A Segment-Based Speaker Verification System Using *SUMMIT*

by

Sridevi Vedula Sarma

B.S., Cornell University, 1994

Submitted to the Department of Electrical Engineering
and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

April 1997

Author . . . .
Department of Electrical Engineering
and Computer Science
September 19, 1997

Certified by . . . .
Victor W. Zue
Senior Research Scientist
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

A Segment-Based Speaker Verification System Using *SUMMIT*

by

Sridevi V. Sarma

Submitted to the Department of Electrical Engineering
and Computer Science
on September 19, 1997 in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis describes the development of a segment-based speaker verification system. Our investigation is motivated by past observations that speaker-specific cues may manifest themselves differently depending on the manner of articulation of the phonemes. By treating the speech signal as a concatenation of phone-sized units, one may be able to capitalize on measurements for such units more readily. A potential side benefit of such an approach is that one may be able to achieve good performance with unit (i.e., phonetic inventory) and feature sizes that are smaller than what would normally be required for a frame-based system, thus deriving the benefit of reduced computation.

To carry out our investigation, we started with the segment-based speech recognition system developed in our group called SUMMIT [44], and modified it to suit our needs. The speech signal was first transformed into a hierarchical segment network using frame-based measurements. Next, acoustic models for each speaker were developed for a small set of six phoneme broad classes. The models represented feature statistics with diagonal Gaussians, which characterized the principle components of the feature set. The feature vector included averages of MFCCs, plus three prosodic measurements: energy, fundamental frequency (F0), and duration. The size and content of the feature vector were determined through a greedy algorithm optimized on overall speaker verification performance.

To facilitate a comparison with previously reported work [19, 2], our speaker verification experiments were carried out using 2 sets of 100 speakers from the TIMIT corpus. Each speaker-specific model was developed from the eight SI and SX sentences. Verification was performed using the two SA sentences common to all speakers. To classify a speaker, a Viterbi forced alignment was determined for each test utterance, and the forced alignment score of the purported speaker was compared with those obtained with the models of the speaker's competitors. Ideally, the purported speaker's score should be compared to scores of every other system user. To reduce the computation, we adopted a procedure in which the score for the purported speaker is compared only to scores of a cohort set consisting of a small set of acoustically similar speakers. These scores were then rank ordered and the user was accepted if his/her model's score was within the top N scores, where N is a parameter we varied in our experiments. To test for false acceptance, we used only the members of a speaker's cohort set as impostors. We have found this method to significantly reduce computation while minimally affecting overall performance.

We were able to achieve a performance of 0% EER in a clean domain and 7.47% EER in a noisy domain, with a simple system design. We reduced computation significantly through the use of a small number of features representing broad-classes, diagonal Gaussian speaker models, and using only cohort sets during testing.

Thesis Supervisor: Victor W. Zue

Title: Senior Research Scientist

Acknowledgments

The Spoken Language Systems (SLS) group provides the ideal research environment, a place where one can grow extensively both academically and emotionally. I cannot even begin to describe how much I have learned from all members of this group. My experience begins with my thesis advisor Dr. Victor Zue, for whom I have the deepest respect and gratitude. Victor not only gave me the wonderful opportunity to be a part of the SLS group, but he also provided his full support and confidence these last 16 months. We communicated frequently with productive meetings, and I often found Victor pondering over my project in his own free time along with myself. We worked as a team and without his optimism and motivation, I may not have had the strength to dig myself out of the bad times.

I would also like to thank my good friend Rajan Naik for all his support, love, and encouragement in my academics. He lifted my spirits up when I was down and stood by my side unconditionally.

I also wish to thank all the other members of the Spoken Language Group for all of the support and friendship they have provided me. In particular I would like to thank:

Jane Chang for her time and patience in teaching me everything there is to know about the SLS libraries and programming languages to successfully pursue my research. I am deeply in debt to her and hope to help another group member half as much as she has helped me.

Giovanni Flammia for all his help in brushing up my skills in C and Latex and for patiently teaching me Perl. In addition to sharing his vast background in all programming languages and the Internet, he also shared a sense of humor which often made stressful times durable.

Michelle Spina for being such a good friend and colleague. She always encouraged me and guided me through rough times by sharing her experiences in academia. She also patiently helped me out during my learning stages of programming in C and C++.

Mike McCandless for his patience with my pestering him with technical questions about how to use the recognizer and of course always having a solution to my bugs.

Ben Serridge for being a great office-mate and for giving me a beautiful bouquet of flowers after I took my oral qualifiers!

Tim Hazen for teaching me details about various methods of speech and speaker recognition.

Ray Lau, Alex Manos, and Ray Chun for all the help they've provided and questions they've answered.

Jim Hugunin for introducing me to Python.

Jim Glass for acting as another supportive advisor by discussing my work, offering great advice, and for relating his experiences in the field.

Lee Hetherington for his spur of the moment advice and ideas.

Stephanie Seneff for her advice about being a female at M.I.T, and of course for making me realize how much I have yet to learn!

David Goddeau, Joe Polifroni, and Christine Pao for keeping our systems up and running, which was a difficult task this year.

Vicky Palay and Sally Lee for everything they have done for the group and for dealing patiently with my concerns.

I would also like to thank colleagues outside SLS who have helped me tremendously in adjusting to M.I.T. Babak Ayazifar taught me how to survive and succeed here and is a reliable mentor, not to mention an excellent teacher. Likewise, I cannot thank Stark Draper enough as he was my sole partner my first semester at M.I.T. We worked well together and I learned a great deal from him.

I would also like to acknowledge my two best childhood friends Paulina Vaca and Garrett Robbins. They make me laugh until my stomach hurts, and they allow me to escape this technical world to catch a glimpse of the world of literature and art.

Finally, I wish to thank my family for all their love and support. Without their encouragement to succeed in life in all my endeavors, and their stress of importance for higher education, I may not have had the strength to come this far.

This research was supported partially by NSF and DARPA.

Contents

List of Figures

List of Tables

Chapter 1

Speaker Verification

1.1 Introduction

Speaker verification involves the task of automatically verifying a person's identity by his/her speech through the use of a computer. The outcome of speaker verification is a binary decision as to whether or not the incoming voice belongs to the purported speaker. Speaker verification has been pursued actively by researchers, because it is presently a palpable task with many uses that involve security access authorizations. In the past, applications for speaker verification systems mainly involved physical access control, automatic telephone transaction control (*e.g.*, bank-by-phone), and computer data access control. However, due to the revolution in telecommunications, uses for speaker verification systems also include Internet access control, and cellular telephone authorizations.

Figure 1-1 illustrates the basic components of a speaker verification system. The feature extraction component attempts to capture acoustic measurements from the user's speech signal that are relevant to inter-speaker differences. During training, the acoustic features are used to build speaker-specific models. During testing, measurements extracted from the test data are scored against the stored speaker models to see how well the test data match the reference models. The speaker is accepted or rejected based on this score. Of course, many details are left out of the block diagram, such as the *type of text* the system prompts, the *features* the system extracts, and

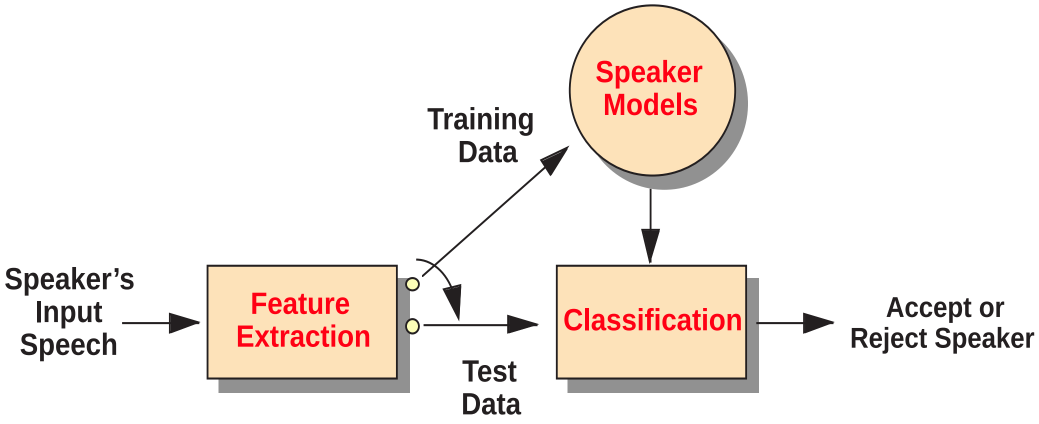


Figure 1-1: General Speaker Verification System

the *speaker models* and *classifiers* the system implements. For detailed tutorials on speaker verification, refer to [27, 6].

1.2 Previous Research

Research in speaker verification has been active for many years. In this section, we describe general approaches to speaker verification research in the last 3 decades, and illustrate these methods with a few specific examples.

During the late 1960's and 1970's, researchers mainly used knowledge-based approaches to speaker verification research. Since many of the researchers are speech scientists knowledgeable of the acoustic-phonetic encoding of speech, they focused their attention on the discovery of features, typically measured across speech segments. Speech segments, or phone units, were believed to be the appropriate choice of units, because speaker-specific cues may manifest themselves differently depending on the manner of articulation of phones. While these features may be sound on theoretical grounds, algorithms for automatically computing these features were inadequate. Consequently, investigators resorted to manually segmenting speech data and estimating features to conduct their studies, which constrained the amount of data observed, and the statistical validity of their results.

One example of research done in this era is the doctoral thesis of Wolf [41]. Wolf found specific segmental measurements that discriminated well among speakers. He investigated 17 different features such as, fundamental frequency (F0), glottal source spectral slopes, duration, and features characterizing vowel and nasal spectra. During training, 21 male speakers repeated 6 short sentences 10 times. Nine of the repetitions of each utterance were used to develop speaker templates consisting of means and variances of the features. The remaining sentences were used to test the speakers. During testing, Euclidean distances between test data and speaker templates were used to classify speakers. Wolf used the F-ratio analysis of variance to evaluate the speaker-discriminating abilities of the measurements. The F-ratio is a weighted ratio of the variance of speaker means to the average of speaker variances. Wolf found that features with high F-ratios resulted in 100% speaker classification accuracy.

Wolf's study showed that segment-based features discriminate well among speakers. Using phonetic units is also advantageous, because the verification can be independent of the particular words the users says. However, Wolf extracted the features from manually segmented speech data. Consequently, he could not build an automated speaker verification system that derived the benefits of his knowledge-based approach. Other studies that also used knowledge-based approaches to speaker verification are described in [38, 14].

In the 1980s, researchers abandoned the notion of using segment-based measurements for speaker verification, because algorithms to automatically segment speech remained inadequate. Instead, investigators began using measurements that are easily computed automatically, such as features extracted from speech frames. Frame-based features may not necessarily distinguish speakers well. However, these measurements allowed researchers to build automated systems. These systems typically modeled speakers with word templates. The templates represented speech frames of words with feature centroids. Just as before, speakers were classified with distances computed between test feature vectors and centroids.

One of the earliest automated speaker verification systems was implemented in the early 1980's at Texas Instruments (TI) corporate headquarters in Dallas, Texas [6].

The system automatically computed features from 6 frames for each word, regardless of the word's duration. Specifically, each frame used the output of a 14 channel filter bank, uniformly spaced between 300 and 3000Hz, as a 14x1 spectral amplitude feature vector. During training, templates for 16 words were constructed for each speaker. During testing, the system prompted 4-word utterances constructed randomly from the 16 word bank. A Euclidean distance between measurements of test frames and reference frames was then computed, and used to make a verification decision. At the time, the system achieved 99.1% acceptance rate of valid users, and 0.7% acceptance rate of impostors. Similar speaker verification systems that use template matching classification techniques are described in [15, 9].

As mentioned above, these pioneering systems typically modeled words with templates for each speaker. Templates do not capture variations in the acoustic feature space, because each frame is represented by a fixed acoustic centroid. Consequently, the templates are not robust models of speech. In addition, the system is dependent on the words the users says during verification.

In the early 1990s, statistical models of speech became popular for speech recognition, because the models represent the acoustic feature space with a distribution, rather than a fixed centroid. As a result, researchers began applying the technology to speaker verification. Specifically, speaker verification research focused on investigating hidden Markov models (HMMs), because HMMs were becoming very successful in speech recognition [32]. Many investigators simply modified existing speech recognition systems for speaker verification, in hopes of achieving high performance. HMMs are developed from frame-based features; therefore, investigators neglected to further explore segment-based features. In fact, most of the studies use frame-based cepstral measurements, and compare different HMM speaker models to each other.

An HMM models speech production as a process that is only capable of being in a finite number of different states, and each state generates either a finite number of outputs or a continuum of outputs. The system transitions from one state to another at discrete intervals of time, and each state produces a probabilistic output [27]. In a speaker verification system, each speaker is typically represented by an HMM, which may capture statistics of any component of speech such as a sub-phone, phone, subword, word etc. To verify the speaker, the test sentence is scored by the HMM. The score represents the probability of an observation sequence, given a test sequence and a speaker HMM.

Furui and Matsui investigated various HMM systems for speaker verification. In one study [25], they built a word-independent speaker verification system and compared discrete HMM to continuous HMM speaker models. The speaker verification system computed frame-based cepstral features, and the corpus consisted of 23 male and 13 female speakers, recorded during three sessions over a period of 6 months. Ten sentences were used to train both continuous and discrete HMMs for each speaker, and 5 sentences were used to test the speakers. During testing, the purported speaker's cumulative likelihood score was used to make a verification decision. Furui and Matsui reached a performance of 98.1% speaker verification rate, using continuous HMMs. Other studies that are based on HMMs include [24, 36, 35].

Recently, investigators have applied other statistical methods, such as neural networks, to speaker verification. Neural networks have also been successful in other tasks, such as speech and handwriting recognition. They are statistical pattern classifiers that utilize a dense interconnection of simple computational elements, or nodes [20]. The layers of nodes operate in parallel, with the set of node outputs in a given layer providing the inputs to each of the nodes in a subsequent layer. In a speaker verification system, each speaker is typically represented by a unique neural network. When a test utterance is applied, a verification decision is based on the score for the speaker's models. Some examples of systems that use neural networks to represent and classify speakers are [42, 3, 18, 28, 37].

1.3 Discussion

Thirty years ago, researchers manually computed segment-based acoustic features, and modeled the speech signal with templates consisting of acoustic centroids. Presently, systems automatically compute frame-based acoustic features, and use statistical models to represent the speech signals, such as HMMs and neural networks. As Matsui and Furui showed in one of their studies [25], most statistical methods give improved performance over template methods. In addition, frame-based measurements are easy to compute and are successful in speaker verification. However, segment-based features have been proven to carry speaker-specific cues, and may result in equivalent performance with less dimensionality.

1.4 Thesis Objective and Outline

The ultimate goal of speaker verification research is to develop user-friendly, high performance systems, that are computationally efficient and robust in all environments. In this study, we strive to develop a competitive segment-based speaker verification system; and, after developing a viable system, we explore two methods to reduce computation.

We automatically compute segment-based measurements, and use statistical models of speech to represent speakers. Therefore, we combine two successful approaches to speaker verification, knowledge-based and statistical. As a result, we hope to achieve competitive speaker verification performance. We do not investigate robustness issues specifically. However, we explore acoustic features that have been proven to be robust in the past, such as fundamental frequency and energy [42, 17].

To achieve our goal, we modified SUMMIT, a state-of-the-art speech recognition system developed at MIT [44], for speaker verification. We chose SUMMIT for the following reasons. First, SUMMIT treats the speech signal as a concatenation of segments, which allows us to capitalize on the speaker-discriminating abilities of such phonetic-size units. Second, SUMMIT allows us to model the features statistically; therefore we can also capture feature-varying attributes in the speech signal. Finally, SUMMIT employs search algorithms, which allows us to modify the algorithms to conduct a search for an optimal feature set. We search for an optimal feature set from an initial pool of measurements, which include cepstral and prosodic measurements. The feature search, described in 4, is one of the two computationally efficient methods explored.

Details of our speaker verification system and its design are given in chapter 2. The system description is followed by a summary of the system's performance. Chapter 4 describes two methods used to reduce computation during training and testing, and reports performance when the methods are employed. Finally, chapter 5 summarizes conclusions of our system results, and proposes future work that remains in the area.

Chapter 2

System Description

2.1 Introduction

In this chapter, we describe the components of our speaker verification system. Figure 2-1 summarizes our system with a block diagram, whose building blocks are components of SUMMIT, modified to suit our needs. Initially, signal processing transforms the speech samples to frame-based acoustic features. These features are then used to propose a segmentation network for the utterance. Next, the acoustic measurements are averaged across segments, and rotated into a space that de-correlates them, via principal components analysis (PCA) (section 2.5). During training, diagonal Gaussian speaker models are developed. During testing, the speaker models are used to compute forced alignment scores (section 2.6.2) for test utterances. Finally, the scores (section 2.6.2) are used to classify speakers, and make a verification decision.

This chapter begins with a description of the corpus used to train and evaluate our system. Next, the acoustic features selected to represent the speech signal are discussed. Thereafter, the algorithm used to create a segmentation network from the frame-based features is described, and followed by a discussion of a search for an optimal set of segment-based measurements. Finally, details are given on how speaker models were developed, and how speakers were classified.

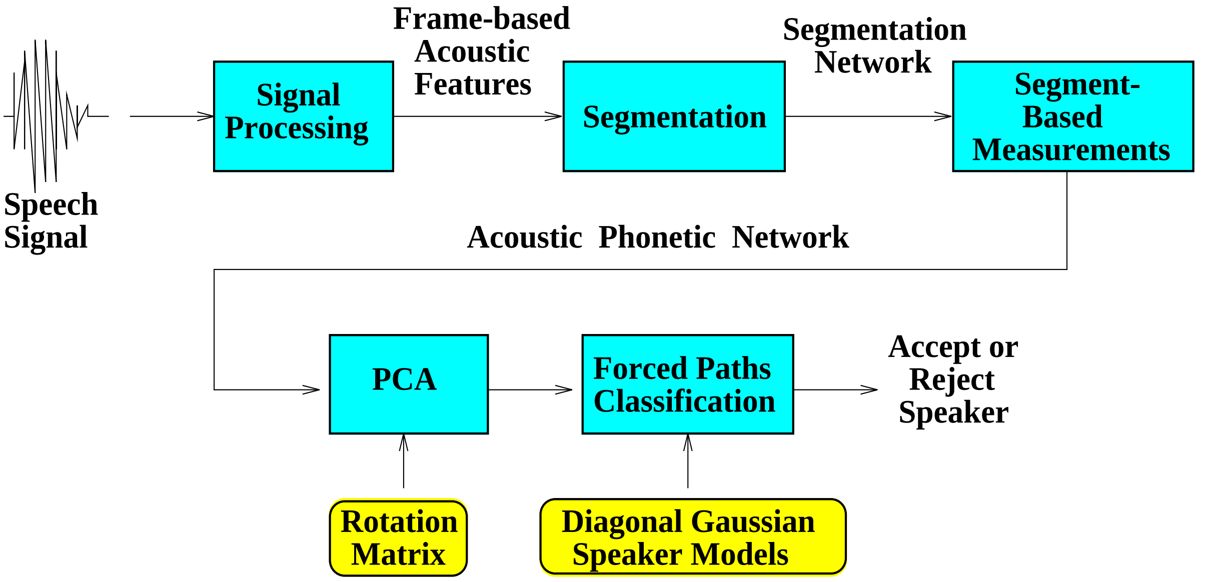

Speech
Signal
Signal Processing
Frame-based Acoustic Features
Segmentation
Segmentation Network
Segment-Based Measurements
Acoustic Phonetic Network
PCA
Forced Paths Classification
Accept or Reject Speaker
Rotation Matrix
Diagonal Gaussian Speaker Models

Figure 2-1: Speaker Verification System

2.2 Corpus

Many researchers in speaker verification use a variety of existing corpora, while others collect their own data. We chose to use the TIMIT corpus for a variety of reasons [10]. First, TIMIT is publicly available and widely used. Therefore, it facilitates a direct comparison of our work with that of others. Second, TIMIT contains data for many speakers, and provides time-aligned phonetic transcriptions. Thus, TIMIT allows us to easily develop phonetic models for each speaker. In addition, TIMIT consists of sentences, which create a more natural environment for users than, for example, passwords or digit combinations. YOHO, a corpus specifically designed for speaker verification, contains large amounts of data per speaker and a large number of speakers. However, the corpus consists solely of digits [4]. Finally, NTIMIT, a corpus obtained by transmitting TIMIT over a telephone network, is also publicly available [16]. Since our work includes investigating speaker verification performance in noisy environments, such as the telephone domain, the availability of NTIMIT allows us to replicate experiments under noisy conditions, and to make meaningful comparisons to clean speech (TIMIT) results.

2.2.1 TIMIT

TIMIT consists of 630 speakers, 70% male and 30% female, who represent 8 major dialect regions of the United States. We selected a subset of 168 speakers (TIMIT's standard NIST-test and NIST-dev sets) for evaluation. Each speaker read a total of 10 sentences, 2 dialect (SA), 5 phonemically rich (SX), and 3 other (SI) sentences. The 2 SA utterances are the same across all speakers, while the 3 SI sentences are unique to each speaker. A collection of 450 SX sentences in TIMIT are each read by 7 speakers, whereas 1890 sentences from the Brown corpus were each read by one speaker. We used 8 sentences (SX,SI) to develop each speaker model, and the remaining 2 SA sentences to test each speaker. Since 8 utterances may not adequately model a speaker's sound patterns, it is necessary to compensate for the lack of training data. In this study, the complexity of the speaker models is reduced by forming broad phonetic classes.

2.2.2 Broad Classes

As mentioned above, 8 utterances do not contain enough tokens to adequately model all phones separately. Therefore, we increased the number of tokens per model by collapsing phones into broad classes. For the speaker verification task, the broad classes should capture speaker-specific cues. Since past observations have shown that speaker trends are easily captured in the broad manner classes [30, 41], we chose to collapse the 61 TIMIT-labeled phones into 6 broad manner classes. As a result, each speaker is represented by 6 broad class distributions, as opposed to 61 phone distributions, and the average number of tokens per model increases by a factor of 10.[1]

The manner classes are obtained based on our knowledge about acoustic phonetics, and consist of vowels, nasals, weak fricatives, strong fricatives, stops, and silence. The exact content of each manner class is shown in Table 2-1.

[1] The average number of tokens per phone is 5, whereas the average number of tokens per broad class is 50.

CLASS	**PHONES**
Vowels	iy,ih,eh,aa,ay,ix,ey,oy,aw,w,r,l,el,er,ah,ax,ao,ow,uh,axr,ax-h,ux,ae
Stops	b,d,g,p,t,k
Nasals	m,em,n,en,nx,ng,eng,dx,q
Strong Frics	s,sh,z,zh,ch,jh
Weak Frics	f,th,dh,v,hh,hv
Silence	pcl,tcl,kcl,bcl,dcl,gcl,pau,epi,h#

Table 2-1: Phone Distributions of Broad Manner Classes

The selection of the classes affects the performance of each feature set. For example, voiced and unvoiced stops are clustered together into one stop class. Voiced and unvoiced stops differ significantly in duration, because voiceless stops have added aspiration. Thus, speaker distributions for the stop class, using duration as a feature, will have large variances. These large variances make it difficult to distinguish among the users, and may result in poor speaker verification performance.

2.3 Signal Representations

After choosing a corpus, we collected 17 features to represent the speech signal. The features include measurements that are commonly used in speaker verification systems, such as MFCCs, in addition to three prosodic measurements: fundamental frequency, energy and duration. Below, we describe why the above features were selected for the speaker verification task, and how we computed them.

2.3.1 MFCCs

Mel-frequency-based cepstral coefficients (MFCCs) are perhaps the most widely used features in speaker verification. MFCCs are cepstral features obtained from a system that approximates the frequency response of the human ear. Presumably, MFCCs have been successful in speaker verification because they capture inter-speaker differences. It can be shown via cepstral analysis of speech [29] that MFCCs carry vocal tract information (i.e., formant frequency locations), as well as fundamental frequency information. The vocal tract system function is dependent on the shape and size of the vocal tract, which is unique to a speaker and the sound that is being produced. Fundamental frequency (F0) also carries speaker-specific information, because F0 is dependent on accents, different phonological forms, behavior and other individualistic factors [42, 1].

To compute MFCCs, the speech signal was processed through a number of steps. First, the digitized utterances were initially passed through a pre-emphasis filter, which enhances higher frequency components of the speech samples, and attenuates lower frequency components. Next, a short time Fourier transform (STFT) of the samples was computed at an analysis rate of 200 Hz, using a 20.5 ms Hamming window. The STFT thus produced one frame of spectral coefficients every 5 seconds. Then, each of the coefficients was squared component-wise to produce the power spectral density (PSD) for each frame. Thereafter, the logarithm of the PSD was computed and the resulting coefficients were processed by an auditory filter bank, which produced mel-frequency spectral coefficients (MFSCs). Finally, the MFSCs were rotated by the discrete cosine transform (DCT) matrix. The matrix transformed the mel-frequency spectral coefficients (MFSCs) to 14 less correlated MFCCs. More details are given in Appendix A.

2.3.2 Prosodic Features

In addition to MFCCs, we decided to explore three prosodic features: fundamental frequency (F0), energy and duration. These features attempt to measure psychophysical perceptions of intonation, stress, and rhythm, which are presumably characteristics humans use to differentiate between speakers [6]. Prosodic features have also proven to be robust in noisy environments [42, 17, 1]. Therefore, these features show great potential for the speaker verification task.

To estimate F0, we used the ESPS tracker, in particular the FORMANT function [7]. For each frame of sampled data, FORMANT estimates speech formant trajectories, fundamental frequency, and other related information. The ESPS formant tracker implements the linear prediction analysis method, described in Appendix B, to estimate F0. FORMANT also uses dynamic programming and continuity constraints to optimize the estimates of F0 over frames. Although the tracker also estimates probabilities of voicing for each frame, we retained F0 information for every frame, regardless of whether the underlying sounds were voiced or unvoiced.

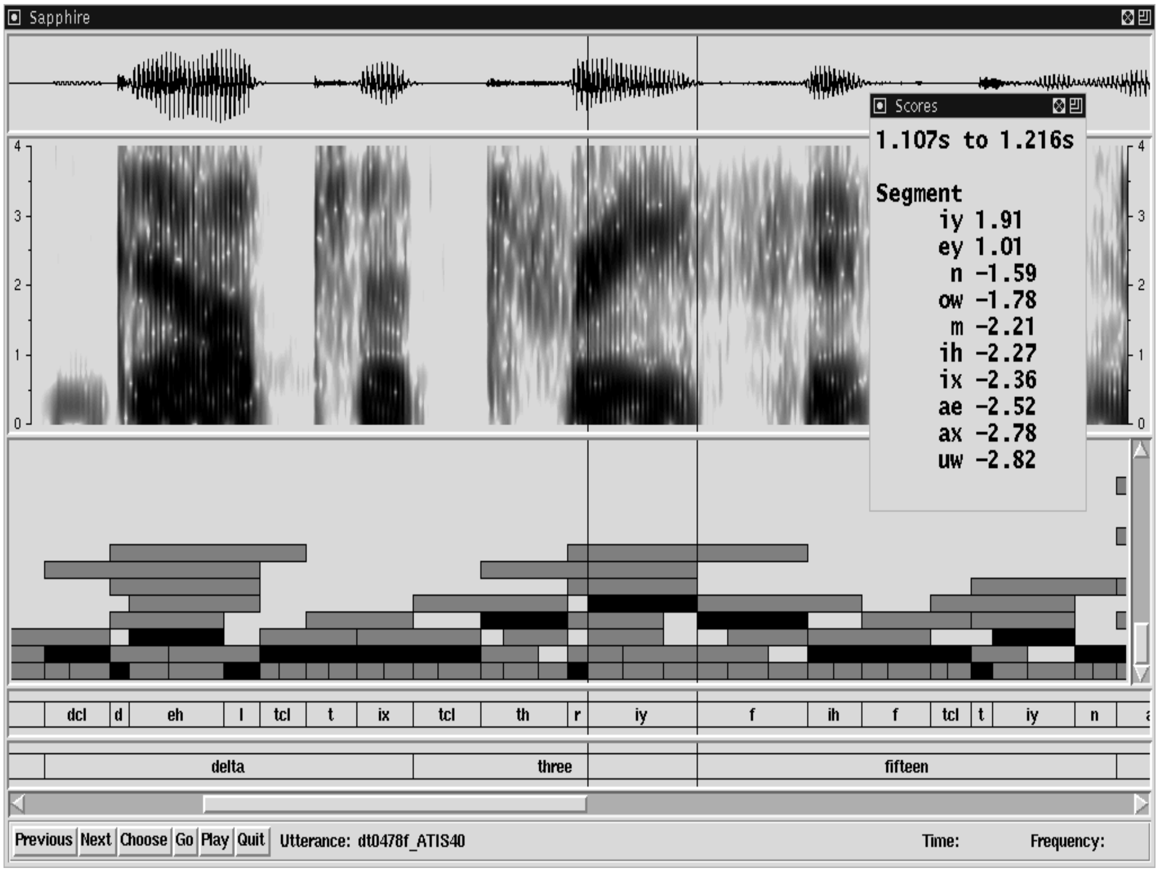
To compute energy, the power spectral density coefficients for each frame, obtained in the same manner as described in section 2.3.1, were summed. We computed the logarithm of this sum to convert energy to the decibel (dB) scale. The logarithm of duration was also computed in our experiments.

2.4 Segmentation

Once frame-based acoustic features are computed, the system proposes possible segmentations for the utterance. The goal of the segmenter is to prune the segment search space using inexpensive methods, without deleting valid segments. During segmentation, frame-based MFCCs are used to first establish acoustic landmarks in the utterance. Then, a network of possible acoustic-phonetic segments are created from the landmarks.

Acoustic landmarks are established in two steps. First, the algorithm identifies regions of abrupt spectral changes, and places primary landmarks at these locations. Next, secondary landmarks are added to ensure that a specified number of boundaries are marked within a given duration. To create the network of possible acoustic-phonetic segments, the procedure then fully connects all possible primary landmarks for every deleted secondary landmark.

An analysis of the networks proposed using this algorithm shows that on a development set, there are an average of 2.4 landmarks proposed for every transcription landmark, and 7 segments hypothesized for every transcription segment [12]. The multi-level description of the segmentation is illustrated in Figure 2-2 for the utterance "Delta three fifteen". The segmentation algorithm is described in more detail in [11].



```
1.107s to 1.216s

Segment
     iy 1.91
     ey 1.01
      n -1.59
     ow -1.78
      m -2.21
     ih -2.27
     ix -2.36
     ae -2.52
     ax -2.78
     uw -2.82
```

Utterance: dt0478f_ATIS40

Figure 2-2: Segmentation Network Proposed by SUMMIT: The waveform of the utterance is displayed at the top of the figure. Below the speech waveform is a spectrogram, and the segmentation network proposed is illustrated below the spectrogram. Finally, the phonetic labels of the utterance are given underneath the segmentation network.

2.5 Speaker Models

During training, statistical models of segment-based acoustic features are developed for each speaker. Specifically, the speaker models consist of diagonal Gaussian probability density functions (pdfs). We chose to represent the acoustic space with Gaussian distributions because features of speech data, such as cepstral coefficients, fit these bell-shaped curves well [39]. Diagonal distributions were implemented because they have few parameters to train (diagonal covariance matrices), and thus do not require much training data to accurately estimate the parameters. However, features that are correlated are not modeled well with diagonal covariance matrices.

To ensure that the features fit the diagonal models better, principal components analysis (PCA) was performed on the acoustic features before developing the models. PCA rotates a d-dimensional space to a set of orthogonal dimensions (less than or equal to the dimension d). As a result, the full covariance matrix of the original space is transformed to a diagonal matrix in the new space. In principle, PCA also allows us to reduce the dimensionality of the feature vectors. However, in our experiments, we did not reduce dimensionality with PCA since the feature search already prunes the number of features used in the system.

The Gaussian distributions that model the acoustic features for each speaker are developed using the maximum likelihood (ML) estimation procedure. The mathematical expressions for the ML estimates for the means, variances and the *a priori* class probability estimates for a particular speaker model are shown below. An example of a speaker model developed using the ML procedure is shown in Figure 2-3. Figure 2-3 illustrates a histogram of a speaker's training data and the corresponding model developed. It is apparent that a single diagonal Gaussian cannot completely model the data for each class. Mixtures of diagonal Gaussians may fit the data better. However, there are more parameters to train mixtures of Gaussians, which require more data than are available.

j = the jth broad class

n_j = the number of tokens for class j

$n =$ the total number of tokens for all classes

$x_{j,k} =$ the kth data token for class j

$\overline{\mu}_j =$ the ML estimate of the mean for class j

$\overline{\sigma}_j^2 =$ the ML estimate of the variance for class j

$P(j) =$ *a priori* probability for class j

$$\overline{\mu}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} x_{j,k}$$

$$\overline{\sigma}_j^2 = \frac{1}{n_j} \sum_{k=1}^{n_j} (x_{j,k} - \overline{\mu}_j)^2$$

$$P(j) = \frac{n_j}{n}$$

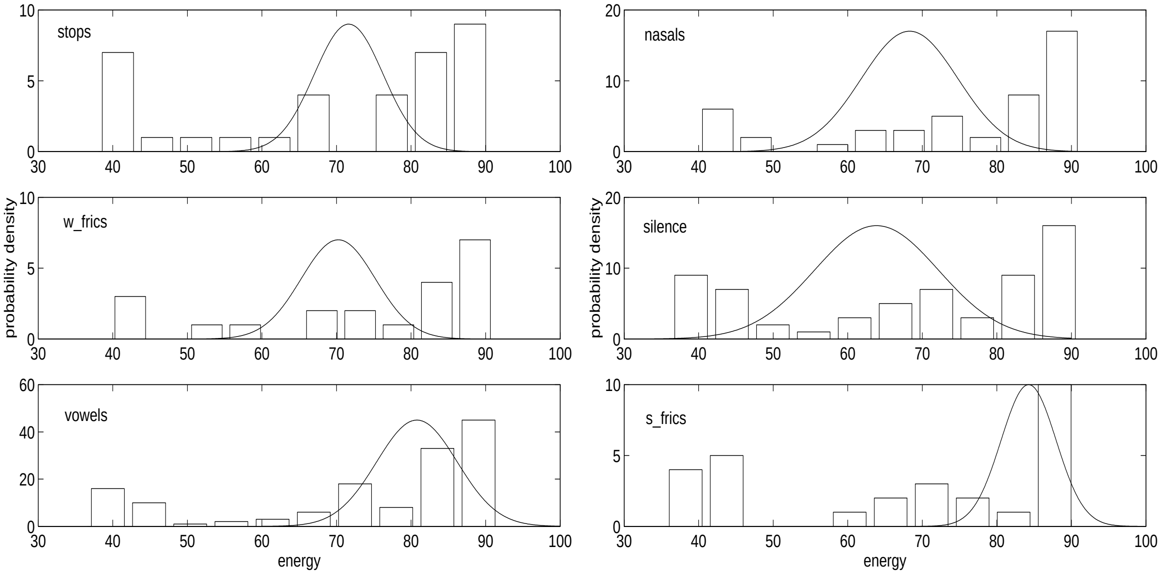


Figure 2-3: Histogram of Data and Corresponding ML Model of a Speaker

2.6 Speaker Classification

Once speaker models are developed, test utterances are scored against these models to classify speakers and make verification decisions. Below we describe the verification process and conclude this chapter with a description of how scores are computed.

2.6.1 Verification Process

To accept or reject a speaker, we compute forced alignment scores, described in 2.6.2, for the purported speaker's two test utterances. The scores are computed from all 100 speaker models. These scores are then sorted, and the speaker is accepted if the score using his/her model is in the top N scores of the 100 results.[2] The verification procedure is illustrated in Figure 2-4.

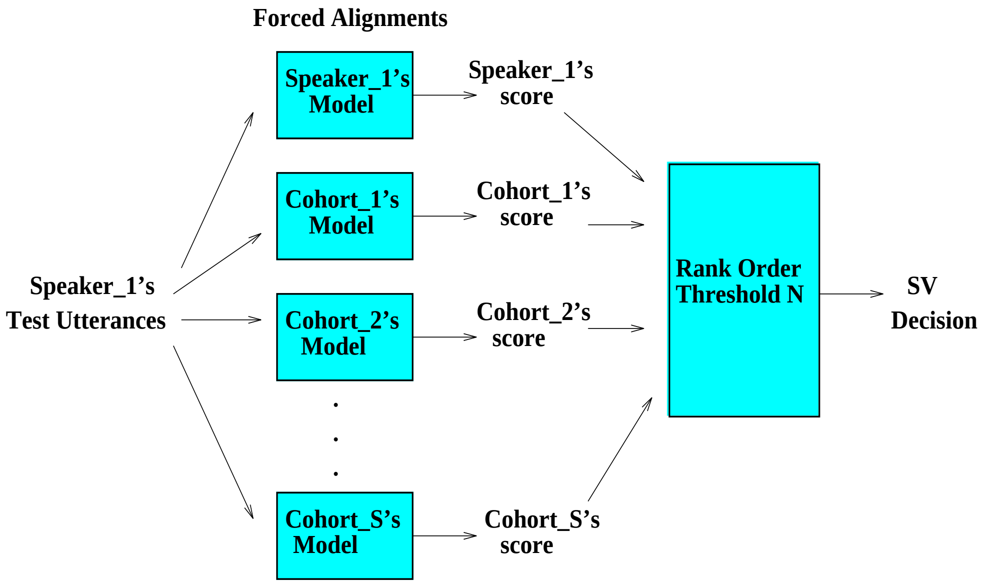


Figure 2-4: Speaker Verification Testing Procedure

False acceptance rates are obtained as they would be in a real impostor situation. If $speaker_a$ poses as $speaker_b$, $speaker_a$'s test utterances are scored by the 100 speaker models. These scores are then sorted and rank ordered. If the score using $speaker_b$'s model is in the top N scores, he/she is falsely accepted.

2.6.2 Scoring

The scores used to classify speakers correspond to likelihood probabilities accumulated from paths of speech segments. Specifically, a score reflects the probability of

[2] The rank threshold, N, is a parameter that we varied for each feature set.

observing feature vectors across the path of segments in a forced alignment, for given broad class labels.

A forced alignment is the result of a constrained search, that assigns an utterance's broad class labels to a path of segments. The search is constrained because the broad class labels are known *a priori*; thus it is not necessary to consider all possible classes for each proposed segment. During the search, each possible alignment for an utterance accumulates likelihood scores. These likelihood scores reflect the probabilities of observing feature vectors across the segments in the alignment, for the given labels. The path of segments that corresponds to the highest likelihood score, which is used to classify speakers, is chosen as the forced alignment for the test utterance.

Normally, likelihood scores are accumulated along all possible paths. However, the system implements the Viterbi algorithm to find the forced alignment without scoring all possible segmentation paths. The Viterbi algorithm is based on dynamic programming methods, and prunes the search without any loss in optimality. Details of the Viterbi algorithm can be found in [33, 5].

Chapter 3 reports the system performance, and compares our system's results to that of 3 other systems. Results using 2 computationally efficient methods are then described in Chapter 4.

Chapter 3

Performance

3.1 Overview

In this chapter, we first summarize the system performance of using only cohort sets during testing and using the entire speaker set during testing. Next, performance effects of using cohort normalization to reduce computation during testing are illustrated. Thereafter, performance discrepancies between clean and noisy environments are discussed, which is followed by an evaluation of the advantages of selecting features from a feature search. Finally, we compare our results to those of three other competitive systems that are also evaluated on the TIMIT and NTIMIT corpora.

3.2 System Performance

3.3 Performance Comparison

In order to evaluate the advantages and disadvantages of our approach to the speaker verification task, it is necessary to compare our system's performance and design to those of other systems. Often, it is difficult to compare systems unequivocally because the data used to evaluate the systems and the evaluation methods differ. In order to make somewhat meaningful comparisons, we compare our system with three other systems, described below, that also use the TIMIT and/or NTIMIT corpora.

3.3.1 HMM Approach

A state-of-the-art HMM speech recognition system, built by Lamel and Gauvain [19], was recently modified for speaker recognition. The system extracts frame-based acoustic features, which include 15 MFCCs, first derivatives of the MFCCs, energy, and the first and second derivative of energy. During training, 8 utterances (2 SA, 3 SX and 3 SI) were used to build speaker models. To develop the speaker models, a general speaker-independent model of 40 phonetic classes was trained on the 462 speakers in the TIMIT NIST-train set. This model then served as a seed model to be used to adapt, via the maximum a posteriori procedure (MAP discussed in section 5.2.5), each speaker-specific model. During adaptation, the speaker models were modified to represent 31 broad phonetic classes, rather than 40. During testing, 168 speakers from TIMIT's NIST-test and NIST-dev sets were evaluated. The 168 speaker models were combined in parallel into one large HMM, which was used to recognize the test speech of the remaining 2 SX sentences of each user. To classify speakers, the system used the phone-based acoustic likelihoods produced by the HMM on the set of 31 broad class models. The speaker model with the highest acoustic likelihood was identified as the speaker.

Lamel and Gauvain reported 98.8% speaker identification accuracy using 1 test utterance and 100% accuracy using 2 test utterances. Since we perform mini-speaker identification tests in our system, these HMM results can be compared to our results when we use all speakers during testing. Essentially, if we were to convert the HMM speaker identification system above into a speaker verification system that implements our decision algorithm, the system achieves 0% EER.

Lamel's system is evaluated on 168 test speakers (nist-test and nist-dev sets), which is 1.68 times as large as out test set. However, the sentences used during testing are two SX, whereas we test each speaker using the 2 SA utterances. Unlike the SA sentences, the SX sentences are each repeated 7 times by 7 different speakers. Thus, a test sentence may be included in the training set, suggesting that the system may have seen the same sequence of phones (spoken by different speakers) in both testing and training stages. As a result, better performance may result over a system which tests completely different orthography than the training data.

3.3.2 Neural Network Approach

Another competitive system that uses the TIMIT corpus is a neural network-based speaker identification system built by Younes Bennani [2]. The system computes 16 frame-based cepstral coefficients derived from linear prediction coefficients (LPCs). Before training, acoustic vectors computed from the 5 SX utterances for 102 speakers were grouped together into homogeneous classes, via a non-supervised k-means algorithm. [1] Each of the 102 test speakers was then assigned to the class to which the majority of the speaker's acoustic vectors belonged. During training, a typology detector and a set of expert modules (neural networks), which discriminate between speakers of the same typology, were developed. During testing, 102 speakers were evaluated using 3 SI sentences. To classify speakers, a score computed from a weighting of scores of the typology detection module with those of the expert modules is used.

Bennani's neural network system achieved a performance of 100% identification accuracy. Again, if the system implements the speaker verification decision algorithm we use, it would result in 0% EER.

3.3.3 Gaussian Mixture Models Approach

Recently, a competitive system was proposed by Doug Reynolds [34], which uses Gaussian mixture models (GMMs) to represent speakers. The system computes MFCCs from speech frames, followed by channel equalization via blind deconvolution. During training, 8 utterances were used to develop 168 speaker models (NIST-test and NIST-dev sets), each consisting of 32 mixtures of diagonal Gaussians. The models represent broad acoustic classes selected in an unsupervised fashion. However, each

[1] Exact speaker set is not reported in reference paper, but the set is not exactly the same as our test set of 100 speakers.

mixture does not necessarily model each of the 32 broad acoustic classes, since multiple Gaussians can act together to represent a single acoustic class. During testing, each speaker acts as a claimant for the remaining speakers, along with being tested as themselves. The system classifies speakers with probability likelihoods. Specifically, the test utterances are scored by the claimed speaker models in addition to background speaker models (which are previously selected). The background speaker models' scores effectively represent the probability that the test utterances are not from the claimed speaker. Thus, a speaker is accepted or rejected if the ratio of the two scores is greater than or less than a threshold, respectively. The system achieves 0.24% EER in the TIMIT domain and 7.19% EER in the NTIMIT domain.

3.4 Discussion

As illustrated above, there are high performance speaker verification systems that are evaluated on the TIMIT corpus. Although all of the systems described above use TIMIT and/or NTIMIT, they evaluate the systems on either a different set of sentences, or a different set of speakers than our sets. The different sets makes direct comparisons between our system and the three systems described above difficult. However, we may still make some meaningful comparisons concerning system design and computation during training and testing. Table 3-1 summarizes the design and performance of our system and those of the three systems discussed above.

Like the HMM and neural network system discussed above, our system achieves ideal performance (0% EER) in the clean domain. However, we designed our system to be computationally efficient, and reduced computation in a variety of ways. First, we used only 17 acoustic features, as opposed to 32 in the NN system and 30 in the GMM system (TIMIT), to represent the speech signal. Second, we developed speaker models of 6 broad phonetic classes, as opposed to 31 for the HMM system and ≤ 32 for the GMM system. Third, each of the 6 broad classes is represented by a single diagonal Gaussian distribution, as opposed to mixtures of Gaussians or the nonlinear distributions that neural networks typically produce. The two latter models have

Parameter	**HMM**	**Neural Network**	**GMM**	**SUMMIT**
Speaker Models	Mixtures of Gaussians	Neural Network	Mixtures of Gaussians	Diagonal Gaussians
Classifier	HMM scores	neural network output	probability likelihood scores	forced-alignment scores
# of System Users	168	102	168	100
Type of Measurements	frame-based	frame-based	frame-based	segment-based
Model Modifications	MAP adaptation	-	-	-
# of Broad Classes	31	-	≤ 32	6
Feature Vector Size	32	16	30 (TIMIT), 19 (NTIMIT)	6
Selected Features	MFCCs, delta MFCCs, Energy, and delta Energy	Cepstral Coefficients	MFCCs (no c[0])	MFCCs
TIMIT Performance	0% EER	0% EER	0.24% EER	0% EER
NTIMIT Performance	- EER	-	7.19%	8.36% EER

Table 3-1: Comparing SUMMIT to an HMM, Neural Network, and GMM System

System	**# of Params**
HMM	$\leq$4312685
GMM	$\leq$327936
SUMMIT	$\leq$35280

Table 3-2: Number of Training Parameters

more parameters to estimate, and hence require more computation during training. Finally, we reduce computation during testing by using only a set of speaker models (for NTIMIT) similar to the purported speaker's model, as opposed to using all the speaker models in the system, like the HMM and NN systems.

Computation, in terms of the number of training parameters[2], is approximated in Table 3-2. As illustrated in Table 3-2, the HMM and GMM systems estimate (with the same amount of training data) on the order of 10^6 and 10^5 parameters, respectively. While, we estimate on the order of 10^4 parameters. Not enough information is given for the neural network system to approximate the number of training parameters reliably.

[2] The number of parameters were computed assuming 168 system users, and the estimates are approximating an upper bound for every system.

Chapter 4

Computational Issues

4.1 Overview

In the previous chapter, we showed that a segment-based speaker verification system is viable and efficient in that it requires few parameters to estimate during training. We now explore two methods, listed below, to further reduce computation during both training and testing.

1. **Feature Search**

2. **Cohort Normalization**

This chapter first describes the motivation behind conducting a feature search. The details of the feature search algorithm implemented are then discussed. Next, our technique of cohort normalization is described. Note that we implemented both methods simultaneously, *i.e.*, we used cohort normalization in all of the feature searches. Finally, we conclude with all of the feature search results, and determine whether the 2 methods are viable.

4.2 Feature Search

Each of the segments proposed by the segmentation algorithm is described by a set of acoustic features. The set of 17 measurements, discussed in 2.3, represents a pool of possible features to characterize segments. We may not want to use all 17 measurements in the system for the following reasons. First, some features may be useful in discriminating speakers well, while others may not. Second, some of the measurements may be correlated or essentially carry the same information. In addition, training models with high dimensionality may be a problem since not much data is available per speaker. Finally, computation increases as the number of features increases, which may become expensive if all 17 measurements are used in the system.

To find a (sub)-optimal subset of the 17 features, we conducted a greedy search, because an exhaustive search is computationally prohibitive. A greedy search may not always produce an optimal solution. However, it significantly prunes large search spaces without much loss in optimality [5]. At every decision point in a greedy algorithm, the best choice, based on some optimality criterion, is selected. Our search criterion is the speaker verification performance of each proposed feature set. Performance is measured in terms of the equal error rate (EER) described in detail in section 4.2.1. Below, we describe the greedy feature search, which is also illustrated in Figure 4-1 for an initial pool of 5 features.

The search algorithm begins by obtaining EERs for the set of speakers, using each of the 17 features. Thus we obtain 17 performance results corresponding to each measurement. The feature that results in the smallest distance measure (best performance) is chosen as the best 1-dimensional measurement. Next, the best 1-dimensional feature is combined with each of the remaining measurements. Two-dimensional feature sets are grouped in this fashion, and are each used to test the set of speakers. The best 2-dimensional feature vector, in terms of speaker verification performance, is then used for the next stage of the search. The search continues to accumulate dimensions in the feature set until there is no longer significant improvement in speaker verification performance, or if performance actually degrades as more features are added.

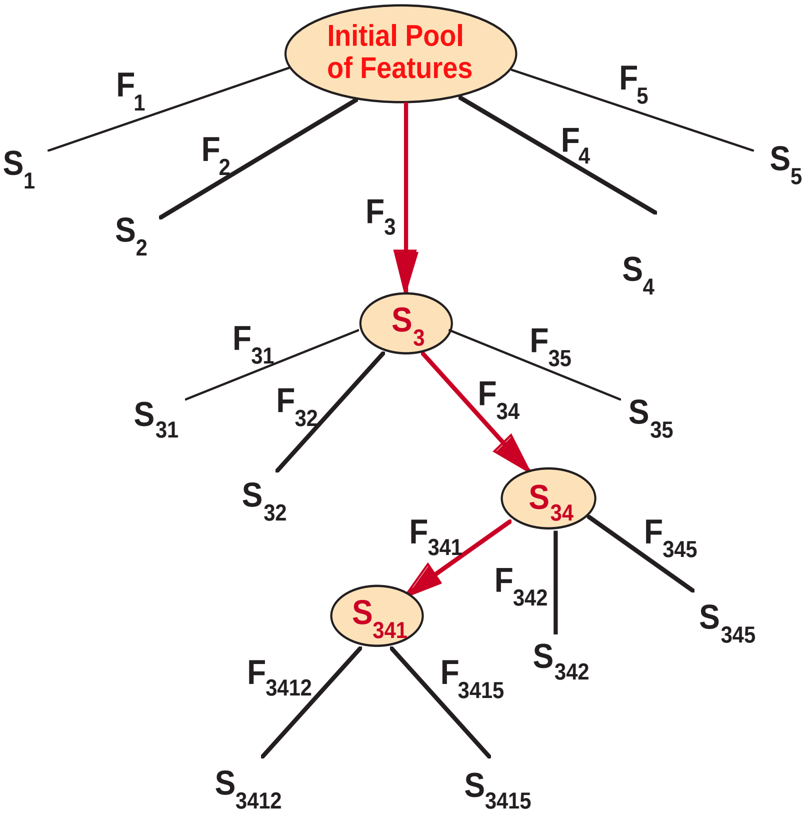


Figure 4-1: Illustrative Example of our Greedy Feature Search: F_{ijk} is the set of features i, j, and k. S_{ijk} is the corresponding verification score in terms of a distance measure. First, each feature is individually tested, and feature #3 results in the best speaker verification performance. Next, feature #3 is combined with each of the 4 remaining features to form 2-dimensional sets. Features #3,4 then result in the best performance (which is significantly better than the 1-dimensional set). This 2-dimensional set is then combined with the 3 remaining measurements to form 3-dimensional sets. Finally, features #3,4,1 is the optimal set, because performances of the two 4-dimensional sets fail to significantly improve over the 3-dimensional set.

4.2.1 Performance Measure

The performance of a speaker verification system is typically measured in terms of two types of errors: false rejections of true users (FR) and false acceptances of impostors (FA). These errors often illustrated with conventional receiver operating characteristic (ROC) curves, which plot the rates of FR versus the rates of FA for some varying parameter.

A popular single number measure of performance is the equal error rate (EER), which is the rate at which the two errors (FR and FA) are equal. EER is thus the intersection between the ROC curve and the line FR=FA. Many researchers design speaker verification systems to minimize the EER. Therefore, we use EER as our search criterion to facilitate comparing our work with that of others. However, minimizing this measure does not allow for different costs to be associated with FA and FR. For high security applications such as bank-by-phone authorizations, minimizing false acceptances of impostors is the first priority. Rejecting a true user may annoy the user. However, accepting an impostor may be costly to the customer. Figure 4-2 uses the ROC curve for MFCC1 as an example to illustrate the EER.

4.3 Cohort Normalization

During testing, it is ideal to compare the utterances to all speaker models in the system, and accept the purported speaker if his/her model scores best against the test data. However, computation becomes more expensive as speakers are added to the system. Since speaker verification is simply a binary decision of accepting or rejecting a purported speaker, the task should be independent of the user population size.

To keep our system independent of the number of users and computationally efficient, we implemented a technique called cohort normalization. For each speaker, we pre-detected a small set of speakers, called a cohort set, who are acoustically similar to the purported speaker. During testing, we only test the speakers in the cohort set for the purported speaker. Speakers outside the cohort set are considered

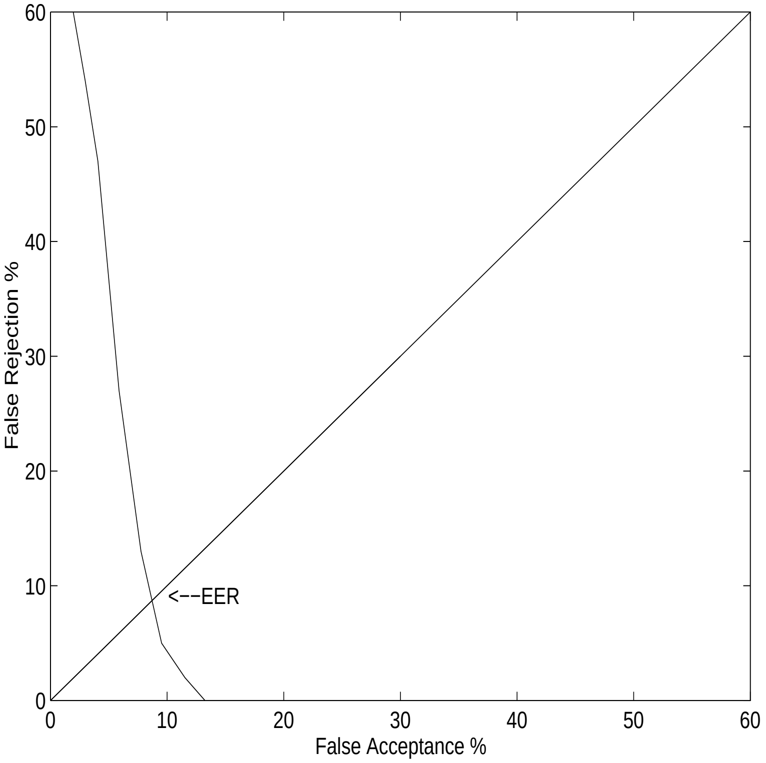


Figure 4-2: ROC Curve for MFCC1 and Corresponding EER

outliers that have low probabilities of scoring well against the purported speaker's test data. If this is the case, the ROC curves corresponding to performance using all speakers during testing can be obtained from the ROC curves using only cohort sets during testing, via normalization. The normalization divides the number of false acceptances obtained for a feature set, using for each speaker only the speakers in his/her cohort set as impostors, by the number of possible false acceptances when all the remaining speakers pose as impostors for each speaker (100 speakers x 99 impostors in the case of a population size of 100).

For each feature set, we found 14 nearest neighbors (cohorts) for each speaker using the Mahalanobis distance metric [40]. Specifically, $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$, σ_1^2 and σ_2^2, are d-dimensional mean vectors and dxd-dimensional covariance matrices for two speaker models, respectively. The Mahalanobis distance squared, D^2, between the two speakers is then

$$D^2(1,2) = \sum_{i=1}^{d} \frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_i^2}$$

where

$$\sigma_i^2 = \frac{n_1 \sigma_{1i}^2}{n_1 + n_2} + \frac{n_2 \sigma_{2i}^2}{n_1 + n_2}$$

and n_1 and n_2 are the number of data vectors for speaker one and speaker two, respectively.

Once the speaker models were developed, this metric was applied to every possible pair of speakers. The distances were then sorted for each speaker, and the cohorts were chosen to be the 14 closest neighbors to each speaker.

An example of a female speaker and her cohorts' models for the 6 broad classes, using F0 as a feature, is shown in Figure 4-3. Distributions of 4 cohorts are plotted along with models of 2 outliers (1 female and 1 male) of the cohort set for that speaker. As expected, the cohort models are very similar to the speaker's model, while there is more disparity between the true speaker's models and the models of the outliers.

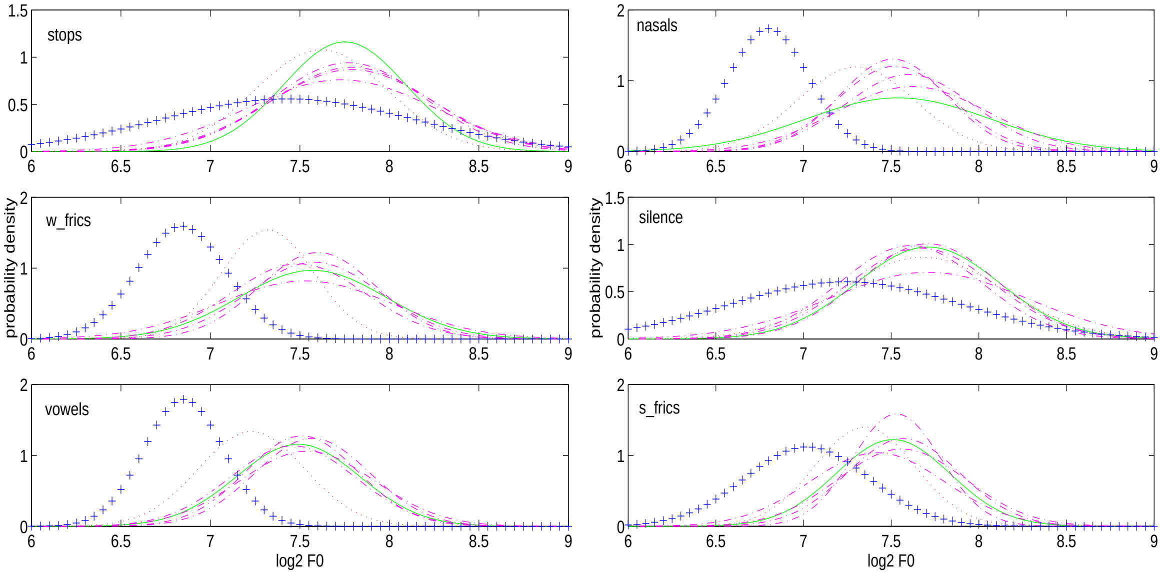


Figure 4-3: Speaker F0 Models and Cohorts: '-' models represent the true female speaker and '-.' models are 5 of her cohorts. The '.' models represent a female outlier and the '+' models represent a male outlier of the true speaker's cohort set.

During verification, to accept or reject a speaker, the scores are computed from the purported speaker's model and his/her 14 cohort models. These scores are then sorted, and the speaker is accepted if the score using his/her model is in the top N scores of the 15 results.[1] The verification procedure is illustrated in Figure 4-4 for a cohort size equal to S.

False acceptance rates are obtained as they would be in a real impostor situation. If $speaker_a$ poses as $speaker_b$, $speaker_a$'s test utterances are scored by $speaker_b$'s model and $speaker_b$'s cohort models. These scores are then sorted and rank ordered. If the score using $speaker_b$'s model is in the top N scores, he/she is falsely accepted.

4.4 Results

4.4.1 TIMIT

In this section, we present the results of two greedy feature searches conducted on 100 development and 100 test speakers from the TIMIT corpus. Both searches were

[1] The rank threshold, N, is a parameter that we varied for each feature set.

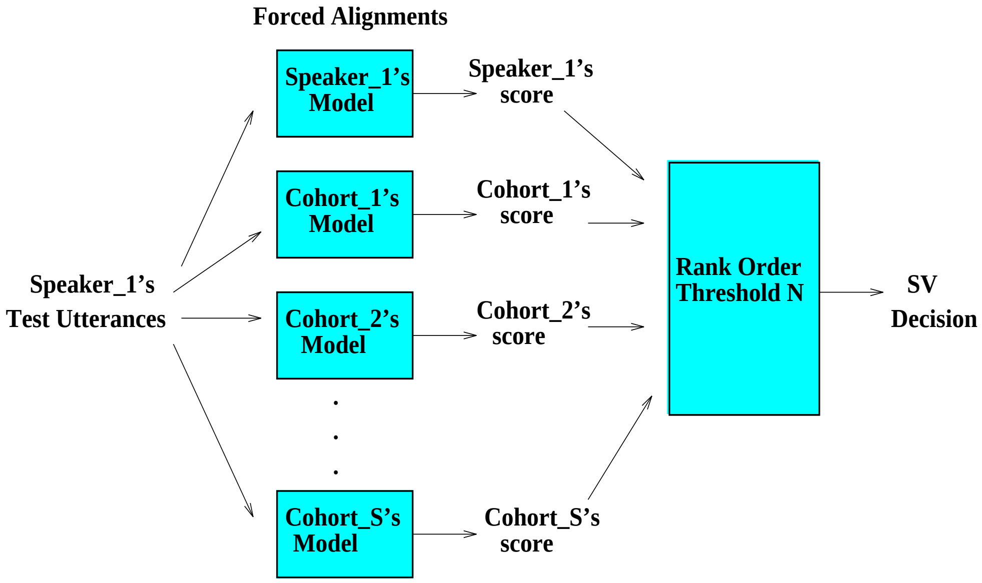


Figure 4-4: Speaker Verification Testing Procedure

performed using a cohort set size of 14, and the EER results reported are measured from the normalized ROC curves. Thus, the rates given are estimates of performance when all speakers are using during testing.

Results Using Development Speakers

The first stage of the search evaluates the speaker verification performance of individual features. Performances of the one-dimensional measurements are given in Table 4-1, and are illustrated in Figure 4-5. Figure 4-5 shows that the top 14 features, particular MFCCs, F0, and energy, result in similar EERs. We disregarded the three lowest-ranked features for subsequent search stages, because they resulted in significantly worse performance than the top 14 features. We realized that such pruning will result in a search that is not greedy in the strictest sense of the word.

Past observations have shown that MFCCs and prosodic features are useful for speaker verification [19, 42, 17]. In our search, we found that two of the three prosodic measurements investigated performed well individually. Specifically, energy ranked seventh in the set of 17 features, and F0 ranked third. However, duration ranked last

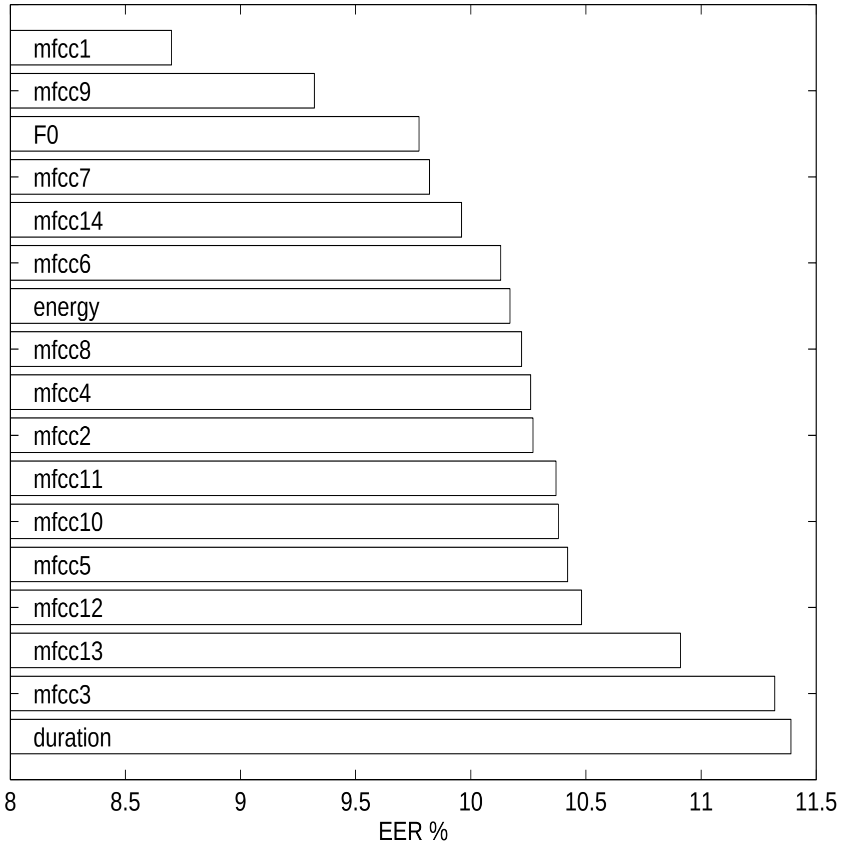


Figure 4-5: EERs for One-Dimensional Feature Sets (TIMIT DEV)

FEATURE SET	**EER %**
F0	9.78
Energy	10.17
Duration	11.39
MFCC1	8.71
MFCC2	10.28
MFCC3	11.32
MFCC4	10.26
MFCC5	10.42
MFCC6	10.13
MFCC7	9.82
MFCC8	10.22
MFCC9	9.32
MFCC10	10.38
MFCC11	10.37
MFCC12	10.48
MFCC13	10.91
MFCC14	9.96

Table 4-1: One-Dimensional Feature Set Results (TIMIT DEV)

in the first stage of the feature search. Perhaps duration performed poorly because of the manner in which the broad classes were formed. Some of the 6 broad classes selected, such as stops and weak fricatives, consist of both voiced and unvoiced phones, which are mainly distinguished by duration. Consequently, the variances of duration for these classes are large for all distributions, and the speaker models are often indistinguishable if the means do not differ by much. As a result, speaker verification performance is poor.

Figure 4-6 illustrates these large variances (on the order of 10^5) of 4 speakers' duration models of stop consonants and weak fricatives. The 4 speakers are within a cohort set. Thus, during testing, these speakers are compared to each other and the remaining members of the cohort set. As shown in Figure 4-6, it is difficult to reliably distinguish among the 4 distributions. In fact, we computed the average Mahalanobis distance between the 4 cohort models for energy, which resulted in a significantly smaller EER, and duration. These distances are shown in Table 4-2, which illustrates that the duration models are more similar (smaller distance) to each other than the energy models, discussed below.

Perhaps duration as a measurement could have performed better if our broad classes were selected knowing *a priori* that duration was to be a measured feature. An appropriate selection of broad classes would then be voiced stops, unvoiced stops, voiced fricatives, unvoiced fricatives, long vowels, short vowels etc. Essentially, the classes would have similar duration characteristics.

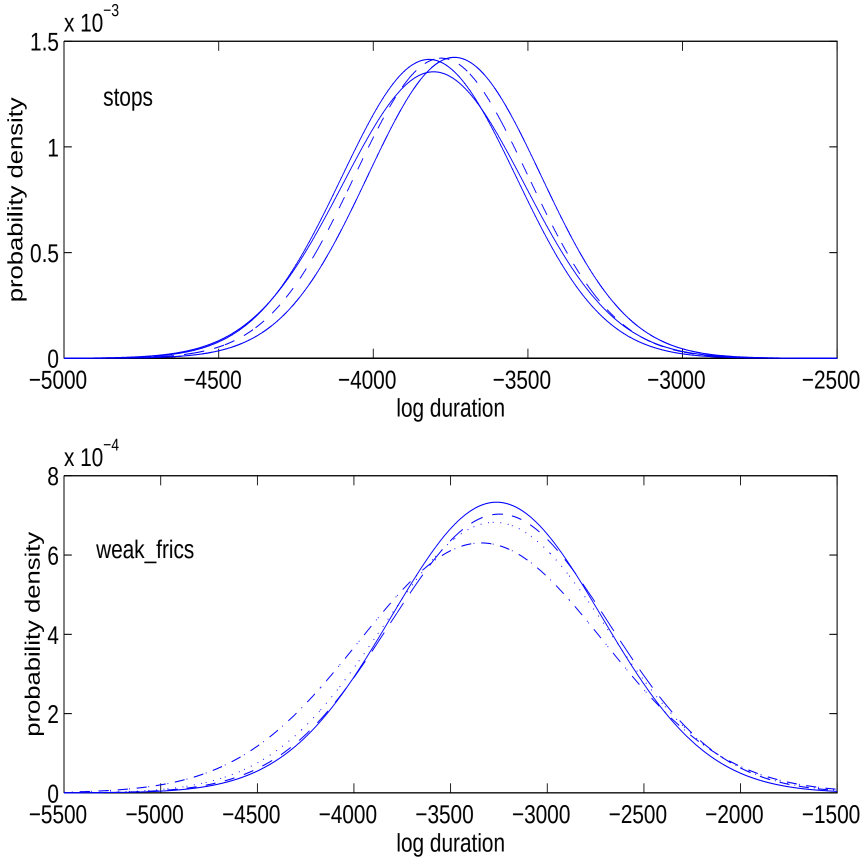


Figure 4-6: Duration models of 4 Speakers

Energy, on the other hand, performed significantly better in the first stage of the search, suggesting that the energy characteristics within classes are similar. Thus, we expect the opposite trends in the statistics of energy. For example, the energy of strong fricatives is much larger than the energy of weak fricatives. Therefore, the strong fricatives' and weak fricatives' models for energy have smaller variances than the duration models for these same two classes. When the variances are small for every speaker model, it is possible to distinguish between speakers with different means.

Figure 4-7, is a similar plot of 4 speakers' (within a cohort set) energy models for weak fricatives and stop consonants. Figure 4-7 illustrates the larger differences between the 4 speaker models for energy than the models for duration, suggesting that distinguishing between speakers is easier using energy as a feature. In fact, as shown in Table 4-2, the average Mahalanobis distance for the energy models is approximately twice that of the duration models, implying that the energy speaker models are significantly more different than the duration speaker models within cohort sets.

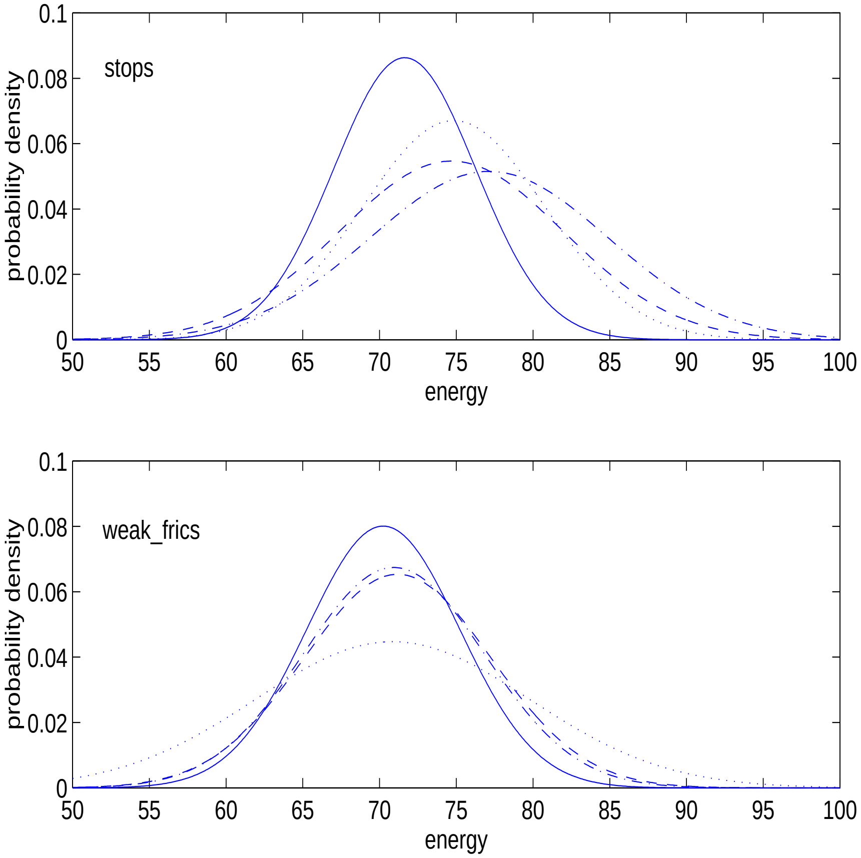


Figure 4-7: Energy models of 4 Speakers

During the second stage of the search, we explored pairs of features, combining

Feature	Distance
Duration	0.479
Energy	0.926

Table 4-2: Average Mahalanobis Distances for 4 Speakers' Duration and Energy Models

MFCC1 with each of the remaining 13 features. The 2-dimensional results are shown in Figure 4-8, which illustrates that MFCC1 combined with MFCC9 result in the best speaker verification performance. Furthermore, there is a noticeable improvement in performance by the addition of another measurement to the feature set, since the EERs are significantly smaller for the 2-dimensional sets than for individual measurements. The performance improvement suggests that the additional features carry further speaker-specific information. Also, the dimension of the feature set is small enough that model parameters can be sufficiently estimated from the 8 training utterances available per speaker.

During the third search stage, MFCC1 and MFCC9 were combined with the 12 remaining measurements. The 3-dimensional results are shown in Figure 4-9, which illustrates that the best 3-dimensional feature set is MFCC1 and MFCC9 combined with MFCC12. We continued our feature search, since performance continued to significantly improve, and accumulated dimensions to the feature vector in the manner illustrated above. The results of the feature sets for all stages of this search are given in Appendix C, thus we eliminate discussion of stages 4-9.

The 9-dimensional feature set that resulted in the best speaker verification performance included MFCC1, MFCC9, MFCC12, MFCC6, MFCC2, MFCC11, energy, MFCC5, and MFCC14. These measurements were then combined with each of the 5 remaining features in the 10th search stage. The 10-dimensional results are shown in Figure 4-10, which illustrates that the best 10-dimensional feature set consists of MFCC1, MFCC9, MFCC12, MFCC6, MFCC2, MFCC11, Energy, MFCC5, MFCC14, and MFCC4.

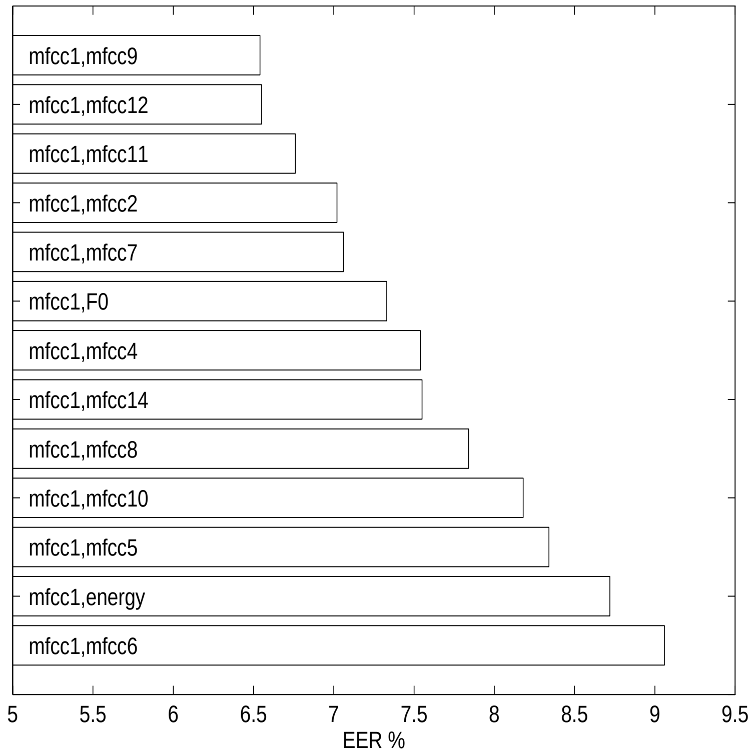


Figure 4-8: EERs for Two-Dimensional Feature Sets (TIMIT DEV)

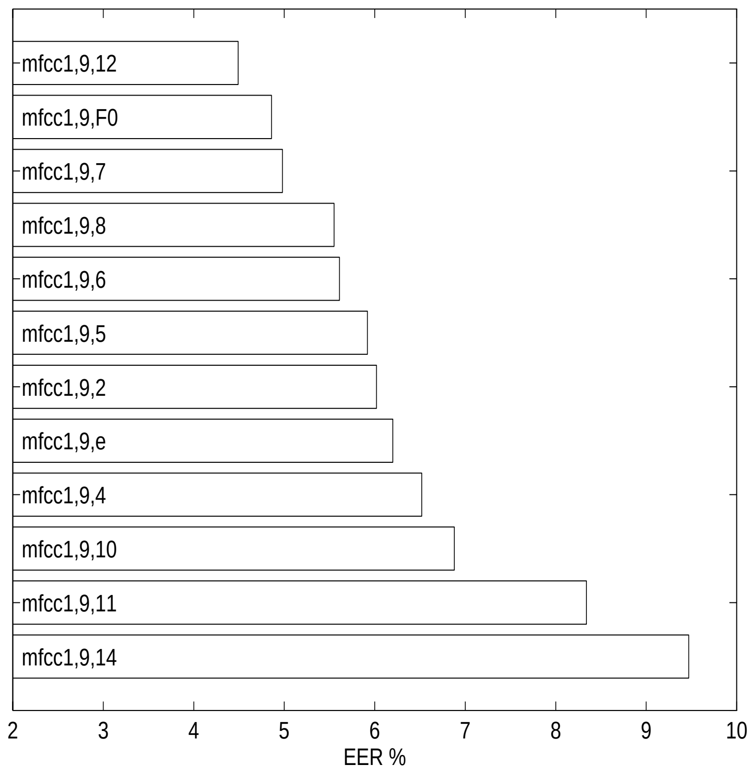


Figure 4-9: EERs for Three-Dimensional Feature Sets (TIMIT DEV)

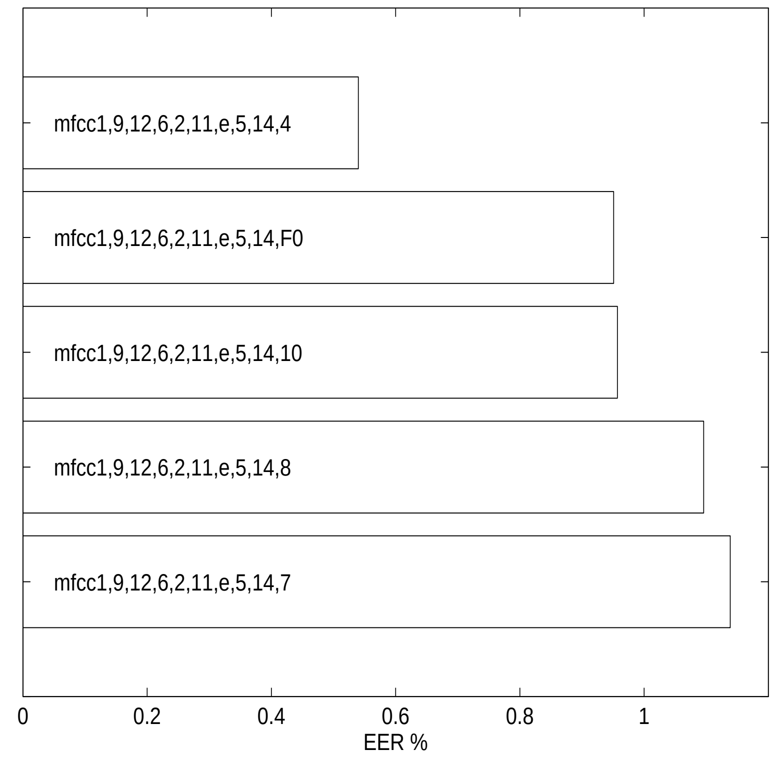


Figure 4-10: EERs for Ten-Dimensional Feature Sets (TIMIT DEV)

We conducted the remaining stages of the search to see if additional features improved performance over the (sub)-optimal set of 10 features. The best results for each search stage along with results for the 11-17 dimensional sets are shown in Figure 4-11. Figure 4-11 illustrates that all of the feature sets 11 dimensions and higher performed worse (according to EERs) than the best 10-dimensional feature set.

Overall, speaker verification performance initially improves as more measurements are added to the feature set, because the additional features contribute further speaker-specific information. Also, there are sufficient amounts of training data to accurately estimate the model parameters. However, adding features eventually degrades performance, presumably because not enough training data are available to accurately estimate the model parameters.

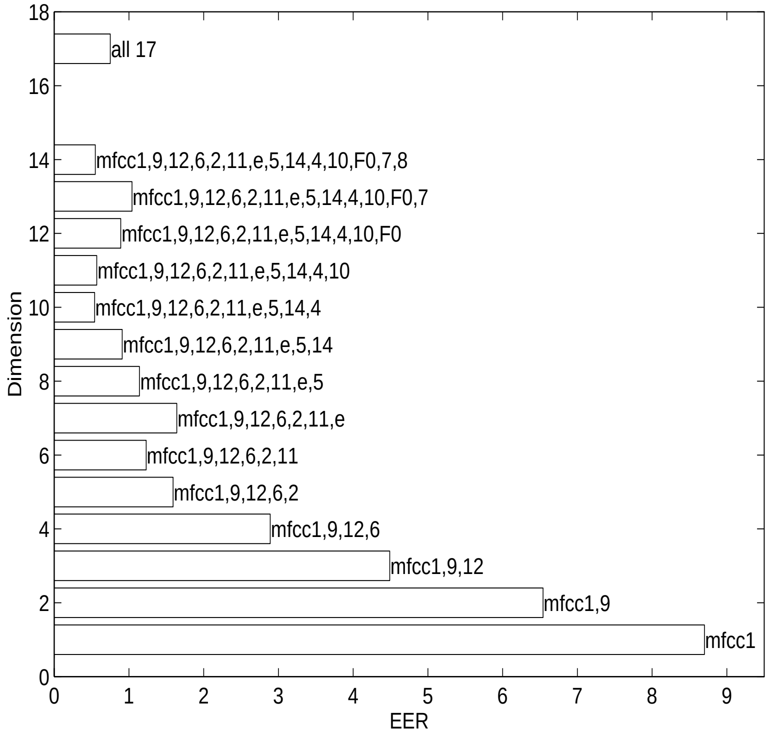


Figure 4-11: EERs for Best Feature Sets of Each Search Stage (TIMIT DEV)

Since performance degraded after the tenth search stage, the best 10-dimensional set, listed below, was considered the (sub)-optimal subset of the 17 collected features.

1. MFCC1
2. MFCC9
3. MFCC12
4. MFCC6
5. MFCC2
6. MFCC11
7. Energy
8. MFCC5
9. MFCC14
10. MFCC4

Results Using Test Speakers

The optimal 10-dimensional feature set found using 100 development speakers should be independent of the user population. To ensure that the feature search does not produce significantly different results using another set of speakers, we conducted an identical search using 100 test users. This speaker set does not contain any speakers in the original development set. However, the test set has the same ratio of males to females (2 to 1) as the development set.

As before, we began the search by testing individual features from the initial pool of 17 measurements. Table 4-3 and Figure 4-12 illustrate the results of the first search stage. The results are similar to those obtained previously in the first stage. Specifically, 13 of the top 14 features from this search are included in the top 14 features from the first search. To replicate our search experiments, we eliminated the 3 worst features, according to EER, and kept the top 14 measurements for the remaining stages of the search. The top ranking features still consisted of energy, F0, and the particular MFCCs; and as before, duration performed the worst.

FEATURE SET	**DEV SET**	**TEST SET**	**DIFF**
	EER %	EER %	(MAG)
F0	9.78	10.01	0.23
Energy	10.17	10.38	0.21
Duration	11.39	11.85	0.46
MFCC1	8.71	9.54	0.83
MFCC2	10.28	9.58	0.70
MFCC3	11.32	11.82	0.50
MFCC4	10.26	9.33	0.93
MFCC5	10.42	9.98	0.44
MFCC6	10.13	9.55	0.58
MFCC7	9.82	9.86	0.04
MFCC8	10.22	10.73	0.51
MFCC9	9.32	9.78	0.46
MFCC10	10.38	9.63	0.75
MFCC11	10.37	10.16	0.21
MFCC12	10.48	9.26	1.22
MFCC13	10.91	9.64	1.27
MFCC14	9.96	9.36	0.60

Table 4-3: One Dimensional Feature Set Results (TIMIT TEST)

Since MFCC12 is the best 1-dimensional feature, it was combined with the remaining 13 features. These 2-dimensional sets were evaluated, and performance results are illustrated in Figure 4-13. Figure 4-13 shows that the best pair of features is MFCC12 and MFCC1.

Continuing the search, MFCC12 and MFCC1 were then combined with the remaining measurements to form 3-dimensional feature sets. Figure 4-14 illustrates that the best 3 features are MFCC12, MFCC1, and MFCC9, which is the same best 3-dimensional set found from the search on the development set.

We continued to accumulate dimensions to the feature set as performance continued to increase. However, we eliminate discussion of stages 4-9. The results for each stage of this search are shown in Appendix D. The 10-dimensional feature sets are as shown in Figure 4-15, which illustrates that the optimal feature set consisted

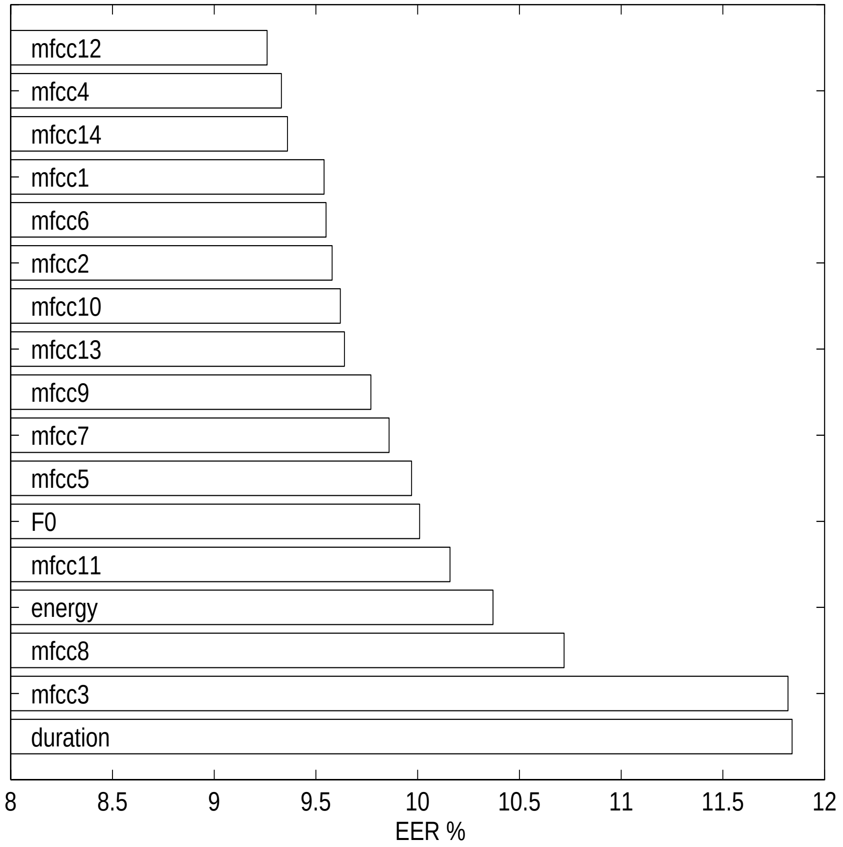


Figure 4-12: EERs for One-Dimensional Features (TIMIT TEST)

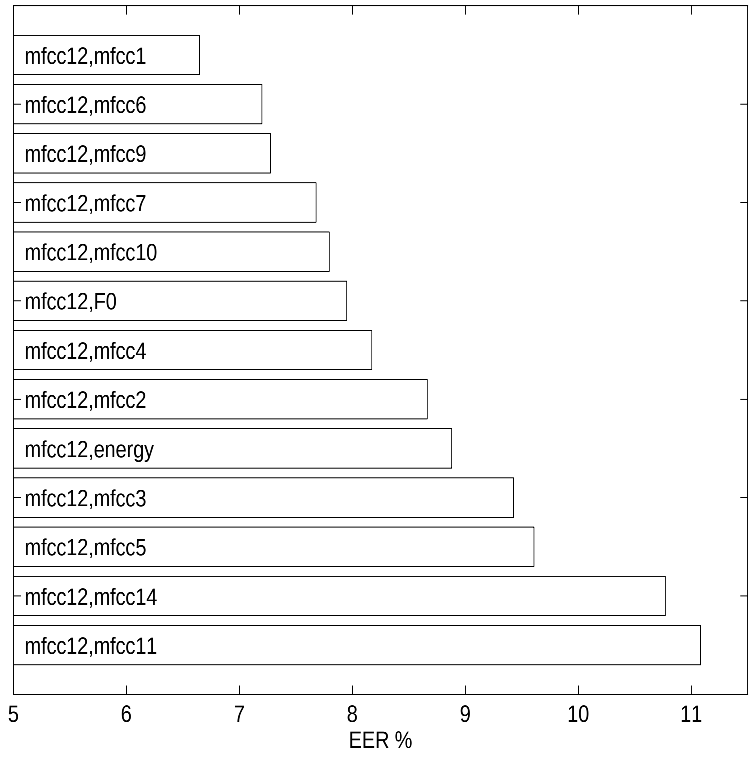


Figure 4-13: EERs for Two-Dimensional Features (TIMIT TEST)

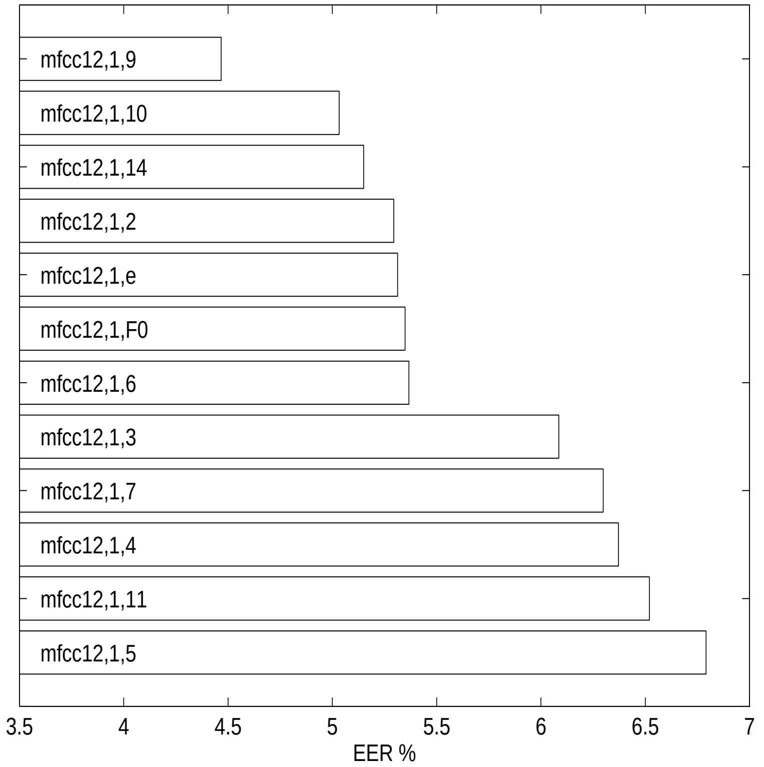


Figure 4-14: EERs for Three-Dimensional Features (TIMIT TEST)

of MFCC12, MFCC1, MFCC9, MFCC14, MFCC6, MFCC11, MFCC3, MFCC2, MFCC5, and MFCC10. Eight of the ten features are included in the optimal feature set found using the development speakers, which suggests that feature selection is essentially independent of speaker population.

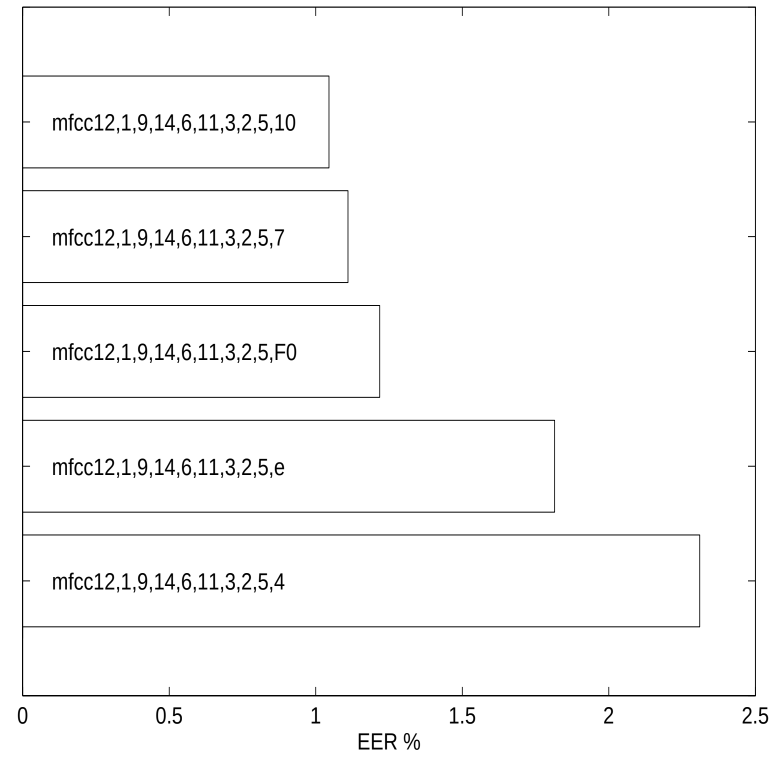


Figure 4-15: EERs for Ten-Dimensional Features (TIMIT TEST)

As before, EERs initially decreased as feature set size increased presumably because the additional features contribute further speaker-specific information. However, adding features to the 10-dimensional set failed to significantly improve performance, presumably because not enough training data are available to accurately estimate the model parameters. Figure 4-16 illustrates this trend by plotting the EERs for the best feature sets of each search stage.

Unlike the search on the development speaker set, the estimated EER using all 17 features is smaller than the EER for the best 10- dimensional set of this search. In section 4.5, we compute the actual system performance for the optimal 10-dimensional

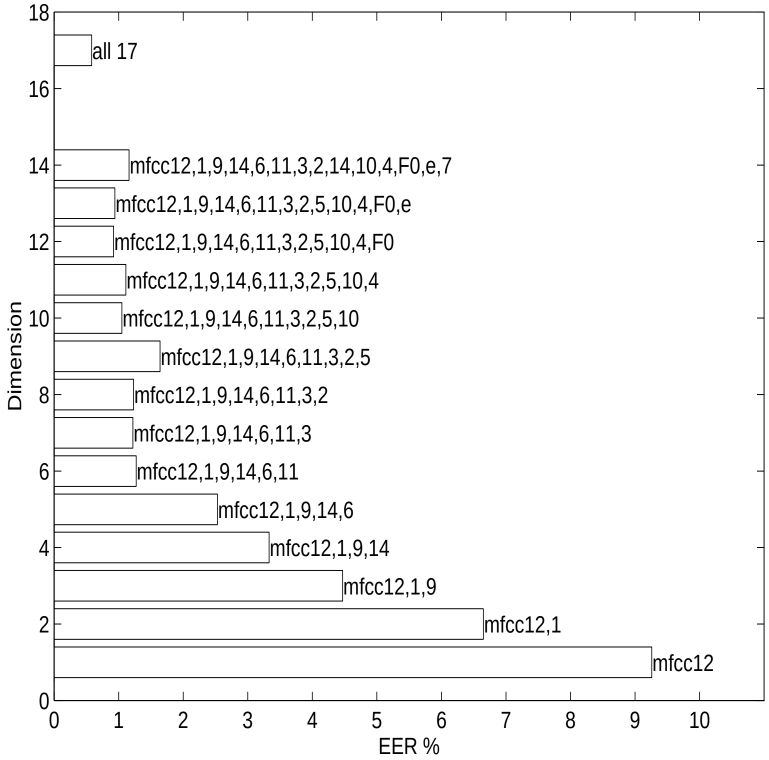


Figure 4-16: EERs for Best Features of Each Search Stage (TIMIT TEST)

and the 17-dimensional feature sets by using all development and test speakers during testing, as opposed to using only cohorts during testing. Depending on the trade-offs between performance and computation, we will determine which feature set to use in the system. Computational issues are addressed in section **??**.

The best 10-dimensional set found using the test speakers is listed below and consists of 8 of the top 10 features found using the development set.

1. MFCC12
2. MFCC1
3. MFCC9
4. MFCC14
5. MFCC6
6. MFCC11
7. MFCC3
8. MFCC2
9. MFCC5
10. MFCC10

4.4.2 NTIMIT

In this section, we present the results of a greedy feature search using the development speakers in an approximated telephone domain. The experiments are evaluated on the NTIMIT corpus, which simulates the telephone environment by transmitting TIMIT data through a telephone channel. Since our feature selections using TIMIT were essentially independent of speaker population (both searches performed the best using either 10 selected features or all 17 features), we felt that the features selected using the development set would consist of the (sub)-optimal feature set for the noisy domain. As before, the search was conducted using a cohort set size of 14, and the EERs reported are measured from the normalized ROC curves.

Results Using Development Speakers

The first stage of the search evaluates the speaker verification performance of individual features. Performances of the one-dimensional measurements are given in Table 4-4, and are illustrated in Figure 4-17. Figure 4-17 shows that the top features in the noisy domain consist of F0, particular MFCCs, energy, and duration. In general, the EER's are considerably higher than those in the TIMIT domain.

FEATURE SET	**NTIMIT**	**TIMIT**	**DIFF**
	EER %	EER %	(MAG)
F0	9.86	9.78	0.08
Energy	11.41	10.17	1.24
Duration	11.52	11.39	0.13
MFCC1	11.90	8.71	3.19
MFCC2	11.51	10.28	1.23
MFCC3	11.38	11.32	0.06
MFCC4	11.77	10.26	1.51
MFCC5	11.48	10.42	1.06
MFCC6	11.63	10.13	1.50
MFCC7	11.31	9.82	1.49
MFCC8	11.86	10.22	1.64
MFCC9	11.69	9.32	2.37
MFCC10	11.82	10.38	1.44
MFCC11	11.93	10.37	1.56
MFCC12	10.94	10.48	0.46
MFCC13	11.01	10.91	0.10
MFCC14	11.69	9.96	1.73

Table 4-4: One-Dimensional Feature Set Results (NTIMIT DEV)

Past observations have shown that prosodic features are robust and useful for speaker verification [42, 17]. In our search, we found that all prosodic measurements investigated performed well in the new domain. Specifically, F0 ranked first, energy ranked sixth, and duration ranked ninth in the set of 17 features.

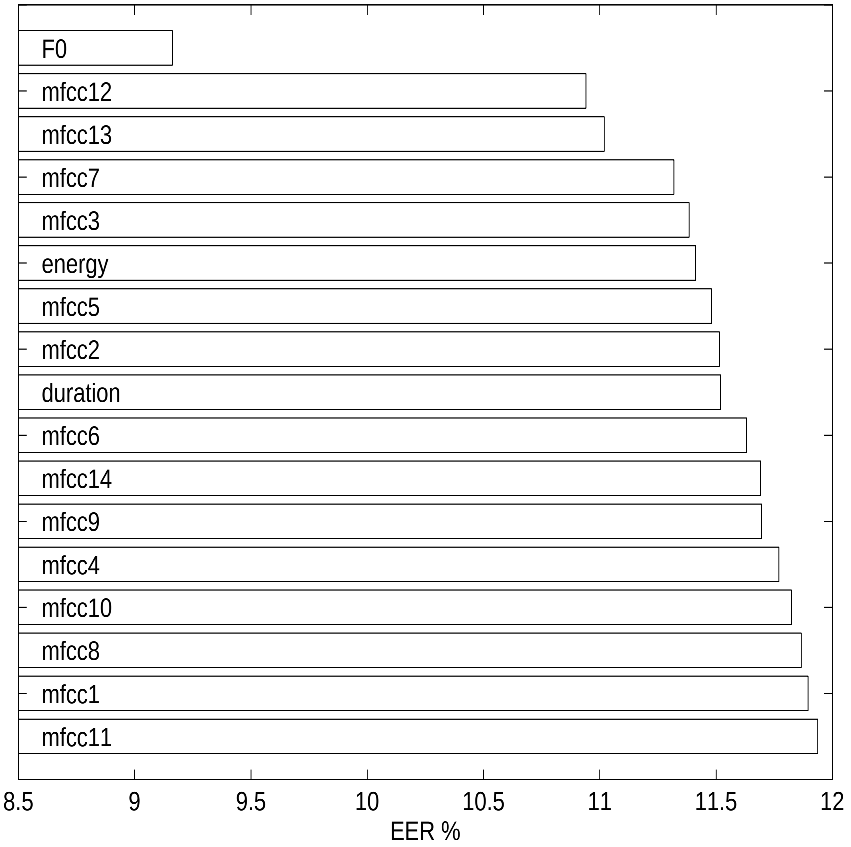


Figure 4-17: EERs for One-Dimensional Feature Sets (NTIMIT DEV)

Since F0 is the best 1-dimensional feature, it was combined with the remaining 16 features. The 2-dimensional sets were evaluated, and performance results are illustrated in Figure 4-18. Figure 4-18 shows that the best pair of features is F0 and MFCC13.

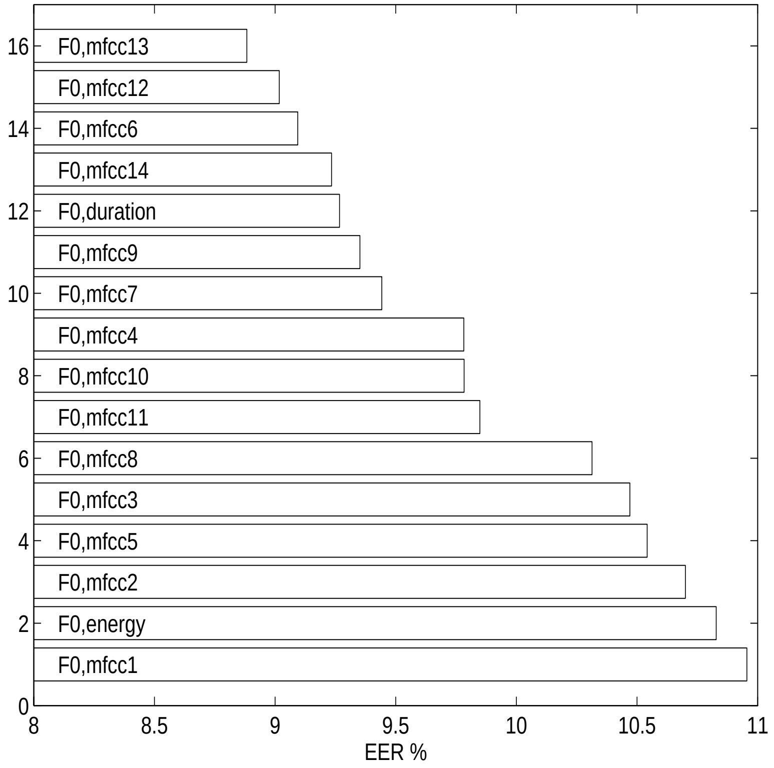


Figure 4-18: EERs for Two-Dimensional Features (NTIMIT DEV)

Continuing the search, F0 and MFCC13 were then combined with the remaining features to form 3-dimensional feature sets. Figure 4-19 shows that the best 3 features are F0, MFCC13, and duration. Unlike in the TIMIT domain, performance does not significantly improve as features are added in the early stages of the search.

However, we continued to accumulate dimensions to the feature sets. The results for all stages are shown in Appendix E, thus we eliminate discussion of stages 4-5. Figure 4-20 illustrates the results of the 6th search search stage and shows that the best 6-dimensional feature set consists of F0, MFCC13, duration, MFCC7, MFCC8, and MFCC6.

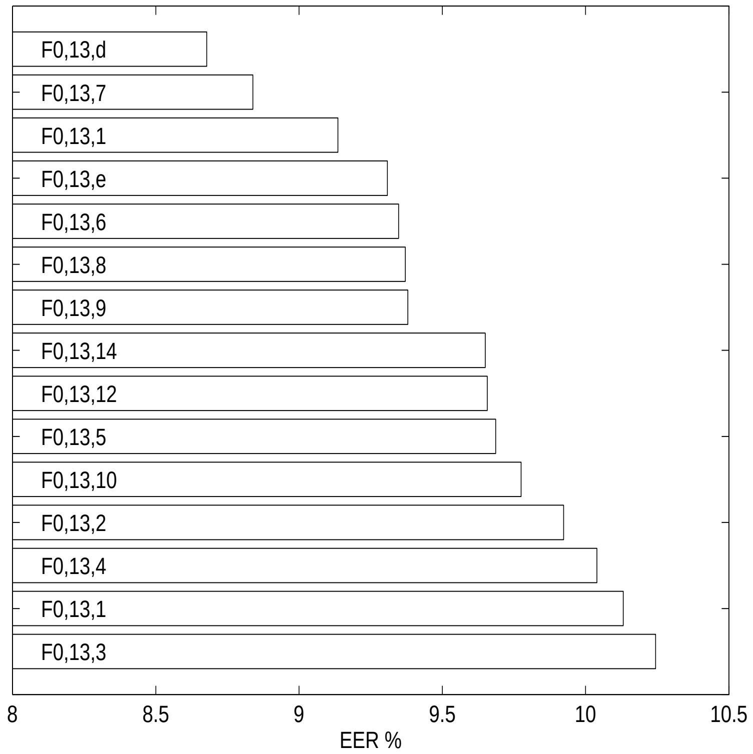


Figure 4-19: EERs for Three-Dimensional Features (NTIMIT DEV)

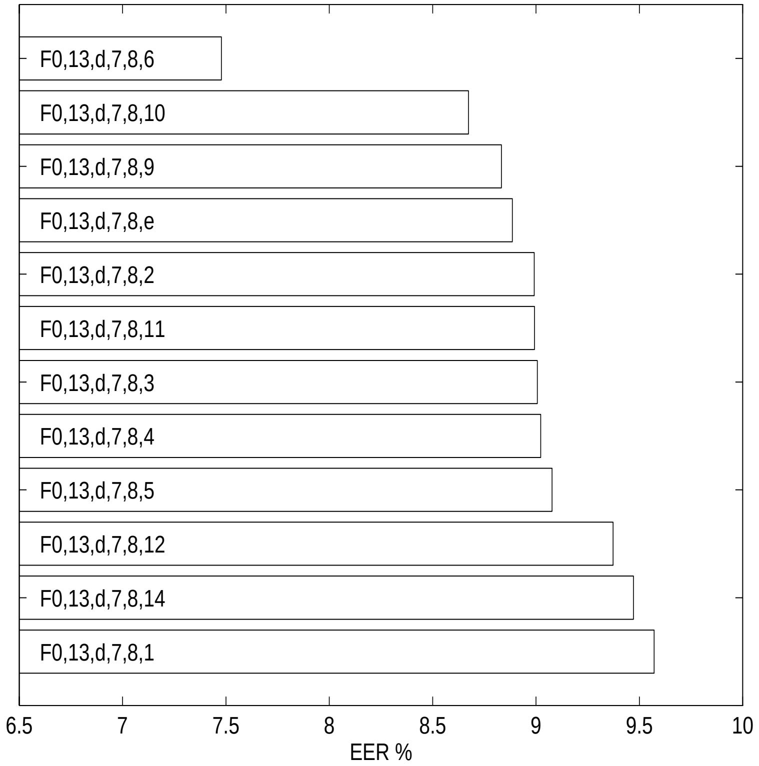


Figure 4-20: EERs for Six-Dimensional Features (NTIMIT DEV)

In general, performance decreased as more features were added to the 6-dimensional feature set. Figure 4-21 illustrates this trend and plots the best feature sets of each search stage. As before, EERs initially decrease as feature set size increase presumably because the additional features contribute further speaker-specific information. However, adding features eventually degrades performance, presumably because not enough training data is available to accurately estimate the model parameters.

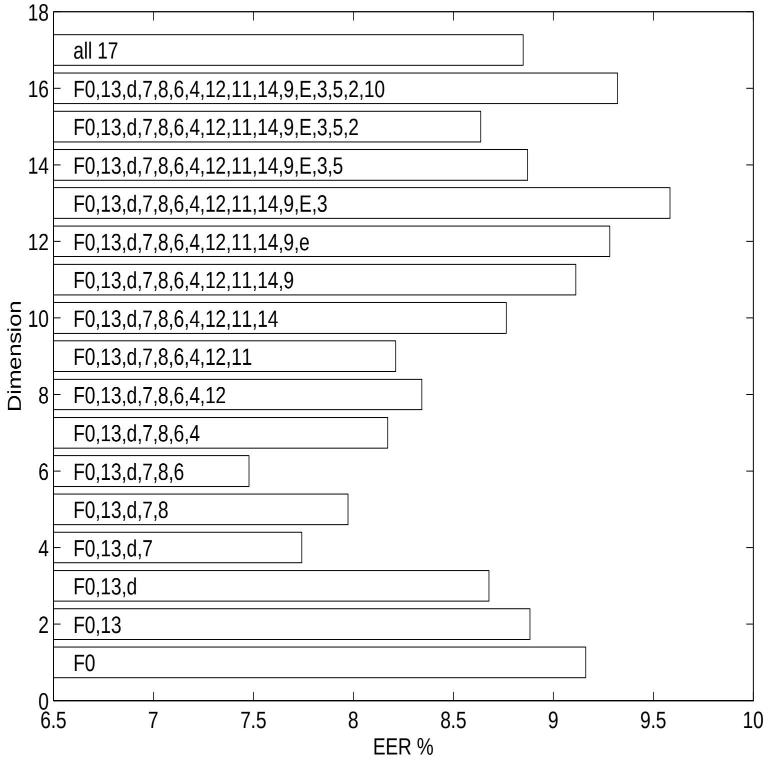


Figure 4-21: EERs for Best Features of Each Search Stage (NTIMIT DEV)

The (sub)-optimal 6-dimensional feature set is listed below, and consists of features that are essentially differ from those found from the previous TIMIT searches.

1. F0

2. MFCC13

3. Duration

4. MFCC7

5. MFCC8

6. MFCC6

The performances of the optimal feature sets selected from the three searches describe above are summarized in Chapter 3. Next, we analyze some trends observed as the feature vector size increases from 1 to 17.

4.4.3 Analysis

In this section, we investigate the sensitivity of equal error rates, Mahalanobis distances, forced scores, and average speaker ranks to feature vector size for both the TIMIT and NTIMIT domains[2].

EERs vs. Dimension

Figure 4-22 summarizes the feature search results by illustrating the EERs for both domains as a function of feature dimension. As discussed earlier, EERs initially decrease as features are added, presumably because the new features carry additional speaker information. However, as the feature vector size exceeds approximately 6 dimensions (NTIMIT) and 8 dimensions (TIMIT), EERs fail to significantly change, presumably because there is not enough training data to reliably estimate the model parameters. Although these trends are apparent in both domains, EERs using NTIMIT data fail to significantly change with dimension. As a result, performance remains poor in the noisy domain, suggesting that more robust features, models, and/or classifiers are necessary to improve performance.

The performance is poor in the telephone domain, because the limited bandwidth ($\sim$300Hz-3500Hz) makes it difficult to distinguish between broad phonetic classes. For example, strong fricatives and weak fricatives are distinguishable by the total energy in the phones. However, the energy for fricatives is largely distributed in frequencies above 3 KHz. Thus, in the telephone domain, it is difficult to discriminate

[2] The results reported in this section are those obtained when evaluating the development speaker set.

between strong fricatives and weak fricatives using energy as a feature. As a result, the speaker models have similar distributions for these classes, which in turn results in poor speaker verification performance.

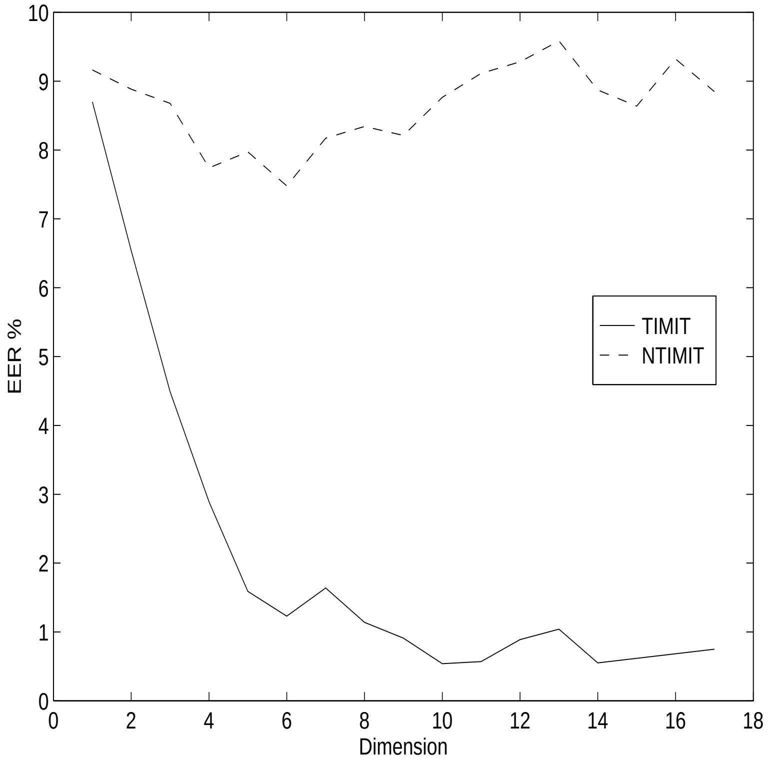


Figure 4-22: EER versus Feature Dimension

Mahalanobis Distances vs. Dimension

To observe the degree of similarity between the speaker models in both domains, we computed the average Mahalanobis distance between speaker models (within cohort sets) for each dimension. Figure 4-23 illustrates the higher rate of increase in distance for dimensions 1-8 than that for dimensions 9-17. The distances fail to increase at the initial rate after approximately 8 dimensions (as expected since EERs fail to significantly change), presumably because the added features do not further separate speakers in the acoustic space. In addition, the NTIMIT distances are considerably smaller than the TIMIT distances for every dimension, illustrating the degree of similarity and difficulty in discriminating between speakers in the telephone domain.

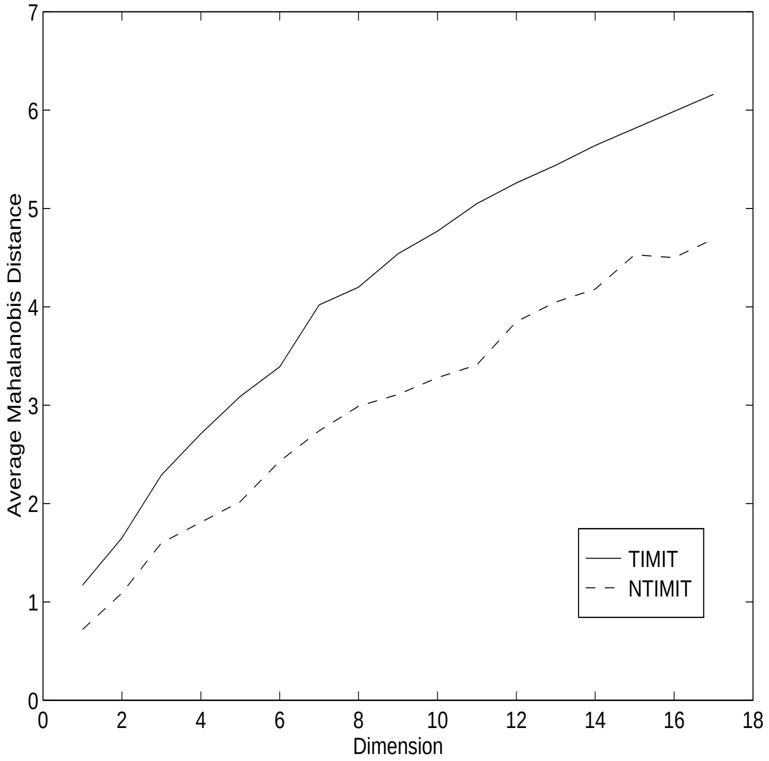


Figure 4-23: Mahalanobis Distance versus Feature Dimension

Forced Scores vs. Dimension

Next, we observe the sensitivity of the average forced alignment scores to feature vector size. The forced scores are correlated to how well the model parameters are trained. A higher score produces a better forced path, which implies that the models are better trained than those which produce a lower score. Figure 4-24 plots the average forced scores of the test utterances for the best feature sets of each search stage. As illustrated in Figure 4-24, increasing the number of features initally improves the forced paths, which suggests that the model parameters are becoming more reliable. However, after approximately 8 dimensions (TIMIT) and 6 dimensions (NTIMIT), the average scores decrease, illustrating the expected sparse data effects.

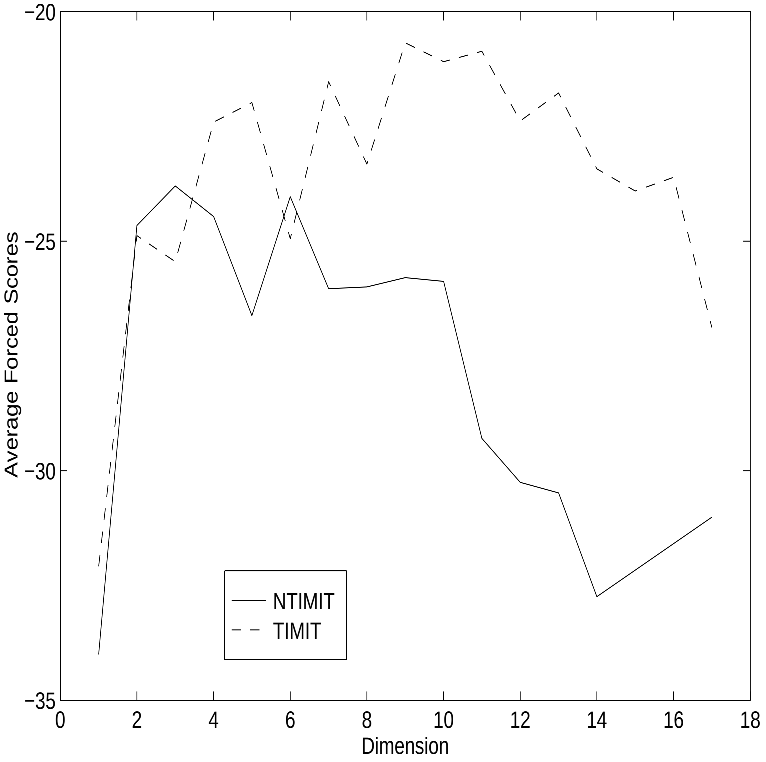


Figure 4-24: Average Forced Scores versus Feature Dimension

Average Speaker Rank vs. Dimension

Although the model parameters for 17 dimensions may be less accurate, according to forced scores, than the models for 10 dimensions, distinguishing between speakers may not be affected significantly. This phenomenon was observed while evaluating the TIMIT test speakers, when all 17 features resulted in a smaller EER than the 10-dimensional feature set. The speaker models may be poorly trained using all 17 features. However, the true speaker's model scores relatively higher on his/her test utterances than the other models. The relative scores between speakers can be inferred from the average rank of the user's score within his/her cohort scores. Figure 4-25 plots the average normalized speaker rank for the TIMIT and NTIMIT domains. The normalized ranks are ratios of the true speakers' ranks divided by the impostors' ranks. Therefore, the smaller the normalized rank the better performance. As expected, the normalized ranks of the 10 and 17-dimensional feature sets are the smallest in the TIMIT domain, and the rank of the 6-dimensional set is the smallest in the NTIMIT domain.

4.5 Summary

In the feature searches described in the previous chapter, we used a cohort set size of 14 and the EERs reported were already normalized. The normalized approximations, described in section 4.3, approach true performance as the speaker's score and his/her cohort scores approach the top 15 of 100 scores. To verify whether these normalized approximations are reasonably close to performance using all speakers during testing, we repeated experiments on particular feature sets using all speakers during testing. The cohort results (Coh) and the results using all speakers (NCoh) are summarized in Table 4-5.

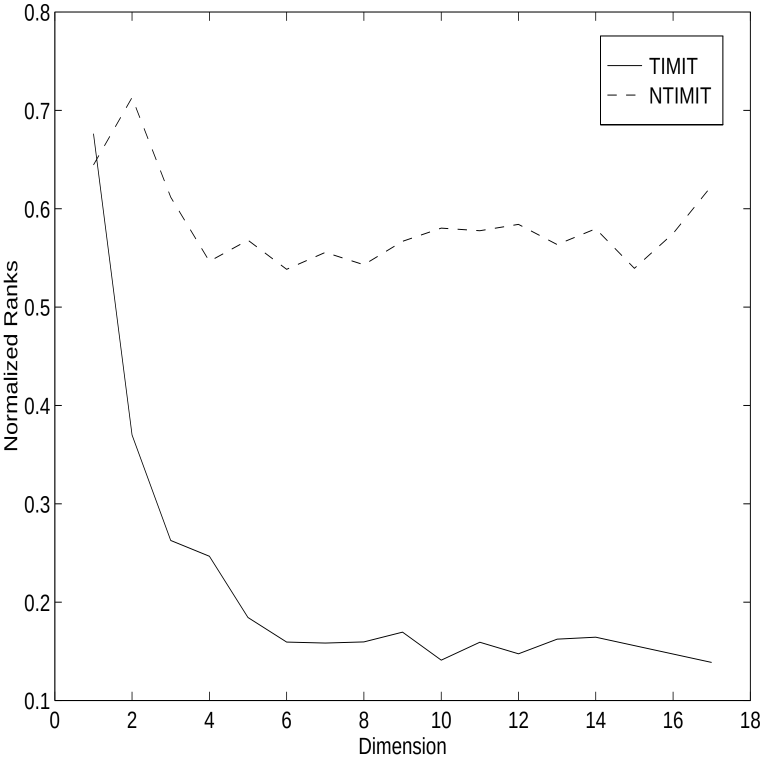


Figure 4-25: Average Speaker Ranks versus Feature Dimension

FEATURE SELECTION	**SPEAKER SET**	**TIMIT**		**NTIMIT**	
		Coh	**NCoh**	**Coh**	**NCoh**
all 17	dev	0.55%	0.01%	8.85%	20.19%
all 17	test	0.00%	0.00%	8.33%	21.56%
MFCC1_9_12_6_2_11_Energy_5_14_4	dev	0.54%	0.01%	10.08%	25.28%
MFCC1_9_12_6_2_11_Energy_5_14_4	test	1.10%	0.34%	9.88%	28.30%
10 random	dev	1.13%	0.24%	− − −	− − −
MFCC12_1_9_14_6_11_3_2_5_10	test	1.05%	0.11%	9.39%	27.08%
F0_MFCC13_Duration_7_8_6	dev	− − −	− − −	7.47%	17.93%
F0_MFCC13_Duration_7_8_6	test	− − −	− − −	8.36%	17.44%

Table 4-5: Performance

4.5.1 Cohorts vs. No Cohorts

As Table 4-5 illustrates, the normalized cohort approximations do not match true performance well, suggesting that the speaker's score and his/her cohorts' scores are not necessarily the top 15 of 100 scores. Some or all of the purported speaker's cohorts' scores may fall below the top 15 scores for the following reason. During training, we select cohorts using the Mahalanobis distance, whereas during testing we compare speakers using forced-paths scores. Although the models and forced scores are presumably correlated (i.e., similar models produce similar scores), the correlation is not 100%. Therefore, speakers who are most similar in terms of model parameters, may not be the most similar (out of the 100 speakers) in terms of forced scores. The lack of strong correlation between the distance metric and the forced scores may be due to the fact that when we computed the Mahalanobis distances, each broad class was weighted equally. Forced scores, on the other hand, take into account the frequency of the broad class by using the *a priori* probabilities. Therefore, in the case of two equidistant potential cohort speakers, it might be more appropriate to select the potential cohort that is closest to the true speaker with respect to the more common broad classes. For example, a similarity in vowel models might be much more important than a similarity in nasal models, since vowels occur much more frequently than nasals, and will thus contribute more heavily to the final forced score.

In addition to discrepancies between estimated and actual speaker verification performance, results using cohort sets are much better than those using all speakers during testing in the NTIMIT domain. The estimated performance may be better than the true performance for the following reason. The cohorts were not selected in a manner which maximizes the spread around each speaker, as they are in [34]. Spreading the cohorts around the true speaker prevents distant impostors from scoring highly with the true speaker's models without also scoring highly on at least some of the cohort models. Thus, a spread of cohorts prevents distant impostors from being falsely accepted, which in turn improves performance when all speakers may pose as impostors.

However, in the TIMIT domain, actual performance exceeds estimated performance using cohorts. This suggests that the impostor ranks within cohort sets are higher than their ranks within the 100 scores, which allows them to be falsely accepted more easily when using only cohorts during testing.

As described above, there are reasons why the estimated performances are either better or worse than the actual performances. In our current system, reducing computation with cohort normalization is beneficial in the noisy domain, while in the clean environment, all speakers should be used during testing.

4.5.2 TIMIT vs. NTIMIT

Overall, performance in the clean domain is significantly better than that in the telephone domain. As discussed in section 4.4.3, there are many reasons for such performance discrepancies. First, the models estimated using NTIMIT data are not as accurate as those trained on TIMIT data, since the NTIMIT forced scores are less than those using TIMIT data. Second, the average Mahalanobis distances between NTIMIT speaker models are significantly smaller than those using TIMIT data, which makes it difficult for the system to discriminate between speakers within a cohort set. Finally, the normalized speaker ranks (true speaker rank/impostor rank) within cohort sets are much higher in the telephone domain that in the clean domain, which implies that impostors score well on the true speaker's models.

In addition to the performance discrepancies in the two domains, the features selected from the searches in each environment differ. In the TIMIT domain, particular MFCCs and energy performed well. However, in the NTIMIT environment, F0, duration and other MFCCs performed well. These results suggests that prosodic features tend to be more robust in noisy environments than MFCCs. However, overall system performance remains poor in the telephone domain, requiring modifications to the system. Future work, discussed in Chapter 5, describes some areas of exploration that may improve performance in noisy environments.

4.5.3 Selected Features vs. All Features

Table 4-5 summarizes results for the features selected from each feature search along with results using all 17 features and randomly selected features. The development set for TIMIT selected a (sub)-optimal 10-dimensional set (energy,MFCC1,9,12,6,2,11,5,14,4), which results in 0.54% EER using cohorts and 0.01% EER using all speakers during testing. This exceeds the performance of computing all 17 features on the same speaker set. However, performance using 17 features on the test set actually performs better than using the optimal 10-dimensional set selected from the test set search (MFCC12,1,9,14,6,11,3,2,5,10). Therefore, performance remains consistent across all speakers using 17 features and achieves a performance 0% EER on the TIMIT test set. Since computation is not prohibitive (shown in section **??**) using 17 features, the optimal feature set in the TIMIT domain consists of all 17 measurements. However, in the NTIMIT domain, the (sub)-optimal 6-dimensional feature set (F0,duration,MFCC13,7,8,6) performs on average much better than the 17-dimensional feature set.

To ascertain whether the selected subsets of the 17 measurements are significantly better than random subsets, we tested the development speakers for TIMIT using 10 random features and compared the results to those using the 10 selected features. As shown in Table 4-5, performance decreased from 0.54% EER to 1.13% EER using cohorts during testing, and from 0.01% EER to 0.24% EER using all speakers during testing. Thus, if computation is prohibitive, selecting a subset of features from a feature search provides much better performance than computing a random subset of features of the same dimension.

Chapter 5

Conclusions & Future Work

5.1 Summary

This thesis attempted to achieve two goals. The first was to build a competitive segment-based speaker verification system, and the second goal was to build a computationally efficient system. Often, these goals cannot be achieved simultaneously. Systems that achieve 0% error may not be computationally efficient. Below, we briefly discuss how we significantly reduced computation while maintaining competitive speaker verification performance.

As described in section 4.5, our system achieves a performance of 0% EER in the TIMIT domain and 8.36% EER in the NTIMIT domain. We significantly reduced computation in many ways. As previously mentioned, the system uses a small number of features, a small number of phonetic models per speaker, few model parameters, and few competing speakers during testing. We believe that the system is able to achieve good performance with a simple design because we treated speech as a concatenation of segments, rather than frames. Past observations show that speech segments carry speaker-specific information. Therefore, by considering the speech signal as a concatenation of phone-size units, we capitalized on measurements for such units more readily.

5.2 Future Work

In this section, we discuss future work in connection with our research. This work includes exploring robustness issues, conducting an exhaustive search for optimal acoustic features, selecting broad classes based on acoustic criteria, representing features with more complex distributions, and adapting speaker models. Finally, we plan to incorporate our speaker verification system into a web-based information access system called GALAXY.

5.2.1 Robustness Issues

An important future topic to investigate is the robustness of the system in various acoustic environments. Although we achieved ideal speaker verification performance on the TIMIT corpus, the training environment matched the testing environment. In reality, these two environments usually differ. For example, training data may be collected in a quiet environment over a microphone, while test data are transmitted through a noisier environment over a telephone. The noisy environment and limited bandwidth cause feature statistics to change; thus test data are mis-matched to trained models.

To appreciate the magnitude of the degradation in performance due to mismatched environments, we evaluated our system using speaker models trained on TIMIT and tested on NTIMIT. As mentioned in section 2.2, NTIMIT is TIMIT transmitted over a telephone network. Figure 5-1 gives an indication of how SV performance degrades when testing on mis-matched data. The EER for a randomly-selected feature set degrades from approximately 5% EER when training and testing on TIMIT to 40% EER when training on TIMIT and testing on NTIMIT. This significant decrease in performance suggests that a robust system is necessary for mis-matched environments. Perhaps an algorithm could be adopted to re-estimate the speaker model parameters trained on clean speech to better fit noisy test data. Alternatively, it may be necessary to search for better acoustic features for the noisy environment.

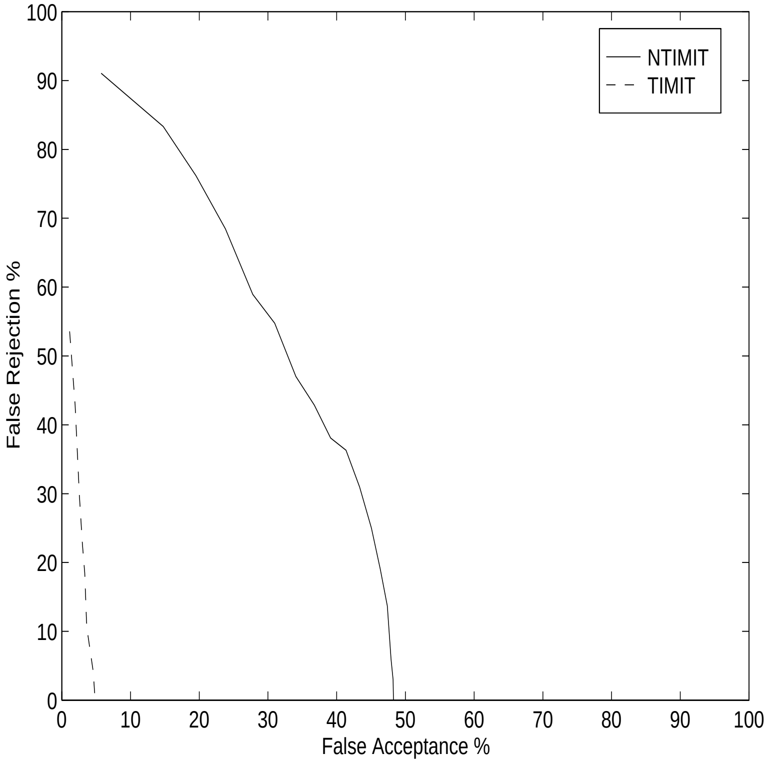


Figure 5-1: ROC curves for Optimal Feature Set: Models are trained on TIMIT data and tested on either TIMIT or NTIMIT.

5.2.2 Exhaustive Search for Robust Features

In this thesis, we conducted a greedy search with pruning for a (sub)-optimal set of acoustic features. Since we did not explore all possible feature sets formed from the 17 selected measurements, we do not know whether or not the best feature set found from the greedy search is optimal. To ensure that a feature set formed from a pool of measurements is optimal, an exhaustive search without pruning should be conducted. Optimality may be more important in domains where performance degrades significantly with different feature sets, as in noisy environments. As illustrated above, our (sub)-optimal feature set results in good performance when train and test environments are clean. However, performance degrades significantly with the same feature set, when testing in a noisy domain.

In the future, we plan to use a program called SAILS to help us extract optimal and robust features. SAILS [31] was originally used to extract optimal acoustic attributes that signify phonetic contrasts for speech recognition. It allows the user to vary parameters such as frequency range and time interval for measuring any set of features for selected speakers' phonemes, and their left and right phonetic contexts. For example, if the algorithm explores MFCCs, SAILS finds optimal places to start and end measuring the coefficients (SAILS specifies a range in the segment to compute over, such as 30%-70% of the segment), as well as which coefficients best discriminate between speakers' phonemes.

5.2.3 Feature-Motivated Broad Class Selections

As observed in this thesis, our selection of the broad manner classes affected the performance of various features, especially duration. In the past, duration has been proven to be robust and speaker-specific [42]. However, the classes we selected did not reflect different duration characteristics. As a result, the variances of duration were large for all speakers models, and the performance scores using duration ranked last in the scores for the 1-dimensional stage for both searches conducted. Thus, duration was eliminated in the search for optimal features.

In order to prevent disregarding potentially useful features for speaker verification, and to ensure that each broad class has small variances, we plan to select broad classes by using an unsupervised clustering algorithm. Unsupervised clustering algorithms, such as the K-means algorithm, group phones into classes based on acoustic characteristics. Thus, unlike the manner classes, each broad class should have similar acoustic statistics. As a result, the speaker models will have small variances for all features, which makes distinguishing between speakers easier than if the models have large variances. In turn, we hope to improve speaker verification performance.

5.2.4 Representing Features With More Complex Distributions

Future work also includes exploring more complex feature distributions than diagonal Gaussians. We chose to represent the broad class acoustic statistics with diagonal Gaussians, which have few parameters to train, to reduce computation. As a result we traded model accuracy for computation. Essentially, we forced the acoustic features for each class to be represented by a mean vector and a diagonal covariance matrix, which assumes that the features are uncorrelated random variables. Features may be more accurately modeled with mixtures of diagonal Gaussians, or full covariance Gaussians. Given enough data, more complex models may improve speaker verification performance. However, computation increases, since complex models have many parameters to estimate during training.

5.2.5 Adaptation of Models

Often, little training data are available per speaker. As a result the speaker models estimated from the data are not reliable. Ideally, one would like to obtain accurate models from little training data so that users will not be required to speak many utterances before being able to use the system. To reliably represent speakers with little training data, many investigators apply adaptation techniques to the speaker models. Specifically, the means, variances, and *a priori* broad class probabilities are typically adapted from the statistics of a well-trained speaker-independent model.

As a first attempt to observe performance effects due to adaptation of speaker models, we modified the *a priori* class probabilities of each speaker model. Specifically, we first trained a speaker-independent (SI) model using data from the 462 speakers' data from the NIST-train set of TIMIT. These estimates were then adapted to each speaker model. This simple technique forced the *a priori* estimates to be accurate and consistent across all speakers. The *a priori* class probabilities should be independent of speakers since the probability of observing a particular broad class in a segment is dependent only on the lexicon in the corpus.

Figure 5-2 illustrates the performance, before and after applying our adaptation method, evaluated on the original 168 test speakers using the optimal 6-dimensional feature set. As shown in the figure, there is no significant improvement in performance when we only adapt the *a priori* class probabilities. Perhaps the *a priori* estimates did not differ significantly from speaker to speaker before adaptation, resulting in little performance differences. The insignificant improvement after adaptation of the *a prioris* suggests that more complex adaptation techniques that modify means and variances are required to improve the speaker models, and in turn improve speaker verification performance.

In the future, we plan to implement the maximum *a-posteriori* probability (MAP) adaptation procedure, a common method for adapting all the statistics of models. MAP provides a way to incorporate prior information into the estimation process, by assuming an *a-priori* distribution of the parameters that are being estimated. Details on the MAP technique can be found in [40, 23].

5.2.6 Incorporating into GALAXY

Finally, we plan to incorporate our speaker verification system into the GALAXY conversational system [13]. GALAXY is a system currently under development in our group that enables information access using spoken dialogue. Presently, GALAXY can access the information sources on the Internet via speech for four applications: weather reports, airline travel information, automobile sales information, and the

Figure 5-2: ROC curves for the Optimal Feature Set Before and After Adaptation

Boston city guide.

Appendix A

Mel-frequency Cepstral Coefficients

To extract MFCCs from speech, speech samples are initially modulated by a Hamming window of approximately 25 msec in duration. The discrete Fourier transform (DFT) of the modulated interval of speech is then computed and squared component-wise to obtain the power spectral density (PSD or energy) of the speech interval. The samples are then transformed logarithmically and filtered by the mel-frequency-based banks. These auditory triangular filter banks consist of 40 constant-area filters designed to approximate the frequency response of the human ear. The filters are on a mel-frequency scale, which is linear up to 1000 Hz and logarithmic thereafter. These filters are shown for a particular range of frequencies in Figure A-1 below.

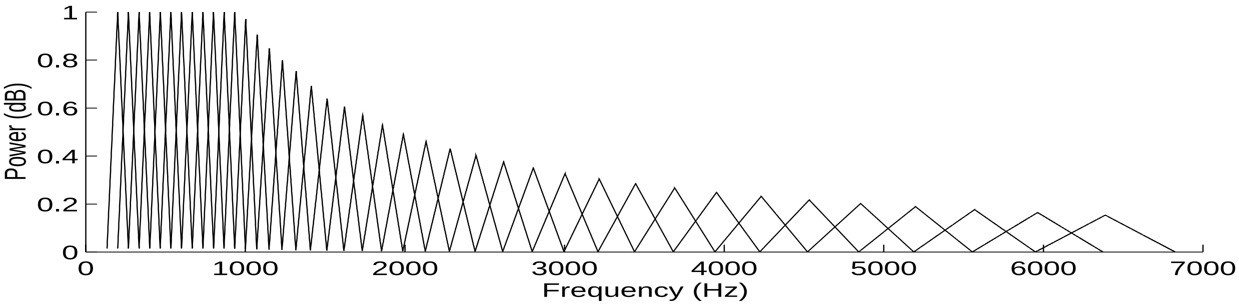


Figure A-1: MFSC Filter Banks

Collectively these coefficients form the N-dimensional mel-frequency-based spectral coefficient (MFSC) vector for the windowed speech. Finally, M (a number not necessarily equal to N) MFCCs are calculated from these spectral coefficients via the following discrete cosine transform (DCT),

$$Y_i = \sum_{k=1}^{N} X_k cos[(k - \frac{1}{2})\frac{\pi}{N}]$$

where X_k for k = 1,2,..N are the mel-frequency spectral coefficients (MFSCs), and Y_i for i = 1,2,...M are the mel-frequency cepstral coefficients (MFCCs). The details of the signal processing described above is summarized in the block diagram below. More details on computing MFCCs can be found in [26]. Some SV systems that compute MFCCs are [30, 21, 8].

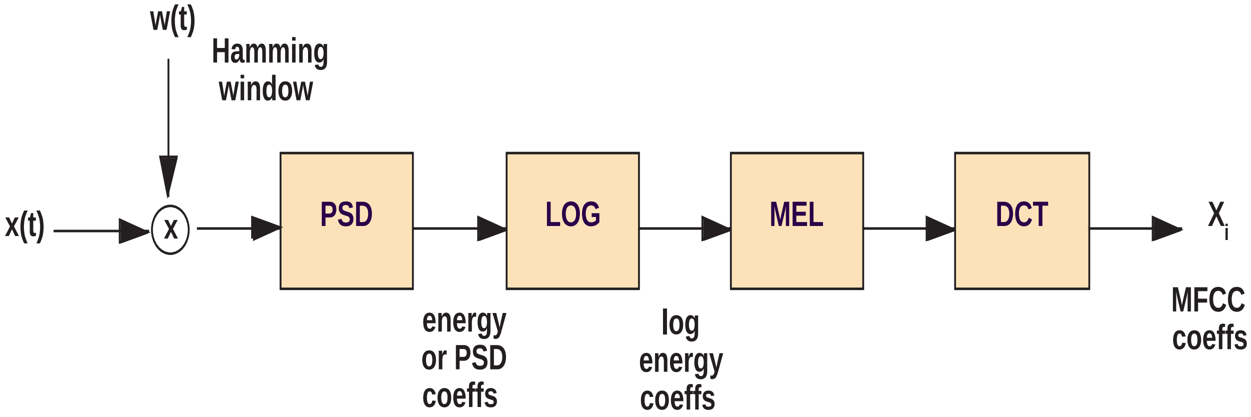


Figure A-2: Block Diagram for Computing MFCCs

Appendix B

Linear Prediction Analysis

The principles of linear prediction involve modeling the vocal tract system with an all-pole system function. The processing of a speech signal is shown in Figure B-1.

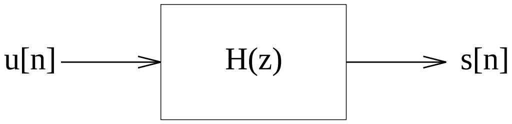


Figure B-1: Production of Speech Signals

The speech signal, shown as the output of the discrete-time system in Figure B-1, is produced by exciting the vocal tract system with a wide-band excitation $u[n]$. The vocal tract, $H(z)$, changes slowly with time, hence for short time intervals, the vocal tract can be modeled as a fixed pth-order all-pole system. Specifically,

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

Thus, by cross multiplying and taking the inverse Bilateral z-transform of both sides, we obtain:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Gu[n]$$

The goal of linear prediction analysis is to estimate the a_k's and G from $s[n]$. The predicted signal is defined as:

$$\tilde{s}[n] = \sum_{k=1}^{p} \alpha_k s[n-k]$$

which leaves a residual error $e[n] = s[n] - \tilde{s}[n] \simeq Gu[n]$ (this is approximately true when the estimates α_k's are very good). The α_k's are chosen to minimize the residual error. SV systems that use LPCs for acoustic features are [18, 43, 3]. For a tutorial on LPC analysis, refer to [22].

B.1 Estimation of Fundamental Frequency

There are many methods to approximate F0, such as cepstral analysis and LPC analysis. We describe the approximation of F0 using linear prediction analysis below. Refer to [29] for the method of approximating F0 from cepstral coefficients. In order to estimate the fundamental frequency using LPC analysis, the autocorrelation of the error function is computed. During a fixed time interval of the speech signal, $s[n]$ can be assumed to be N points in length, which makes the autocorrelation function of the error, $R_e[k]$, a finite sum for each n. Specifically,

$$R_e[k] = \sum_{n=1}^{N-1-k} e[n]e[n+k]$$

When the speech signal is voiced, $u[n]$ is assumed a train of narrow glottal pulses. The signal is then windowed over an interval, and a few of the pulses remain in the interval if the window is larger than a few pulse periods. Note that the fundamental frequency is simply the reciprocal of the fundamental period of the pulses. The autocorrelation function of the residual error exhibits local maxima where the pulses occur. An example of the error autocorrelation function for a voiced time interval is illustrated in Figure B-2. These functions are plotted for every frame, and the distance between the first two peaks in $R_e[k]$ are estimates of the fundamental period (1/F0).

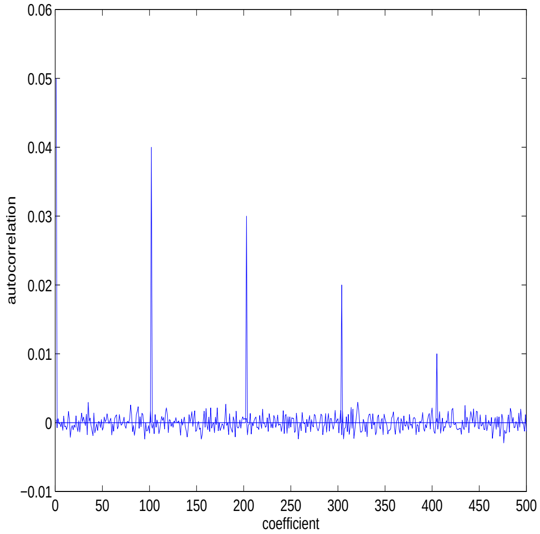


Figure B-2: The Autocorrelation Function of the Error Residual for a Short Time Interval

Appendix C

TIMIT: Development Set

FEATURE SET	EER %
F0	9.78
Energy	10.17
Duration	11.39
MFCC1	8.71
MFCC2	10.28
MFCC3	11.32
MFCC4	10.26
MFCC5	10.42
MFCC6	10.13
MFCC7	9.82
MFCC8	10.22
MFCC9	9.32
MFCC10	10.38
MFCC11	10.37
MFCC12	10.48
MFCC13	10.91
MFCC14	9.96

Table C-1: One-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9	6.54
MFCC1,12	6.55
MFCC1,11	6.76
MFCC1,2	7.02
MFCC1,7	7.06
MFCC1,F0	7.33
MFCC1,4	7.54
MFCC1,14	7.55
MFCC1,8	7.84
MFCC1,10	8.18
MFCC1,5	8.34
MFCC1,E	9.72
MFCC1,6	9.06

Table C-2: Two-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12	4.49
MFCC1,9,F0	4.86
MFCC1,9,7	4.98
MFCC1,9,8	5.55
MFCC1,9,6	5.61
MFCC1,9,5	5.92
MFCC1,9,2	6.02
MFCC1,9,E	6.20
MFCC1,9,4	6.52
MFCC1,9,10	6.88
MFCC1,9,11	8.34
MFCC1,9,14	9.47

Table C-3: Three-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6	2.89
MFCC1,9,12,F0	2.98
MFCC1,9,12,14	3.53
MFCC1,9,12,8	3.69
MFCC1,9,12,10	3.84
MFCC1,9,12,4	3.89
MFCC1,9,12,2	4.28
MFCC1,9,12,7	4.49
MFCC1,9,12,E	4.69
MFCC1,9,12,5	4.83
MFCC1,9,12,10	5.18

Table C-4: Four-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6,2	1.59
MFCC1,9,12,6,14	2.01
MFCC1,9,12,6,7	2.56
MFCC1,9,12,6,4	2.77
MFCC1,9,12,6,5	2.78
MFCC1,9,12,6,10	2.84
MFCC1,9,12,6,F0	3.30
MFCC1,9,12,6,11	3.52
MFCC1,9,12,6,8	3.70
MFCC1,9,12,6,E	3.99

Table C-5: Five-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
MFCC1,9,12,6,2,11	1.24
MFCC1,9,12,6,2,7	1.94
MFCC1,9,12,6,2,10	2.02
MFCC1,9,12,6,2,8	2.12
MFCC1,9,12,6,2,14	2.12
MFCC1,9,12,6,2,F0	2.16
MFCC1,9,12,6,2,E	2.17
MFCC1,9,12,6,2,4	2.58
MFCC1,9,12,6,2,5	2.78

Table C-6: Six-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
MFCC1,9,12,6,2,11,E	1.64
MFCC1,9,12,6,2,11,4	1.91
MFCC1,9,12,6,2,11,F0	2.21
MFCC1,9,12,6,2,11,14	2.31
MFCC1,9,12,6,2,11,7	2.47
MFCC1,9,12,6,2,11,8	3.28
MFCC1,9,12,6,2,11,10	3.43
MFCC1,9,12,6,2,11,5	3.92

Table C-7: Seven-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
MFCC1,9,12,6,2,11,E,5	1.16
MFCC1,9,12,6,2,11,E,7	1.28
MFCC1,9,12,6,2,11,E,F0	1.50
MFCC1,9,12,6,2,11,E,10	1.50
MFCC1,9,12,6,2,11,E,4	1.55
MFCC1,9,12,6,2,11,E,8	1.56
MFCC1,9,12,6,2,11,E,14	2.00

Table C-8: Eight-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6,2,11,E,5,14	0.91
MFCC1,9,12,6,2,11,E,5,4	0.99
MFCC1,9,12,6,2,11,E,5,F0	1.12
MFCC1,9,12,6,2,11,E,5,8	1.50
MFCC1,9,12,6,2,11,E,5,7	1.52
MFCC1,9,12,6,2,11,E,5,10	1.88

Table C-9: Nine-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6,2,11,E,5,14,4	0.54
MFCC1,9,12,6,2,11,E,5,14,F0	0.95
MFCC1,9,12,6,2,11,E,5,14,10	0.96
MFCC1,9,12,6,2,11,E,5,14,8	1.10
MFCC1,9,12,6,2,11,E,5,14,7	1.14

Table C-10: Ten-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6,2,11,E,5,14,4,10	0.58
MFCC1,9,12,6,2,11,E,5,14,4,F0	0.90
MFCC1,9,12,6,2,11,E,5,14,4,7	0.91
MFCC1,9,12,6,2,11,E,5,14,4,8	1.06

Table C-11: Eleven-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	**EER %**
MFCC1,9,12,6,2,11,E,5,14,4,10,F0	0.89
MFCC1,9,12,6,2,11,E,5,14,4,10,7	1.17
MFCC1,9,12,6,2,11,E,5,14,4,10,8	1.20

Table C-12: Twelve-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
MFCC1,9,12,6,2,11,E,5,14,4,10,F0,8	1.04
MFCC1,9,12,6,2,11,E,5,14,4,10,F0,7	2.00

Table C-13: Thirteen-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
MFCC1,9,12,6,2,11,E,5,14,4,10,F0,8,7	0.55

Table C-14: Fourteen-Dimensional Feature Set Results (TIMIT DEV)

FEATURE SET	EER %
all 17 features	0.75

Table C-15: Seventeen-Dimensional Feature Set Results (TIMIT DEV)

Appendix D

TIMIT: Test Set

FEATURE SET	**DEV SET**	**TEST SET**	**DIFF**
	EER %	EER %	(MAG)
F0	9.78	10.01	0.23
Energy	10.17	10.38	0.21
Duration	11.39	11.85	0.46
MFCC1	8.71	9.54	0.83
MFCC2	10.28	9.58	0.70
MFCC3	11.32	11.82	0.50
MFCC4	10.26	9.33	0.93
MFCC5	10.42	9.98	0.44
MFCC6	10.13	9.55	0.58
MFCC7	9.82	9.86	0.04
MFCC8	10.22	10.73	0.51
MFCC9	9.32	9.78	0.46
MFCC10	10.38	9.63	0.75
MFCC11	10.37	10.16	0.21
MFCC12	10.48	9.26	1.22
MFCC13	10.91	9.64	1.27
MFCC14	9.96	9.36	0.60

Table D-1: One Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1	6.65
MFCC12,6	7.20
MFCC12,9	7.28
MFCC12,7	7.68
MFCC12,10	7.80
MFCC12,F0	7.95
MFCC12,4	8.17
MFCC12,2	8.66
MFCC12,E	8.88
MFCC12,3	9.43
MFCC12,5	9.61
MFCC12,14	10.77
MFCC12,11	11.08

Table D-2: Two-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9	4.47
MFCC12,1,10	5.03
MFCC12,1,14	5.15
MFCC12,1,2	5.29
MFCC12,1,E	5.31
MFCC12,1,F0	5.35
MFCC12,1,6	5.37
MFCC12,1,3	6.09
MFCC12,1,7	6.30
MFCC12,1,4	6.37
MFCC12,1,11	6.52
MFCC12,1,5	6.79

Table D-3: Three-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14	3.33
MFCC12,1,9,F0	3.34
MFCC12,1,9,6	3.80
MFCC12,1,9,E	4.02
MFCC12,1,9,11	4.22
MFCC12,1,9,7	4.26
MFCC12,1,9,4	4.37
MFCC12,1,9,10	4.43
MFCC12,1,9,5	4.67
MFCC12,1,9,2	4.98
MFCC12,1,9,3	5.14

Table D-4: Four-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6	2.53
MFCC12,1,9,14,10	2.55
MFCC12,1,9,14,11	2.94
MFCC12,1,9,14,5	3.03
MFCC12,1,9,14,7	3.06
MFCC12,1,9,14,4	3.23
MFCC12,1,9,14,2	3.90
MFCC12,1,9,14,3	4.13
MFCC12,1,9,14,E	4.18
MFCC12,1,9,14,F0	4.28

Table D-5: Five-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11	1.27
MFCC12,1,9,14,6,4	2.00
MFCC12,1,9,14,6,F0	2.13
MFCC12,1,9,14,6,E	2.15
MFCC12,1,9,14,6,3	2.17
MFCC12,1,9,14,6,2	2.36
MFCC12,1,9,14,6,5	2.48
MFCC12,1,9,14,6,10	2.73
MFCC12,1,9,14,6,7	3.06

Table D-6: Six-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,3	1.22
MFCC12,1,9,14,6,11,2	1.43
MFCC12,1,9,14,6,11,5	1.67
MFCC12,1,9,14,6,11,7	1.71
MFCC12,1,9,14,6,11,4	1.95
MFCC12,1,9,14,6,11,E	2.17
MFCC12,1,9,14,6,11,F0	3.29
MFCC12,1,9,14,6,11,10	3.34

Table D-7: Seven-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,2	1.23
MFCC12,1,9,14,6,11,F0	1.59
MFCC12,1,9,14,6,11,4	1.72
MFCC12,1,9,14,6,11,7	1.88
MFCC12,1,9,14,6,11,10	2.18
MFCC12,1,9,14,6,11,5	2.23
MFCC12,1,9,14,6,11,E	2.33

Table D-8: Eight-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5	1.64
MFCC12,1,9,14,6,11,E	1.71
MFCC12,1,9,14,6,11,10	2.23
MFCC12,1,9,14,6,11,4	2.31
MFCC12,1,9,14,6,11,7	2.57
MFCC12,1,9,14,6,11,F0	2.72

Table D-9: Nine-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5,10	1.05
MFCC12,1,9,14,6,11,5,7	1.11
MFCC12,1,9,14,6,11,5,F0	1.22
MFCC12,1,9,14,6,11,5,E	1.82
MFCC12,1,9,14,6,11,5,4	2.31

Table D-10: Ten-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5,10,4	1.11
MFCC12,1,9,14,6,11,5,10,E	1.33
MFCC12,1,9,14,6,11,5,10,7	1.59
MFCC12,1,9,14,6,11,5,10,F0	1.94

Table D-11: Eleven-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5,10,4,F0	0.92
MFCC12,1,9,14,6,11,5,10,4,7	1.46
MFCC12,1,9,14,6,11,5,10,4,E	1.70

Table D-12: Twelve-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5,10,4,F0,E	0.94
MFCC12,1,9,14,6,11,5,10,4,F0,7	1.05

Table D-13: Thirteen-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
MFCC12,1,9,14,6,11,5,10,4,F0,E,7	1.16

Table D-14: Fourteen-Dimensional Feature Set Results (TIMIT TEST)

FEATURE SET	**EER %**
all 17	0.58

Table D-15: Seventeen-Dimensional Feature Set Results (TIMIT TEST)

Appendix E

NTIMIT: Development Set

FEATURE SET	NTIMIT	TIMIT
	EER %	EER %
F0	9.86	9.78
Energy	11.41	10.17
Duration	11.52	11.39
MFCC1	11.90	8.71
MFCC2	11.51	10.28
MFCC3	11.38	11.32
MFCC4	11.77	10.26
MFCC5	11.48	10.42
MFCC6	11.63	10.13
MFCC7	11.31	9.82
MFCC8	11.86	10.22
MFCC9	11.69	9.32
MFCC10	11.82	10.38
MFCC11	11.93	10.37
MFCC12	10.94	10.48
MFCC13	11.01	10.91
MFCC14	11.69	9.96

Table E-1: One-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13	8.88
F0,12	9.02
F0,6	9.09
F0,14	9.23
F0,D	9.27
F0,9	9.35
F0,7	9.44
F0,4	9.78
F0,10	9.78
F0,11	9.85
F0,8	10.31
F0,3	10.47
F0,5	10.54
F0,2	11.70
F0,E	11.83
F0,1	10.96

Table E-2: Two-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D	8.68
F0,13,7	8.84
F0,13,1	9.13
F0,13,E	9.31
F0,13,6	9.35
F0,13,8	9.37
F0,13,9	9.38
F0,13,14	9.65
F0,13,12	9.66
F0,13,5	9.69
F0,13,10	9.78
F0,13,2	9.92
F0,13,4	10.04
F0,13,1	10.13
F0,13,3	10.24

Table E-3: Three-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7	7.47
F0,13,D,8	8.14
F0,13,D,10	8.22
F0,13,D,E	8.44
F0,13,D,5	8.61
F0,13,D,6	8.77
F0,13,D,1	8.78
F0,13,D,14	8.83
F0,13,D,12	8.90
F0,13,D,4	8.93
F0,13,D,3	9.11
F0,13,D,9	9.24
F0,13,D,11	9.26
F0,13,D,2	9.31

Table E-4: Four-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7,8	7.97
F0,13,D,7,6	8.24
F0,13,D,7,10	8.30
F0,13,D,7,11	8.39
F0,13,D,7,9	8.53
F0,13,D,7,E	8.56
F0,13,D,7,5	8.75
F0,13,D,7,12	8.84
F0,13,D,7,2	8.87
F0,13,D,7,4	8.88
F0,13,D,7,3	9.16
F0,13,D,7,14	9.56
F0,13,D,7,1	11.92

Table E-5: Five-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7,8,6	7.48
F0,13,D,7,8,10	8.67
F0,13,D,7,8,9	8.83
F0,13,D,7,8,E	8.88
F0,13,D,7,8,2	8.99
F0,13,D,7,8,11	8.99
F0,13,D,7,8,3	9.00
F0,13,D,7,8,4	9.02
F0,13,D,7,8,5	9.08
F0,13,D,7,8,12	9.37
F0,13,D,7,8,14	9.47
F0,13,D,7,8,1	9.57

Table E-6: Six-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7,8,6,4	8.17
F0,13,D,7,8,6,9	8.45
F0,13,D,7,8,6,11	8.70
F0,13,D,7,8,6,2	8.79
F0,13,D,7,8,6,10	8.81
F0,13,D,7,8,6,12	9.02
F0,13,D,7,8,6,5	9.34
F0,13,D,7,8,6,1	9.39
F0,13,D,7,8,6,14	9.55
F0,13,D,7,8,6,E	9.66
F0,13,D,7,8,6,3	9.77

Table E-7: Seven-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12	8.34
F0,13,D,7,8,6,4,9	8.64
F0,13,D,7,8,6,4,14	8.85
F0,13,D,7,8,6,4,10	8.96
F0,13,D,7,8,6,4,11	9.06
F0,13,D,7,8,6,4,E	9.28
F0,13,D,7,8,6,4,5	9.38
F0,13,D,7,8,6,4,3	9.77
F0,13,D,7,8,6,4,2	9.96
F0,13,D,7,8,6,4,1	10.08

Table E-8: Eight-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11	8.21
F0,13,D,7,8,6,4,12,9	8.54
F0,13,D,7,8,6,4,12,14	9.00
F0,13,D,7,8,6,4,12,10	9.05
F0,13,D,7,8,6,4,12,5	9.07
F0,13,D,7,8,6,4,12,1	9.18
F0,13,D,7,8,6,4,12,2	9.31
F0,13,D,7,8,6,4,12,3	9.57
F0,13,D,7,8,6,4,12,E	10.02

Table E-9: Nine-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11,14	8.76
F0,13,D,7,8,6,4,12,11,1	8.96
F0,13,D,7,8,6,4,12,11,5	8.97
F0,13,D,7,8,6,4,12,11,9	9.15
F0,13,D,7,8,6,4,12,11,10	9.53
F0,13,D,7,8,6,4,12,11,2	9.57
F0,13,D,7,8,6,4,12,11,E	9.84
F0,13,D,7,8,6,4,12,11,3	10.09

Table E-10: Ten-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11,14,9	9.11
F0,13,D,7,8,6,4,12,11,14,5	9.46
F0,13,D,7,8,6,4,12,11,14,1	9.73
F0,13,D,7,8,6,4,12,11,14,3	9.86
F0,13,D,7,8,6,4,12,11,14,E	9.91
F0,13,D,7,8,6,4,12,11,14,2	10.11

Table E-11: Eleven-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11,14,9,E	9.28
F0,13,D,7,8,6,4,12,11,14,9,3	9.57
F0,13,D,7,8,6,4,12,11,14,9,5	10.03
F0,13,D,7,8,6,4,12,11,14,9,2	10.07
F0,13,D,7,8,6,4,12,11,14,9,1	10.08

Table E-12: Twelve-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11,14,9,E,3	9.58
F0,13,D,7,8,6,4,12,11,14,9,E,1	9.87
F0,13,D,7,8,6,4,12,11,14,9,E,2	10.07
F0,13,D,7,8,6,4,12,11,14,9,E,5	10.14

Table E-13: Thirteen-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	EER %
F0,13,D,7,8,6,4,12,11,14,9,E,5	8.87
F0,13,D,7,8,6,4,12,11,14,9,E,10	9.59
F0,13,D,7,8,6,4,12,11,14,9,E,2	10.44

Table E-14: Fourteen-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7,8,6,4,12,11,14,9,E,5,2	8.64
F0,13,D,7,8,6,4,12,11,14,9,E,5,10	9.33
F0,13,D,7,8,6,4,12,11,14,9,E,5,1	9.71

Table E-15: Fifteen-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
F0,13,D,7,8,6,4,12,11,14,9,E,5,2,10	9.32
F0,13,D,7,8,6,4,12,11,14,9,E,5,2,1	9.77

Table E-16: Sixteen-Dimensional Feature Set Results (NTIMIT DEV)

FEATURE SET	**EER %**
all 17	8.85

Table E-17: Seventeen-Dimensional Feature Set Results (NTIMIT DEV)

Bibliography

[1] B. Atal. Automatic speaker recognition based on pitch contours. *JASA*, 52:1687–1697, 1972.

[2] Y. Bennani. Speaker identification through modular connectionist architecture: evaluation on the timit database. In *Proceedings of the 1992 International Conference on Spoken Language Processing*, pages 607–610, 1992.

[3] Y. Bennani and P. Gallinari. On the use of tddn-extracted features information in talker identification. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 385–388. IEEE, 1991.

[4] J. Campbell. Testing with he yoho cd–rom voice verification corpus. In *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*, pages 341–344. IEEE, 1995.

[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms.* The MIT Press, 1990.

[6] G. Doddington. Speaker recognition-identifying people by their voices. *Proceedings of the IEEE*, 73(11):1651–1663, November 1985.

[7] G. Doddington and B. Secrest. An integrated pitch tracking algorithm for speech systems. In *Proceedings of the 1983 International Conference on Acoustics, Speech, and Signal Processing*, pages 1352–1355, 1983.

[8] K. Farrel and R. Mammone. Speaker identification using neural networks. In *Proceedings of the 1994 International Conference on Acoustics, Speech, and Signal Processing*, pages 165–168. IEEE, 1994.

[9] D. Gaganelis. A novel approach to speaker verification. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 373–376. IEEE, 1991.

[10] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren. Darpa timit acoustic-phonetic continuous speech corpus cd-rom. In *National Institute of Standards and Technology*, 1990.

[11] J. Glass. *Finding acoustic regularities in speech: applications to phonetic recognition.* PhD thesis, Massachusetts Institute of Technology, 1988.

[12] J. Glass, J. Chang, and M. McCandless. A probabilistic framework for feature-based speech recognition. In *Proceedings of the 1996 International Conference on Spoken Language Processing*, 1996.

[13] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue. Galaxy: A human-language interface to on-line travel information. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, 1994.

[14] U. Goldstein. *An Investigation of Vowel Formant Tracks for Purposes of Speaker Identification.* PhD thesis, Massachusetts Institute of Technology, 1975.

[15] S. Hangai and K. Miyauchi. Speaker identification based on multipulse excitation and lpc vocal-tract model. In *Proceedings of the 1990 International Conference on Spoken Language Processing*, pages 1269–1272, 1990.

[16] C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. N-timit:a phonetically balanced, continuous speech, telephone bandwidth speech database. In *ICAASP*, pages 109–112, 1990.

[17] C. Jankowski, T. Quatieri, and D. Reynolds. Measuring fine structure in speech: Application to speaker identification. In *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*, pages 325–328. IEEE, 1995.

[18] I. Jou, S. Lee, and M. Lin. A neural network based speaker verification system. In *Proceedings of the 1990 International Conference on Spoken Language Processing*, pages 1273–1276, 1990.

[19] L. Lamel. A phone-based approach to non-linguistic speech feature identification. *Computer Speech and Language*, 9:87–103, 1995.

[20] R. Lippman. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 1987.

[21] M. Lund, C. Lee, and C. Lee. A distributed decision approach to speaker verification. In *Proceedings of the 1994 International Conference on Acoustics, Speech, and Signal Processing*, pages 141–144. IEEE, 1994.

[22] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4), April 1975.

[23] J. Makhoul, R. Schwartz, and G. Zavaliagkos. Adaptation algorithms for bbn's phonetically tied mixture system. In *ARPA SCSTW*, pages 82–87, 1995.

[24] T. Matsui and S. Furui. A text independent speaker recognition method robust against utterance variations. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 377–380. IEEE, 1991.

[25] T. Matsui and S. Furui. Comparison of text-independent speaker recognition methods using vq distortion and discrete/continuous hmms. *IEEE Transactions on Signal Processing*, 2(157), 1992.

[26] H. Meng. The use of distinctive features for automatic speech recognition. Master's thesis, Massachusetts Institute of Technology, 1991.

[27] J. Naik. Speaker verification:a tutorial. *IEEE Communications Magazine*, pages 42–47, January 1990.

[28] J. Oglesby and J. Mason. Radial basis function networks for speaker recognition. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 393–396. IEEE, 1991.

[29] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing.* PTR Prentice Hall, Inc., 1989.

[30] E. Parris and M. Carey. Discriminative phonemes for speaker identification. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 1843–1846, 1994.

[31] M. Phillips and V. Zue. Automatic discovery of acoustic measurements for phonetic classification. In *Proceedings of the 1992 International Conference on Spoken Language Processing*, pages 795–798, 1992.

[32] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[33] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition.* PTR Prentice Hall, Inc., 1993.

[34] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1):91–108, August 1995.

[35] Rosenburg, DeLong, Lee, Juang, and Soong. The use of cohort normalized scores for speaker verification. In *Proceedings of the 1992 International Conference on Spoken Language Processing*, pages 599–602, 1992.

[36] A. Rosenburg, C. Lee, and S. Gokcen. Connected word talker verification using whole word hidden markov models. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 381–384. IEEE, 1991.

[37] L. Rudasi and S. Zahorian. Text-independent talker identification with neural networks. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 389–392. IEEE, 1991.

[38] M. Sambur. *Speaker Recognition and Verification using Linear Prediction Analysis.* PhD thesis, Massachusetts Institute of Technology, 1972.

[39] Y. Tohkura. A weighted cepstral distance measure for speech recognition. *Computer Speech and Language*, 35:1414, 1987.

[40] H. Van Trees. *Detection, Estimation, and Modulation Theory (Part I).* John Wiley and Sons, Inc., 1968.

[41] J. Wolf. *Acoustic Measurements for Speaker Recognition.* PhD thesis, Massachusetts Institute of Technology, 1969.

[42] B. Yegnanarayana, S. Wagh, and S. Rajendra. A speaker verification system using prosodic features. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 1867–1870, 1994.

[43] H. Yin and T. Zhou. Speaker recognition using static and dynamic cepstral feature by a learning neural network. In *Proceedings of the 1990 International Conference on Spoken Language Processing*, pages 1277–1280, 1990.

[44] V. Zue, M. Phillips, and S. Seneff. The mit summit speech recognition system: A progress report. In *Proceedings DARPA Speech and Natural Language Workshop*, pages 179–189, 1989.