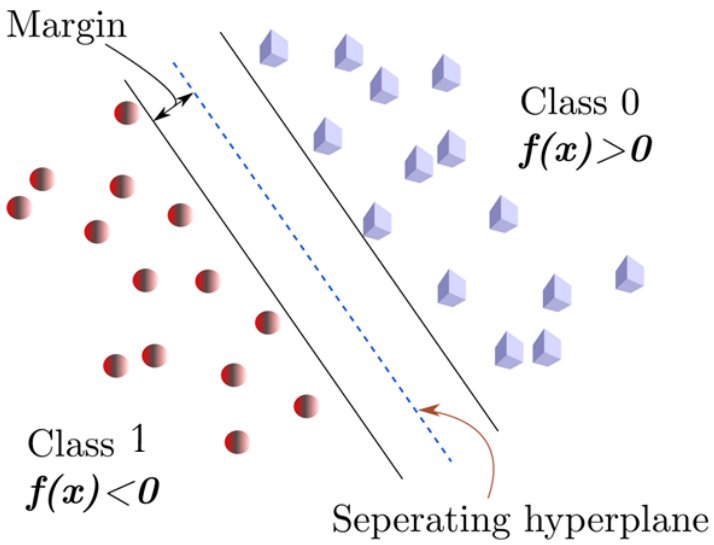

Figure 2.7: Support Vector Machine concept [15].

postor speakers in Figure 2.6. If $f(x)$ is an SVM for a target speaker,

$$f(\mathbf{x}) = \sum_{i \in \{i|t_i=1\}} \alpha_i K(\mathbf{x}, \mathbf{x_i}) - \sum_{i \in \{i|t_i=-1\}} \alpha_i K(\mathbf{x}, \mathbf{x_i}) + d \tag{2.16}$$

The first sum is a per-utterance-weighted target score [32].

The main idea for constructing a train/test kernel is illustrated in Figure 2.8. The basic approach is to compare two utterances by training a model on one utterance and then scoring the resulting model on another utterance. This process produces a number that measures the similarity between the two utterances.

SVM can be represented as a two class problem: target and nontarget speaker. If ω is a random variable representing the hypothesis, then $\omega = 1$ represents target present and $\omega = 0$ represents target not present. A score is calculated from a sequence of observations $y_1 ... y_n$ extracted from the speech input. The scoring function is based on the output of a generalized linear discriminant function of

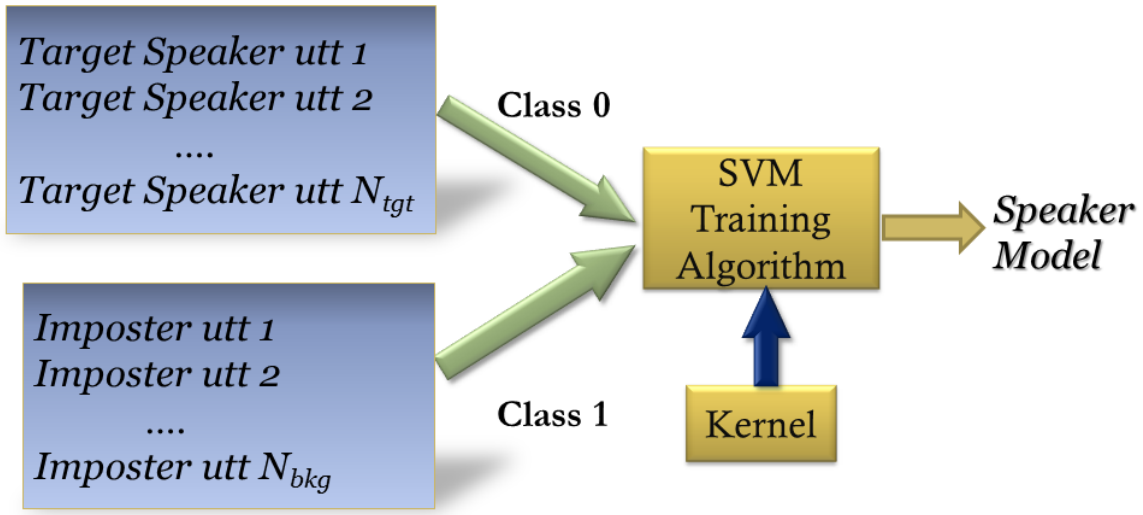

Figure 2.8: General train/test sequence kernel [15].

the form $g(\mathbf{y}) = \omega^t b(y)$, where ω is the vector of classifier parameters and b is an expansion of the input space into a vector of scalar functions [33]:

$$\mathbf{b}(\mathbf{y}) = \begin{bmatrix} \\ b_1(\mathbf{y})b_2(\mathbf{y})...b_n(\mathbf{y}) \\ \\ \end{bmatrix}^t . \tag{2.17}$$

If the classifier is trained with a mean-squared error training criterion and ideal outputs of 1 for $\omega = 1$ and 0 for $\omega = 0$, then $g(y)$ will approximate the posteriori probability $p(\omega = 1|y)$ [34]. We can then find the probability of the entire sequence, $p(y_1...y_n|\omega = 1)$ as follows:

$$p(\mathbf{y_1}...\mathbf{y_n}|\omega) = \prod_{i=1}^{n} p(\mathbf{y_i}|\omega) \tag{2.18}$$

$$p(\mathbf{y_1}...\mathbf{y_n}|\omega) = \prod_{i=1}^{n} \frac{p(\omega|\mathbf{y_i})p(\mathbf{y_i})}{p(\omega)}. \tag{2.19}$$

Taking log on both sides [33], we get the discriminant function:

$$d^{'}(\mathbf{y}_1^n|\omega) = \sum_{i=1}^{n} \log\left(\frac{p(\omega|\mathbf{y_i})}{p(\omega)}\right), \tag{2.20}$$

For the purpose of classification, we discard $p(\mathbf{y_i})$. Using $\log(x) \approx x - 1$

$$d(\mathbf{y}_1^n|\omega) = \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{p(\omega|\mathbf{y_i})}{p(\omega)}\right). \tag{2.21}$$

Assuming $g(y) \approx p(\omega = 1|y)$,

$$d(\mathbf{y}_1^n|\omega = 1) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbf{w}^t\mathbf{b}(\mathbf{y_i})}{p(\omega = 1)}$$

$$= \frac{1}{np(\omega = 1)}\mathbf{w}^t\left(\sum_{i=1}^{n}\mathbf{b}(\mathbf{y_i})\right)$$

$$d(\mathbf{y}_1^n|\omega = 1) = \frac{1}{p(\omega = 1)}\mathbf{w}^t\bar{\mathbf{b}}_y \tag{2.22}$$

where the mapping $y_1^n \rightarrow \mathbf{b}_y$ is

$$\mathbf{y}_1^n \rightarrow \frac{1}{n}\sum_{i=1}^{n}\mathbf{b}(\mathbf{y_i}). \tag{2.23}$$

In the scoring method, for a sequence of input vectors $x_1, x_2, ..., x_n$ and a speaker model w, **b** can be constructed using (2.23) [35]. For speaker verification, if score is above a threshold, then we declare the identity claim valid; otherwise, the claim is rejected as an impostor attempt.

In Chapter 4, we will observe the application of Gaussian Mixture Model and Support Vector Machine for training and testing of a speaker verification system.

Chapter 3

Statement of Problem

The goal is to implement a speaker verification system based on the MIT LL 2008 Speaker Recognition System. For performing this task, we will use NIST SRE 2008 training corpus. For training and testing the dataset, we will apply Adaptive Gaussian Mixture model and Support Vector Machine. Our goal is to observe the performance using different classification methods. For measuring performances we will follow the methods described in [15]. Further, we will extend our research to compare our results with NIST SRE 2008 evaluation and observe how our system performs.

Our plan is to develop a similar system as MIT 2008 NIST Speaker Verification System. The task is to detect whether a specified speaker is speaking during a given segment of conversational speech. We will focus on cepstral based, Gaussian mixture modeling (GMM) and support vector machine (SVM) systems. For this purpose, we will calculate MFCC from the speech samples and train the system using extracted features. Then we will compare the Speaker model with the Imposter model to verify the speaker's identity.

Chapter 4

Methods of Research

4.1 Data Selection

For the experiment, we used a part of NIST Speaker Recognition Evaluation (SRE) 2008 database. For training and testing the system, we used English speech data only for both male and female speakers. The gender of each target speaker is provided in the dataset. We performed text independent, gender dependent speaker verification. In the database, there were speech data from both native and non-native English speakers. Our experiment associated only with native English speakers. Firstly, we listen to all the speech and chose 30 male and 30 female native English speakers' data.

The NIST SRE 2008 database consists of speech data from telephone conversation. In the database, each record contains two fields. The first field is the speaker identifier. The second includes speech files that are to be used to train the model. For the two channel training conditions, each speech file label also specifies whether the target speakers speech is on the 'A' or the 'B' channel of the speech file. An example record looks like: '32324 mrpvc.sph:B'. This means 32324 is the

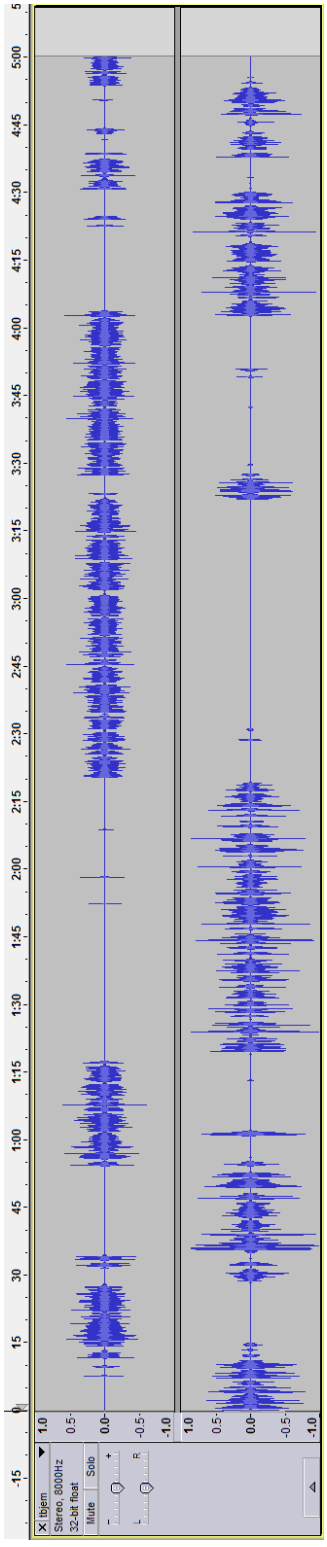

Figure 4.1: Five minutes of telephone conversation between two speakers.

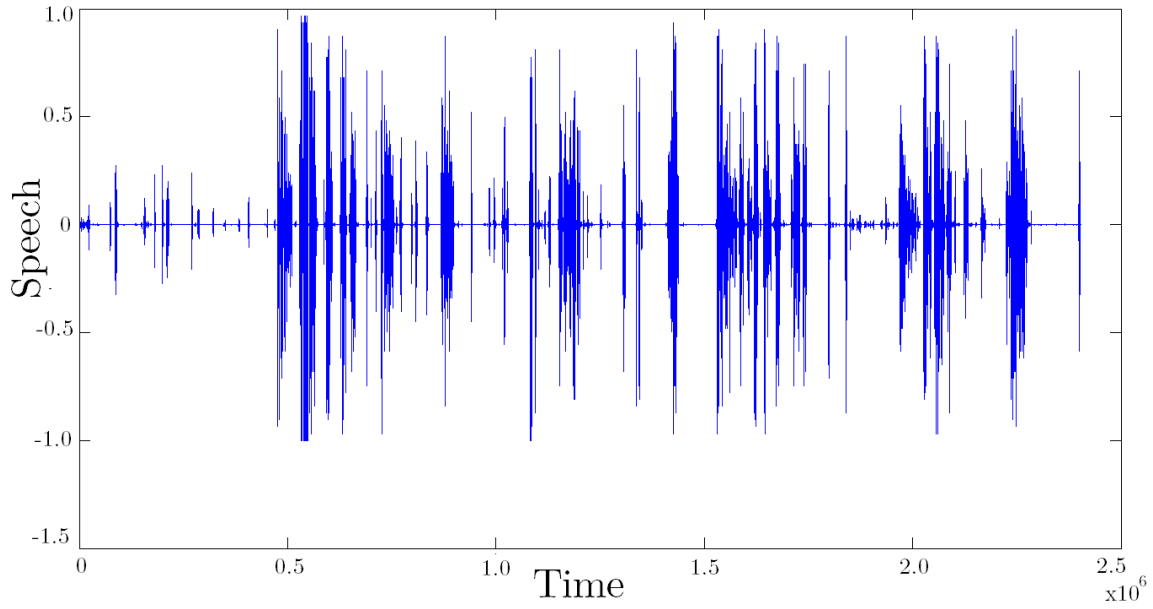

Figure 4.2: Speech of target speaker from five minutes of telephone conversation.

speaker identification number and and the target speaker's speech is in the channel 'B'. In this way, we separated the speech of target speaker from the conversational speech.

All the speech files are five minutes long conversations of two persons. A typical conversation is presented in Figure 4.1. In the first step, we separated the target speaker from the conversation. Figure 4.2 depicts the speech of target speaker only. Then we automatically removed silences from the speech. A method described in [36] is used to discard the silence. This method uses the short-term energy and sets a threshold based on the noise energy to decide the voiced components of the recording. The utterances were concatenated together. We calculated energy for each 10ms of frame and set a threshold is equal to 2.5 times of noise energy. Figure 4.3 shows the speech of target speaker after removing the silence.

We trained the system with 60 speakers (30 male and 30 female). We divided

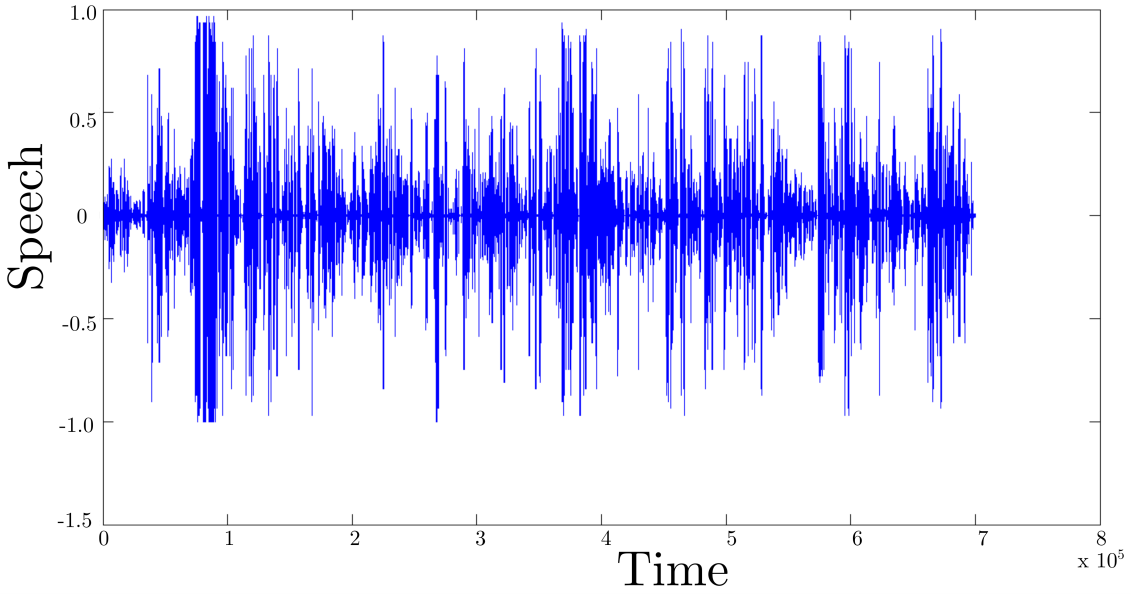

Figure 4.3: Speech of target speaker after removing silence.

the experiment into 6 parts. In each part, there were 10 speakers in a set. For each sets, we used one speaker as target, while other 9 speakers are treated as imposters. For example, the system is trained such that when the speaker 01 is target, speaker 02 through speaker 10 are imposters, and so on. No cross gender trails are performed, the gender of the hypothesized and background speakers are the same.

4.2 Gaussian Mixture Model

The code was written in MatLab. For classification we used Mel-Frequency Cepstral Coefficients (MFCCs) as features of the target speaker. For calculating MFCC, we used MatLab function ”mfcc” from Auditory toolbox [37]. Using this function we extracted a 13 dimensional feature vector for each frame. For our

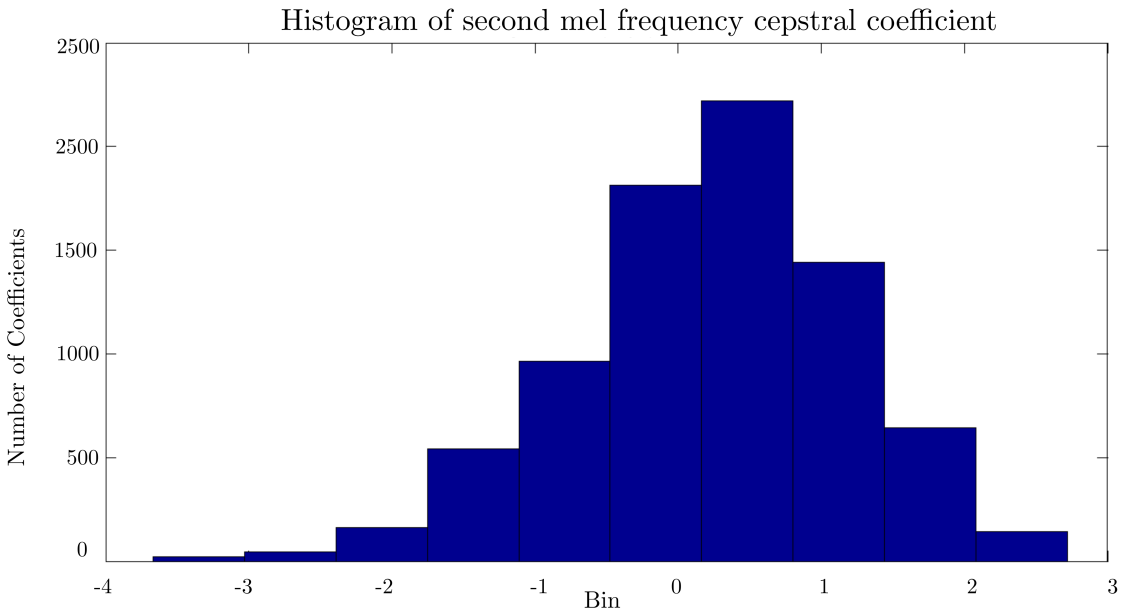

Figure 4.4: Histogram of one MFCC.

experiments, we discarded the first coefficient and used the other 12 coefficients (2-13 features for each frame).

Using these coefficients, we trained a model for each speaker. Figure 4.4 shows the histogram of one coefficient over all frames. We plotted a histogram of the second MFCC over all the frames. The histogram looks like a Gaussian distribution. Therefore, we can apply Gaussian mixture model to train the system.

For Gaussian Mixture Model (GMM) Log-Likelihood Ratio (LLR) classifier, we used MatLab function ”gmdistribution” from the Statistics toolbox. The function ”gmdistribution.fit” trains the model by maximum likelihood, using the Expectation-Maximization (EM) algorithm. The testing method was based on minimum distance classification. We used 64 Gaussian Mixtures for this experiment.

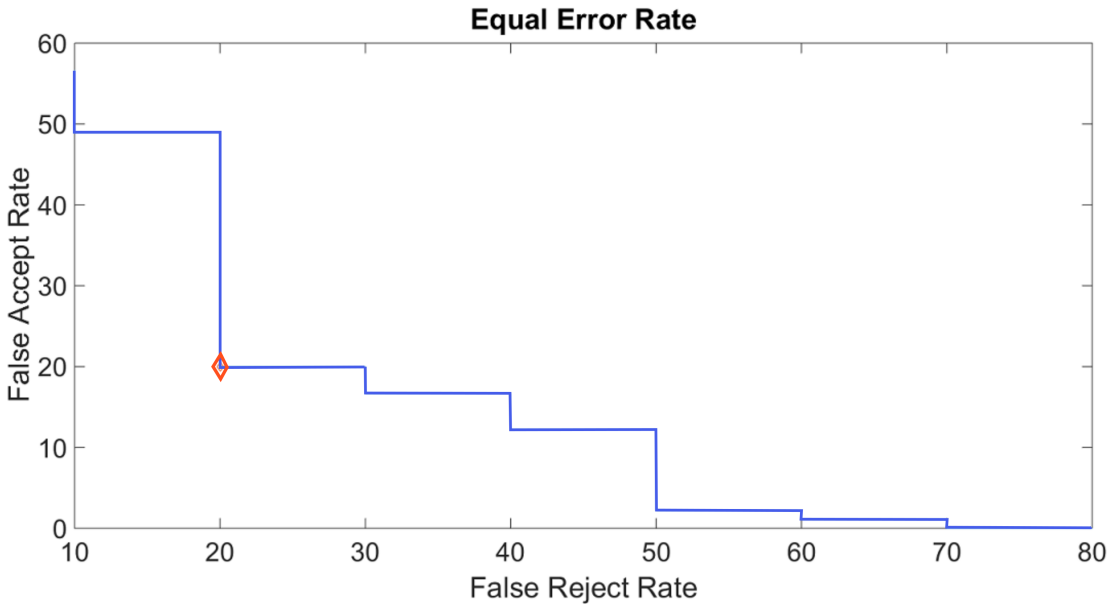

Figure 4.5: Speaker independent verification performance

4.2.1 Text independent speaker independent verification

We conducted a speaker independent Verification experiment as described in [27]. For 10 different hypothesized speakers, 10 independent trails are conducted. All scores are computed using the equation (2.13). All scores where the hypothesized speaker is the speaker in the test utterance are pooled into a set of target scores. The other scores are pooled into a set of nontarget scores. A single, speaker independent, threshold is swept over the two sets of scores. The False Reject Rate (FRR) and the False Accept Rate (FAR) are calculated for each threshold. We automatically chose a threshold for which we got the minimum Equal Error Rate (EER). In our case, we calculated 20% of EER which is similar performance to [27]. As we were looking for better results, we moved forward to speaker dependent verification.

4.2.2 Text independent speaker dependent verification

On an average we obtained 27000 feature vectors for each speaker. We trained the system in two steps: in the first step, we used first 9000 feature vectors for computing the target GMM while we had 9000 imposter vectors (1000 from each 9 speakers) for computing the background speaker model. Then in the second step of training, using these models we computed the 36 scores for 36 segments of speech data in which 18 were from target speaker's speech and 18 were from imposters. Each of the 36 segments contains 500 feature vectors. Based on these 36 scores, we selected a single threshold for individual speaker that provided the minimum equal error rate.

For the verification test we took another 36 segments of speech data in which 18 segments are from target utterance and 18 are from imposters. In this case, each of the 36 segments contains 500 feature vectors also. In this test, we used the threshold as a boundary between the target and the imposters. Using the speaker dependent threshold, we calculated the False Reject Rate (FRR) and the False Accept Rate (FAR) for the same 30 male and 30 female speakers.

An example of speaker dependent verification for a typical speaker is described below. Table 4.1 shows the scores for 18 segments of a target speaker in the first pool and the scores for 18 segments of imposters in the second pool. The pool 1 scores are sorted from low to high and the pool 2 scores are sorted from high to low. Based on these scores, the system automatically set a threshold at 67.89 where we obtained the minimum equal error rate of 5.56%.

We used the computed threshold as a boundary between that individual target speaker and non-target speakers. We took another 18 segments of target speaker and another 18 segments of imposters to verify the system. Table 4.2 shows

Pool 1	***Pool 2***
36.45	255
40.21	255
40.79	247.70
43.46	241.18
43.54	218.94
45.23	193.25
46.70	171.89
47.48	162.63
47.53	158.61
47.58	154.96
48.43	152.17
48.83	145.42
50.16	142.21
50.75	132.69
51.73	124.43
53.33	113.77
67.98	68.63
68.35	67.42

Table 4.1: Scores in Pool 1 and Pool 2 for setting threshold.

Target scores	*Decision*	*Imposters' score*	*Decision*
36.81	Accept	112.32	Reject
56.79	Accept	123.82	Reject
46.83	Accept	233.41	Reject
55.64	Accept	224.30	Reject
61.74	Accept	184.35	Reject
51.39	Accept	160.83	Reject
39.28	Accept	5.05	Accept
55.17	Accept	15.37	Accept
12.16	Accept	137.66	Reject
55.86	Accept	171.96	Reject
46.95	Accept	212.57	Reject
63.71	Accept	240.17	Reject
53.23	Accept	240.27	Reject
58.76	Accept	219.00	Reject
62.98	Accept	115.05	Reject
49.01	Accept	153.74	Reject
55.63	Accept	167.84	Reject
60.59	Accept	161.62	Reject

Table 4.2: Verification based on speaker dependent threshold.

how we used the threshold to accept a segment of speech as a target speaker or reject a segment of speech as an imposter speaker. For this individual speaker we obtained 0% of False Reject Rate (FRR) and 11.11% of False Accept Rate (FAR).

4.2.2.1 Speaker Dependent Thresholds

For training our speaker dependent GMM verification system, we calculated a threshold for each individual speaker based on equal error rate. The thresholds corresponding to 30 male speakers are presented in the Table 4.3 and the thresholds corresponding to 30 female speakers are presented in the Table 4.4.

Speaker No.	*EER*	*Threshold*
speaker 01	5.56%	61.33
speaker 02	0	52.04
speaker 03	0	67.50
speaker 04	0	86.08
speaker 05	5.56%	57.66
speaker 06	0	89.32
speaker 07	0	90.99
speaker 08	0	74.79
speaker 09	0	123.2
speaker 10	0	74.98
speaker 11	0	131.5
speaker 12	0	71.84
speaker 13	0	70.31
speaker 14	0	78.79
speaker 15	0	67.81
speaker 16	0	93.13
speaker 17	0	96.44
speaker 18	0	75.99
speaker 19	0	118.60
speaker 20	0	84.12
speaker 21	5.56%	57.07
speaker 22	0	108.30
speaker 23	0	75.29
speaker 24	0	83.17
speaker 25	0	122.42
speaker 26	5.56%	67.89
speaker 27	0	75.07
speaker 28	0	94.93
speaker 29	5.56%	115.92
speaker 30	5.56%	77.57

Table 4.3: Thresholds of male speakers.

Speaker No.	*EER*	*Threshold*
speaker 01	0	97.94
speaker 02	0	76.78
speaker 03	0	85.69
speaker 04	0	91.90
speaker 05	0	60.98
speaker 06	0	63.15
speaker 07	0	98.82
speaker 08	5.56%	56.59
speaker 09	0	60.79
speaker 10	0	87.51
speaker 11	0	69.87
speaker 12	0	62.64
speaker 13	0	85.27
speaker 14	0	67.14
speaker 15	11.11%	99.99
speaker 16	5.56%	80.72
speaker 17	0	123.10
speaker 18	0	107.12
speaker 19	0	83.21
speaker 20	0	51.15
speaker 21	0	83.30
speaker 22	0	113.68
speaker 23	0	53.98
speaker 24	0	71.08
speaker 25	5.56%	63.84
speaker 26	0	86.91
speaker 27	0	95.57
speaker 28	0	52.28
speaker 29	0	66.96
speaker 30	5.56%	82.61

Table 4.4: Thresholds of female speakers.

4.3 Support Vector Machine

For speaker verification, a one-versus-all strategy was used. For a given target speaker, we took all of target speaker's utterances into class 0 and we took all the remaining nine nontarget speakers into class 1, and then trained a speaker model. Both polynomial and rbf kernels were used for the train/test kernel. In the first experiment, the classifier is a polynomial discriminant function of degree 2. For this purpose, MatLab Machine Learning toolbox is used for coding.

For training we used 18000 feature vectors of target speakers and 18000 feature vectors of imposters (2000 vectors from each nine imposters). For the verification tests, we used 36 segments in which 18 segments are from target utterance and 18 are from imposters. Each of the 36 segments contains 500 feature vectors. We used the same segments of speech for SVM which we used in GMM. If the majority of these 500 vectors are labeled as class 0 by the system, the segment of the speech is classified as target speaker, and if the majority of these 500 vectors are labeled as class 1 by the system, the segment of the speech is classified as an imposter. We calculated the False Reject Rate (FRR) and the False Accept Rate (FAR) for the same 30 male and 30 female speakers and compared with the results of GMM.

For example, SVM is applied on the same target speaker we used in the GMM system. In this experiment, we used the same segments of data we used previously. We present the output of polynomial and rbf kernels, respectively, in Table 4.5 and Table 4.6. For SVM with polynomial kernels we attain 0% of FRR and 38.89% of FAR. However, for SVM with rbf kernels we achieved 0% of FRR and 11.11% FAR which is similar to the results of GMM speaker dependent system.

Verify Target			Verify Imposters		
Target	***Imposter***	***Decision***	***Target***	***Imposter***	***Decision***
382	118	Accept	249	251	Reject
387	113	Accept	281	219	Accept
345	155	Accept	161	339	Reject
363	137	Accept	205	295	Reject
407	93	Accept	196	304	Reject
376	124	Accept	228	272	Reject
405	95	Accept	264	236	Accept
404	96	Accept	215	285	Reject
448	52	Accept	377	123	Accept
402	98	Accept	494	6	Accept
423	77	Accept	271	229	Accept
367	133	Accept	196	304	Reject
374	126	Accept	175	325	Reject
394	106	Accept	120	325	Reject
403	97	Accept	245	255	Reject
340	160	Accept	205	295	Reject
404	96	Accept	281	219	Accept
372	128	Accept	333	167	Accept

Table 4.5: Output of SVM with polynomial kernels of order 2.

Verify Target			Verify Imposters		
Target	***Imposter***	***Decision***	***Target***	***Imposter***	***Decision***
369	131	Accept	205	295	Reject
381	119	Accept	227	273	Reject
307	193	Accept	107	393	Reject
348	152	Accept	173	327	Reject
396	104	Accept	149	351	Reject
358	142	Accept	161	339	Reject
377	123	Accept	196	304	Reject
358	142	Accept	171	329	Reject
427	73	Accept	345	155	Accept
352	148	Accept	463	37	Accept
377	123	Accept	150	350	Reject
332	168	Accept	97	403	Reject
334	166	Accept	163	337	Reject
342	158	Accept	118	382	Reject
359	141	Accept	247	253	Reject
326	174	Accept	204	296	Reject
372	128	Accept	240	260	Reject
339	161	Accept	157	343	Reject

Table 4.6: Output of SVM with 'rbf' kernels.

Chapter 5

Comparison of Results for GMM and SVM Based Systems

In this chapter, we compare the False Reject Rate (FRR) and the False Accept Rate (FAR) in GMM and SVM classification method for the same 30 male and 30 female speakers. In the case of SVM classifier, we present the results for both polynomial and rbf kernels. The following tables contain the 6 sets of results. Each set includes 10 speakers.

5.1 Performance of GMM vs. SVM Classifier

In the training phase of the GMM verification system, we used Equal Error Rate (EER) for selecting individual threshold. However, in the testing phase we calculated False Reject Rate and False Accept rate as a measure of verification.

From the Tables 5.4 and 5.8, we can observe that on an average, GMM classifier with speaker dependent threshold performed better than SVM classifier for the selected speakers. For speaker dependent GMM system we obtained 1.85%

equal error rate for the male speakers and 0.93% false reject rate and 1.48% false accept rate on an average for female speakers. These results are better than for SVM using polynomial kernels. However, SVM using 'rbf' kernels performed better in terms of FRR although speaker dependent GMM performed better than 'rbf' in terms of FAR both for male and female speakers.

5.2 Performance of Polynomial vs.'rbf' Kernels for SVM Classifier

Using Polynomial kernels we obtained 5.74% FRR, 5% FAR for male and 2.59% FRR, 9.26%FAR for female speakers. However, using 'rbf' kernels we achieved 1.11% FRR, 2.59%FAR for male and 0.74% FRR, 3.15% FAR for female speakers. From the results presented in the following tables, it is observed that for the case of the SVM classifier, the system trained using 'rbf' kernels performed much better than polynomial kernels for almost every speaker.

5.3 Comparison between Male and Female Speakers' Performance

In the case of male speakers we obtained both the average FRR and FAR equal to 1.85% [Table: 5.4] for GMM classifier. However, in the case of female speakers we did not get the same percentage of FRR and FAR. For the female speakers FRR is equal to 0.93% and FAR is equal to 1.48%, both of which are less than 1.85% [Table: 5.8]. Based on these results, we can say that speaker dependent GMM classifier performed better for female speakers than male speakers.

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 01	16.67	16.67	22.22	22.22	11.11	11.11
speaker 02	0	0	0	0	0	0
speaker 03	0	0	0	5.56	0	0
speaker 04	0	0	38.89	0	16.67	0
speaker 05	0	0	16.67	0	0	0
speaker 06	0	0	0	0	0	0
speaker 07	0	0	11.11	0	0	0
speaker 08	0	0	0	0	0	0
speaker 09	0	0	0	0	0	0
speaker 10	0	0	0	0	0	0
Average	1.67	1.67	8.89	2.78	2.78	1.11

Table 5.1: GMM and SVM performance comparison for male speakers of set 1

In SVM, using both polynomial and 'rbf' kernels provided lower FRR for female speakers than for male speakers. However, for male speakers we attained lower FAR than for female speakers. Therefore, in terms of FRR, SVM performed better for female speakers than for male speakers although in terms of FAR, SVM performed better for male speakers than for female speakers.

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 11	0	0	0	0	0	0
speaker 12	0	0	22.22	0	0	0
speaker 13	0	0	11.11	0	0	0
speaker 14	0	11.11	0	0	0	5.56
speaker 15	5.56	5.56	0	11.11	0	0
speaker 16	0	0	0	0	0	0
speaker 17	0	0	11.11	0	0	0
speaker 18	0	0	0	0	0	0
speaker 19	0	0	16.67	0	0	0
speaker 20	0	0	5.56	0	0	0
Average	0.56	1.67	6.67	1.11	0	0.56

Table 5.2: GMM and SVM performance comparison for male speakers of set 2

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 21	5.56	11.11	0	27.78	0	22.22
speaker 22	0	0	0	0	0	0
speaker 23	0	0	11.11	11.11	0	0
speaker 24	16.67	0	0	0	0	0
speaker 25	11.11	0	0	11.11	0	5.56
speaker 26	0	11.11	0	38.89	0	11.11
speaker 27	0	0	0	0	0	0
speaker 28	0	0	5.56	0	5.56	0
speaker 29	0	0	0	11.11	0	11.11
speaker 30	0	0	0	11.11	0	11.11
Average	3.33	2.22	1.67	11.11	0.56	6.11

Table 5.3: GMM and SVM performance comparison for male speakers of set 3

GMM			SVM-Polynomial		SVM-rbf	
Error Rate	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
Average	1.85	1.85	5.74	5	1.11	2.59

Table 5.4: Average performance comparison for 30 male speakers.

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 01	0	0	0	0	0	0
speaker 02	0	0	0	11.11	0	11.11
speaker 03	0	0	0	0	0	0
speaker 04	0	0	0	5.56	0	0
speaker 05	0	5.56	0	11.11	0	0
speaker 06	0	0	0	5.56	0	0
speaker 07	0	0	22.22	0	0	0
speaker 08	16.67	0	5.56	22.22	0	11.11
speaker 09	0	0	0	0	0	0
speaker 10	0	0	0	5.56	0	0
Average	1.67	0.56	2.78	6.11	0	2.22

Table 5.5: GMM and SVM performance comparison for female speakers of set 4

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 11	0	0	0	0	0	0
speaker 12	0	0	0	16.67	0	0
speaker 13	0	0	16.67	0	0	0
speaker 14	0	0	0	0	0	0
speaker 15	11.11	11.11	0	11.11	0	0
speaker 16	0	11.11	0	5.56	0	0
speaker 17	0	0	0	27.78	0	5.56
speaker 18	0	0	0	0	0	0
speaker 19	0	0	0	0	0	0
speaker 20	0	0	0	0	0	0
Average	1.11	2.22	1.67	6.11	0	0.56

Table 5.6: GMM and SVM performance comparison for female speakers of set 5

GMM			SVM-Polynomial		SVM-rbf	
ID	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
speaker 21	0	0	0	16.67	0	0
speaker 22	0	0	0	22.22	0	5.56
speaker 23	0	0	0	27.78	0	11.11
speaker 24	0	0	33.33	5.56	22.22	5.56
speaker 25	0	11.11	0	33.33	0	22.22
speaker 26	0	0	0	5.56	0	0
speaker 27	0	0	0	0	0	0
speaker 28	0	0	0	22.22	0	5.56
speaker 29	0	5.56	0	16.67	0	16.67
speaker 30	0	0	0	5.56	0	0
Average	0	1.67	3.33	17.28	2.22	6.67

Table 5.7: GMM and SVM performance comparison for female speakers of set 6

GMM			SVM-Polynomial		SVM-rbf	
Error Rate	***FRR***	***FAR***	***FRR***	***FAR***	***FRR***	***FAR***
Average	0.93	1.48	2.59	9.26	0.74	3.15

Table 5.8: Average performance comparison for 30 female speakers.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We developed a speaker independent GMM verification system and achieved 20% of equal error rate which is similar to MIT Lincoln Lab's result [15]. In the quest of obtaining better performance we introduced speaker dependent threshold for each individual speaker. This method secured much better performance. Using speaker dependent GMM system we obtained 1.85% equal error rate for the male speakers and 0.93% average false reject rate and 1.48% average false accept rate for female speakers. For US-English speakers, MIT LL reported 2% of EER in [6]. They used a fusion of GMM and SVM Classifier.

We also developed a SVM verification system using the same set of data. We achieved better performance using 'rbf' kernels than using polynomial kernels. Firstly, we implemented SVM with polynomial kernels and observed 5.74% FRR, 5% FAR for male and 2.59% FRR, 9.26%FAR for female speakers. For pursuing better performance, we used SVM with 'rbf' kernels and achieved 1.11% FRR, 2.59% FAR for male and 0.74% FRR, 3.15% FAR for female speakers.

6.2 Future Work

This thesis project focused on speaker verification system based on Gaussian Mixture Model and Support Vector Machine. In the future, a similar system can be developed by using Inner Product Discriminant Functions (IPDFs), Joint Factor Analysis (JFA), SVM GMM super vector system (SVM GSV), or Total Variability system.

From the system performance, we observed that FAR is higher than the FRR in every experiment except speaker dependent GMM for male speakers. Future work could develop a system with equal FRR and FAR.

We concentrated on low level spectral features for all the experiments. For using in the real world, this project can be extended to include high level features to develop an speaker verification system.

Appendices

Appendix A Male speakes' ID in NIST SRE 2008 database

We have chosen 30 US-English speaking male speakers for the experiments. In the NIST SRE database these speakers have corresponding speaker IDs. However, we tagged them as speaker 01, speaker 02, ..., speaker 30. The nomenclature is included in the following table.

Our speaker No.	*Database speaker ID*
speaker 01	12054
speaker 02	15693
speaker 03	17601
speaker 04	18325
speaker 05	22073
speaker 06	23885
speaker 07	26317
speaker 08	28239
speaker 09	34128
speaker 10	34766
speaker 11	36037
speaker 12	37017
speaker 13	37222
speaker 14	40530
speaker 15	41657
speaker 16	44537
speaker 17	44931
speaker 18	45941
speaker 19	50650
speaker 20	54944
speaker 21	62745
speaker 22	65038
speaker 23	67233
speaker 24	68170
speaker 25	69142
speaker 26	70280
speaker 27	72717
speaker 28	75263
speaker 29	76453
speaker 30	76857

Table 1: Male speakers' ID

Appendix B Female speakes' ID in NIST SRE 2008 database

We have chosen 30 US-English speaking female speakers for the experiments. In the NIST SRE database these speakers have corresponding speaker IDs. However, we tagged them as speaker 01, speaker 02, ..., speaker 30. The nomenclature is included in the following table.

Our speaker No.	*Databe speaker ID*
speaker 01	15417
speaker 02	17053
speaker 03	17709
speaker 04	18262
speaker 05	18788
speaker 06	31241
speaker 07	33186
speaker 08	33509
speaker 09	35963
speaker 10	38585
speaker 11	46223
speaker 12	47522
speaker 13	48019
speaker 14	53137
speaker 15	55550
speaker 16	58414
speaker 17	59140
speaker 18	64473
speaker 19	66158
speaker 20	70066
speaker 21	70602
speaker 22	71076
speaker 23	72055
speaker 24	74580
speaker 25	78263
speaker 26	79091
speaker 27	82451
speaker 28	83588
speaker 29	90240
speaker 30	90863

Table 2: Female speakers' ID

Bibliography

[1] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 430–451, Jan. 2004.

[2] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, pp. IV–4072 – IV–4075, May 2002.

[3] D. Sturim, W. Campbell, and D. Reynolds, "Classification methods for speaker recognition," in *Speaker Classification I* (C. Muller, ed.), vol. 4343 of *Lecture Notes in Computer Science*, pp. 278–297, Springer Berlin Heidelberg, 2007.

[4] J. Deller, J. Proakis, and J. Hansen, *Discrete-time processing of speech signals*. Macmillan Pub. Co., 1993.

[5] D. Sturim, W. Campbell, N. Dehak, Z. Karam, A. McCree, D. Reynolds, F. Richardson, P. Torres-Carrasquillo, and S. Shum, "The MIT LL 2010 speaker recognition evaluation system: Scalable language-independent speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5272–5275, IEEE, 2011.

[6] D. E. Sturim, W. M. Campbell, Z. N. Karam, D. A. Reynolds, and F. S. Richardson, "The MIT Lincoln Laboratory 2008 speaker recognition system.," in *INTERSPEECH*, pp. 2359–2362, 2009.

[7] S. S. Kajarekar, N. Scheffer, M. Graciarena, E. Shriberg, A. Stolcke, and L. Ferrer, "The SRI NIST 2008 speaker recognition evaluation system," 2009.

[8] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[9] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*. Pearson Education India, 2002.

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

[11] J. Lyons, "Mel frequency cepstral coefficient (mfcc) tutorial @ONLINE."

[12] J. Tierney, "A study of lpc analysis of speech in additive noise," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 389–397, 1980.

[13] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[14] J. P. Campbell Jr, "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[15] M. Hebert, "Text independent speaker recognition," in *Springer Handbook of Speech Processing* (J. Benesty, M. Sondhi, and Y. Huang, eds.), pp. 743–762, Springer Berlin Heidelberg, 2008.

[16] G. R. Doddington *et al.*, "Speaker recognition based on idiolectal differences between speakers.," in *INTERSPEECH*, pp. 2521–2524, 2001.

[17] J. Navrátil, Q. Jin, W. D. Andrews, and J. P. Campbell, "Phonetic speaker recognition using maximum-likelihood binary-decision tree models," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 4, pp. IV–796, IEEE, 2003.

[18] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Advances in neural information processing systems*, 2003.

[19] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "High-level speaker verification with support vector machines," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 1, pp. I–73, IEEE, 2004.

[20] B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. A. Reynolds, and B. Xiang, "Using prosodic and conversational features for high-performance speaker recognition," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 4, pp. IV–792, IEEE, 2003.

[21] A. G. Adami, R. Mihaescu, D. A. Reynolds, and J. J. Godfrey, "Modeling prosodic dynamics for speaker recognition," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 4, pp. IV–788, IEEE, 2003.

[22] R. C. Rose and D. A. Reynolds, "Text independent speaker identification using automatic acoustic segmentation," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pp. 293–296, IEEE, 1990.

[23] D. A. Reynolds, *"Gaussian mixture modeling approach to text-independent speaker identification"*. PhD thesis, Georgia Institute of Technlogy, 1990.

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

[25] D. A. Reynolds, "Automatic speaker recognition using gaussian mixture speaker models," in *The Lincoln Laboratory Journal*, Citeseer, 1995.

[26] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification.," in *Eurospeech*, 1997.

[27] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models.," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[29] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, pp. 210–229, 2006.

[30] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000.

[31] R. Collobert and S. Bengio, "Svmtorch: Support vector machines for large-scale regression problems," *The Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[32] A. E. Rosenberg, J. DeLong, C.-H. Lee, B.-H. Juang, and F. K. Soong, "The use of cohort normalized scores for speaker verification.," in *ICSLP*, vol. 92, pp. 599–602, 1992.

[33] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, pp. I–161, IEEE, 2002.

[34] J. Schnermann, *Pattern Classification*. John Wiley and Sons, Inc., 1996.

[35] S. Fine, J. Navratil, and R. A. Gopinath, "A hybrid gmm/svm approach to speaker identification," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, 2001.

[36] L. R. Rabiner and M. R. Sambur, "Approaches to speaker detection and tracking in conversational speech.," *The Bell System Technical Journal*, vol. 54, no. 0005-8580, pp. 297 – 315, 1975.

[37] M. Slaney, "Auditory toolbox," 1994.