Text-Independent, Open-Set Speaker Recognition

THESIS

Stephen V. Pellissier
Captain, USA

AFIT/GE/ENG/96M-01

DTIC QUALITY INSPECTED 1

19961212 069

Text-Independent, Open-Set Speaker Recognition

THESIS

Stephen V. Pellissier
Captain, USA

AFIT/GE/ENG/96M-01

Text-Independent, Open-Set Speaker Recognition

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Stephen V. Pellissier, B.S. Electrical Engineering

Captain, USA

March 1996

Acknowledgements

Table of Contents

List of Figures

List of Tables

Abstract

Closed-set speaker recognition systems abound, and the overwhelming majority of research in speaker recognition in the past has been limited to this task. A realistically viable system must be capable of dealing with the open-set task. This effort attacks the open-set task, identifying the best features to use, and proposes the use of a fuzzy classifier followed by hypothesis testing as a model for text-independent, open-set speaker recognition.

Using the TIMIT corpus and Rome Laboratory's GREENFLAG tactical communications corpus, this thesis demonstrates that the proposed system succeeded in open-set speaker recognition. Considering the fact that extremely short utterances were used to train the system (compared to other closed-set speaker identification work), this system attained reasonable open-set classification error rates as low as 23% for TIMIT and 26% for GREENFLAG.

Feature analysis identified the liftered linear prediction cepstral coefficients with or without the normalized log energy or pitch appended as a robust feature set (based on the 17 feature sets considered), well suited for clean speech and speech degraded by tactical communications channels.

Finally, in contrast to previous efforts which have used codebooks consisting of 32–512 codewords, codebook analysis revealed that relatively small codebooks (with as few as 8–10 codewords) are adequate, if not optimal, in terms of classification accuracy and computational complexity for vector quantization-based classification techniques.

Text-Independent, Open-Set Speaker Recognition

I. Introduction

1.1 Motivation

Speaker recognition, like other biometric personal identification techniques (e.g. finger prints, retinal patterns, and face recognition), depends upon a person's intrinsic characteristics [15]. While speaker recognition has been applied in a variety of applications, such as: forensic science, controlling access to a secure facility, surveillance and intelligence gathering, conducting transactions by phone, labeling dictation, etc., its true usefulness has yet to be exploited. For example, when a terrorist calls in a bomb threat, wouldn't it be nice to automatically develop a ranked list of suspects? Or, when a drug kingpin is conducting "business" on the phone, wouldn't it be nice to identify the voice to a level of accuracy which could be used to prosecute and convict? The courts, which still rely predominantly on experts reading spectrograms [21] [23], could benefit greatly from more current methods. In general, as man's need to interact with machines continues to grow, so does the need for further research in speaker recognition.

1.2 Background

During the evolutionary process, humans have developed the capability to recognize and classify patterns. This capability, and the complexity involved, is often taken for granted since every day we encounter and classify (correctly, more often than not) thousands of different patterns. The speech signal conveys not only information about what was said, but also who said it, and humans, acting as the existence proof, can effectively accomplish both speech and speaker recognition. One logical extension is to use computers to recognize a speaker, and for decades researchers have pursued the idea of automated speaker recognition.

A closed-set speaker recognition system is constrained to a fixed population of speakers on which the system was trained, and various approaches have performed quite well for this task [11] [54] [56]. Considerably less work has been done for the open-set task. The open-set speaker recognition system must contend with speakers whom it has never "heard." In this manner, the open-set system must deal

1

with both the closed-set speaker identification task *and* a form of the speaker verification task. That is, not only must the system determine, from a population of speakers on which it was trained, the most likely speaker of an utterance, but it must also determine whether that utterance matches the speaker "close enough" (i.e. within an acceptable degree of tolerance). If the match is within tolerance, the classification is accepted; otherwise, it is rejected.

Can machines perform better than humans in this task? Atal [7] cites a speaker verification study conducted by Rosenberg in 1973, wherein a two second, all-voiced utterance was spoken by 40 speakers. Rosenberg found that human listeners achieved an accuracy of 96%, while an automated method using pitch, formant, and intensity data achieved 98% accuracy. While these results may lead one to believe that the problem was solved, Atal pointed out that the verification simply involved determining whether a pair of test and reference utterances were spoken by the same or different speakers, and he warned that such studies provide only a rough estimate of performance. The true difficulty of the task is noted by Nolan who claimed that no current speaker identification system is reliable and that absolute speaker recognition is not theoretically possible [40]. This inherent difficulty may contribute to the reason why the vast majority of speaker recognition systems built to date deal only with the closed-set task.

Since it is unrealistic for a system to be trained on *all* speakers, closed-set speaker recognition systems are extremely limited in real-world application. The only viable system is, therefore, the open-set speaker recognition system.

1.3 Problem Statement

1. Develop a text-independent, open-set speaker recognition system.

2. Identify the best features for a speaker recognition system.

1.4 Research Objectives

The best features, found by comparing 17 feature sets in terms of classification accuracy, will be used when testing the text-independent, open-set speaker recognition system in clean and noisy environments. Proper operation of the open-set system will entail rejecting initially mis-classified

utterances (e.g. out-of-set speakers' utterances) into an "Others" class. Based on the training limitation of using extremely short utterances, an acceptable error rate[1] will be 40%.

1.5 Scope and Assumptions

This effort focuses on text-independent, open-set speaker recognition as applied to the TIMIT corpus [1] and to Rome Laboratory's GREENFLAG tactical corpus [66]. The following list details the scope and assumptions which apply to this effort:

- Each corpora consists of speakers' utterances which were collected over a relatively short time span (i.e. days, rather than months).

- Speakers' utterances may be corrupted by noise (tactical communications channels for the GREENFLAG corpus).

- Channel effects and background noise are not directly removed from an utterance. That is, rather than attempting to extract only the voiced portion of the utterance, the entire utterance is used. If channel effects and background noise are speaker specific, they may actually be used to advantage, assisting in speaker identification. In such a situation, the system is actually performing a combination of speaker and channel recognition.

- One arbitrary utterance, 2-4 seconds in duration, is used to train the system on a speaker. By comparison to many speaker recognition systems, this training is extremely short; Reynolds and Rose, for example, used 30, 60, and 90 seconds of speech for training [55].

- The sentences used from the TIMIT corpus are the phonetically compact *sx* sentences. GREEN-FLAG utterances less than 0.5 seconds (which were primarily static) were discarded.

- Small speaker populations are used. The system is pre-initialized (or trained) with 10 speakers. Open-set classification tasks include utterances from the 10 speakers on which it was trained plus new speakers' utterances (typically five new speakers).

- The features considered here are based on common by-frame analysis feature extraction methods which have been successfully used in both speaker and speech recognition.

---

[1]The sponsor of this research, the U.S. Army Communications-Electronics Command (USACECOM), Fort Monmouth, NJ, indicated that a system with this error rate would be serviceable. These research objectives and the list of Scope and Assumptions (Section 1.5) partially support system requirements for USACECOM.

1.6 Approach/Methodology

This research consists of two major goals. The first goal is to determine the optimal or best set of features (among the group of commonly used features summarized in Figure 1) for a speaker recognition system. The second goal, supported by the first, is to develop a text-independent, open-set speaker recognition system. The approach is as follows:

1. To find those feature subsets which provide optimal (in terms of classification accuracy) discrimination amongst speakers, the extracted features are subjected to a closed-set speaker identification system. A crisp nearest cluster classifier, with the codebooks formed using the Linde, Buzo, and Gray (LBG) Algorithm [35], is used to classify the speakers' utterances. Feature optimality is based on the minimal classification error.

2. The open-set speaker recognition system (see Figure 1) consists of an initial training phase, followed by an operational testing phase.

   - **Training.** Features are extracted from each frame of a speaker's training utterance, and the feature vectors are then vector quantized to produce the speaker's codebook. These same training feature vectors are also subjected to the fuzzy classifier, and each frame is classified based on a maximum membership function value. A majority voting scheme of the frames is then used to classify the utterance, and those by-frame membership function values corresponding to the winning speaker are stored as that speaker's reference membership function values ($refU$). When this has been done for all training utterances, the system is in its initialized state.

   - **Testing.** When a test utterance is presented, the features are extracted, and the fuzzy classifier determines the most likely speaker based on a by-frame majority voting scheme. The membership function values corresponding to the winning speaker ($testU$) are then statistically compared to the speaker's reference membership function values ($refU$) for the final decision of whether to accept or reject the classification.

1.7 Thesis Organization

This chapter has identified the need for more work in open-set speaker recognition and defined the scope and goals of this research effort. Chapter II provides background information on the techniques

FEATURES:
- Linear Prediction Cepstral Coefficients
- Reflection Coefficients
- Mel-Frequency Cepstral Coefficients
- RelAtive SpecTrA Coefficients
- Delta Coefficients
- Acceleration Coefficients
- Pitch
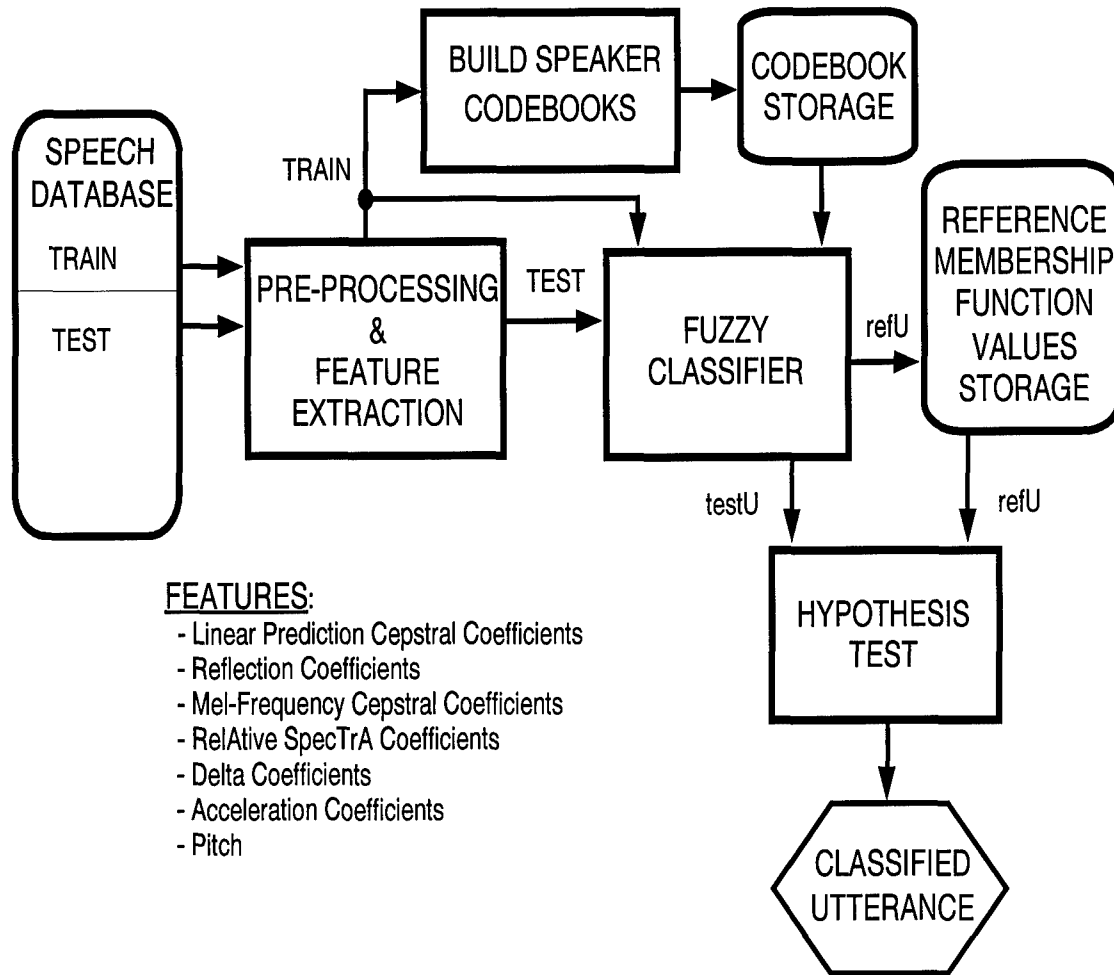
Figure 1. Open-Set Speaker Recognition System Overview.

used to accomplish the goals, focusing on feature extraction, classification, and hypothesis testing, and includes a literature review of significant past research in closed-set speaker recognition. Chapter III describes the methodology and experimental procedures used in this effort. Chapter IV presents and discusses the results obtained from the experiments. Chapter V provides a summary of the results and the conclusions of this research. Appendix A provides additional background in support of Chapters II and III. It may be helpful for the novice in speaker recognition to review this appendix first. Appendix B provides additional experimental results. Appendix C provides the results of baseline testing (in terms of closed-set speaker identification) to justify the use of the by-frame majority voting method for utterance classification.

5

II. Background and Literature Review

2.1 Introduction

This chapter examines background information relevant to this effort. It first provides a background of speech analysis, signal processing, pattern recognition, and statistical inference techniques applicable to open-set speaker recognition. Feature extraction and classification (based on clustering analysis) techniques are initially discussed. Additional details for these topics are provided in Appendix A. Next, a review of the statistical inference methods used for hypothesis testing is provided. This chapter concludes with a brief review of some of the more pertinent work in closed-set speaker recognition. (Overwhelmingly, the literature available specific to speaker recognition deals only with the closed-set task.)

2.2 Feature Extraction

Feature extraction, in terms of speaker recognition, is the process of creating a compact set of parameters characteristic of a speaker. The goal is to preserve information relevant to the speaker's identity, while producing minimal intra-speaker variance and maximal inter-speaker variance. Many of the techniques discussed below have been successfully applied in closed-set speaker and speech (or word) recognition applications.

2.2.1 Linear Prediction Analysis. Linear prediction (LP) is a procedure for encoding the speech signal by representing it in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the excitation [6] [36]. The application of LP analysis derives a set of predictor coefficients, obtained by minimizing the total squared error, E, between the actual signal value and its predicted value [36]. The predictor coefficients (or reflection coefficients, derived intermediately) represent the combined information about the formant frequencies, their bandwidth, and the glottal wave [7].

2.2.2 Cepstral Analysis. In recent years, the cepstrum (particularly, the real cepstrum) has gained widespread use in successful speech and speaker recognition systems. The cepstrum of a signal is the Fourier transform of the logarithm of its magnitude spectrum. The uniqueness of the cepstrum (see Figure 13, page 52) is that it provides a means to separate the speech signal's two components:

6

the slowly varying spectral envelope and the rapidly varying pitch harmonic peaks [13]. Atal [7] found the cepstrum function, derived from linear prediction, to be a most effective feature (compared to the predictor coefficients, the impulse response of the all-pole filter, the autocorrelation function, and the area function) for speaker recognition, with an accuracy nearly 7% greater than the next closest feature (the predictor coefficients).

2.2.2.1 Mel-Warped Cepstra. Linear prediction cepstrum coefficients (LPCCs), generated from the LP spectrum and distributed along a linear frequency axis, form a less than optimal representation of an auditory signal since a logarithmic function of frequency better approximates the perception of the human ear to frequencies [33]. The mel or Bark scale is often used to approximate the resolution of the human auditory system [13] [46]. To obtain mel-frequency cepstral coefficients (MFCCs), the magnitude spectrum is mel-scaled prior to taking the FFT to obtain the cepstral coefficients [22]. The mel-scaling can be accomplished either by applying a simulated mel-scaled triangular filter bank (shown in Figure 14, page 54) or by applying a bilinear transform (given by Equation 17, page 53, and illustrated in Figure 15, page 55). Davis and Mermelstein [12] compared MFCCs generated from the filter bank approach to the linear frequency cepstral coefficients (computed from the log magnitude of the discrete Fourier transform) and the LPCC (computed from the linear prediction coefficients) and found that the MFCCs performed best for word recognition.

2.2.2.2 Transitional Coefficients. Delta and acceleration coefficients, obtained via linear regression techniques, from the cepstral coefficients provide temporal information and information about the spectral changes from frame to frame. Fenstermacher and Smith [18] found these coefficients to be useful for speaker identification. Furui [20] preferred the polynomial approximation, showing that a first-order polynomial characterization of spectral change is adequate. Soong and Rosenberg [62] found that while instantaneous features carry more speaker relevant information, the transitional features are less affected by the transmission channel. Also, the instantaneous linear prediction coefficients and their delta coefficients are relatively uncorrelated and can be used together in order to improve speaker recognition accuracy. Lee [33], on the other hand, found that the formant slopes are relatively invariant across speakers.

2.2.2.3 Liftering.   By applying a window function $w(k)$ to the cepstral coefficients, liftering reduces or removes undesirable components (noise), while retaining the essential characteristics of the formants [26]. It essentially accounts for the sensitivity of the low-order cepstral coefficients to the overall spectral slope and the high-order cepstral coefficients to the noise [5]. In doing so, liftering attempts to minimize the mismatch between speech collected under different environments and/or from different communications channels. Juang *et al* [26] found increased recognition accuracy in isolated digit recognition using the raised sinusoid window function of Equation 1 to lifter the cepstral coefficients;

$$w(k) = 1 + \frac{L}{2} \sin \frac{\pi k}{L} \tag{1}$$

where k $(1 \leq k \leq L)$ is the index of the cepstral coefficients.

2.2.2.4 RASTA.   Similar to liftering, the RelAtive SpecTrA (RASTA) process is a means of reducing the cepstral coefficients' sensitivity to noise [24] [25]. Hermansky *et al* [24] describe the RASTA process as the equivalent to bandpass filtering each channel (i.e. each index of the cepstral coefficients over all the feature vectors) through an IIR filter with a transfer function given by:

$$H(z) = \frac{0.1(2.1 + 1.0z^{-1} - 1.0z^{-1} - 2.0z^{-1})}{z^{-1}(1 - 0.98z^{-1})} \tag{2}$$

The high-pass portion of the filter described by Equation 2 pacifies the effect of convolutional noise introduced by the communications channel, while the low-pass filtering smoothes the frame-to-frame spectral changes [24]. In addition to compensating for the time-varying channel bias, RASTA processing also removes the global mean of the feature vectors [55]. Many have found improved performance by applying the RASTA process as described in Equation 2 [27] [41] [56].

2.2.3 Fundamental Frequency.   The fundamental frequency or pitch is robust in transmission and resists distortion by telephone and similar channels [7] [40]; however it has often been disregarded since it varies with respect to a speaker's stress, emotion, and intonation. Still, the pitch is constrained by the physics of a speaker's larynx [13]. It is uncorrelated to the information conveyed by the cepstrum [20] and independent of the predictor coefficients [7]; thus, it is a viable feature which deserves consideration.

2.3 Classification

The goal in speaker recognition is for the system to make an accurate, reliable decision of an unknown speaker's identity. Classification, the final stage in a closed-set pattern recognition system, is the decision-based process in which the system chooses the most probable or closest matching class, based on a minimum distortion measure or maximum probability. The distortion measure is commonly the distance measured between two templates or models, such as the Euclidean (which is appropriate for cepstral coefficients [49]).

Due to the limited amount of training data used in this effort (see Section 1.5), a vector quantization (VQ) based classification approach was chosen over other, more popular approaches, such as Hidden Markov Models (HMMs). When less training data are available, Matsui and Furui [38] found that the HMM parameters are not well estimated and that a VQ based method is less adversely affected.

2.3.1 Vector Quantization.

Vector quantizers are often designed using the Generalized Lloyd Algorithm, which is described by Linde, Buzo, and Gray [35] and commonly referred to as the LBG Algorithm. The cluster centers (a.k.a. codewords) are found by an iterative method which terminates in a local minimum when the average distortion (based on the distance from the cluster centers to the data points within the clusters) stops changing significantly.

Ideally, the feature space consists of small clusters (i.e. with small variance) each formed by repetitions of the features taken from a speaker's utterance, with the different speakers' clusters widely separated. Using the codebooks (the set of a speaker's codewords) created by vector quantization, the pattern classifier needs only to compare the test samples to the representative codewords, rather than the entire training set of data for classification. Thus, classification entails finding the minimum distortion between an unknown test speaker's utterance and the set of reference speakers' codebooks.

2.3.1.1 Codebook Initialization.

A variety of codebook initialization schemes have been investigated in hopes of providing accelerated convergence of the LBG Algorithm, achieving a better local minimum, and providing flexibility (in terms of the number of cluster centers). Linde *et al* [35] suggest a splitting method, whereby the LBG Algorithm is applied at each power of two (giving codebook sizes of 1, 2, 4, 8, 16...). Katsavounidis *et al* [28] proposed a maxi-min method, while

9

recently, DeSimio *et al* [14] proposed a Karhunen-Loève initialization scheme, whereby the cluster centers are placed along the principal component axes of the training data's covariance matrix.

2.3.1.2 Codebook Size. Codebook sizes often seem to be an arbitrarily chosen number of codewords. Ramachandran *et al* [51], for example, simply used a codebook size of 32 codewords, while Assaleh and Mammone [5] based their codebook size on the number of phonemes and used 46 codewords. Matsui and Furui [38] analyzed codebook sizes of 32, 64, 128, 256, and 512, but found little improvement in speaker identification rate above 64 codewords.

Increasing the number of codewords, N, significantly increases the computational complexity, not only for building codebooks, but also for classification of utterances. To build a codebook, for example, for M training vectors (or frames of speech), each iteration requires M distance calculations, resulting in a complexity of $O(MN)$ for one Lloyd iteration [28]. Table 1 illustrates the computational complexity involved in building codebooks then classifying speakers' utterances (closed-set speaker identification) for different values of N. The complexity is measured by the number of floating point operations (flops) required. Ten GREENFLAG speakers were chosen for this example and the proposed speaker recognition system[1] was used to first build the speakers' codebooks using one utterance per speaker, then classify the remaining 46 utterances. As shown, the number of flops required for both building the codebooks and for classifying the utterances increases dramatically as N increases. Moreover, an improvement in the classification error rate for $N = 8$ does not occur until $N = 128$, clearly indicating that the cost of increased computational complexity for improved accuracy may not be worthwhile. In light of these results, prudence suggests a conservative approach, in which fewer codewords may in fact be preferable, when choosing the codebook size.

2.3.2 Fuzzy Logic Techniques. Similar to the LBG algorithm, the fuzzy c-means algorithm incorporates the calculation of the degree of class membership for an evaluated input data vector [9]. The membership function value is defined by [9] [29]:

$$U_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{(x_k - v_i)^T A (x_k - v_i)}{(x_k - v_j)^T A (x_k - v_j)} \right)^{\frac{2}{m-1}}} \tag{3}$$

---

[1]The proposed speaker recognition system will be described in detail in Section 3.5, and a full understanding of it is irrelevant to the present discussion. The focus here is to illustrate the computational complexity with respect to different codebook sizes.

Table 1. Computational Complexity, in terms of floating point operations (flops), involved in Building Codebooks and Classifying speakers' utterances. Ten GREENFLAG speakers were applied to the system described in Section 3.5, operating in a closed-set mode, to generate these results. One utterance per speaker was used to build his codebook, and 46 test utterances were classified. As shown, computational complexity increases by orders of magnitude for increasing N, and it is not until $N = 128$ that the error is corrected.

# of Codewords N	Build Codebooks (Megaflops)	Classify (Megaflops)	Pr(Error)
8	16.6	412	0.02
16	34.9	1,486	0.07
32	58.7	5,652	0.02
64	116.7	22,057	0.02
128	198.4	87,154	0.00

where $0 \leq U_{ik} \leq 1$ for all i, k. For c classes, x represents the data (or feature vector) and is indexed by k, v is a vector of the cluster centers and the cluster centers for each class are indexed by i, A is the identity matrix for Euclidean distance, and m represents the degree of fuzziness ($m > 1$, increasing m increases the fuzziness).

Equation 3 can be used either for designing a fuzzy codebook or for classification based on the maximum membership function value (minimal distance $\Rightarrow$ maximal membership). For fuzzy logic classification, the decision is based on the maximum membership function value. For a small (three speaker), Telugu (a Dravidian language spoken in southern India) corpus, Pal and Majumder [44] achieved a 97% speaker identification accuracy using a fuzzy set classification technique.

2.3.3 Feature Vector Size. Simply concatenating features results in extremely large numbers of free parameters within the classifier. With regard to classification, it is desirable to limit the number of elements in a feature vector since a large number of free parameters in the classifier may cause the classifier to "memorize" the training data, resulting in degraded performance with previously unseen test data [17] [60]. Multiple codebooks of relatively small dimensional feature vectors can serve as an alternative to a single codebook of concatenated feature vectors. Classification using multiple codebooks can be achieved by either a simple voting method (in which each individual classification result is equally weighted) or a weighted voting method [56] [62].

11

Table 2. Decision Table for Hypothesis Testing

STATE	DECISION	
	Accept H_0	Reject H_0
H_0 is TRUE	Correct Decision	False Reject (Type I Error)
H_0 is FALSE	False Accept (Type II Error)	Correct Decision

2.4 Hypothesis Testing

The open-set task requires a means of determining whether the results of classification are "close enough." Hypothesis testing is one method to accomplish this task. Hypothesis testing, as referred to in this work, is a form of statistical inference which involves comparing two unknown populations, with two complementary outcomes: the null hypothesis H_0 and the alternative hypothesis H_1 [31]. A determination must be made whether to accept or reject the null hypothesis. The testing procedure involves observing a computed test statistic, which is a random variable, and deciding which hypothesis to accept [16]. Two types of errors may occur, with probabilities defined in Equations 4 and 5 and explained in Table 2.

$$\alpha = \text{Pr(Reject } H_0 | H_0 \text{ True)} \tag{4}$$

$$\beta = \text{Pr(Accept } H_0 | H_0 \text{ False)} \tag{5}$$

The challenge in hypothesis testing is in finding an acceptable balance between the two types of error. Finding this balance begins by varying α over a range of values. For each value of α, the critical statistic x_α is computed via optimization techniques, which numerically integrate the distribution (see, for example, Figure 2) at different values of x until the result equals α. Figure 2(a) shows the F-distribution, used for analysis of variance (ANOVA), with four degrees of freedom in the numerator (DoF1) and eight degrees of freedom in the denominator (DoF2). Alpha (α) is the shaded area under the tail of the curve. For values of $x \leq x_\alpha$ the null hypothesis is accepted, while for values of $x > x_\alpha$ the null hypothesis is rejected. Figure 2(b) shows that when $DoF1 = 1$, the F-distribution is not bounded and numerical integration cannot be accurately accomplished.
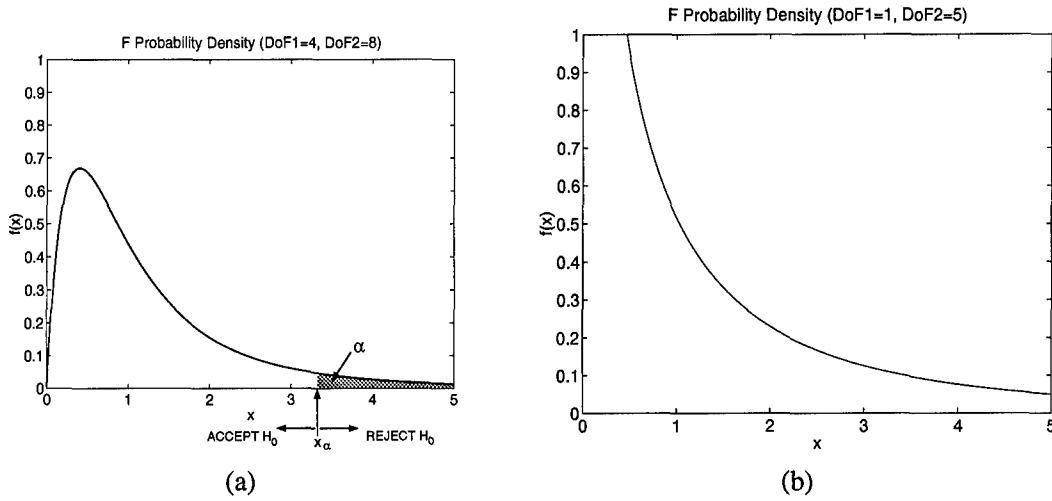
Figure 2. The F-Distribution, (a) shows α and the regions where H_0 is accepted and rejected; (b) shows the F-distribution when $DoF1 = 1$.

2.4.1 The Smirnov Two-Sample Test. In non-parametric (or distribution-free) tests, the underlying population variables are not assumed to be normally distributed, as in parametric tests such as the ANOVA and the Chi-Square Goodness of Fit Tests. Free of such assumptions, non-parametric tests can be more robust and are often fast and efficient to implement [16].

The Smirnov Test for Common Distributions is similar to the Kolmogorov-Smirnov Test, which tests whether a set of observations is from a normal population [34] [45], except that it is a non-parametric test. The null hypothesis for the Smirnov Test is that the two populations have the same distribution. By comparing the sample cumulative distribution functions, the test statistic, S, is found as the difference of greatest magnitude between the two cumulative sample distributions [63]:

$$S = \sup_{x} |F_1(x) - F_2(x)|. \tag{6}$$

By using the absolute value in defining S, the test is two-sided. The equation for the value of the critical test statistic, S_α, is given by [68]:

$$S_\alpha = \lambda_\alpha \sqrt{\frac{1}{N_1} + \frac{1}{N_2}} \tag{7}$$

13

where λ_α satisfies

$$\sum_{i=-\infty}^{\infty} (-1)^i e^{-2i^2 \lambda_\alpha^2} = 1 - \alpha \tag{8}$$

and N_1 and N_2 are the sizes of the two populations, with $N_1, N_2 \to \infty$ so that $\frac{N_2}{N_1} \to \rho > 0$. Also, Equation 7 does not require that $N_1 = N_2$.

In sum, the Smirnov Test considers the shapes of the distribution, not just their means and variances [37]. Large values of S are evidence against the null hypothesis, and the null hypothesis is rejected when $S > S_\alpha$.

2.5 Closed-Set Speaker Recognition

This section briefly reviews some of the more significant, related work reported in closed-set speaker recognition. It is included because the closed-set task, while simpler, is inherently related to the open-set task. Basztura [8] is one of the few who has confronted the open-set task using error and risk probability analysis connected with Bayesian decision criterion for selecting a discrimination threshold and approximating the conditional distributions of unknown voices. He obtained overall classification error rates of approximately 5% and 15% for text-dependent experiments conducted on small and large populations (10 closed set and 10 out-of-set speakers, and a hold-out method for 100 speakers, respectively).

Reynolds [53] used mel-frequency cepstral coefficients (mel-scaled via triangular filter bank) and a Gaussian Mixture Model (GMM) classifier for large population (all 630 TIMIT speakers) text-independent, closed-set speaker identification, obtaining 99.5% accuracy.

Ricart *et al* [56] applied a speaker recognition system to Rome Laboratory's tactical GREEN-FLAG database. This speaker identification system used on-line training and incorporated both feature set fusion and classifier fusion. The feature sets consisted of 14th order liftered LP cepstra, RASTA liftered cepstra, delta cepstra, and acceleration cepstra. The fused classifier techniques were LBG vector quantization, multi-layer perceptron, and k-nearest neighbor. Ricart *et al* concluded that combining the results of the different feature sets and classifiers produced significant increases in performance, with the best results (93% accuracy) obtained from a feature combination of the liftered cepstra and delta cepstra.

14

2.6 Conclusion

This chapter provided background information and explored much of the current literature in speaker recognition. The fact that most speaker identification literature deals only with the closed-set task supports the need for further work in open-set speaker recognition. The next chapter describes the methodology used in this effort to find the best set of features and to attack the open-set problem.

III. Methodology

3.1 Introduction

This effort focuses on text-independent, open-set speaker recognition as applied to the TIMIT corpus and to Rome Laboratory's GREENFLAG tactical corpus. Speakers' utterances are text-independent (i.e. not constrained by the text spoken) and may be corrupted by noise (tactical communications channels for the GREENFLAG corpus). The goals in this effort are to identify the best feature set and to develop a text-independent, open-set speaker recognition system. This effort proposes fuzzy classification followed by hypothesis testing for open-set speaker recognition.

This chapter is organized as follows: Speech processing techniques generic to all work conducted are first defined. Again, Appendix A contains additional information. Next, the methodology followed to find the best set of features and the optimal codebook size for a speaker identification system is described. Finally, the approach used to develop the open-set speaker recognition system is described.

3.2 Speech Processing

3.2.1 Pre-Processing.
Pre-processing entails those measures taken to prepare the speech signal for analysis. A pre-emphasis filter, $P(z) = 1 - 0.97z^{-1}$, is applied to the digitized utterance to increase the relative energy of the high frequency spectrum. Then, a 20 ms Hamming window, whose spectral sidelobes are attenuated by 30 dB, is applied every 10 ms. The overlapping frames overcome the shortfalls from window edges.

3.2.2 Feature Extraction.
In terms of speaker recognition, feature extraction is the process of creating a compact set of parameters characteristic of a speaker. The goal is to preserve information relevant to speaker's identity, with minimal intra-speaker variance and maximal inter-speaker variance.

The basic feature vectors consist of: 12th order linear prediction cepstral coefficients (LPCEPSTRA), 12th order mel-frequency cepstral coefficients (MFCC), and 12th order reflection coefficients (LPREFC). The LP model order is 24, and the triangular filter bank approach is used for obtaining the MFCC. Liftering (see Equation 1) is applied to the LPCEPSTRA and MFCC features, with $L = 22$. Pitch (F0) is extracted and examined separately; however, it or the normalized log energy (E) may be appended to the basic features. Delta (D) and acceleration (A) coefficients are obtained from the

16

basic feature vectors via linear regression techniques. The RASTA features are obtained from the LPCEPSTRA by applying the filter given in Equation 2 to each channel.

Table 3 summarizes the features examined in this effort. The appended feature sets are those wherein either the normalized log energy or pitch are appended to the basic features. For example, LPCEPSTRA_E consists of the 12 LPCEPSTRA features, appended with the normalized log energy (giving 13 elements in each feature vector). The derived feature sets consist of the 12 delta or acceleration coefficients for each basic feature and the 12 RASTA features derived from the LPCEPSTRA. Pitch is in the Basic Feature column since it was extracted separately.

Table 3. Summary of the 17 Features.

Basic Features	Appended Feature Sets	Derived Feature Sets
LPCEPSTRA	LPCEPSTRA_E LPCEPSTRA_F0	LPCEPSTRA_D LPCEPSTRA_A RASTA
LPREFC	LPREFC_E LPREFC_F0	LPREFC_D LPREFC_A
MFCC	MFCC_E MFCC_F0	MFCC_D MFCC_A
F0	-	-

3.2.3 Classification. The goal in speaker recognition is to make an accurate, reliable decision of an unknown speaker's identity. In general, classification of a test pattern is based on a minimum distortion measure, or as in the case of a fuzzy classifier, a maximum membership function value. Throughout this work, the Euclidean distance is used to calculate distortions. The open-set task further requires that following the classification, the system must determine whether to accept or reject the classification.

17

3.3 Feature Analysis

An exhaustive search of all possible features is impractical; therefore, only those features shown in Table 3 were considered in searching for the best features. The goal was to find those feature sets which provide optimal discrimination amongst speakers, with optimality defined in terms of minimal classification error rates. LNKnet was used to accomplish this closed-set speaker identification task. The k-means algorithm was used for finding the cluster centers to initialize the nearest cluster classifier. Classification entailed finding the minimum distortion (on a frame-by-frame basis) between an unknown test utterance and the set of reference speakers' codebooks. The best features were those which provided minimal classification error.

3.4 Codebook Analysis

In addition to finding the best features, the analysis described above also lended itself to the determination of the optimal codebook size. Three factors were considered in this clustering analysis: classification error rate, computational complexity, and cluster distortion. The most important factor for this application was classification error rate. The number of codewords at which the classification error stops decreasing and may begin to diverge (i.e. when the classifier is "memorizing" the training data) indicates the largest acceptable codebook size. To limit the computational complexity involved, the least number of codewords which provided an acceptable (yet stable) classification error was optimal. In terms of cluster distortion, the number of codewords at which the change in the cluster distortion becomes insignificant, indicates an optimal number of codewords.

3.5 The Open-Set Task

Recalling the brief overview in Section 1.6, the open-set speaker recognition system (see Figure 1) is initialized during training. This training entails creating the reference speakers' codebooks and obtaining the reference membership function values used for hypothesis testing. Once training is completed, the system is ready for normal operation.

3.5.1 Building The Codebooks.
For the open-set task, clusters of the training samples (the speaker's codebook) are formed using the Karhunen-Loève initialization [14], followed by the LBG Algorithm [35]. The Karhunen-Loève initialization was chosen to initialize the codebooks

because it disperses the desired number of codewords along the principal component axes of the data's covariance matrix, allowing simple, efficient implementation of *any* number of codewords. Figure 3 provides a two-dimensional example of building codebooks for four classes of synthetically generated data. Figure 3(a) shows how the Karhunen-Loève initialization disperses the codewords (symbolized as "x,o,*,+") amongst the data along the principal component axes. Figure 3(b) shows the final codebooks after applying the LBG Algorithm. The pattern classifier needs only to compare (on a frame-by-frame basis) the test utterance's features to the representative codewords.
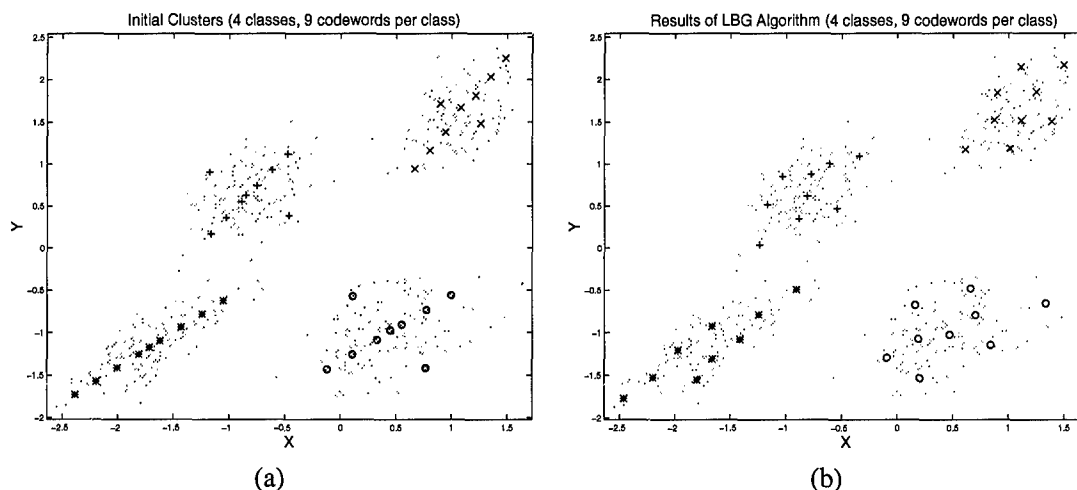
Figure 3.   Creating codebooks for four classes of two-dimensional synthetically generated data, using (a) the Karhunen-Loève initialization, followed by the LBG Algorithm resulting in (b). (The codewords representing the four classes are shown as "x,o,*,+.")

3.5.2 Fuzzy Classification.   Fuzzy classification, which is used for the open-set task, considers the degree of class membership for each test frame. The results of baseline testing (see Appendix C) substantiate the use of the by-frame majority voting scheme described below.

The class membership function value (U), computed with Equation 3, provides a measure of similarity by which a soft decision can be made with a degree of confidence. Throughout this work, the degree of fuzziness (m) is $m = 2$ [29]. Each frame is classified, based on the maximum membership function value. Table 4(a) shows the by-frame results of closed-set speaker identification of 10 GREENFLAG speakers, one test utterance per speaker, using a fuzzy classifier (with eight codewords) and the LPREFC feature set.

Table 4. (a) By-Frame Confusion Matrix for Closed-Set Speaker Identification for 10 GREENFLAG speakers using a Fuzzy Classifier and the LPREFC feature set. (b) By-Utterance Confusion Matrix obtained from (a). The by-utterance classification error (0.10) is now more clearly observable.

ACTUAL SPEAKER	COMPUTED SPEAKER									
-	0	1	2	3	4	5	6	7	8	9
0	48	3	1	9	2	4	.	2	7	.
1	7	196	10	15	1	2	10	27	26	4
2	4	19	239	52	77	91	89	136	92	39
3	2	21	3	32	11	11	3	7	14	2
4	4	4	2	41	81	15	23	18	21	10
5	6	8	142	60	62	261	66	12	38	38
6	2	9	38	5	12	44	39	32	5	4
7	.	4	4	9	9	1	3	74	11	3
8	12	23	25	42	16	37	38	44	106	6
9	6	2	19	37	44	27	8	28	65	176

(a)

ACTUAL SPEAKER	COMPUTED SPEAKER									
-	0	1	2	3	4	5	6	7	8	9
0	1	.	.	.	.	.	.	.	.	.
1	.	1	.	.	.	.	.	.	.	.
2	.	.	1	.	.	.	.	.	.	.
3	.	.	.	1	.	.	.	.	.	.
4	.	.	.	.	1	.	.	.	.	.
5	.	.	.	.	.	1	.	.	.	.
6	.	.	.	.	.	1	.	.	.	.
7	.	.	.	.	.	.	.	1	.	.
8	.	.	.	.	.	.	.	.	1	.
9	.	.	.	.	.	.	.	.	.	1

(b)

Each utterance is classified as belonging to the speaker to whom the majority of the frames are classified, and the membership function values associated with those winning frames are used in hypothesis testing. For example, Table 4(a) shows that the majority of $Speaker_1$'s frames (196) were classified as belonging to $Speaker_1$. The membership function values associated with those 196 frames are then used for hypothesis testing (the others are discarded) to determine whether that utterance classification should be accepted.

When classifying multiple utterances per speaker, it is more common to display the by-utterance classification results (which will be used henceforth). Table 4(b) shows the by-utterance results obtained by summarizing the results of Table 4(a). For this example, the by-frame classification error is 0.62, and the by-utterance classification error, obtained by the majority voting scheme of the by-frame classifications, is 0.10.

3.5.3 Hypothesis Testing. The Smirnov Test for Common Distributions establishes whether or not two populations of sample data have the same distribution. In this case, the two populations are the reference[1] and test membership function values ($refU$ and $testU$, respectively). Thus, the null hypothesis is that $refU$ and $testU$ have the same distribution.

Prior to implementing the Smirnov Test, the sample populations ($refU$ and $testU$) are pre-processed with a histogram "optimization" technique. This histogram optimization ensures that each

---

[1]Recall that the reference membership function values are obtained by classifying the training utterances.

Table 5. Histogram Optimization. Prior to implementing the Smirnov Test, the $refU$ and $testU$ sample populations are pre-processed as illustrated in this simple example to remove outlier bin locations. The population is initially distributed in bin locations b_i with frequency f_i. Bins are combined (as indicated by the "}") to ensure $f_i \geq 5$, resulting in a new distribution given by $\hat{b}_i$ and $\hat{f}_i$.

BEFORE		AFTER	
b_i	f_i	$\hat{b}_i$	$\hat{f}_i$
0.1	1 ⎫	.	.
0.2	0 ⎬	.	.
0.3	4 ⎭	0.3	5
0.4	6	0.4	6
0.5	12	0.5	12
0.6	15	0.6	15
0.7	8	0.7	8
0.8	10	0.8	10
0.9	2 ⎫	0.9	5
1.0	3 ⎭	.	.

bin location contains at least five entries; thereby removing "outliers" by effectively smoothing the histogram and eliminating the need for arbitrarily choosing a threshold value for the membership function values. The process is basically identical to the pre-processing techniques described by Lapin [31] for the Chi-Square Goodness of Fit Test. Afifi and Azen [4] recommend at least five entries per bin location. If, for example, a population is initially distributed in bin locations b_i with frequency f_i as shown in Table 5, the histogram optimization will combine the outliers to ensure the frequency in each bin is at least five, resulting in $\hat{b}_i$ and $\hat{f}_i$. The optimized histogram bin locations are then the values to which the Smirnov Test is applied. The sample cumulative distribution functions of the pre-processed $refU$ and $testU$ are compared, using Equation 6 to obtain S.

Based on the value for the Smirnov significance level, α, S_α is computed using Equation 7 and compared to the calculated value of S. Figure 4 illustrates the Smirnov Test values of S and S_α for an arbitrarily chosen speaker's utterance. If $S \leq S_\alpha$, the classification is accepted, while if $S > S_\alpha$, the classification is rejected. Ideally, when the classifier correctly classifies an utterance, the null hypothesis is always accepted (i.e. $S \leq S_\alpha$ over the range of α). On the other hand, all out-of-class speakers' utterances should be rejected into an "Others" class. Thus, in addition to the

two types of error (false acceptance and false rejection), it is possible to have a correct rejection. This occurs when the classifier mis-classifies the utterance, but the hypothesis testing correctly rejects the classification. Since the classifier must determine the most likely speaker, it will classify (correctly or incorrectly) each utterance; thus, the hypothesis test's capability to correctly reject the classification is of vital importance to accomplishing the open-set task.
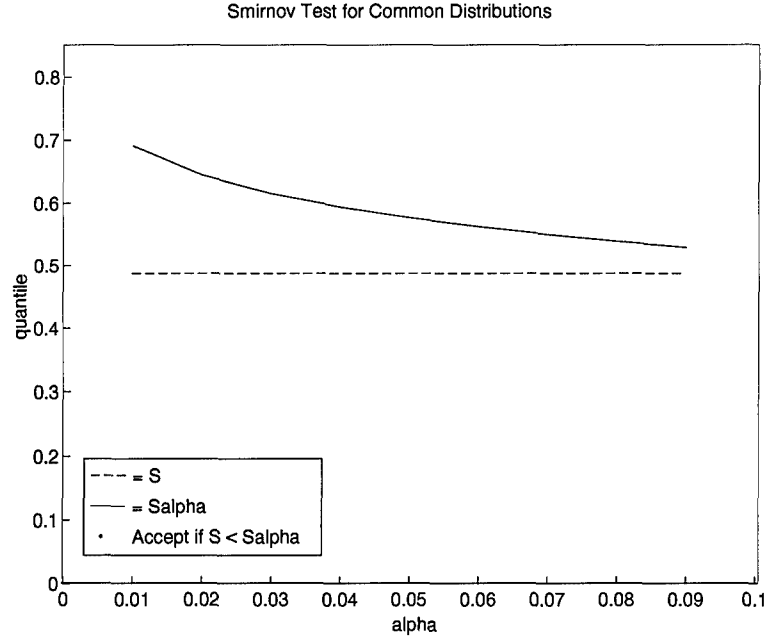
Figure 4. The Smirnov Test values S and S_α for an arbitrary speaker's utterance. While the calculated value of S remains constant, S_α varies. When $S \leq S_\alpha$ the classification is accepted; otherwise, it is rejected.

3.5.4 Performance Measure. After hypothesis testing, overall performance of the system is described by the final classification error rate, $\Pr(Error)$, for a given value of α. Often $\Pr(Error)$ is defined only as a function of the probability of false acceptances and the probability of false rejections. Note, however, that such an approach does not take into account the rate of correct rejections. To do so here, the accuracy, $\Pr(Correct)$, is found by summing all correct classifications (i.e. summing along the confusion matrix diagonal for the closed-set utterances and summing the "Others" class for the out-of-set utterances) and dividing by the number of utterances tested. Thus, the final classification error rate is given by:

$$\Pr(Error) = 1 - \Pr(Correct). \tag{9}$$

3.6 Conclusion

This chapter described the methodology used to find the best features and accomplish the open-set task. A description of the pre-processing, feature extraction, and classification techniques necessary to accomplish each requirement was provided. The next chapter provides the results of the analysis performed.

IV. Results

4.1 Introduction

This chapter presents the experimental results of the methodology described in Chapter III. The results of the search for the best features and the optimal codebook size for the TIMIT and GREENFLAG corpora are first provided. The proposed text-independent, open-set speaker recognition system is then tested using those features found to be best suited for the task.

4.2 Feature Analysis

As discussed in Section 3.3, the best features would be those that provided minimal classification error. The experiments to find the best features used LNKnet's nearest cluster classifier on the feature sets shown in Table 3. Ten TIMIT speakers from Dialect Region 8 and 10 arbitrarily chosen GREENFLAG speakers were used, resulting in 40 and 53 test utterances from each respective corpus. The number of codewords ranged from 1–50 for this analysis.

The results of this feature analysis are summarized in Table 6 and Figure 5. Table 6 shows a ranked order list (ranked according to averaged classification error over the range of 1–50 codewords) for all the features, and Figure 5 further illustrates the averaged classification error and standard deviation for each feature, with the ordinate labeled according to the ranking in Table 6. Thus, based on the minimum averaged closed-set speaker identification error rate, the best features for the TIMIT corpus are the LPCEPSTRA_E, while for the GREENFLAG corpus, the best features are the LPREFC_E. For plotted results showing the classification error versus the number of codewords for all features (from which the mean and standard deviation were calculated), see Appendix B.

4.2.1 Observations.

- As shown in Figure 5, there is very little difference in classification performance for the top three features for TIMIT and the top five features for GREENFLAG. In fact, the difference is not statistically significant, based on ANOVA tests at a significance level of 0.05, with the null hypothesis defined as the means being equal. The details of these ANOVA tests are shown in Table 7.

24

Table 6. Results of Feature Analysis. Ranking the features according to the mean error rate (over the range of 1–50 codewords). As shown, LPCEPSTRA_E are best for TIMIT, and LPREFC_E are best for GREENFLAG. In general, LPCEPSTRA, appended or not, perform well.

| | TIMIT | | | GREENFLAG | | |
| | | Pr(Error) | | | Pr(Error) | |
Rank	Feature	Mean	sd	Feature	Mean	sd
1	LPCEPSTRA_E	0.050	0.082	LPREFC_E	0.051	0.023
2	LPCEPSTRA_F0	0.050	0.064	LPCEPSTRA_E	0.056	0.033
3	LPCEPSTRA	0.080	0.099	LPCEPSTRA_F0	0.059	0.034
4	MFCC_F0	0.116	0.069	LPCEPSTRA	0.061	0.033
5	LPREFC_E	0.121	0.090	LPREFC	0.065	0.029
6	MFCC_E	0.125	0.086	LPREFC_F0	0.073	0.022
7	MFCC	0.126	0.091	MFCC	0.076	0.036
8	LPREFC_F0	0.130	0.083	MFCC_E	0.077	0.035
9	RASTA	0.156	0.125	MFCC_F0	0.096	0.037
10	LPREFC	0.182	0.082	RASTA	0.132	0.121
11	LPCEPSTRA_D	0.364	0.171	LPREFC_D	0.359	0.119
12	LPREFC_A	0.422	0.167	LPREFC_A	0.431	0.127
13	LPREFC_D	0.479	0.135	MFCC_D	0.462	0.121
14	LPCEPSTRA_A	0.507	0.136	MFCC_A	0.507	0.126
15	MFCC_D	0.509	0.139	LPCEPSTRA_D	0.510	0.121
16	MFCC_A	0.562	0.110	LPCEPSTRA_A	0.582	0.107
17	F0	0.668	0.038	F0	0.628	0.043

- While not the best feature set for both corpora, the LPCEPSTRA features (with or without normalized log energy or pitch appended) performed well, indicating that they are a robust set of features.

- The RASTA features always performed better than the transitional features, but not quite as well as the static features.

- The transitional features did not perform as well as the static features from which they were derived. This supports Soong and Rosenberg's findings [62]. In related experiments, the addition by concatenation of the delta and acceleration coefficients gave equal (for MFCC_E and LPCEPSTRA_E) or worse results (for LPREFC_E), suggesting that decision fusion techniques are the best means of capitalizing on the temporal information.

- Pitch, as an independent feature, performed poorly.

25

- Appending the normalized log energy, and sometimes the pitch, can improve the performance of the basic features.

- The fact that many of the feature sets performed well for GREENFLAG utterances may indicate that in addition to speaker recognition, platform recognition (i.e. recognition based on the noise characteristics of the platform and the channel) was also taking place.

Table 7. ANOVA Test Results for Best Features. The difference in performance of the top three TIMIT features (LPCEPSTRA_E, LPCEPSTRA_F0, and LPCEPSTRA) and the top five GREENFLAG features (LPREFC_E, LPCEPSTRA_E, LPCEPSTRA_F0, LPCEPSTRA, and LPREFC) is not statistically significant at a significance level of 0.05.

Corpus	DoF1	DoF2	F	$F_{0.05}$	H_0
TIMIT	2	147	1.30	3.07	ACCEPT
GREENFLAG	4	245	1.41	2.42	ACCEPT

Considering the results of the ANOVA testing described above and observing the stability of the error rates plotted versus the number of codewords shown in Figure 20, page 69, and Figure 21, page 70, the feature sets most used in the remaining experiments were the LPCEPSTRA_E for TIMIT and the LPREFC for GREENFLAG.

Average Classification Error (10 TIMIT Speakers, 40 Test Utterances)

Average Classification Error (10 GREENFLAG Speakers, 53 Test Utterances)
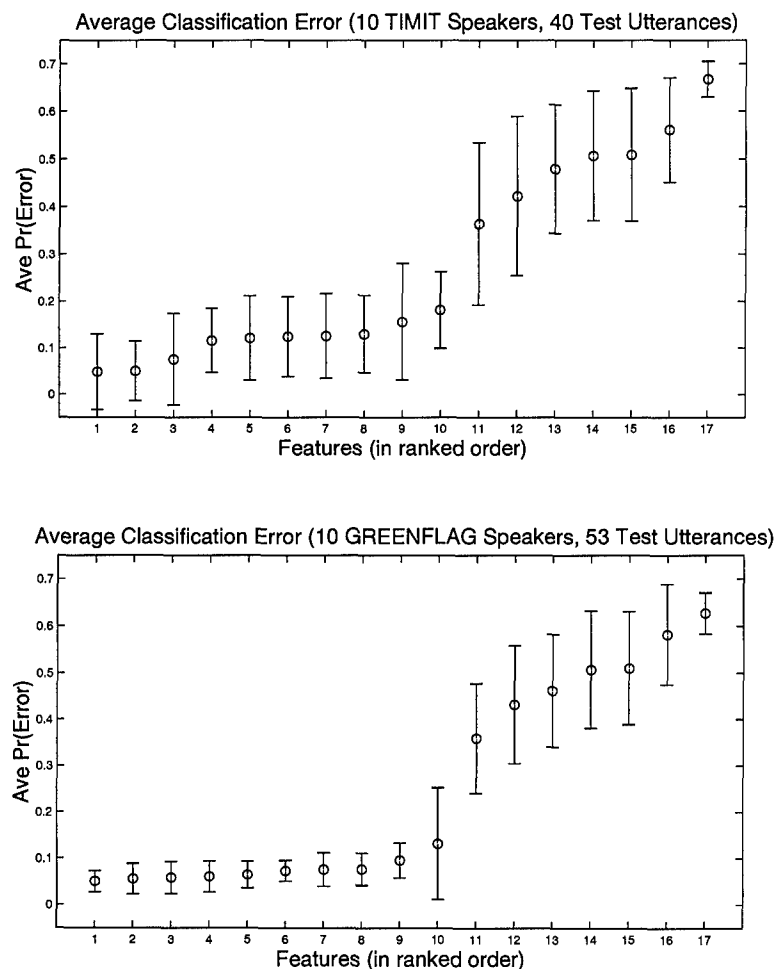
Figure 5.   Results of Feature Analysis. These plots show the average classification error for the 17 feature sets considered for (top) TIMIT and (bottom) GREENFLAG. The features are placed along the ordinate according to the ranking in Table 6. As shown, the best TIMIT feature set is the LPCEPSTRA_E, while for GREENFLAG, the best feature set is LPREFC_E. Note the clear separation in performance between the static and the transitional feature sets.

4.3 Codebook Analysis

As described in Section 3.4, the codebook analysis to find the optimal codebook size was based on minimal classification error, minimal computational complexity, and an insignificant change in cluster distortion. Again, LNKnet's nearest cluster classifier was used for this closed-set task. Cluster distortions were taken from the results of LNKnet's k-means vector quantization. The number of codewords ranged from 1–50 for this analysis.

4.3.1 TIMIT.
Figure 6 shows the experimental results for finding the optimal codebook size for the TIMIT corpus using the LPCEPSTRA_E feature set. Ten speakers chosen from Dialect Region 8, four test utterances each, were used. Based solely on minimal classification error, the optimal codebook size should be in the range from 19–27 codewords. Considering computational complexity, however, a more acceptable range is 9–16 codewords.
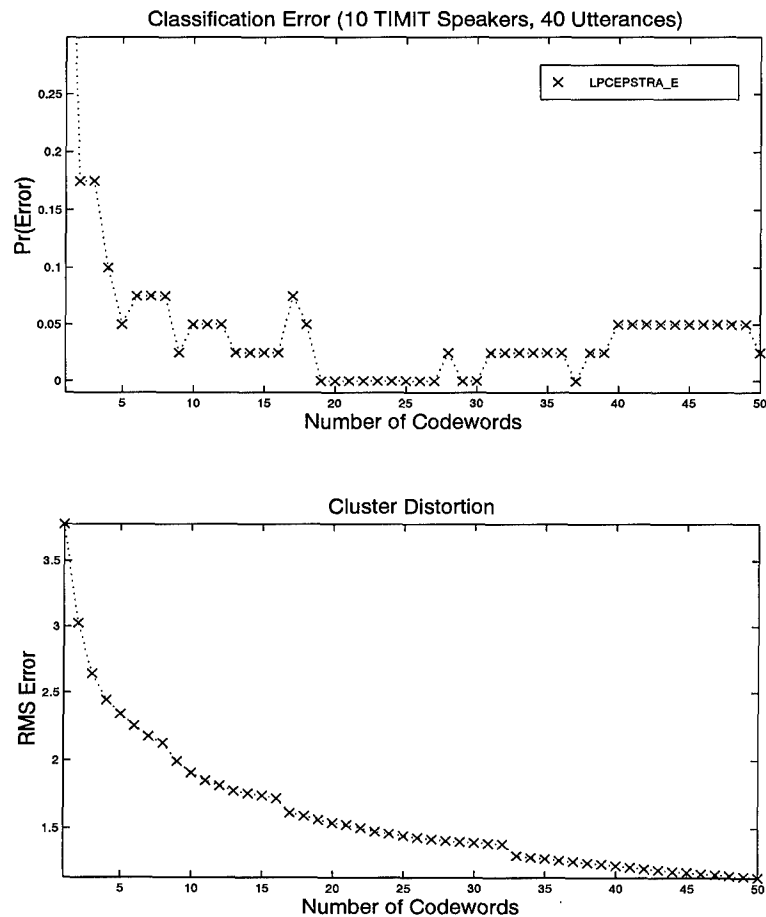
Figure 6. Results of Codebook Analysis for TIMIT. Using LPCEPSTRA_E features extracted from 10 TIMIT speakers (40 test utterances), the classification error (top) shows that an optimal number of codewords, based solely on classification, is in the range of 19–27 codewords. In contrast, considering cluster distortion (bottom) alone may lead one to choose a rather large codebook size. Even though the cluster distortion drops significantly for the first few codewords and gradually less thereafter, its change does not become insignificant until approximately 35 codewords.

4.3.2 GREENFLAG. Figure 7 shows the experimental results for finding the optimal codebook size for the GREENFLAG corpus using the LPREFC feature set. Ten arbitrarily chosen GREENFLAG speakers, with a total of 53 test utterances, were used. Based on minimal classification error, the optimal codebook size should range from 6–10 codewords. This range is also acceptable in terms of computational complexity.

Figure 8 shows the results of additional codebook analysis using the LPREFC_E features extracted from all 41 GREENFLAG speakers (one test utterance per speaker). Here, both training and test utterances were classified. Classification of the training utterances immediately converges to zero errors. Classification of test utterances, however, is the primary concern. Again, small codebooks (with approximately 6–8 clusters) are adequate, if not optimal. Similar results were obtained using LPCEPSTRA_E and MFCC_E.
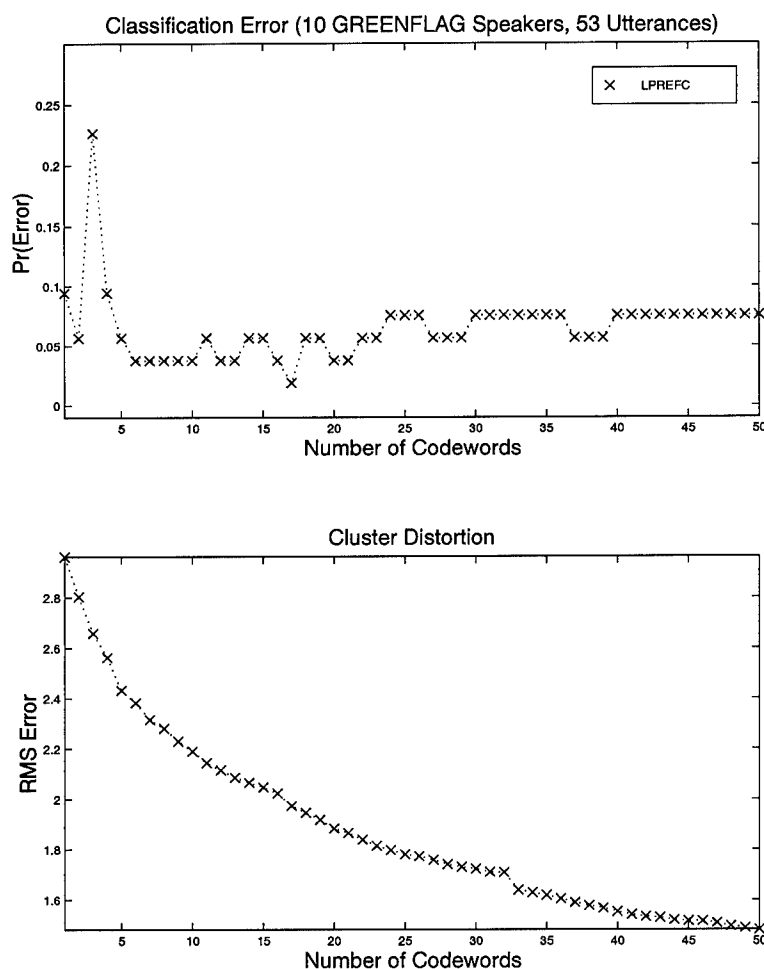
Figure 7. Results of Codebook Analysis for GREENFLAG. Using LPREFC features extracted from 10 GREENFLAG speakers (53 test utterances), the classification error (top) shows that an optimal number of codewords, based solely on classification, is in the range of 6–10 codewords. In contrast, considering cluster distortion (bottom) alone may lead one to choose a rather large codebook size. Even though the cluster distortion drops significantly for the first few codewords and gradually less thereafter, its change does not become insignificant until approximately 45 codewords.
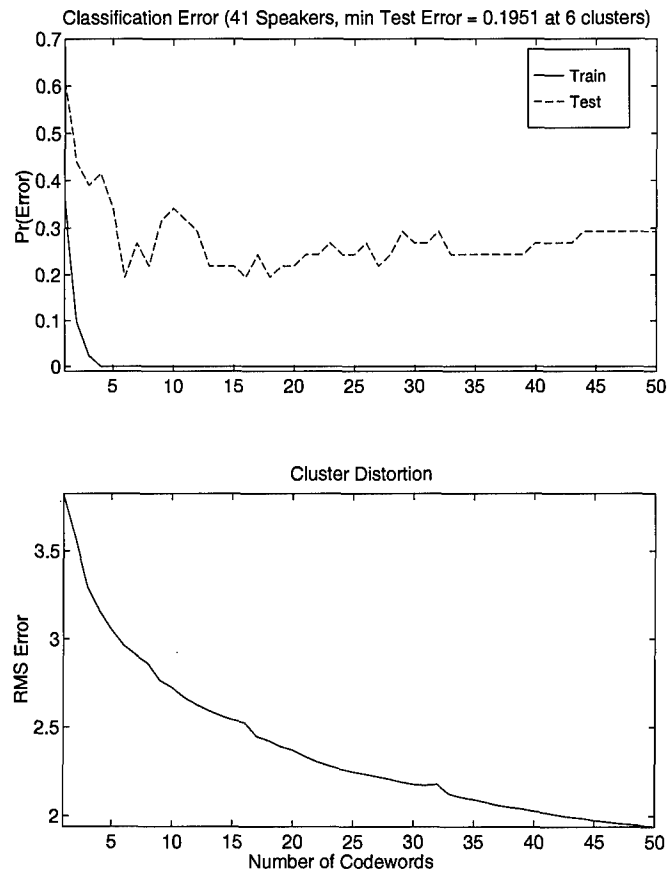
Figure 8. Results of Codebook Analysis (all GREENFLAG speakers). Using LPREFC_E features extracted from the 41 GREENFLAG speakers (one test utterance per speaker), the classification error (top) shows that an optimal number of codewords is six. Again, the cluster distortion (bottom) signifies the use of larger codebooks.

4.3.3 Observations.

- Relatively small codebooks are adequate to achieve reasonable, if not optimal, speaker identification results.

- Increasing the number of codewords can provide slight improvements in classification accuracy; however, since the tradeoff for this minimal additional accuracy is computational complexity, codebook sizes of approximately 8-10 codewords were determined to be optimal.

- In contrast to classification accuracy, cluster distortion results considered alone signify that a large number of codewords may be optimal (as many as 35–45, or depending on how an insignificant change in cluster distortion is defined). However, cluster distortion always drops significantly in the first few codewords. Based on these two observations, a requirement to add codewords until the change in cluster distortion becomes insignificant may actually create codebooks that fit the training data too closely, resulting in a classifier that has "memorized" the training data or lost its ability to generalize for test data. This would explain the divergent trend, seen here, in classification error as the number of codewords increases past the optimal range for classification accuracy. It further indicates that the initial decrease in cluster distortion is all that is required for a significant decrease in classification error. Compared to the factors of classification error and computational complexity, the cluster distortion design criteria had little impact on this cluster analysis.

- It is possible that the cluster analysis results found in this research were influenced by the limitation of using one short training utterance per speaker. Systems which are free to use more training data may find that more codewords produce better speaker identification results.

4.4 The Open-Set Task

Drawing on the closed-set speaker identification results of the feature and codebook analyses, the proposed text-independent open-set speaker recognition system (a fuzzy classifier followed by hypothesis testing) was tested on the two corpora. It is important to recall from Section 1.5 that the amount of training data used in this effort (one 2–4 second utterance for each speaker's codebook) is far less than that used in most text-independent, closed-set speaker recognition systems; hence, directly comparing the results obtained here to such systems would lead to biased conclusions.

4.4.1 TIMIT. This section provides and discusses the results of open-set speaker recognition for the TIMIT corpus. Each dialect region was treated separately, training on one utterance from 10 speakers, and testing on 15 speakers. The reason for testing within dialect regions was because it proved to be more difficult (and realistic) than testing across dialect regions. From the trained speakers, four utterances were used, while five utterances were used for the out-of-class speakers for a total of 65 test utterances. Speaker codebooks, consisting of 10 codewords, and the LPCEPSTRA_E features were used, while the Smirnov significance level, α, ranged from $0.01 \leq \alpha \leq 0.09$.

The results of this test for Dialect Region 8 are shown in Figure 9. (Results for the other dialect regions are similar and can be found in Appendix B.) The lengths of the training utterances used range from 1.8–3.6 seconds, with a mean of 2.7 seconds. The top plot shows the final classification error rate (calculated using Equation 9). In terms of classification accuracy, an optimal value of α is $0.02 \leq \alpha \leq 0.03$. The bottom plot in Figure 9 shows the $\Pr(False Acceptance)$, the $\Pr(False Rejection)$, and the $\Pr(Correct Rejection)$. The equal error rate occurs at approximately $\alpha = 0.06$, where $\Pr(False Acceptance) = 0.12$ and $\Pr(False Rejection) = 0.14$ (actual values). The $\Pr(Correct Rejection)$ indicates the rate at which the Smirnov Test correctly rejects mis-classifications.

Table 8 shows the confusion matrix for Dialect Region 8, obtained at $\alpha = 0.06$. The correct rejections at this value of α occurred at a rate of $\Pr(Correct Rejection) = 0.29$, and the final classification error rate is $\Pr(Error) = 0.26$.

To gain a better appreciation for the results obtained from the individual TIMIT dialect regions, the results from all dialect regions were averaged to produce Figure 10. The top plot shows the averaged final classification error rate, indicating that optimal classification, $\Pr(Error) \approx 0.31$,

34
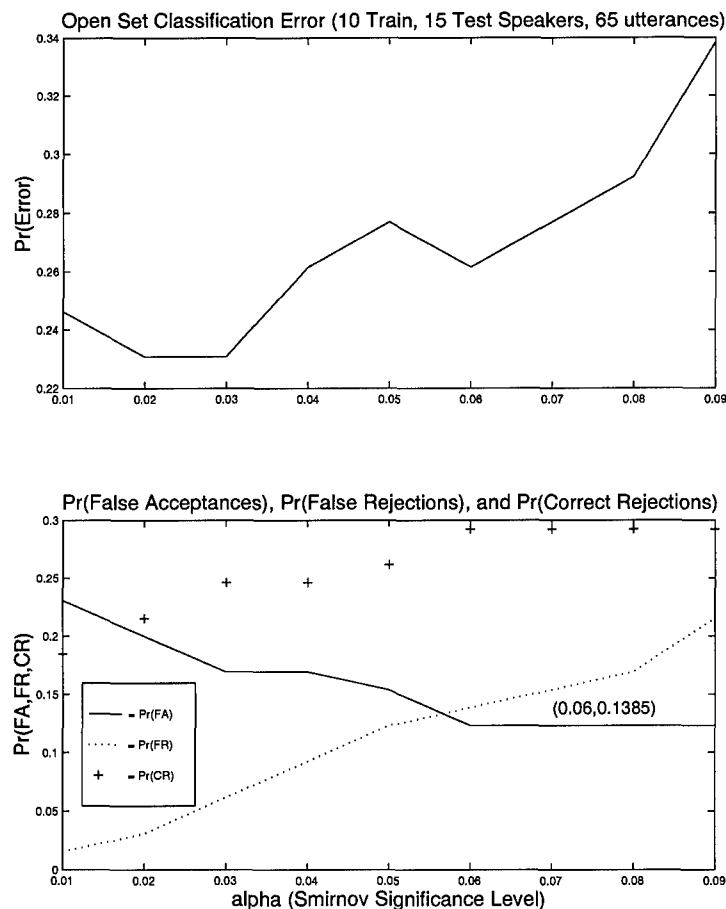
Figure 9. Results of Open-Set Speaker Recognition for TIMIT, Dialect Region 8, using LPCEP-STRA_E and 10 codewords per speaker. For minimal classification error, $0.02 \leq \alpha \leq 0.03$ is optimal, while for an equal error rate, an optimal value is $\alpha \approx 0.06$.

occurs at $\alpha = 0.02$. The bottom plot shows that the equal error rate occurs at $\alpha \approx 0.04$, resulting in $\Pr(Error) \approx 0.35$.

Table 8.   Open-Set Speaker Recognition Results for TIMIT, Dialect Region 8, using LPCEPSTRA_E and 10 codewords per speaker. This confusion matrix is for $\alpha = 0.06$, which is approximately where the equal error rate occurs, resulting in $\Pr(CorrectRejection) = 0.29$ and $\Pr(Error) = 0.26$. The dotted line indicates the division between reference speakers and out-of-set speakers. Notice that most of the out-of-set speakers' utterances are correctly rejected into the "Others" class.

ACTUAL SPEAKER	COMPUTED SPEAKER										
-	0	1	2	3	4	5	6	7	8	9	Others
0	4	.	.	.	.	.	.	.	.	.	.
1	.	4	.	.	.	.	.	.	.	.	.
2	.	.	4	.	.	.	.	.	.	.	.
3	.	1	.	2	.	.	.	.	.	.	1
4	.	.	1	.	3	.	.	.	.	.	.
5	.	.	.	.	.	3	.	.	.	.	1
6	.	.	.	.	.	.	3	.	.	.	1
7	.	.	.	.	.	.	.	1	.	.	3
8	.	.	.	.	.	.	.	.	4	.	.
9	.	.	.	.	.	.	.	.	.	1	3
10	.	.	.	.	1	.	.	.	.	.	4
11	.	3	1	.	.	.	.	.	.	.	1
12	.	.	.	.	.	.	.	.	.	.	5
13	.	.	.	.	.	.	.	.	.	.	5
14	.	.	.	.	.	1	.	.	.	.	4

36
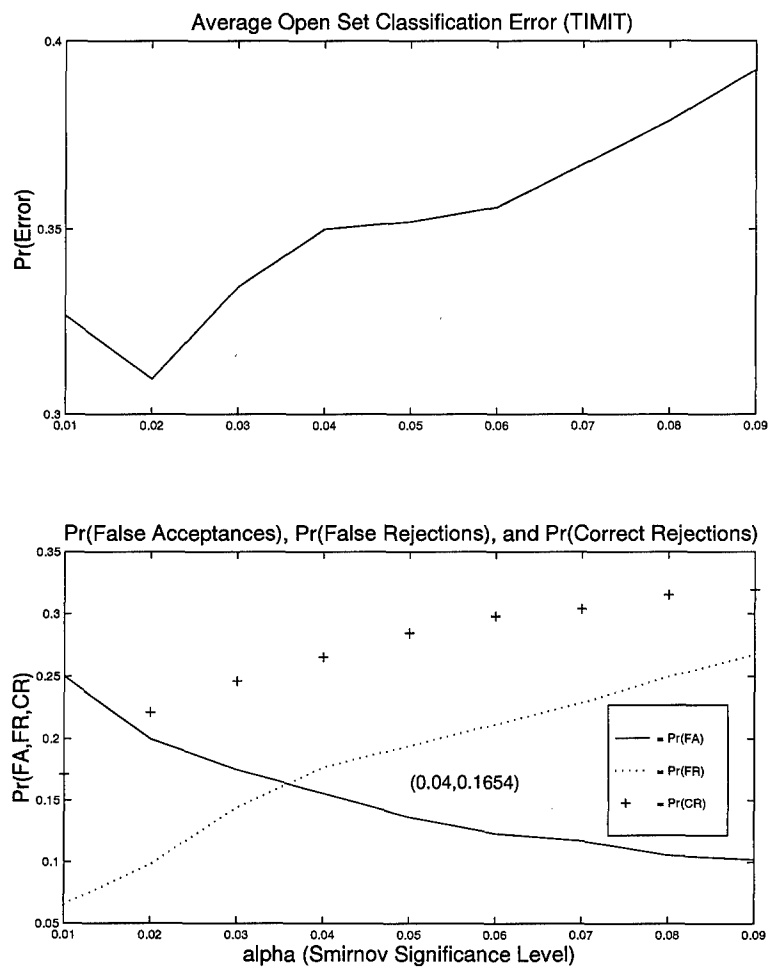
Figure 10. Results of Open-Set Speaker Recognition for all TIMIT speakers used. This figure shows the averaged results obtained from all eight dialect regions (using 65 test utterances per dialect region). For minimal classification error, $\alpha = 0.02$ is optimal, while for an equal error rate, an optimal value is $\alpha \approx 0.04$.

4.4.2 GREENFLAG.   This section provides and discusses the results of open-set speaker recognition for arbitrarily chosen speakers from the GREENFLAG corpus, training on one utterance from each of 10 speakers, and testing on 15 speakers for a total of 73 test utterances. (Results of additional tests with the GREENFLAG corpus, using different groups of 10 speakers for training and 25 for testing, are available in Appendix B.) The lengths of the training utterances used range from 2.0–3.9 seconds, with a mean of 2.8 seconds. Speaker codebooks, consisting of eight codewords per speaker, and the LPREFC features were used, while α ranged from $0.05 \leq \alpha \leq 0.19$.

The results of this test are shown in Figure 11. The top plot shows the final classification error rate. In terms of classification accuracy, $\alpha = 0.09$ is optimal. The bottom plot in Figure 11 shows the $\Pr(FalseAcceptance)$, the $\Pr(FalseRejection)$, and the $\Pr(CorrectRejection)$. The equal error rate occurs at approximately $\alpha = 0.12$, where $\Pr(FalseAcceptance) = 0.16$ and $\Pr(FalseRejection) = 0.15$ (actual values). The $\Pr(CorrectRejection)$ indicates the rate at which the Smirnov Test correctly rejects mis-classifications into the "Others" class.

Table 9 shows the confusion matrix, obtained at $\alpha = 0.12$. The correct rejections at this value of α occurred at a rate of $\Pr(CorrectRejection) = 0.12$, and the final classification error rate is $\Pr(Error) = 0.32$.

38
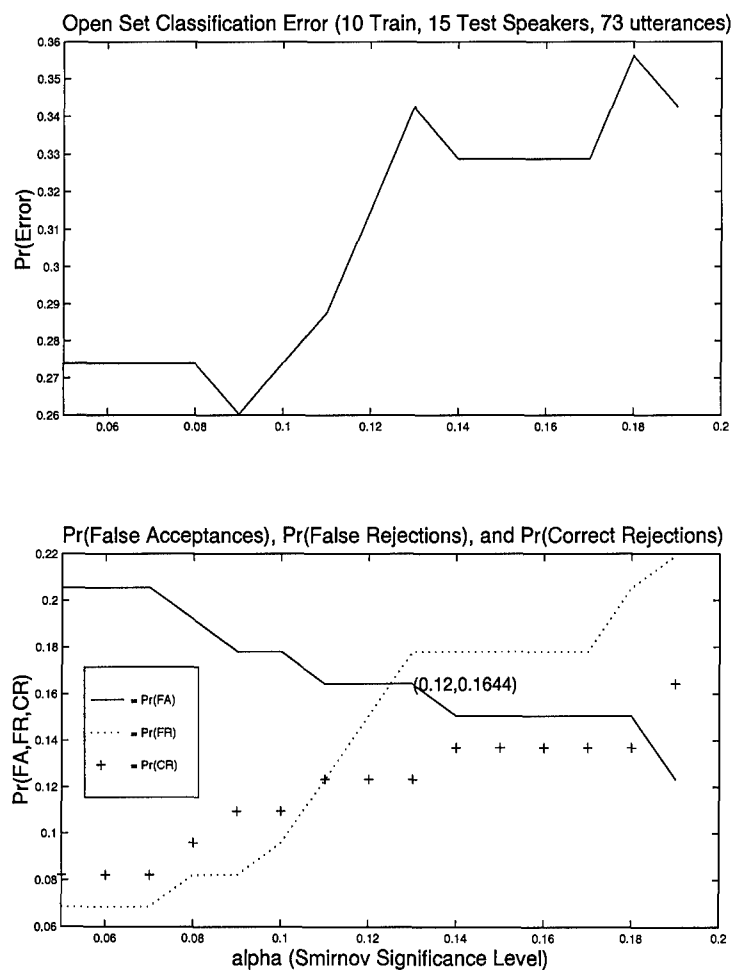
Figure 11. Results of Open-Set Speaker Recognition using the GREENFLAG corpus. For minimal classification error, $\alpha = 0.09$ is optimal. For an equal error rate, an optimal value is $\alpha \approx 0.12$.

Table 9. Open-Set Speaker Recognition Results for GREENFLAG, using LPREFC and eight code-words per speaker. This confusion matrix is for $\alpha = 0.12$, which is approximately where the equal error rate occurs, resulting in $\Pr(CorrectRejection) = 0.12$ and $\Pr(Error) = 0.32$. The dotted line indicates the division between reference speakers and out-of-set speakers.

ACTUAL SPEAKER	COMPUTED SPEAKER										
-	0	1	2	3	4	5	6	7	8	9	Others
0	1	.	.	.	.	.	.	.	.	.	3
1	.	4	.	.	.	.	.	.	.	.	2
2	.	.	4	.	.	.	.	.	.	.	.
3	.	.	.	6	.	.	.	.	.	.	.
4	.	.	.	.	5	.	.	.	.	.	1
5	.	.	.	.	.	4	.	.	.	.	1
6	.	.	.	.	.	.	4	.	.	.	2
7	.	.	.	.	.	.	.	3	.	.	1
8	.	.	.	.	.	.	.	.	7	.	1
9	.	.	.	.	.	.	.	1	.	3	.
10	1	.	.	.	.	.	.	.	.	.	3
11	.	.	.	.	.	.	.	.	.	1	2
12	.	.	1	.	.	.	.	.	.	1	1
13	1	.	2	.	.	.	.	.	.	1	2
14	.	1	1	.	.	.	.	.	.	1	1

40

4.4.3 Observations.

- Based on the objective error rate of 40%, this system attained reasonable results, with classification error rates as low as 23% for TIMIT and 26% for GREENFLAG.

- This application of the Smirnov Test (via Equations 7 and 8) allows the freedom to use any range of the significance level, α. That is, hypothesis testing is not restricted to tabulated values of α.

- As shown in Figures 9, 10, and 11, the value of α at which the minimal classification error occurs, does not necessarily correspond to the value of α at which the equal error rate occurs. Thus, depending on what is more important (e.g. accuracy), or costly (e.g. falsely accepting or falsely rejecting a classification), one must chose the value of α appropriately.

- By *correctly rejecting* utterances into the "Others" class, this system possesses the potential to enhance the closed-set classification rate by correcting mis-classifications, albeit at a possibly high cost of false rejections. More importantly, however, the capability of correctly rejecting mis-classifications is vital in accomplishing the open-set task, for which it performs admirably.

4.5 Conclusion

This chapter showed that the best features, based on an averaged classification error rate, are LPCEPSTRA_E for TIMIT utterances and LPREFC_E for GREENFLAG utterances. Feature analysis further revealed that LPCEPSTRA, with or without the normalized log energy or pitch appended, are a robust feature set. Codebook analysis showed that relatively small codebook sizes (with approximately 8–10 codewords) are optimal. Predominantly, however, this chapter showed that the proposed open-set speaker recognition system (a fuzzy, by-frame majority voting classifier followed by the Smirnov Test) is an effective method for accomplishing text-independent, open-set speaker recognition.

V. Conclusion

5.1 Introduction

The primary objective of this research was to examine the complex task of open-set speaker recognition and develop a method to accomplish this task. The second objective was to determine the best set of features to use for speaker recognition.

5.2 Summary of Results

Both stated objectives were met. The proposed text-independent, open-set speaker recognition system functioned within the objectives and scope outlined in Sections 1.4 and 1.5, attaining reasonable open-set classification error rates as low as 23% for TIMIT and 26% for GREENFLAG. The analysis of features showed that for clean speech from the TIMIT corpus, liftered linear prediction cepstral coefficients with normalized log energy appended (LPCEPSTRA_E) are optimal features (in terms of minimum averaged classification error rate), while for the tactical GREENFLAG corpus, linear prediction reflection coefficients with normalized log energy appended (LPREFC_E) are optimal. This analysis also revealed that the liftered linear prediction cepstral coefficients, with or without normalized log energy or pitch appended, are robust features which should be considered when the speech source (e.g. channel characteristics) is unknown.

5.3 Contributions

- This thesis introduced a fuzzy classifier followed by the Smirnov Test for Common Distributions as a new model for text-independent, open-set speaker recognition. This system has several attributes that make it well suited for open-set speaker recognition: First, the system is robust in that it achieved similar success in both clean and noise corrupted environments (TIMIT and GREENFLAG, respectively). Second, the fuzzy classifier's membership function values are easily derived from commonly used distortion measures, they are limited to the range of values $0 \leq U \leq 1$, and their value, which indicates the degree of class membership, offers a degree of confidence in decision making. Third, the Smirnov Test, being a non-parametric test, is well suited for this application since it did not require an assumption that the populations

42

(the reference and test membership function values) have a particular distribution, nor was this application constrained to tabulated values of the significance level.

- The best features, in terms of minimal speaker identification error rate, from a group of 17 feature sets were identified. All too often, speaker recognition research has relied on feature analyses done for speech recognition. While speech and speaker recognition are similar in many ways, their goals are quite different (one must determine what was said, while the other must determine who said it). To blindly assume that features that work well for one will also work well for the other is an unwise oversimplification. This research focused on finding those features that are well suited specifically for speaker recognition.

- Codebook analysis showed that relatively small codebooks (with approximately 8–10 codewords) are adequate, if not optimal. Much of the past work using vector quantization arbitrarily chose codebook sizes ranging anywhere from 32–512 codewords. As illustrated in Table 1, page 11, using a codebook size of 8, rather than say 64, saves orders of magnitude of floating point operations in both building the codebooks and in utterance classification. These results advocate the use of small codebooks and should provide insight into a better choice of codebook size that will save enormous costs in computation time.

- As a tool for follow-on research, this thesis intentionally provides a significant amount of background information in the area of speaker recognition. Appendix A was specifically written to assist a novice to the field in learning many of the signal processing and speaker recognition fundamentals. The software developed in this effort maximized the use of currently available UNIX based packages (e.g. Matlab, LNKnet, and ESPS) for ease in application or replication of this work for further research.

5.4   Follow-on Research

The research discussed in this thesis is by no means exhaustive. As with any large undertaking, there are many areas left for further research. Future work could include enhancements to the proposed open-set speaker recognition system. Some possible enhancements are listed below:

- Adaptability to new speakers may be a design criteria. When the system's classification response is to reject the classification, the system may need to add that unknown speaker to the reference set. Methods which do not require full retraining will have an advantage in this area.

- The final classification error rate may need to be much lower. Other distance metrics (such as weighted Euclidean or Mahalanobis) or non-vector quantization-based classifiers (such as HMMs, GMMs, or neural networks) may provide better accuracy. Alternatively, feature fusion, classifier fusion, and/or decision fusion techniques may be the answer.

5.5   Conclusion

Closed-set speaker recognition systems abound; however, their application to real-world problems are fundamentally limited since it is unrealistic to train a system on *all* possible speakers. A realistically viable system must be capable of dealing with the open-set task.

Not only does this thesis perform one of the most comprehensive, to date, feature comparisons specifically in the interest of identifying features well suited for speaker recognition, but it also justifiably advocates the use of relatively small codebooks for vector quantization-based classification. Predominantly, however, this thesis introduces a novel approach to accomplishing text-independent, open-set speaker recognition. By acknowledging that training was limited to one short utterance per speaker, while capitalizing on the use of small speaker populations, the system achieved success with utterances from a tactical communications source. This system is, therefore, directly applicable to tactical surveillance applications.

Appendix A. Introduction to Speaker Recognition

A.1 Introduction

In speaker recognition there are two main areas of interest: speaker verification and speaker identification. In *speaker verification*, the computer must verify the identity of an individual by determining whether the test pattern matches a stored reference pattern. An example of speaker verification would be a security system that must determine whether the speaker is who he or she claims to be. The Cable News Network reported (7 May 95: Science and Technology Week) that Texas Instruments has integrated speaker verification into a cellular phone system to prevent criminals from gaining access to and charging calls to unknowing cellular phone customers.

In *speaker identification*, the computer must determine whether the test pattern matches any of the stored reference speakers' patterns. Thus, while speaker verification is essentially a binary decision process (i.e. "Is the speaker who he or she claims to be?" YES/NO), speaker identification is a multi-class decision process in which the probability of mis-identification increases with the size of the speaker population. Automated speaker identification is finding application in labeling recorded dictation (e.g. in court room proceedings). Also, automated speaker identification may eventually prove to be a valuable tool for law enforcement agencies [3] who may, for example, wish to identify individuals discussing criminal activities on the telephone.

When the speaker recognition system is both trained and tested using the same text or phrase (or a subset thereof), the system is *text-dependent*. Again, the example of the security system applies since the individual wishing to gain access is usually required to recite a specific phrase. *Text-independent* systems, on the other hand, are not constrained by the text spoken in either training or testing. In this case, the system lacks the means of capitalizing on the contextual features of what was spoken.

Another means to categorize the speaker recognition system has to do with whether the population size is constrained. *Closed-set* speaker identification describes a system in which the population is constrained to only the N speakers on which the system was trained. Hence, the system will classify the test speaker as one of those N speakers. In an *open-set* speaker identification system, the speaker under test may not match (i.e. be close enough to) any of the speakers on which it was trained. Thus, new speakers should produce a "no match" response from the system. If the system is adaptive, the new speaker may be added to the database and N is incremented by one; otherwise, the new speaker

is simply rejected into an "Others" class. Determination of whether the match is close enough depends on a design threshold within the system.

In a real-world environment, the signal is corrupted by noise; hence, it is necessary to design a system which will operate in the presence of noise. Noise has many sources and is referred to here as anything which degrades or interferes with the signal. The telephone system, for example, degrades the signal not only by adding noise, but also by band-limiting the signal.

The remainder of this appendix focuses on a review of literature in the subjects of speech analysis, signal processing, and pattern recognition and will detail those techniques which apply to automatic speaker recognition. Two introductory articles to the topic of speaker recognition by Gish and Schmidt [22] and O'Shaughnessy [43] are highly recommended. Gish and Schmidt focus on text-independent, closed-set speaker identification, using maximum *a posteriori* probability techniques. The authors introduce some of the fundamentals of speaker identification, particularly feature selection with a description of the mel-warped cepstra method for parameterizing the short-term spectrum. The authors also discuss various "robust" systems which operate well in the presence of noise. O'Shaughnessy deals with the broader topic of automated speaker recognition, providing a description of each of the pattern recognition steps (normalization, parameterization, feature extraction, comparison, and decision) as they apply to speaker recognition. Dynamic time warping and vector quantization are two alternative methods discussed for segmenting speech signals. Various methods for extracting what he states are the best features (pitch, timing cues, and the first three formants) are also discussed. Drawing on the techniques described in the article, the author concludes by describing a possible system design. This appendix concludes with two examples of speaker recognition which apply many of the techniques discussed below.

A.2   Pre-Processing

Pre-processing entails those measures taken to prepare the speech signal for analysis. For system development and speech analysis research, the speech signal is typically in the form of digitized data in a database. One such database (or corpus) is TIMIT [1]. Pre-processing typically consists of pre-emphasizing the speech signal, then windowing the signal to obtain frames of speech. As a precursor to the discussion of these topics, Figure 12 shows an example of a speech signal from the TIMIT corpus, a Hamming window, and a resultant frame of the pre-processed signal.
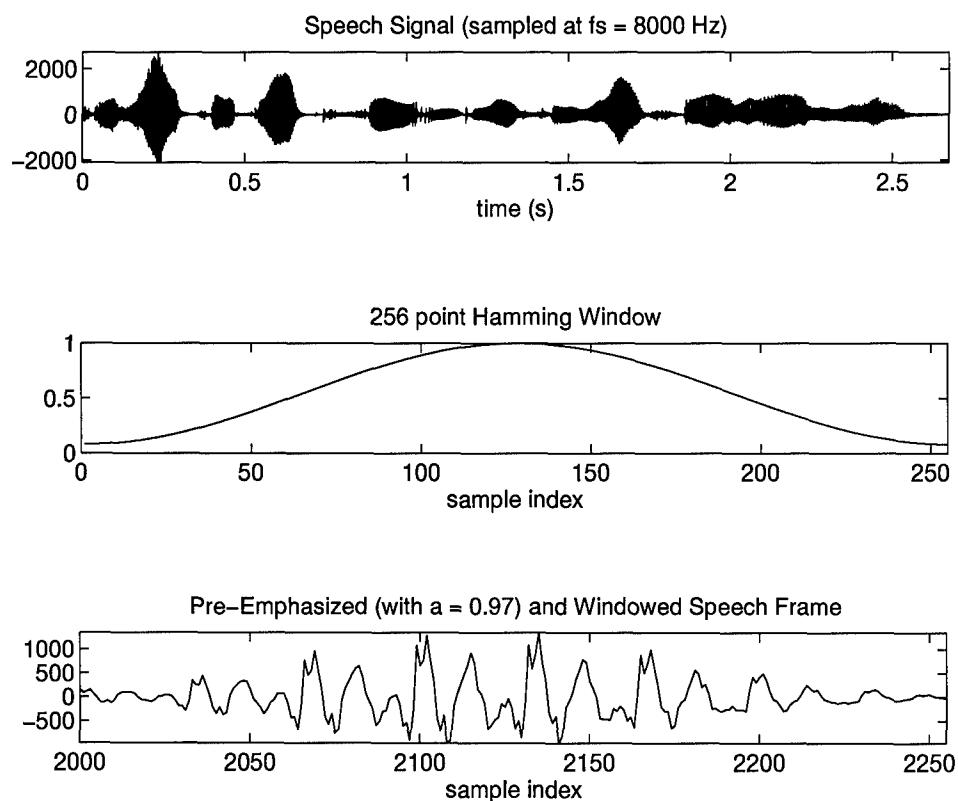
Figure 12. Pre-Processing a Speech Signal. These plots show (top) a TIMIT speech signal sampled at 8000 Hz, (middle) a Hamming window, and (bottom) a resultant speech frame after the speech signal is pre-emphasized (using Equation 10) and windowed (using a 256 point Hamming window).

A.2.1 Pre-emphasis. Given a sampled, digitized corpus, the first step in pre-processing is typically pre-emphasis. Pre-emphasis consists of applying a filter to the speech signal that increases the relative energy of the high frequency spectrum, thereby enhancing the high frequency components and reducing the effects of the low frequency components [13] [46]. The filter shown in Equation 10 creates a zero at $\omega = 0$:

$$P(z) = 1 - az^{-1}, \text{ for } 0.9 \le a \le 1.0, \tag{10}$$

where the value of the pre-emphasis factor is typically $a = 0.97$ [33].

Two reasons for using a pre-emphasis filter are:

1. Pre-emphasis enhances the higher frequency formants in the vocal tract, while reducing the lip and glottal effects. Introducing the zero near $z = 1$ (in addition to the zero near $z = 1$ contributed by the lip radiation characteristic), reduces the spectral effects of the two glottal poles near $z = 1$ [13] [65].

2. By reducing the dynamic range of the signal spectrum, pre-emphasis can prevent numerical instability caused by an ill-conditioned autocorrelation matrix when using linear prediction analysis (discussed in Section A.3.1) [36].

A.2.2 Windowing. Since all analysis must be done in finite time [13], after pre-emphasis, a window function is applied to the speech signal to form a "frame." The resulting short speech frames are typically chosen to be 10-40 ms in length [5]. The two main reasons for windowing the speech signal are:

1. Speech is inherently a non-stationary process [36] [45], but it is *assumed* to be a short-time stationary process up to 50-70 ms [5] [36]. Thus, the speech frames produced by applying the the window are assumed to be stationary. Some researchers, Morgan [39] for example, disagree with this simplification, but it is still widely used.

2. The discrete Fourier transform (DFT), which is widely used in speech analysis, requires a finite number of input samples [30] [42].

There are many window functions from which to chose (e.g. rectangular, Hamming, and Hanning). The window should exhibit a narrow bandwidth mainlobe to resolve the sharp details of the

magnitude spectrum and a large attenuation of the sidelobes to prevent noise from other parts of the spectrum from corrupting the true spectrum at a given frequency [13]. A design tradeoff occurs since a longer window tends to produce a better spectral picture of the signal within a stationary region, while a shorter window resolves signal events better in time [13]. The Hamming window of Figure 12, whose spectral sidelobes are attenuated by 30 dB, provides a suitable tradeoff and is commonly used [46].

Overlapping frames of speech are used to overcome the shortfalls from window edges. (No window is perfect, and some windowed data values are set to nearly zero.) Overlapping the frames is also necessary because finite length transforms are being used to process a long (relative to the short frames) signal. Typically, the frames overlap by $\frac{1}{2}$ to $\frac{2}{3}$ of the frame length [5] [33].

A.3   Feature Extraction

Feature extraction, in terms of speaker recognition, is the process of creating a compact set of parameters characteristic of a speaker. The goal is to preserve information relevant to the speaker's identity, while producing minimal intra-speaker variance and maximal inter-speaker variance. Parsons states, "The ability of a feature to separate two classes depends on the distance [in the feature space] between two classes and the scatter within classes [[46]:page 176]." Further, it is often said that good features make for a good classifier. The optimal features for pattern classification (but not necessarily for pattern reconstruction) are the *a posteriori* conditional probability distribution functions [59]. A significant discovery (although not exploited in this research) was that when the multi-layer perceptron is trained using backpropagation for the multi-class problem, the outputs approximate the *a posteriori* conditional probability distribution functions [58].

A.3.1   Linear Prediction Analysis.   The objective of linear prediction (LP) analysis is to estimate the output sequence (or the forthcoming output sample) from a linear combination of input samples, past output samples, or both [46]. Ignoring nasals and some fricatives, an all-pole filter, excited by either a sequence of quasi-periodic pulses or a white noise source, can accurately model the vocal tract [6]. Linear prediction is a procedure for encoding the speech signal by representing it in terms of time-varying parameters related to the transfer function of the vocal tract and the characteristics of the excitation [6] [36]. The application of LP analysis derives a set of predictor coefficients, obtained by minimizing the total squared error, E, between the actual signal value and its predicted value [36]. The

49

predictor coefficients or reflection coefficients (which are derived intermediately) are representative of the vocal tract (or the data), and they form the feature vectors that are passed to a classifier.

The number of coefficients (or model order, p) required to adequately represent any speech segment can be determined by the number of resonances and anti-resonances of the vocal tract in the frequency range of interest, the glottal volume flow function, and the lip radiation [6]. Atal and Hanauer [6] determined that a value of $p = 12$ is adequate at a sampling frequency of 10 kHz. Specifically, they determined that for $f_s = 10$ kHz, $p = 12$ is adequate for voiced speech and $p = 6$ is adequate for unvoiced speech. Parsons [46] provides a rule-of-thumb which depends on the sampling frequency, f_s, for determining the model order:

$$p = \frac{f_s}{1000} + \gamma \qquad (11)$$

where γ is a "fudge constant," empirically determined, and typically $\gamma = 2$ or 3.

Unfortunately, the assumption of the all-pole model is violated when noise corrupts the speech signal (e.g. signal-to-noise ratios below 5-10 dB), resulting in a serious degradation of the model [46].

A.3.1.1 LP Analysis Methods. The two primary methods for LP analysis, the auto-correlation method and the covariance method, are described below [36]:

- **Autocorrelation Method**. The autocorrelation method assumes that the total squared error, E, is minimized over an infinite duration. This method applies for stationary signals, and it guarantees a stable filter (excluding possible instability due to round-off errors). Problems of parameter accuracy can arise due to the windowing of the signal. For example, Davis and Mermelstein [12] found that for a signal sampled at 10 kHz, better word recognition results were achieved with a 256 point Hamming window than with a window size of 128 points.

- **Covariance Method**. The covariance method, on the other hand, assumes that the total squared error, E, is minimized over a finite interval. This method is more general in application and can be used without restrictions, but it does not guarantee a stable filter.

A.3.1.2 Durbin's Recursion Procedure. Durbin's recursion procedure is used with the autocorrelation method for iteratively determining the predictor coefficients, the reflector coefficients, and the predictor error. The primary advantage to Durbin's procedure is that it significantly reduces

the computational complexity, compared to the straight autocorrelation method (from $\frac{p^3}{3} + O(p^2)$) operations for the autocorrelation method to $p^2 + O(p)$ operations for Durbin's method) [36].

A.3.2 *Cepstral Analysis.*   In recent years, the cepstrum has found widespread use, due to demonstrated performance, in both speaker and speech recognition. Deller *et al* describe the cepstrum as "... the premier feature [as opposed to the LP parameters] in the important 'Hidden Markov Modeling' strategy ... [[13]:page 380]." The cepstrum of a signal is the Fourier transform of the logarithm of its magnitude spectrum, which in equation form is expressed as [22] [46]:

$$\text{cepstrum } = FFT(log|\text{Spectrum}|) \tag{12}$$

The cepstra can be calculated either directly from the Fourier transform or from the linear prediction coefficients (the faster of the two). The uniqueness of the cepstrum is that it provides a means to separate the speech signal's two components: the slowly varying spectral envelope and the rapidly varying pitch harmonic peaks [13] [46]. In fact, Parsons states, "When used with noiseless speech, the cepstrum is unparalleled as a pitch extractor... [[46]:204]."

For illustrative purposes, Figure 13 shows the development of the cepstrum, described in Equation 12, for a synthetically generated 100 Hz pulse train signal. Using 256 samples, the signal is windowed with a 256 point Hamming window to create the frame. Next the signal's spectrum (i.e. the logarithm of the magnitude of the FFT of the frame) is shown. Notice the peak at 100 Hz, which corresponds to the fundamental frequency (or pitch) of the signal. The final plot shows the real cepstrum of the signal. The low-quefrency portion (quefrency less than approximately 0.005 seconds) corresponds to the signal's spectral envelope, while the peaks at 0.01, 0.02, and 0.03 seconds correspond to the pitch and its second and third rahmonic. Thus, for a speech signal, the low-quefrency cepstrum corresponds to the vocal system impulse response, while the high-quefrency cepstrum corresponds to the excitation [13].

A.3.2.1 *The Mel-Warped Cepstrum.*   Linear prediction cepstrum coefficients (LPCC), generated from the LP spectrum and distributed along a linear frequency axis, form a less than optimal representation of an auditory signal since a logarithmic function of frequency better approximates the ability of the human ear to discriminate frequencies [33]. The mel or Bark scale is often used to
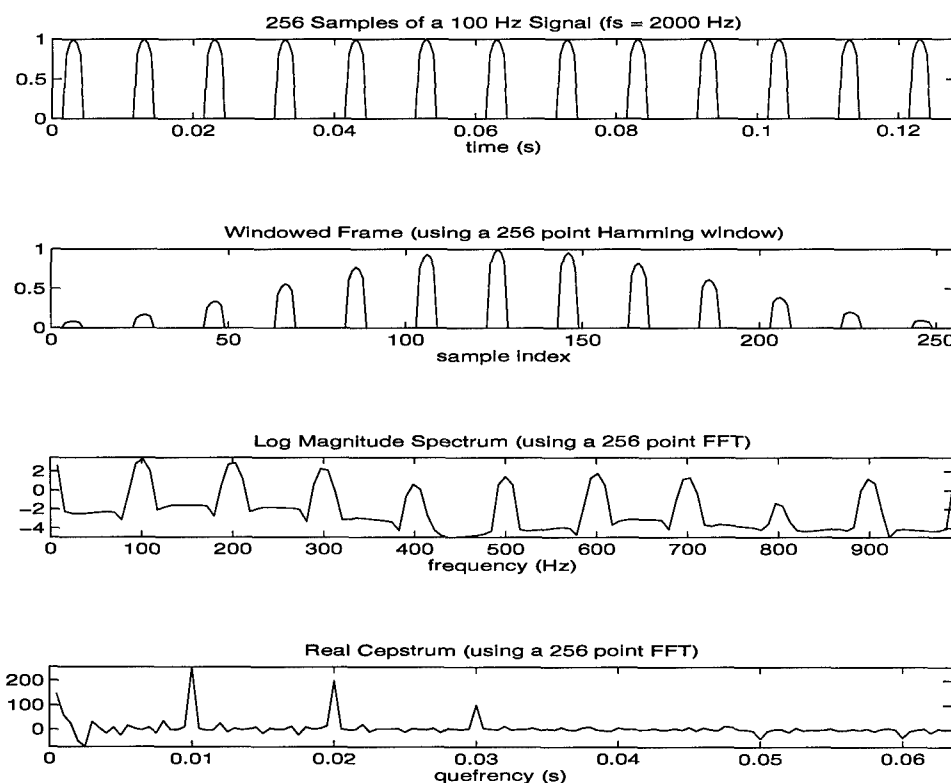
Figure 13.   The development of the cepstrum from a synthetically generated 100 Hz pulse train using Equation 12. The low-quefrency portion (quefrency less than approximately 0.005 seconds) corresponds to the signal's spectral envelope, while the peaks at 0.01, 0.02, and 0.03 seconds correspond to the pitch and its second and third rahmonic. Notice the cepstrum's efficacy in identifying the pitch.

approximate the resolution of the human auditory system's perception of speech [13] [46]. Deller *et al* define the mel as "a unit measure of perceived pitch or frequency of a tone [[13]:380]." An equation for approximating the mel-scale, attributed to Fant (1959), is

$$F_{mel} = \frac{1000}{\log 2} \left(1 + \frac{F_{Hz}}{1000} \right), \tag{13}$$

where F_{mel} is the perceived frequency in mels and F_{Hz} is the actual frequency in Hz [13] [46].

The mel-frequency cepstral coefficients (MFCC) are obtained by mel-warping the spectrum's frequency scale before taking the the last Fourier (or inverse Fourier) transform shown in Equation 12. Since the real cepstrum works directly with the log magnitude spectrum (see Figure 13) of the speech signal, it is well suited for such a computation [13]. Davis and Mermelstein [12] generated MFCCs by applying a simulated mel-scaled triangular filter bank, similar to that shown in Figure 14. The width and spacing of the filters shown in Figure 14 are constant up to 1000 Hz and logarithmic beyond 1000 Hz. The Entropics Signal Processing System (ESPS) Hidden Markov Model Toolkit (HTK) [70], uses the following equation for the mel-scale for mel filter bank analysis:

$$F_{mel} = 2595 \log_{10} \left(1 + \frac{F_{Hz}}{700} \right). \tag{14}$$

HTK then uses a discrete cosine transform (identical to that used by Davis and Mermelstein [12]) applied to the log filter bank outputs, m_j, to calculate the MFCCs:

$$c_i = \sum_{j=1}^{p} m_j \cos \left(\frac{\pi i}{p} (j - 0.5) \right), \tag{15}$$

for $1 \leq i \leq N$, where p is the analysis order and N is the required number of cepstral coefficients.

Lee [33], while crediting Shikano and Oppenheim, describes the bilinear transform (BLT) as a method to transform the linearly scaled LP coefficients to a mel-scale. Lee represents the bilinear transform as

$$z_{new}^{-1} = \frac{(z^{-1} - a)}{(1 - az^{-1})}, \quad \text{for } (-1 < a < 1) \tag{16}$$

$$\omega_{new} = \omega + 2 \tan^{-1} \left(\frac{a \sin\omega}{1 - a \cos\omega} \right) \tag{17}$$
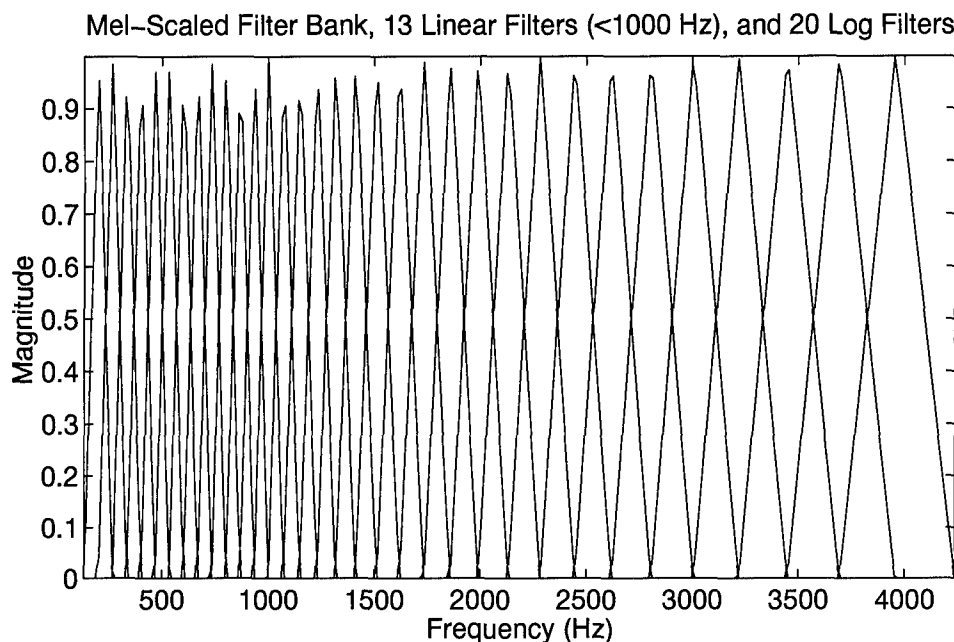
53

Figure 14. Mel-Scaled Triangular Filter Bank. A triangular filter bank can be used in generating mel-frequency cepstral coefficients. The filters are linearly spaced up to 1 kHz and logarithmically spaced after 1 kHz to produce the mel-frequency relationship.

where ω is the sampling frequency, ω_{new} is the converted frequency, and a is the frequency warping parameter. Positive values of a convert the frequency axis into a low-frequency weighted one, and for $0.4 < a < 0.8$, the frequency warping is comparable to that of the mel or Bark scales, obtained from Equation 13 [33]. In his work, Lee chose a value of $a = 0.06$, which, as shown in Figure 15, best approximates the mel-scale up to 1000 Hz.

A.3.3   Choosing the Best Features.   With regard to classification, the best features are those which produce the best classification results. Also, it is often desirable to limit the number of features, thereby preventing what Duda and Hart [17] refer to as "The Curse of Dimensionality." Two reasons for limiting the number of features are [58]:

1. To reduce the time to classify an input vector.

2. A large number of free parameters in the classifier may cause the classifier to "memorize" the training data, resulting in degraded performance with previously unseen test data.
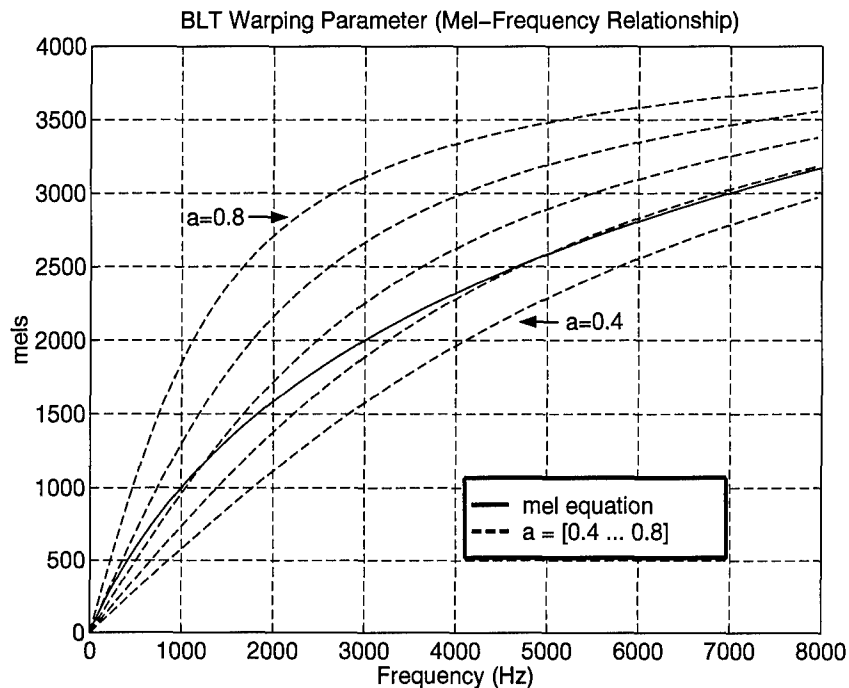
Figure 15.   The mel-frequency relationship, using the bilinear transform (BLT) for $0.4 \leq a \leq 0.8$
to "mel-warp" the frequency scale. The solid line represents the mel-scale given by
Equation 13.

The performance of a feature depends on how well it separates the classes from one another. As applied

to speaker recognition, the best features show little variance for utterances from a single speaker and

large variance for utterances from different speakers. Wolf [69] outlined a set of desirable feature

attributes; while it is unlikely that any feature set will exhibit all of these qualities, the features should:

- occur naturally and frequently in speech,

- be easy to obtain,

- not change over time or be affected by the speaker's health,

- not be affected by reasonable background noise or depend on specific transmission characteristics,

- and not be susceptible to mimicry.

A.3.3.1   Discriminant Analysis.   A method to determine the best features amongst all

the classes of data is the Fisher's Generalized Discriminant Function (the F-ratio) [17] [46]. Parsons

describes a way to characterize this numerically as taking the ratio of the difference of the means (μ) to the standard deviation (σ) of the measurements, defining Fisher's Discriminant as [46]:

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{18}$$

for two classes. For more than two classes, the Generalized Fisher Discriminant function can be described as the ratio of the between class scatter to the within class scatter. In equation form, for n data samples and c different classes, Parsons gives

$$F = \frac{1/(c-1) \sum_{j=1}^{c} (\mu_j - \bar{\mu})^2}{1/c(n-1) \sum_{j=1}^{c} \sum_{i=1}^{n} (x_{ij} - \mu_j)^2} \tag{19}$$

where x_{ij} = the ith sample for class j, μ_j is the mean of all measurements for class j, and $\bar{\mu}$ is the mean of all measurements over all classes. The F-ratio (Equation 19) reduces to the f-ratio (Equation 18) for the two class case. Features can be rank ordered according to their F-ratio, with high F-ratios corresponding to the better features.

A.3.3.2   The Karhunen-Loève Transform.   To achieve a better separation of classes in the feature space and reduce the number of free parameters in the classifier, it is often desirable to remove the correlation between features. The classifier can then operate on a variance vector (or diagonalized covariance), rather than an entire covariance matrix. The Karhunen-Loève transform (KLT) is one method of diagonalizing a covariance matrix [46]. Given a covariance matrix $\mathbf{W}$, the KLT, described by Parsons [46], rotates the feature vector by the matrix $\mathbf{A}$. That is,

$$\mathbf{y} = \mathbf{A}^T \mathbf{x} \tag{20}$$

where $\mathbf{x}$ is the original set of features and $\mathbf{y}$ is the transformed set. The goal is to find the matrix $\mathbf{A}$ such that

$$\mathbf{C_y} = E\{\mathbf{yy}^T\} = E\{\mathbf{A}^T \mathbf{xx}^T \mathbf{A}\} = \mathbf{A}^T E\{\mathbf{xx}^T\}\mathbf{A} = \mathbf{A}^T \mathbf{WA} \tag{21}$$

giving

$$\mathbf{A}^T \mathbf{WA} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \tag{22}$$

56

then $\mathbf{A}$ accomplishes the rotation, and the λ_i are the eigenvalues of the covariance matrix, $\mathbf{W}$. The elements of $\mathbf{y}$ are uncorrelated, and the λ elements give the variances of the y_i (i.e. $\lambda_i = \sigma_{y_i}^2$).

Based on the variance of the original data, the KLT can be used to reduce the number of features by eliminating those features with low variance. The transformed features are rank ordered according to their variances (by examining the λ_i), keeping those with the largest variance.

A.3.3.3   F-ratio and KLT Drawbacks.   It is worth noting some of the drawbacks of the F-ratio and the KLT [32] [46]:

- Both assume the data (features) are Gaussian distributed.

- Since features are thrown away, some information is inevitably lost. Ideally, this lost information is minimal, with the high ranking features retaining the majority of the information.

- Discriminant analysis is not reliable if the mean vectors are near one another.

- The F-ratio evaluates single features, but not necessarily combinations of features. That is, it is possible that two high F-ratio features in combination perform poorer than expected since their information may be redundant. An effective, but very computationally expensive technique, for finding the best combination of features is the Add-On procedure. Starting with just one feature, all subsets of features are successively applied to the classifier, the subset which performs the best is kept.

- The KLT is computationally expensive.

- The KLT is optimal for signal representation (making it optimal for signal reconstruction), but not necessarily optimal with regard to class separability.

A.3.3.4   Foley's Rule.   Part of finding the best features includes determining how many features should be used to classify the data. Foley's rule-of-thumb [19] provides guidance in this area by stating that if the ratio of the number of samples per class, N, to the number of features, L, is greater than three, then the design set error rate is approximately the test set error rate, and the test set error rate is close to the optimum error rate attained by a Bayes classifier. In equation form, Foley's rule is

$$\frac{N}{L} > 3 \implies \text{Optimal Error Rate.} \tag{23}$$

Since Foley's rule assumes that the data are Gaussian distributed, if less is known about the underlying probability structure, an even greater ratio of $\frac{N}{L}$ should be used [19].

A.4 Classification

The goal in speaker recognition is for the system to make an accurate, reliable decision of an unknown speaker's identity. Classification, the final stage in a closed-set pattern recognition system, is the decision-based process in which the system chooses the most probable or closest matching class to accomplish this goal. In general, classification of a test pattern is based on a minimum distortion measure. The distortion measure is commonly the distance measured between two templates or models, such as the Euclidean distance (which is adequate for cepstral coefficients [49]). There are many classification techniques available. Neural networks, one of the more common pattern recognition techniques, are not specifically discussed here; however, suffice it to say that they too have been used in speaker recognition [64]. Three common methods are discussed below.

A.4.1 Vector Quantization. Vector Quantization (VQ) is a form of unsupervised learning. Ideally, the feature space consists of small clusters each formed by repetitions of a speaker's utterances, with the different speakers' clusters widely separated. Vector quantization (VQ) (or clustering) allows for a cluster of data to be represented by a single vector and is therefore a useful means of reducing the amount of data. By vector quantizing the training data (referred to as creating a codebook) the pattern classifier needs only to compare the test sample to the representative cluster centers (a.k.a. codewords), rather than the entire training set of data for classification. Thus, classification entails finding the minimum distortion between an unknown test speaker's utterance and the set of reference speakers' codebooks. In their classic article on VQ, Linde, Buzo, and Gray [35] describe the algorithms for vector quantizing data which is distributed in either a known or unknown distribution. The Generalized Lloyd Algorithm is commonly referred to as the LBG Algorithm, which is similar to a *k*-means algorithm, except that the distortion measure is general (e.g. for LP coefficients, the Itakura-Saito distortion measure may be optimal).

The cluster centers are formed by iteratively computing the nearest neighbor distance (from the data to the cluster center). This iterative method terminates in a local minimum when the average distortion (based on the distance from the cluster centers to the data points within the clusters) stops
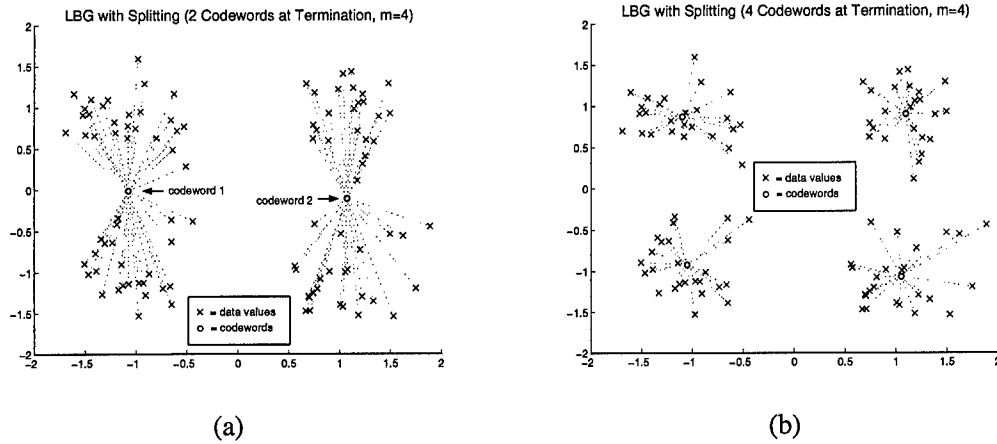
Figure 16. An example of the LBG Algorithm (initialized with splitting) for clustering synthetically generated, unlabeled data, illustrating (a) two codewords (cluster centers), and (b) four codewords. The dotted lines connecting the data points to the codewords form the represented clusters, and m represents the number of iterations.

changing significantly. Linde *et al* [35] discuss a splitting method to initialize the codebook, whereby the LBG Algorithm is applied at each power of two (giving codebook sizes of 1, 2, 4, 8, 16...). Figure 16 is a two-dimensional example of the clustering capability of of the LBG Algorithm using Linde *et al*'s splitting technique on synthetically generated, unlabeled data. Figure 16(a) shows the codeword locations and the clusters formed (dotted lines) of unlabeled data for two cluster centers. In Figure 16(b), the two cluster centers were split into four clusters centers, resulting in a re-clustering of the data for the new codeword locations. As one can see, more codewords can achieve a finer representation of the data; however, in the limit (where the number of codewords equals the number of data points), nothing is achieved by clustering the data.

A.4.2 Dynamic Time Warping. Dynamic Time Warping (DTW) is a form of temporal signal classification. Since speech signals are commonly divided into short, overlapping frames, timing can become important. The segmentation of utterances into meaningful units (e.g., phones) is difficult; thus, templates are usually compared frame-by-frame, which can lead to alignment problems. Linear time normalization is insufficient to treat this problem because the effects of speaking rate changes are nonlinear [43]. A procedure used to address the problem of alignment, called dynamic time warping (DTW), nonlinearly warps one template in an attempt to align similar acoustic segments in the test and
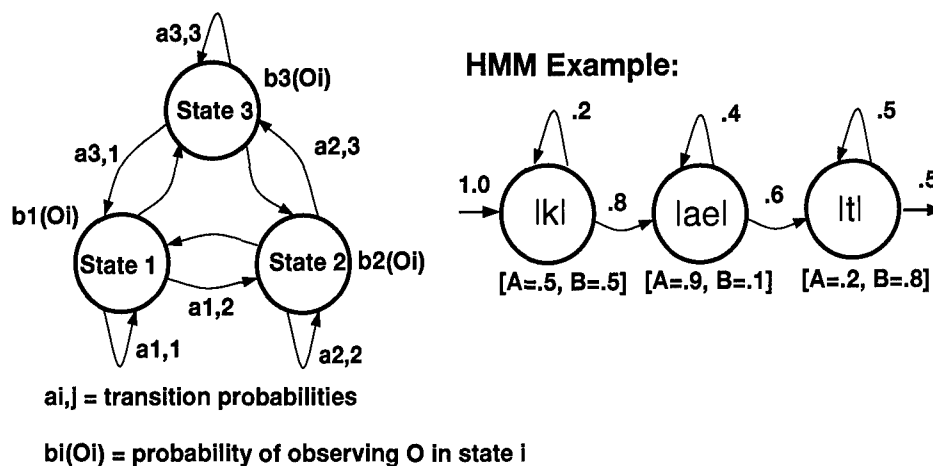
59

Figure 17. Two Hidden Markov Models shown for illustrative purposes. On the left is a fully connected, ergodic HMM; on the right is a left-to-right HMM.

reference templates [13] [50]. DTW combines alignment and distance computation through a dynamic programming procedure [13] [43].

A.4.3 Hidden Markov Models. Another form of temporal signal classification is the Hidden Markov Model (HMM). While DTW creates a template and VQ creates a codebook to represent the training data, an HMM creates a statistical model of the training data which retains information about the distribution of the training data [57]. An HMM is said to be "hidden" because one cannot directly observe which state the model is in – only the features produced by that state. HMMs attempt to identify the steadily or distinctively behaving periods of a signal, then characterize the sequentially evolving nature of the periods, and choose the best model for the periods [48]. Rabiner and Juang define a hidden Markov model as "a doubly stochastic process with an underlying stochastic process that is not observable (i.e. hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [[48]:5]."

Figure 17 shows two types of HMMs. The fully connected HMM, referred to as an ergodic model, shows three states, the state transition probabilities a_{ij} and the probabilities of observing a feature for a given state $b_i(O_i)$. The left-to-right model also shown in Figure 17 is more applicable to time-varying signals such as speech.

For the HMM to be useful in application, three problems must be solved [47] [49]:

1. **Compute** $P(O|\lambda)$. The first problem, referred to as the evaluation problem, deals with evaluating or "scoring" the model. In other words, given a model and a sequence of observations, one must determine the probability that the observed sequence was produced by the model. In solving this problem, one must choose the model that best matches the observations. In other words, the probability of the observation sequence O, given the model λ, $P(O|\lambda)$, must be calculated. The preferred method of calculating $P(O|\lambda)$ is the Forward-Backward Procedure since it reduces the computational complexity (compared to direct computation) by several orders of magnitude.

2. **Find the Optimal State Sequence.** The second problem deals with determining the most likely state sequence (the hidden part of the model) which led to the sequence of observations (i.e. finding the optimal state sequence associated with the given observation sequence). The preferred solution to this problem is the Viterbi Algorithm since direct computation may result in a sequence that does not exist.

3. **Training.** The third problem deals with optimizing the model parameters to best describe the observed sequence. Here, the model is trained to optimally adapt the model's parameters, based on training data. Essentially, the model parameters are adjusted to maximize the probability of the observation sequence, given the model. The solution to this problem is the application of the Baum-Welch Re-estimation Algorithm.

The disadvantage of HMMs is that they require extensive training to develop accurate models; however, their recall is fast. Also, the HMM is designed to model the signal, not specifically to discriminate or classify.

A.5 Speech Corpora and the Channel

In their development, speaker recognition systems are trained and validated using various speech databases, such as the TIMIT corpus mentioned earlier. TIMIT's qualities are that it consists of many speakers (630 speakers each stating 10 sentences), its speech utterances are continuous, and its speakers come from a variety of North American dialects. Its drawback is that it was produced in a low background noise, or clean environment. In a real-world environment, the signal is often corrupted in a variety of ways as it propagates through the channel. In this context, the channel may simply be the telephone system (which not only band-limits the signal, but also adds noise) or a radio

transmission system (with the effects of the signal propagating through free-space). Rome Laboratory's GREENFLAG corpus [66] consists of 41 speakers recorded during an Air Force exercise. The 255 utterances are text-independent (however call-signs are frequently used) and some were recorded on different days. The utterances consist of tactical communications over RF channels, originally sampled at 48 kHz, and later re-sampled at 8 kHz.

Noise can significantly degrade the performance of an speaker recognition system developed solely for noise-free environment. Thus, a speaker recognition system's design must be robust; thereby enabling it to operate in the presence of noise. The GREENFLAG corpus is well suited for testing a system on utterances degraded by noise. There are a variety of alternative approaches to incorporate the effects of a channel in designing a speaker recognition system. One approach, while logistically demanding, is to create the speech database by recording signals which have propagated through the channel. NYNEX Science and Technology created the NTIMIT corpus [2], for example, by transmitting all 6300 original TIMIT utterances through various channels in the NYNEX telephone network. A comparison of the TIMIT and NTIMIT versions of a sentence is shown in Figure 18. Notice that the NTIMIT sentence's spectrum shows the band-limiting effects of the telephone channel. Similarly, Lockheed Sanders produced the cellular-TIMIT or CTIMIT corpus [10] which consists of the TIMIT utterances transmitted over a cellular network. Another approach is to model the channel. Reynolds, for example, compensated for noise with an integrated speech-background model [52]. In this case, the effects of a simulated channel are applied to a clean utterance. Watterson *et al* [67] designed a stationary HF ionospheric channel model validated for bandwidths ranging from 2.5 kHz (nighttime) to 12 kHz (daytime). In Watterson's model, the input signal feeds an ideal delay line and is delivered at several taps with adjustable delays, which represent the ionospheric modal component. To obtain the output signal, each delayed signal is modulated (in amplitude and phase), then the delayed and modulated signals are summed.

A.6 Example 1: Closed-Set Speaker Identification

This section provides an example of applying some of the speaker recognition topics discussed in the previous sections. The software used to obtain these results (and those in the next example) makes maximal use of readily available UNIX based packages, such as: Matlab and the Entropics

Speaker: fcfj0
"She had your dark suit in greasy wash water all year."
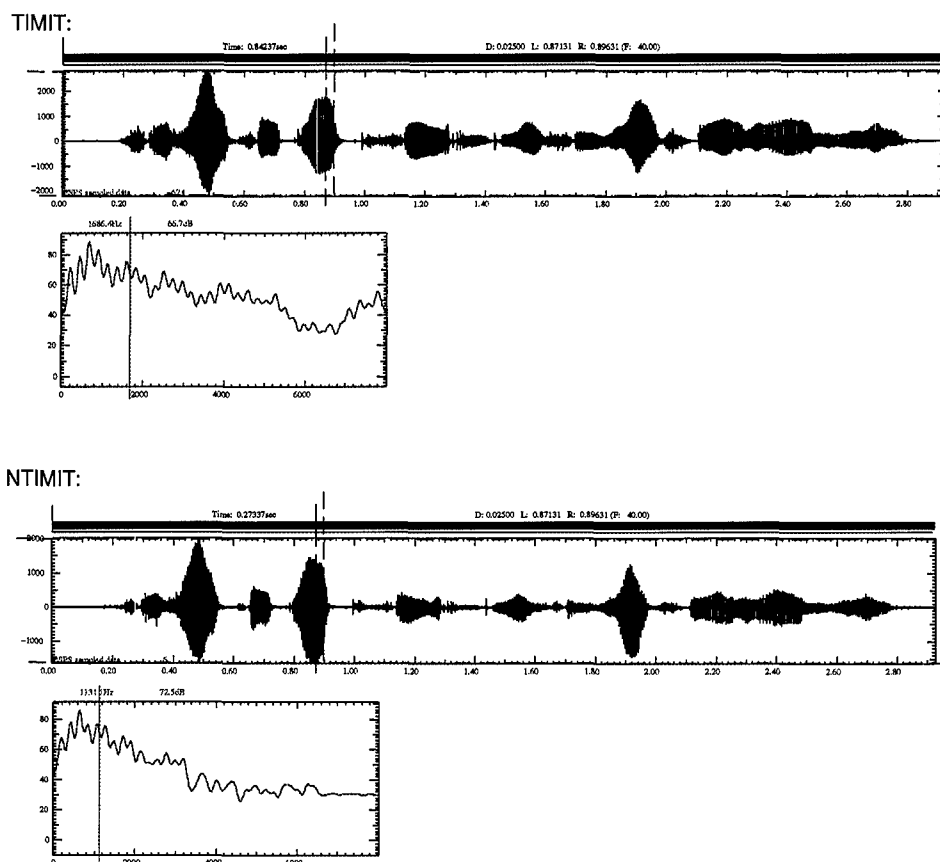
TIMIT:

NTIMIT:

Figure 18.   The TIMIT and NTIMIT versions of the same sentence. Also shown are the spectra, taken from a short time segment of the word "dark." Note the band-limiting effects of the telephone channel, shown in the spectrum of the NTIMIT utterance.

63

Table 10. Summary of Speaker Identification Results, using mel-warped linear prediction cepstral coefficients (MLPCC) and mel-warped cepstral coefficients derived from the real FFT (MFCC) for TIMIT speech.

Dialect Region	Number of Speakers	MLPCC		MFCC	
		# Test Errors	% Accuracy	# Test Errors	% Accuracy
1. Northeast	10	1	90	2	80
2. Northern	10	2	80	1	90
3. North Midland	10	0	100	1	90
4. South Midland	10	2	80	2	80
5. Southern	10	0	100	1	90
6. New York City	10	0	100	1	90
7. Western	10	0	100	2	80
8. Army Brat	10	0	100	1	90
TOTALS	80	5	94	11	86

Signal Processing System (ESPS) and its Hidden Markov Model Tool Kit (HTK). The intent in this example is primarily to illustrate the results of text-independent, closed-set speaker identification.

A.6.1 TIMIT.   Table 10 shows the results for classifying 80 TIMIT speakers (10 speakers from the eight dialect regions, one test utterance per speaker). Each speaker's codebook size is 32 codewords, and the features are 10th order mel-warped linear prediction cepstral coefficients (MLPCC) and mel-warped cepstral coefficients derived from the real FFT (MFCC), obtained by using the first 10 real coefficients from a 1024 point FFT. The mel-warping was done in ESPS with the bilinear transform ($a = 0.5$). The classifier was trained on the *sa1.sd* sentence "She had your dark suit in greasy wash water all year.", and tested on the *sa2.sd* sentence, "Don't ask me to carry an oily rag like that."

In listening to the speakers who resulted in mis-classification, one can understand why they were mis-classified. For example, in Dialect Region 1, speaker *mdpk0* was classified as speaker *mdac0*. For the training sentence, *mdpk0* spoke very clearly, enunciating every phoneme (especially the liquid /r/). For the test sentence, however, *mdpk0*'s New England accent was evident, and he actually did sound more like *mkac0*.

A.6.2 NTIMIT.   Similar tests, with similar parameters, were run using the NTIMIT corpus. It is quite evident from the results shown in Table 11 that noise can have a severe impact on a speaker identification system. Also, in general, it took more iterations for the VQ codebooks to converge for the NTIMIT speech.

Table 11. Summary of Speaker Identification Results, using mel-warped linear prediction cepstral coefficients (MLPCC) and mel-warped cepstral coefficients derived from the real FFT (MFCC) for NTIMIT speech.

Dialect Region	Number of Speakers	MLPCC		MFCC	
		# Test Errors	% Accuracy	# Test Errors	% Accuracy
1. Northeast	10	6	40	7	30
2. Northern	10	6	40	6	40
3. North Midland	10	4	60	5	50
4. South Midland	10	6	40	5	50
5. Southern	10	4	60	7	30
6. New York City	10	6	40	6	40
7. Western	10	5	50	4	60
8. Army Brat	10	2	80	6	40
TOTALS	80	39	51	46	42

A.6.3 Summary of Results. These experiments used an accumulation of small populations from the TIMIT and NTIMIT corpora (i.e. 10 speakers each from within the same dialect region and one utterance per speaker). The results shown in Tables 10 and 11 illustrate that noise can have a major effect on a speaker recognition system.

A.7 Example 2: Open-Set Speaker Recognition

This section provides an example of open-set speaker recognition using the GREENFLAG corpus and a Gaussian classifier[1]. For the open-set task, two errors ($\Pr(FalseAccept)$ and $\Pr(FalseReject)$), which contribute to the overall classification error, are noted. The balance between $\Pr(FalseAccept)$ and $\Pr(FalseReject)$ is controlled by a threshold value, θ. If the classification is within the threshold, it is accepted; thus, mis-classified utterances can be falsely accepted. Alternatively, a correct classification can be falsely rejected. Often, a design criteria is to find a value of θ where $\Pr(FalseAccept)$ equals $\Pr(FalseReject)$, referred to as the equal error rate.

The features used in this example are 12th order liftered linear prediction cepstral coefficients, appended by normalized log energy, averaged over all frames per utterance. The classifier was trained on all of the utterances. Figure 19 shows open-set classification results using a Gaussian classifier on 10 randomly chosen speakers from the GREENFLAG corpus. The average results from 1000 iterations for each threshold value, θ, show that for an equal error rate and highest accuracy, $\theta \approx 28$.

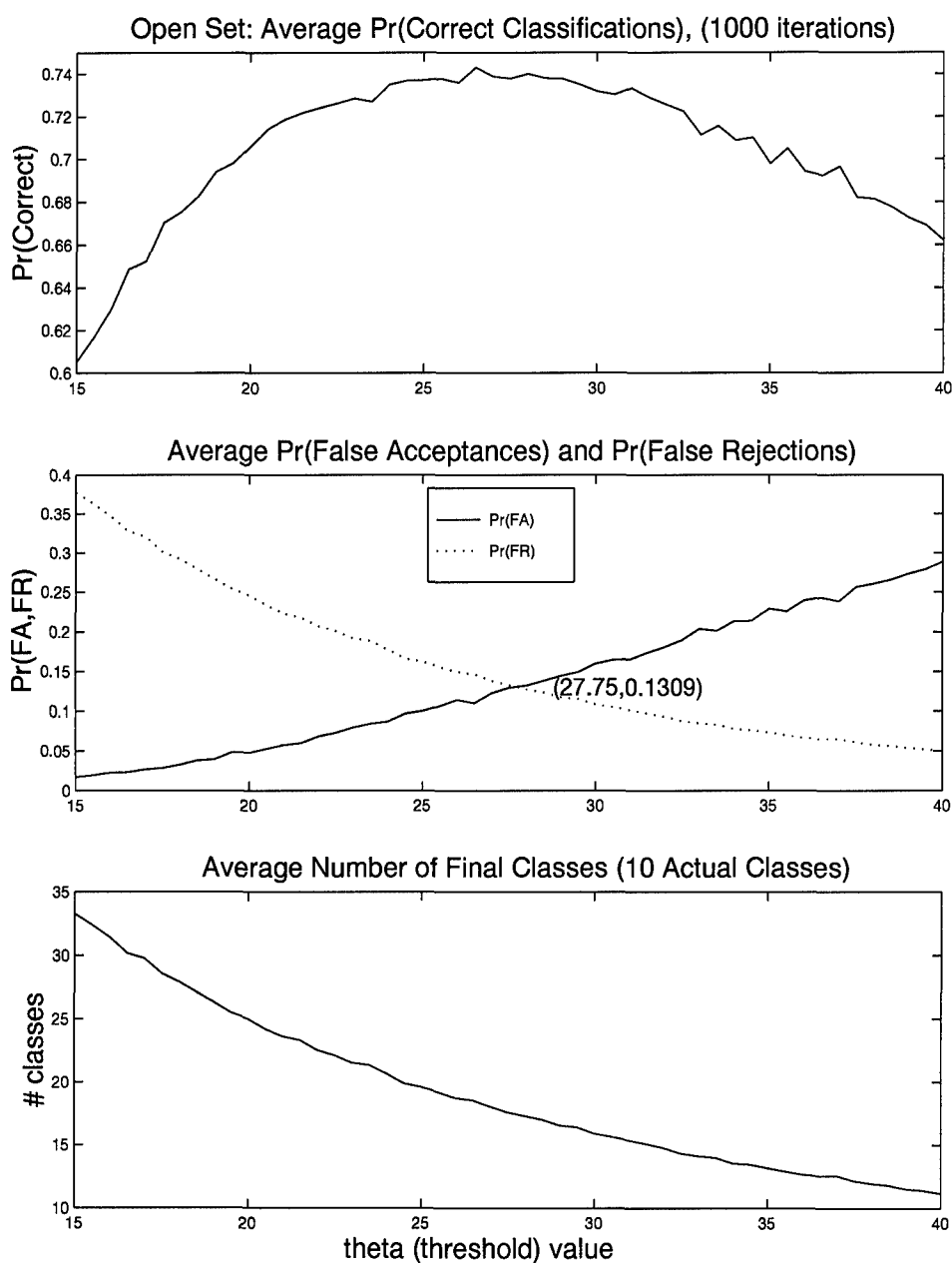[1] Thanks to Maj Ruck for designing this Gaussian classifier.

Figure 19. Results of Open-Set Speaker Recognition with a Gaussian Classifier using the GREEN-FLAG corpus. Based on the average probability of correct classifications (top) and average probabilities of false acceptances and false rejections (middle), an optimal value of theta would be $\theta \approx 28$. Also note this system's ability to adaptively add new speakers (bottom), which corresponds to the $\Pr(FalseRejections)$.

A.8   Conclusion

This appendix provides an overview of speaker recognition, reviewing some of the more pertinent subjects and techniques in the areas of speech analysis, signal processing, and pattern recognition which apply to automatic speaker recognition.

Appendix B. Detailed Results

B.1 Introduction

This appendix provides a more complete view of the results of this effort. Feature analysis results are first provided for both the TIMIT and GREENFLAG corpora. Then, the results of testing the proposed open-set speaker recognition system, a fuzzy classifier followed by hypothesis testing, are presented for both corpora.

B.2 Feature Analysis

The following plots provide the results, presented in the order that the features were ranked according to Table 6, for the TIMIT and GREENFLAG corpora.
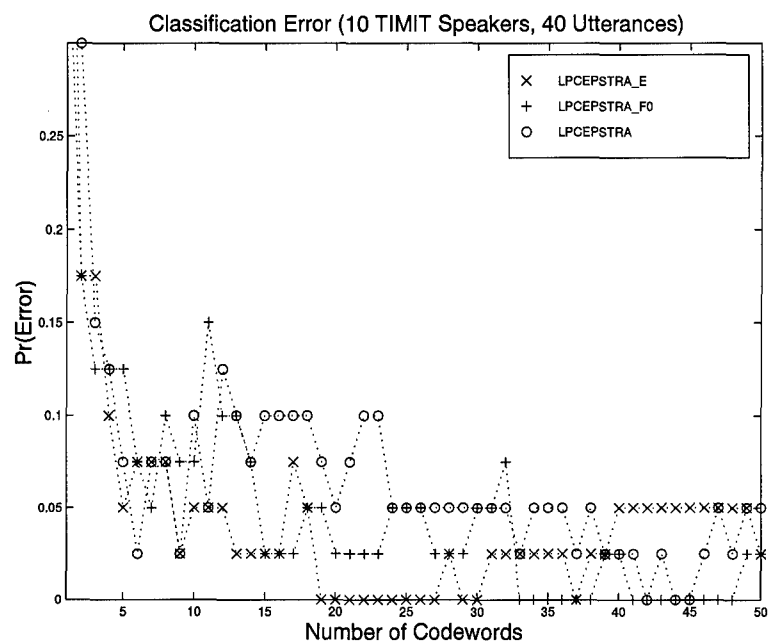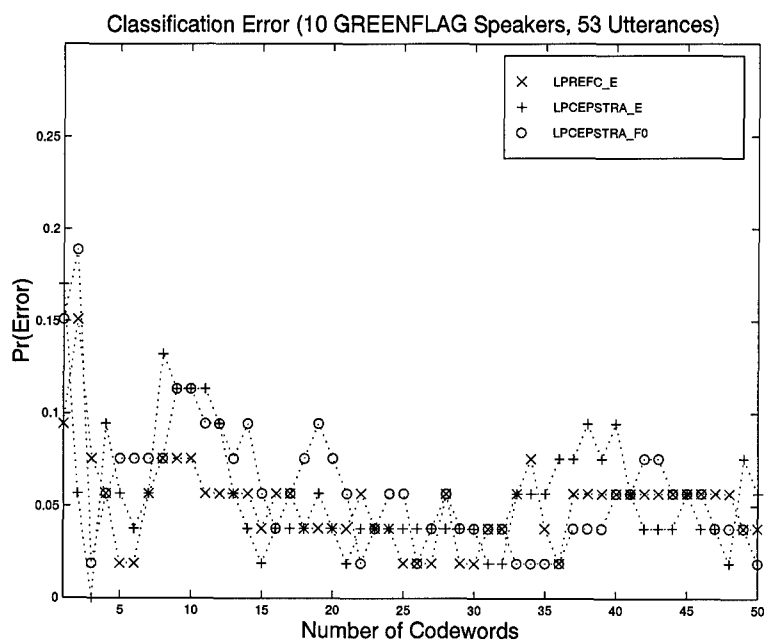
Figure 20. Features Ranked 1, 2, and 3. (a) TIMIT Features: LPCEPSTRA_E, LPCEPSTRA_F0, and LPCEPSTRA. (b) GREENFLAG Features: LPREFC_E, LPCEPSTRA_E, and LP-CEPSTRA_F0.
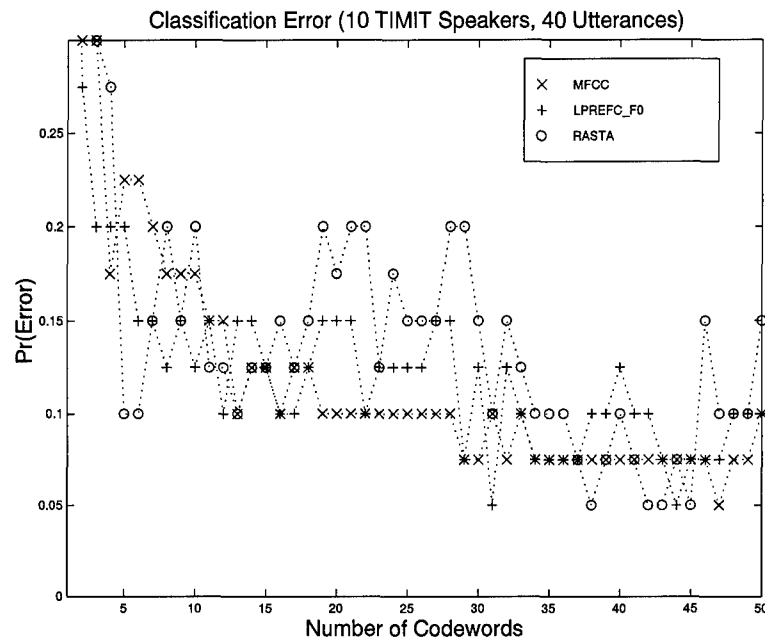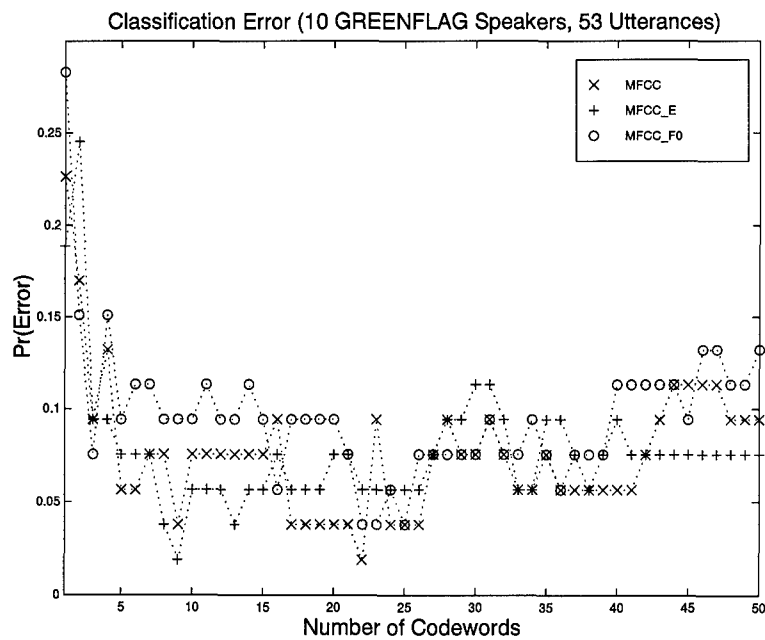
69
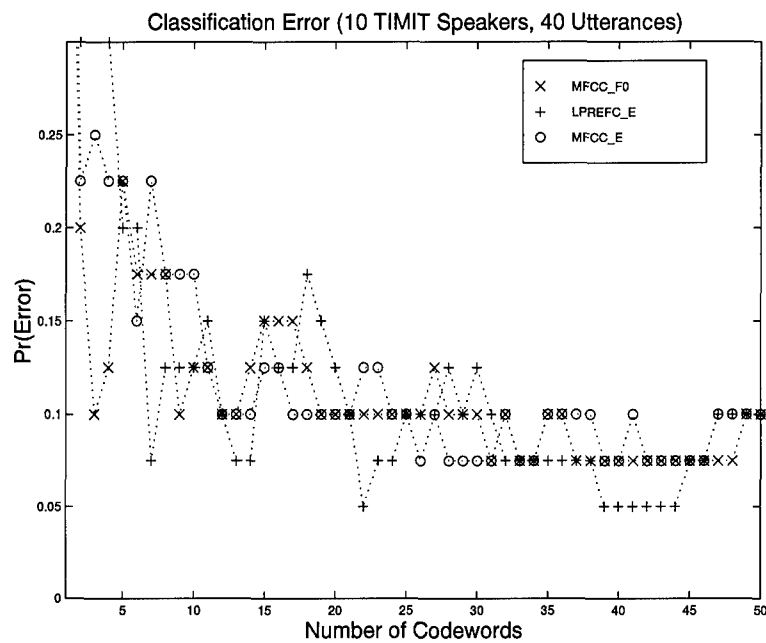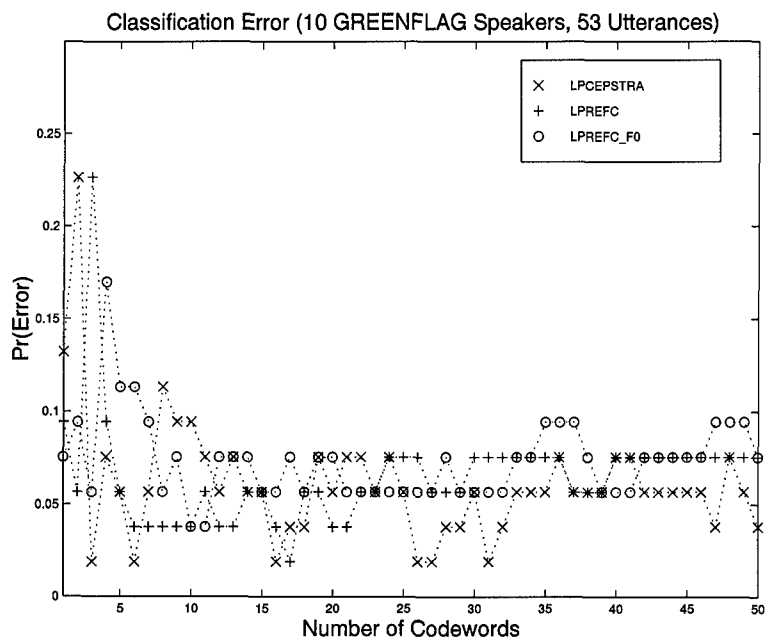
Figure 21.   Features Ranked 4, 5, and 6. (a) TIMIT Features: MFCC_F0, LPREFC_E, and MFCC_E.
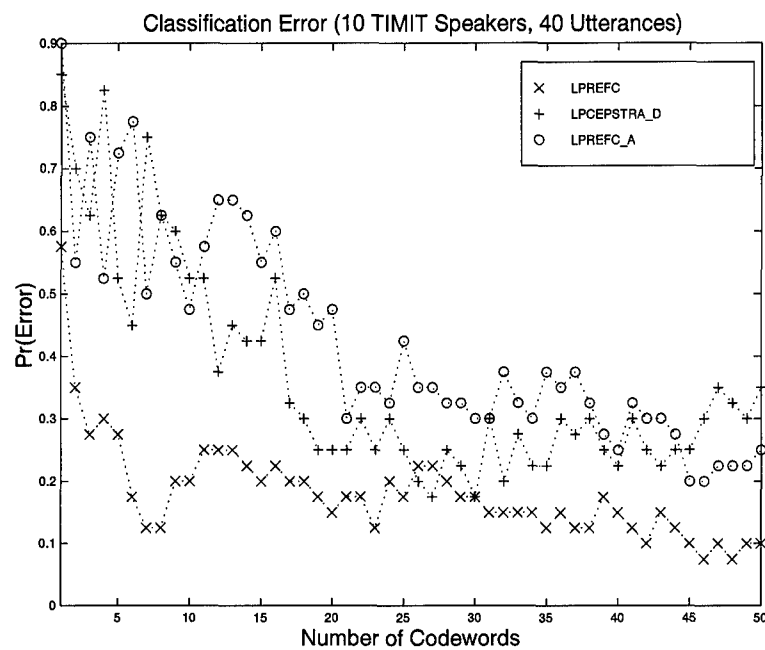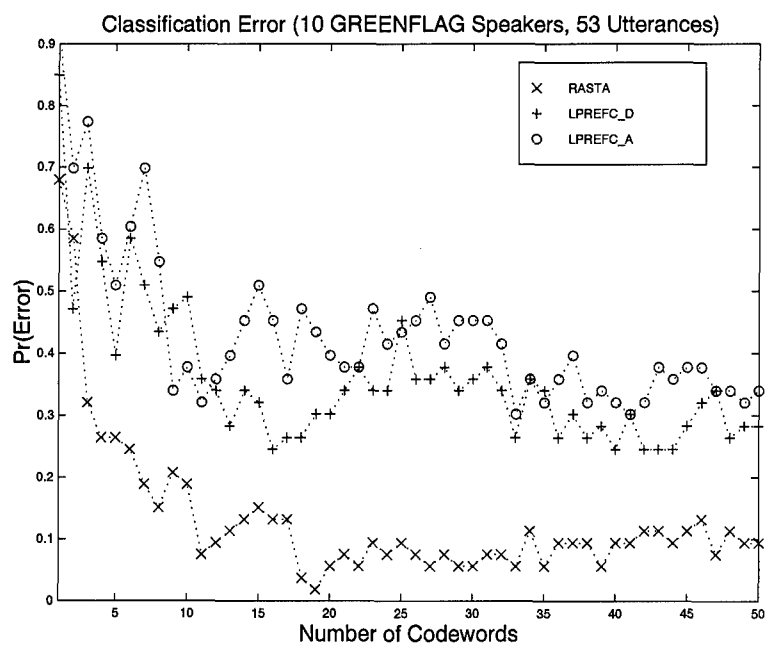(b) GREENFLAG Features: LPCEPSTRA, LPREFC, and LPREFC_F0.

Figure 22.   Features Ranked 7, 8, and 9. (a) TIMIT Features: MFCC, LPREFC_F0, and RASTA. (b) GREENFLAG Features: MFCC, MFCC_E, and MFCC_F0.
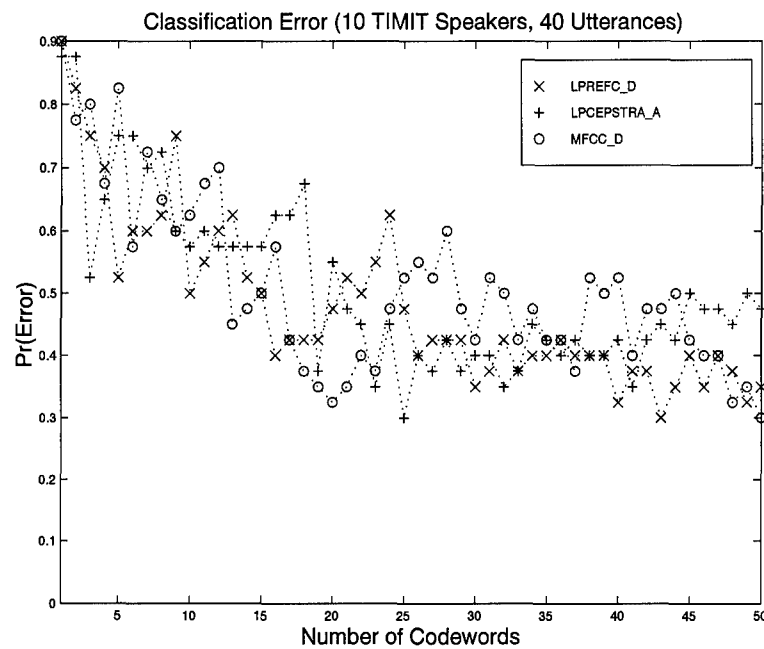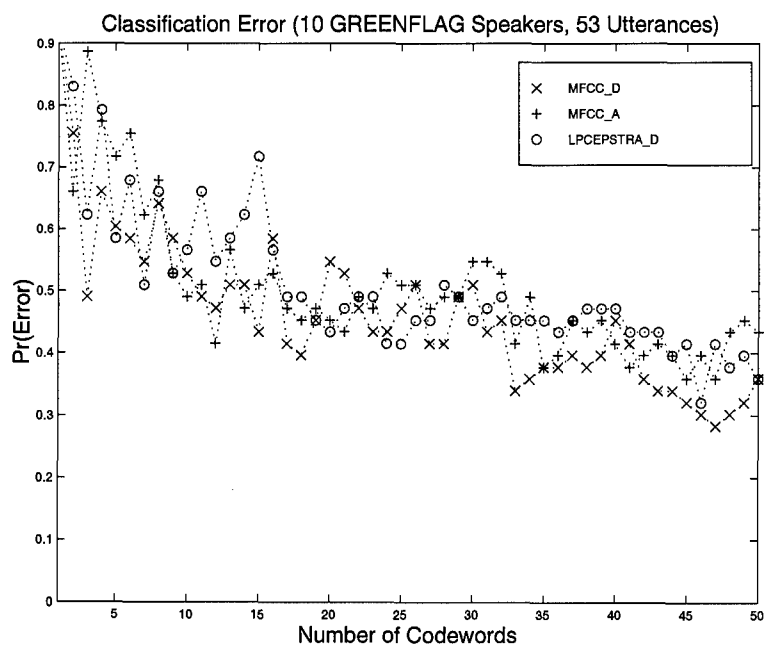
Figure 23. Features Ranked 10, 11, and 12. (a) TIMIT Features: LPREFC, LPCEPSTRA_D, and LPREFC_A. (b) GREENFLAG Features: RASTA, LPREFC_D, and LPREFC_A.

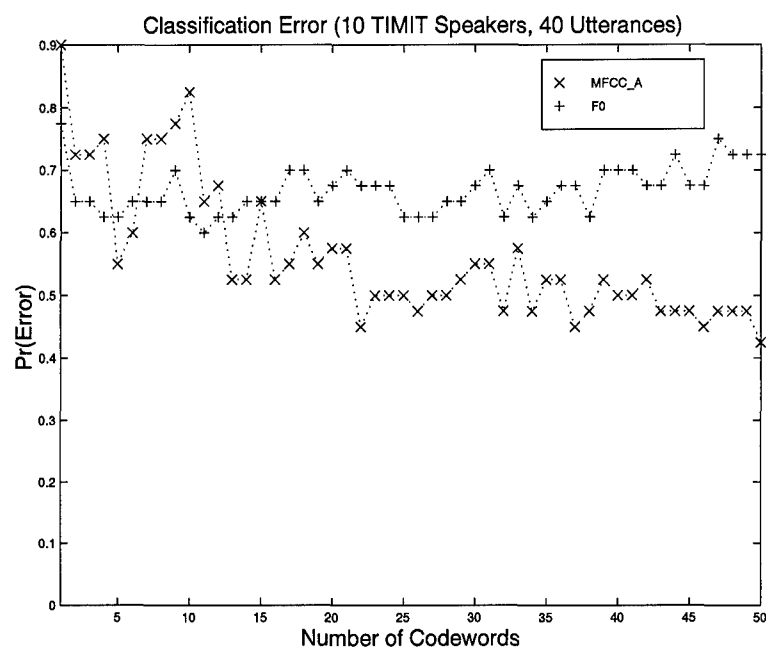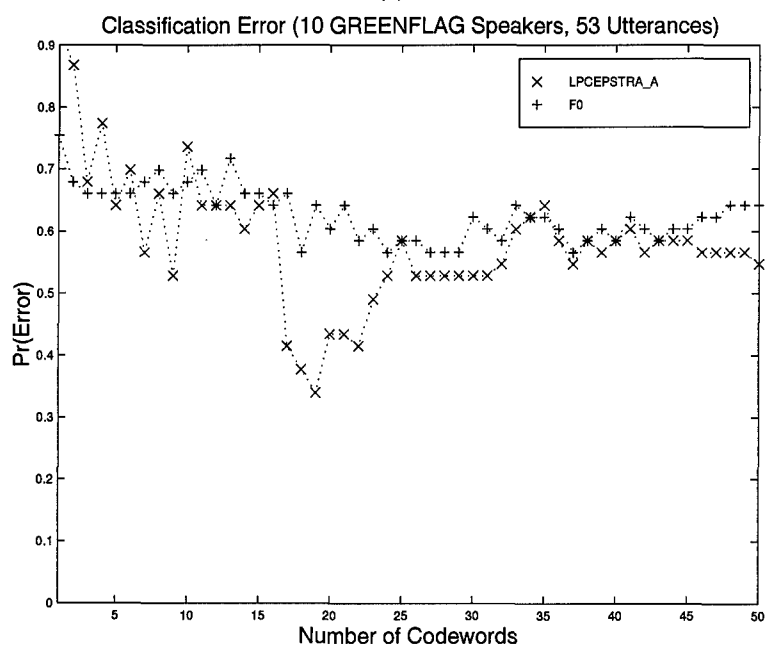Figure 24.   Features Ranked 13, 14, and 15. (a) TIMIT Features: LPREFC_D, LPCEPSTRA_A, and MFCC_D. (b) GREENFLAG Features: MFCC_D, MFCC_A, and LPCEPSTRA_D.

73

Figure 25. Features Ranked 16 and 17. (a) TIMIT Features: MFCC_A and F0. (b) GREENFLAG Features: LPCEPSTRA_A and F0.

B.3 Open-Set Speaker Recognition

The following plots provide the results of testing the proposed text-independent, open-set speaker recognition system for each TIMIT Dialect Region and three arbitrary groups of GREENFLAG speakers.

Figure 26.   Open-Set Speaker Recognition Results for TIMIT, Dialect Region 1, training on 10
speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set
speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-
Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy
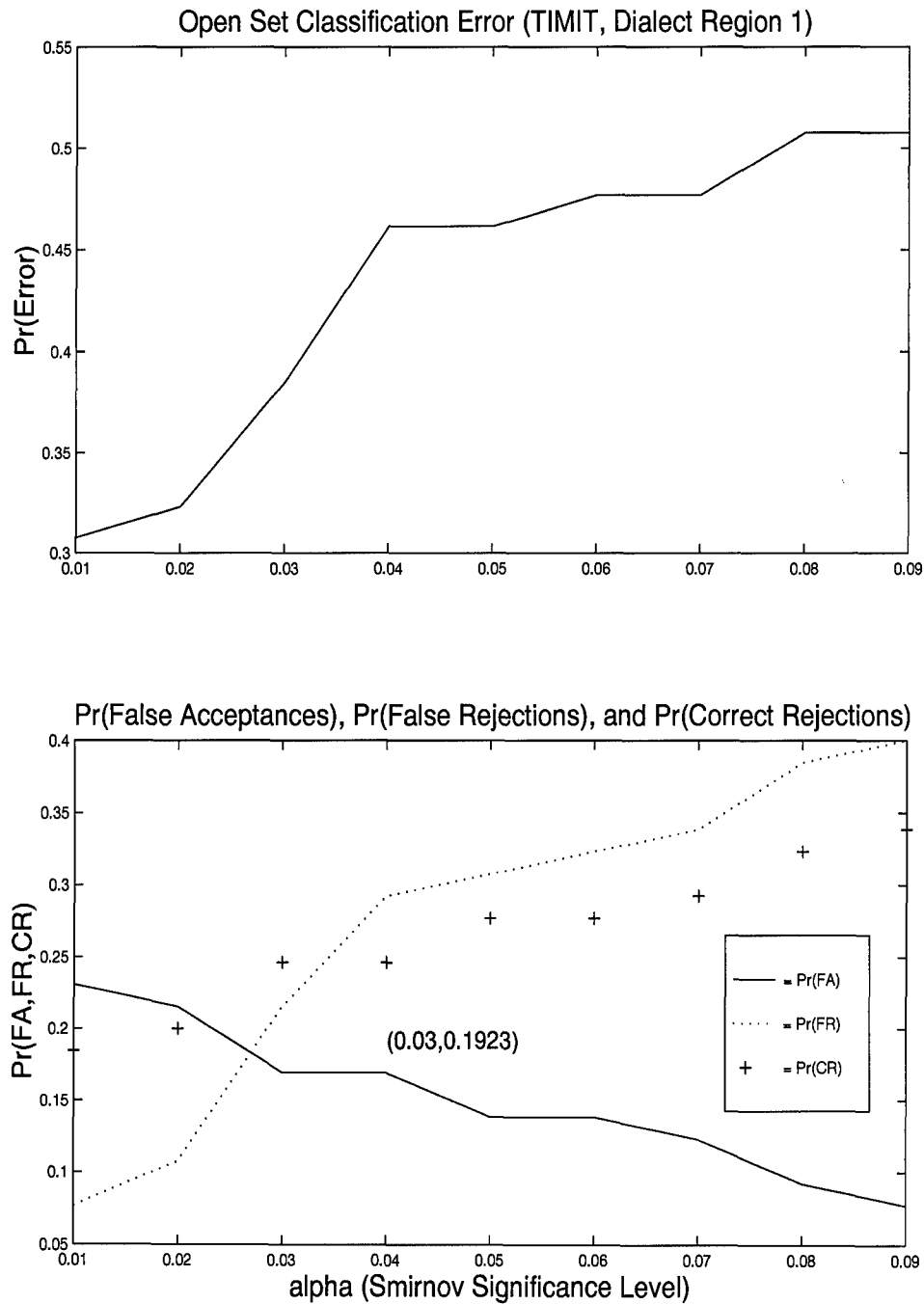classifier followed by the Smirnov Test for Common Distributions.

76

Figure 27. Open-Set Speaker Recognition Results for TIMIT, Dialect Region 2, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.

77

Figure 28. Open-Set Speaker Recognition Results for TIMIT, Dialect Region 3, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
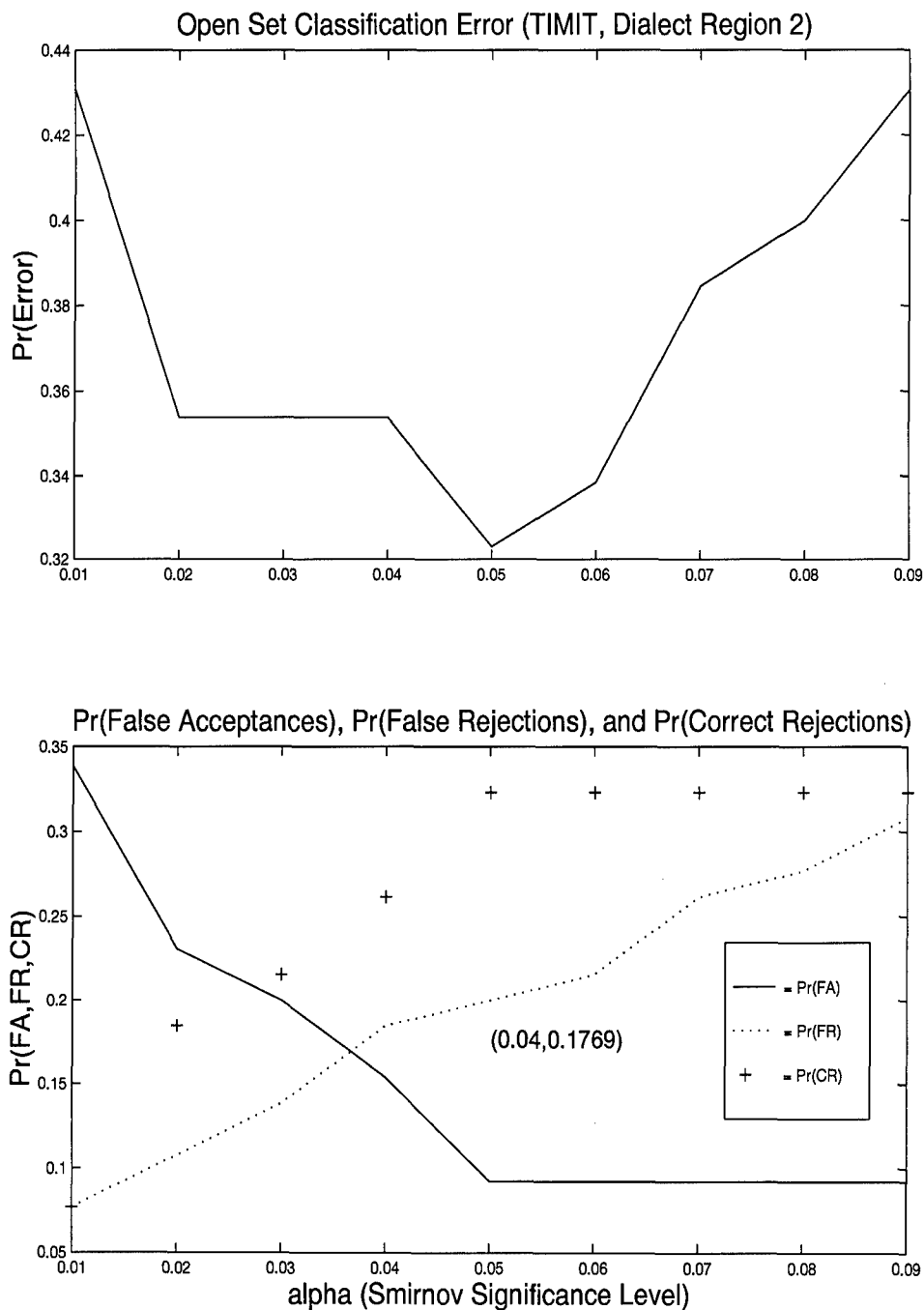
Figure 29. Open-Set Speaker Recognition Results for TIMIT, Dialect Region 4, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
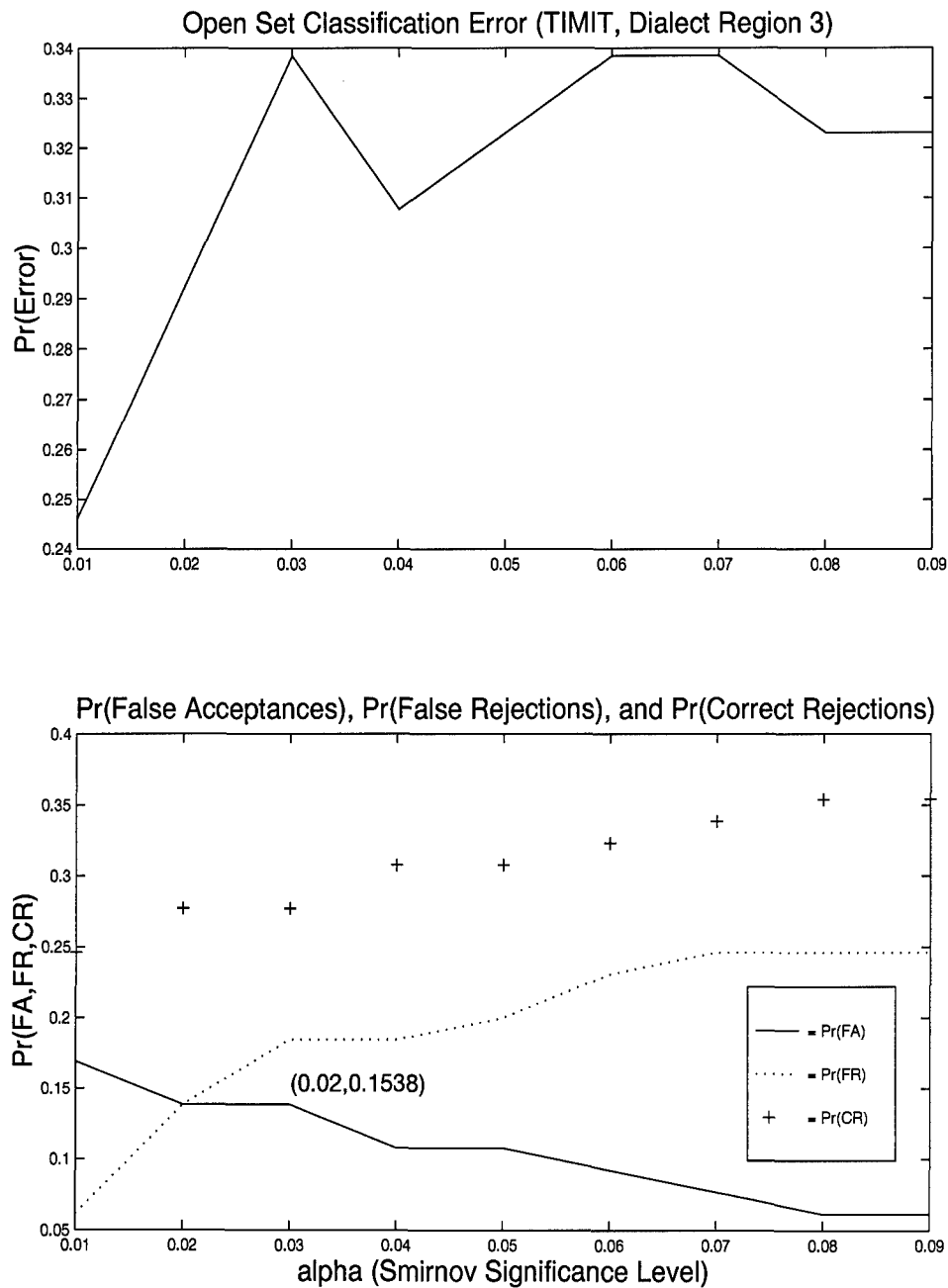
Figure 30.   Open-Set Speaker Recognition Results for TIMIT, Dialect Region 5, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
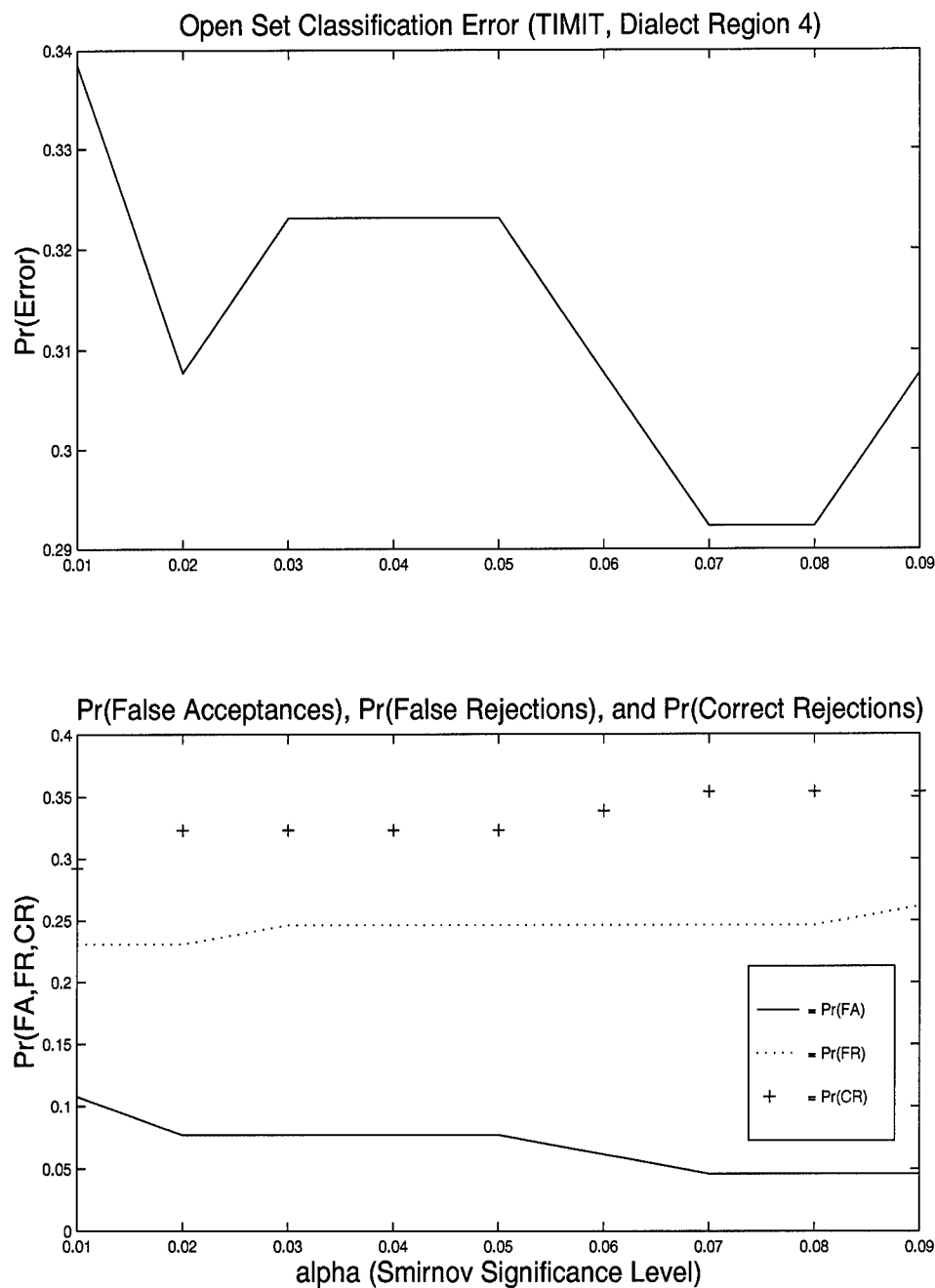
Figure 31.   Open-Set Speaker Recognition Results for TIMIT, Dialect Region 6, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.

81
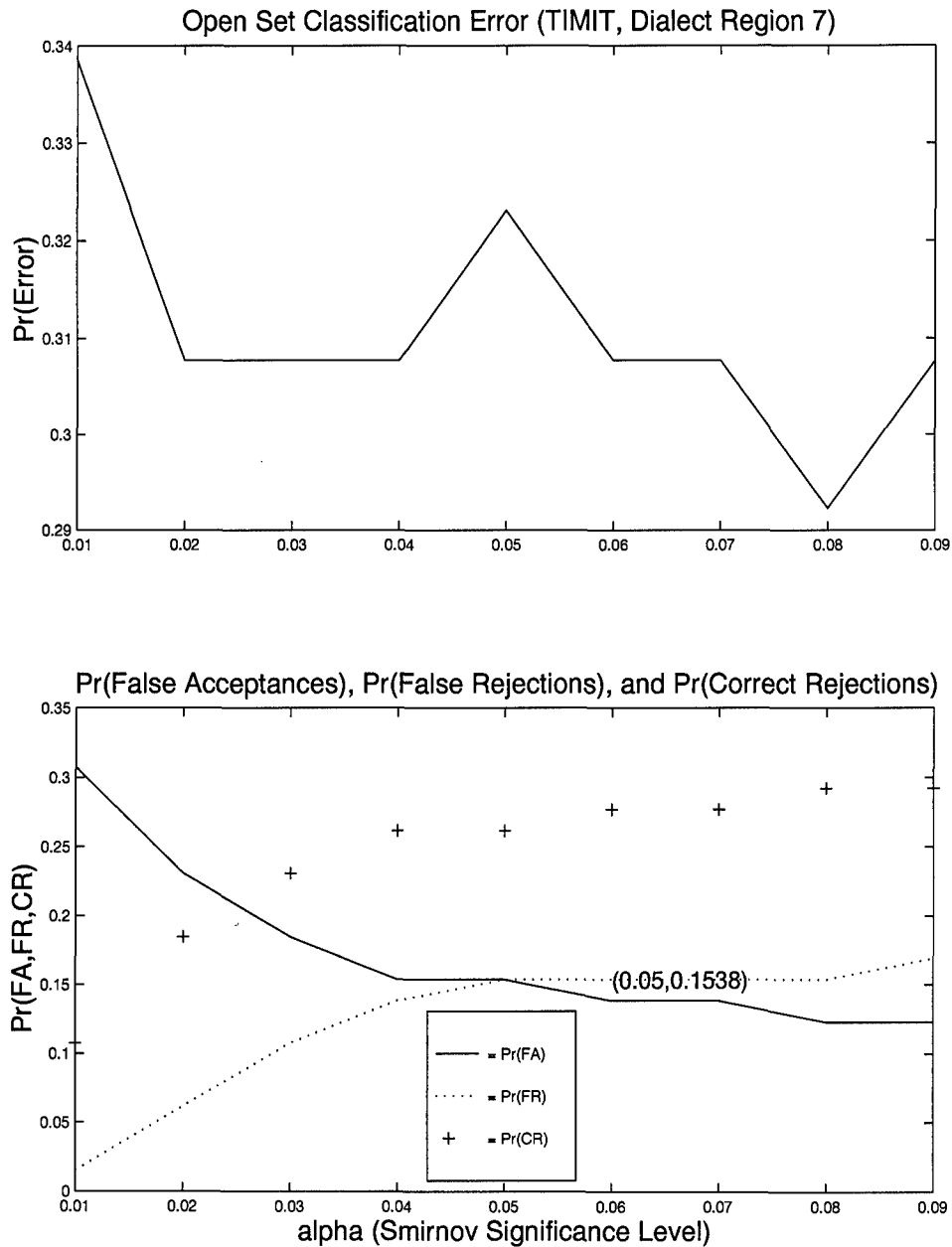
Figure 32. Open-Set Speaker Recognition Results for TIMIT, Dialect Region 7, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
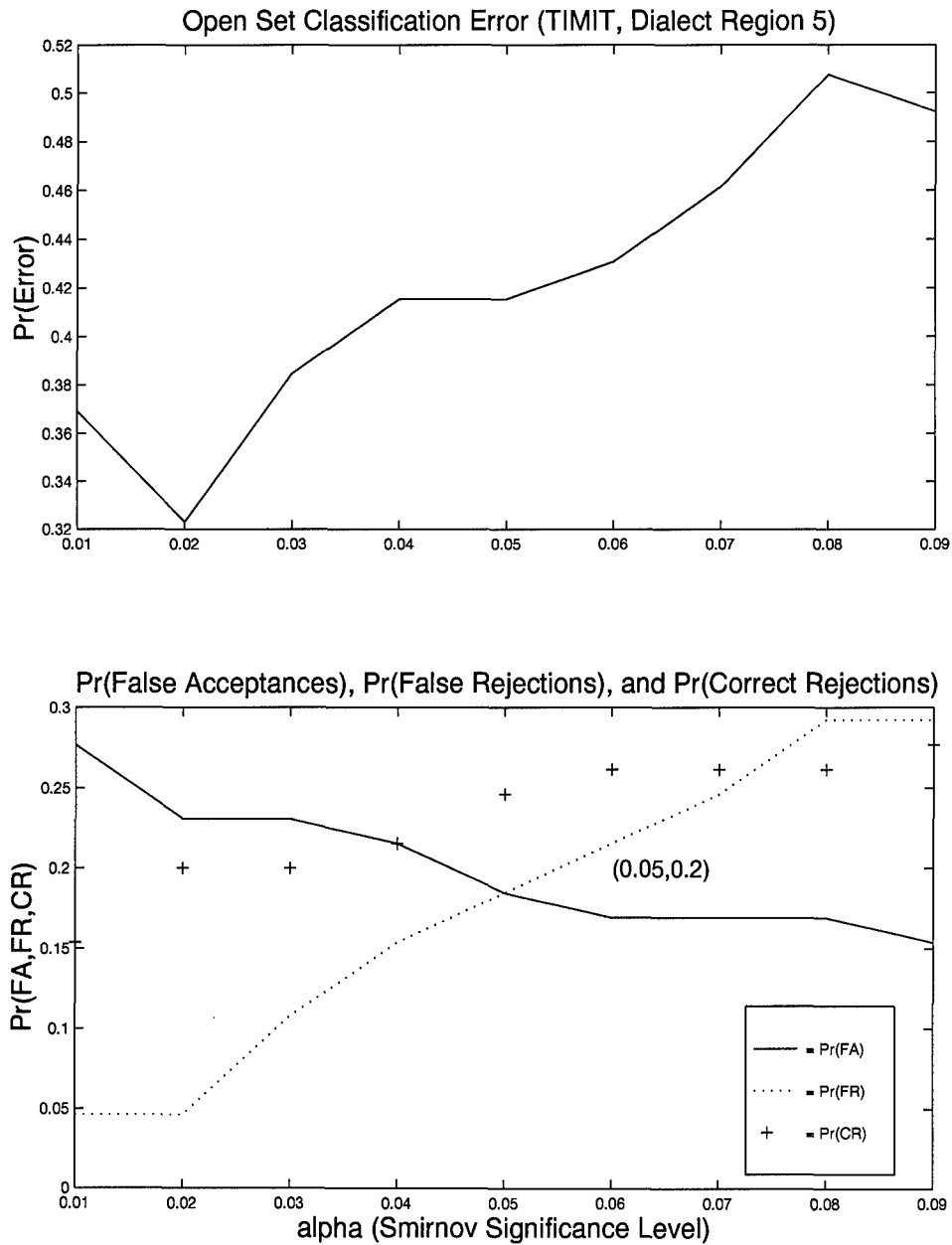
Figure 33. Open-Set Speaker Recognition Results for TIMIT, Dialect Region 8, training on 10 speakers, testing on 15 (the 10 for training, plus five), giving 65 test utterances. Open-set speaker recognition was accomplished using the LPCEPSTRA_E feature set, Karhunen-Loève initialization followed by the LBG Algorithm (10 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
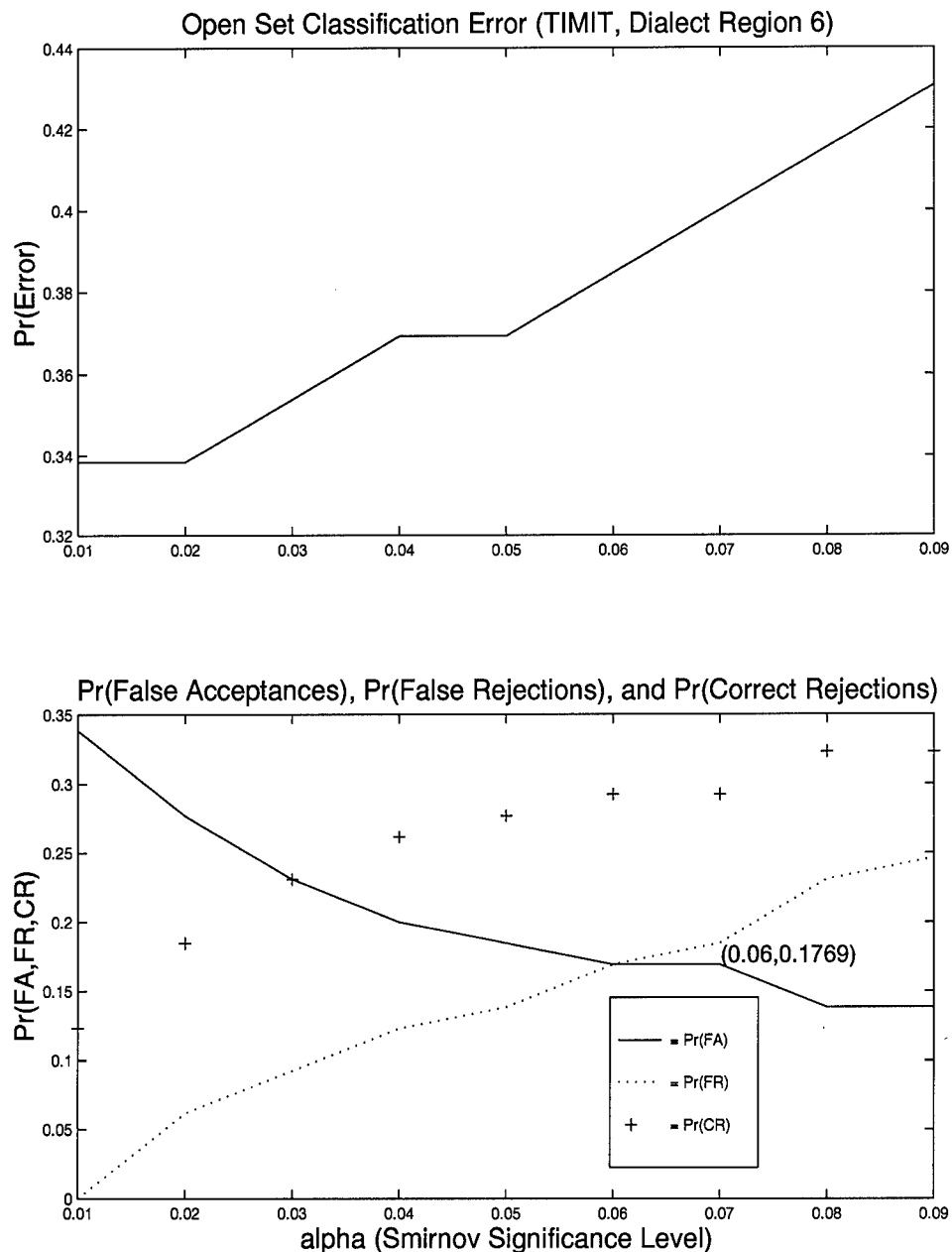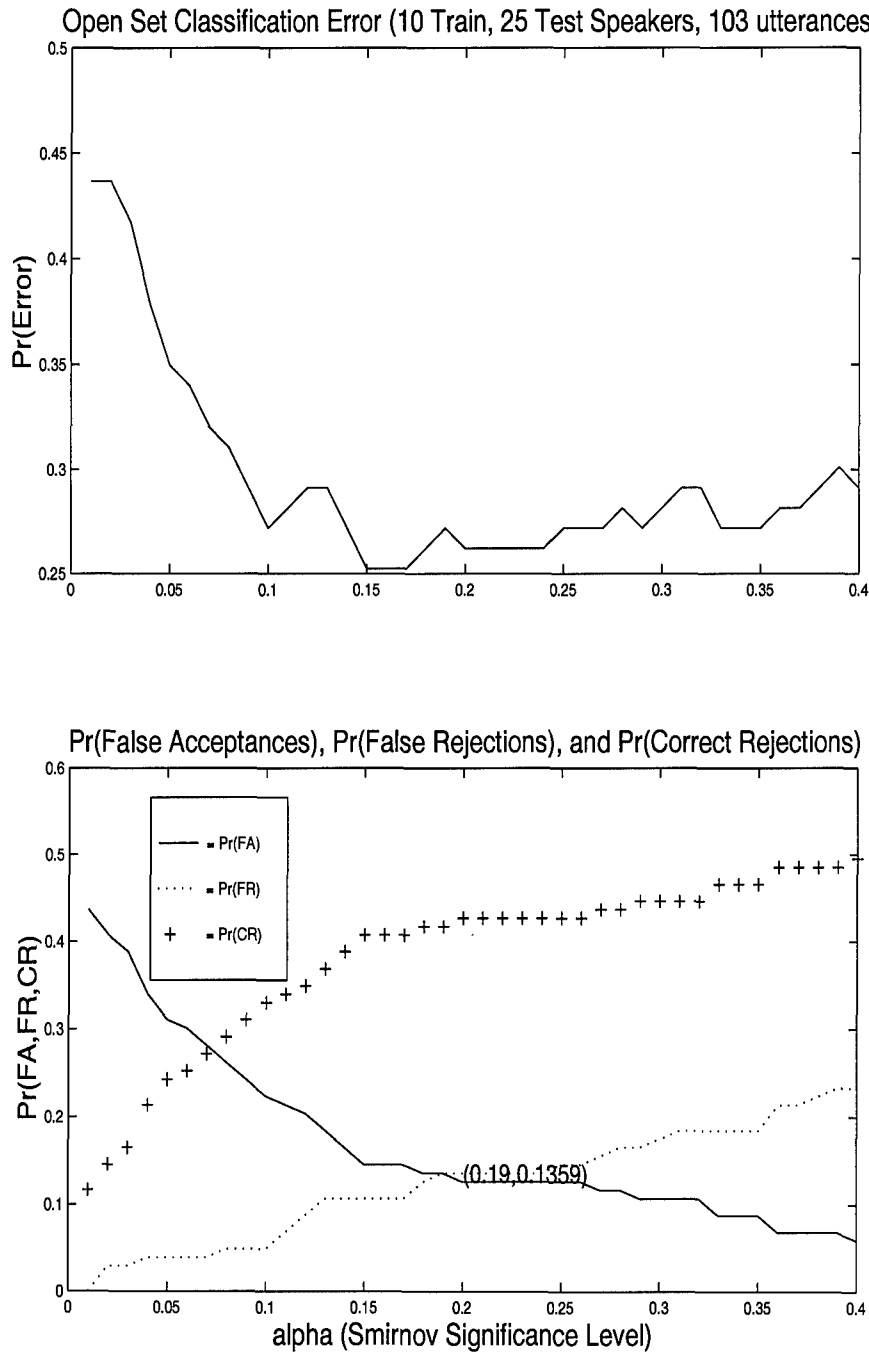
Figure 34. Open-Set Speaker Recognition Results for GREENFLAG, Group 1, training on 10 speakers, testing on 25 (the 10 for training, plus 15), giving 103 test utterances. Open-set speaker recognition was accomplished using the LPREFC feature set, Karhunen-Loève initialization followed by the LBG Algorithm (8 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.

Open Set Classification Error (10 Train, 25 Test Speakers, 109 utterances)

Pr(False Acceptances), Pr(False Rejections), and Pr(Correct Rejections)

(0.16,0.1743)

— = Pr(FA)
···· = Pr(FR)
+ = Pr(CR)

alpha (Smirnov Significance Level)

Figure 35.   Open-Set Speaker Recognition Results for GREENFLAG, Group 2, training on 10 speakers, testing on 25 (the 10 for training, plus 15), giving 109 test utterances. Open-set speaker recognition was accomplished using the LPREFC feature set, Karhunen-Loève initialization followed by the LBG Algorithm (8 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
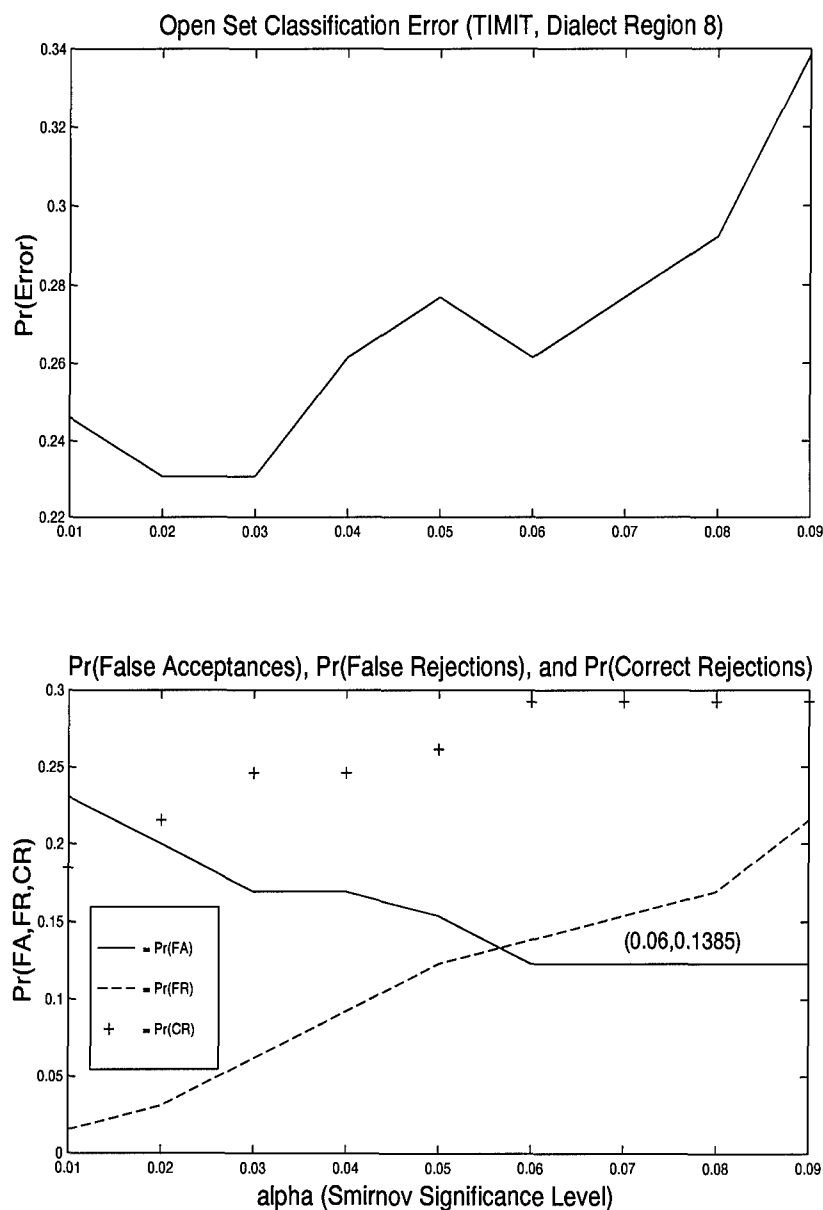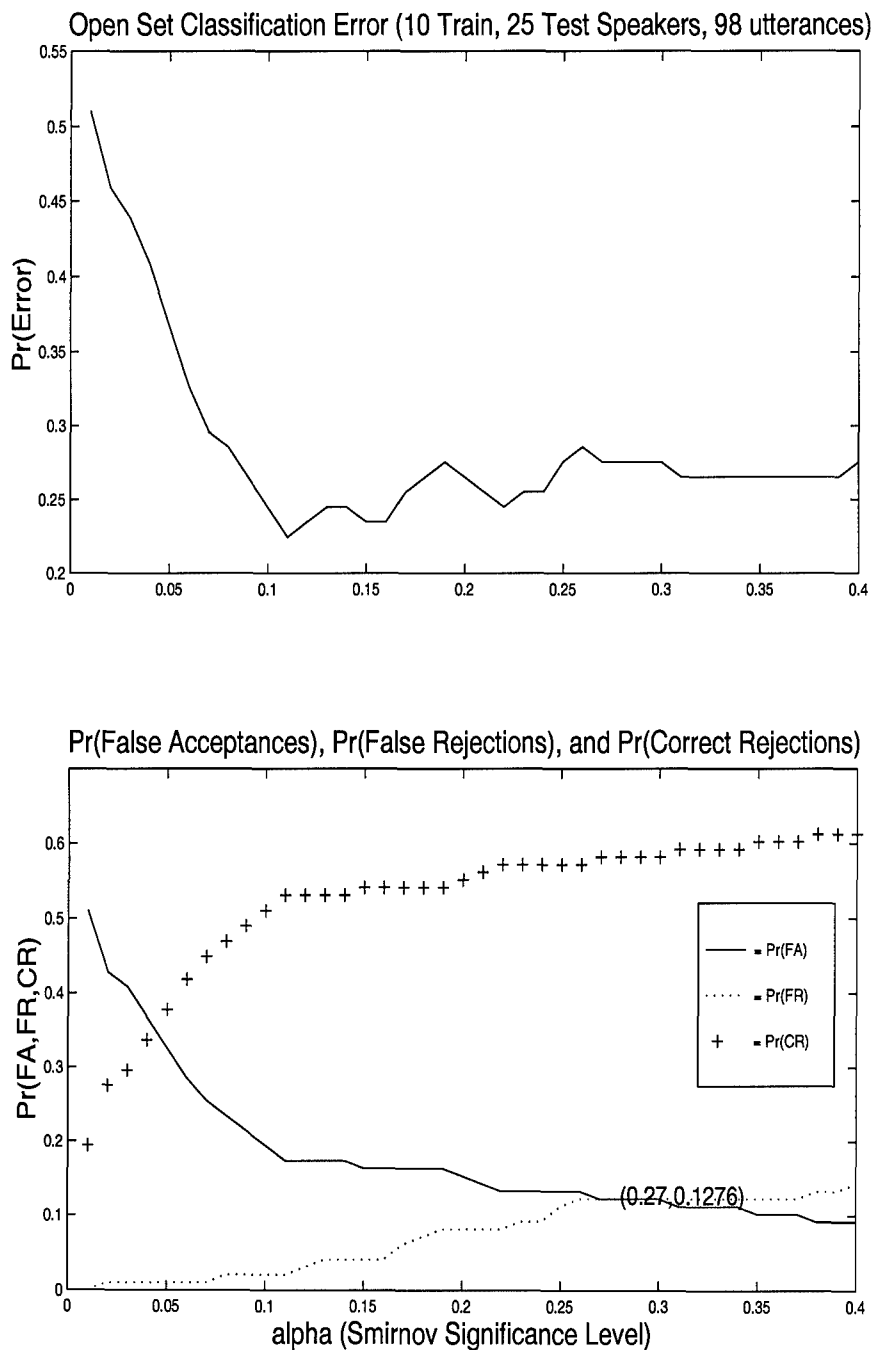
Figure 36. Open-Set Speaker Recognition Results for GREENFLAG, Group 3, training on 10 speakers, testing on 25 (the 10 for training, plus 15), giving 98 test utterances. Open-set speaker recognition was accomplished using the LPREFC feature set, Karhunen-Loève initialization followed by the LBG Algorithm (8 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.
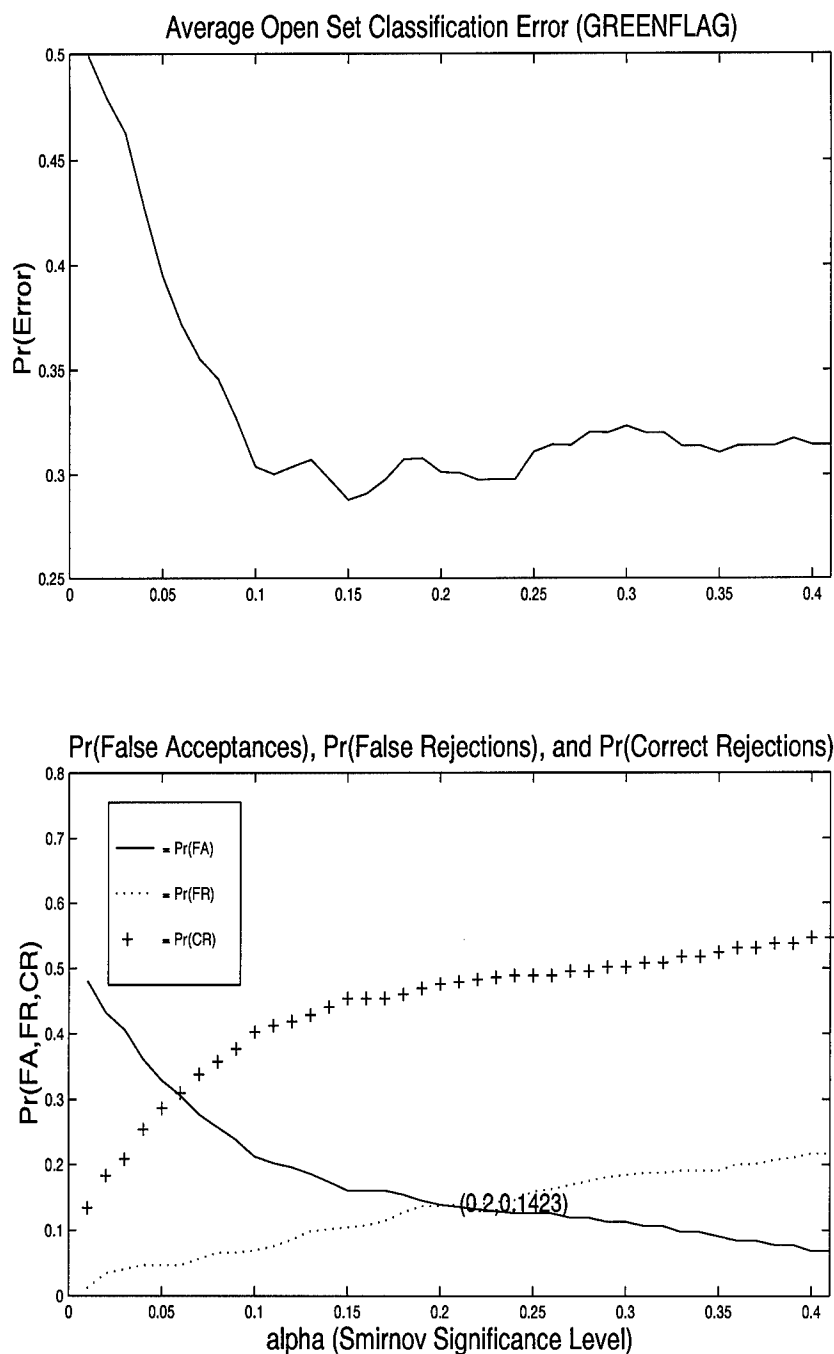
Figure 37. Averaged Open-Set Speaker Recognition Results for GREENFLAG, Groups 1–3 (103, 109, and 98 arbitrary test utterances, respectively). Open-set speaker recognition was accomplished using the LPREFC feature set, Karhunen-Loève initialization followed by the LBG Algorithm (8 codewords), and the fuzzy classifier followed by the Smirnov Test for Common Distributions.

87

B.4   Closed-Set Speaker Recognition

The following plot provides the results of testing the proposed text-independent, open-set speaker recognition system, operating in a closed-set mode, for all GREENFLAG speakers.
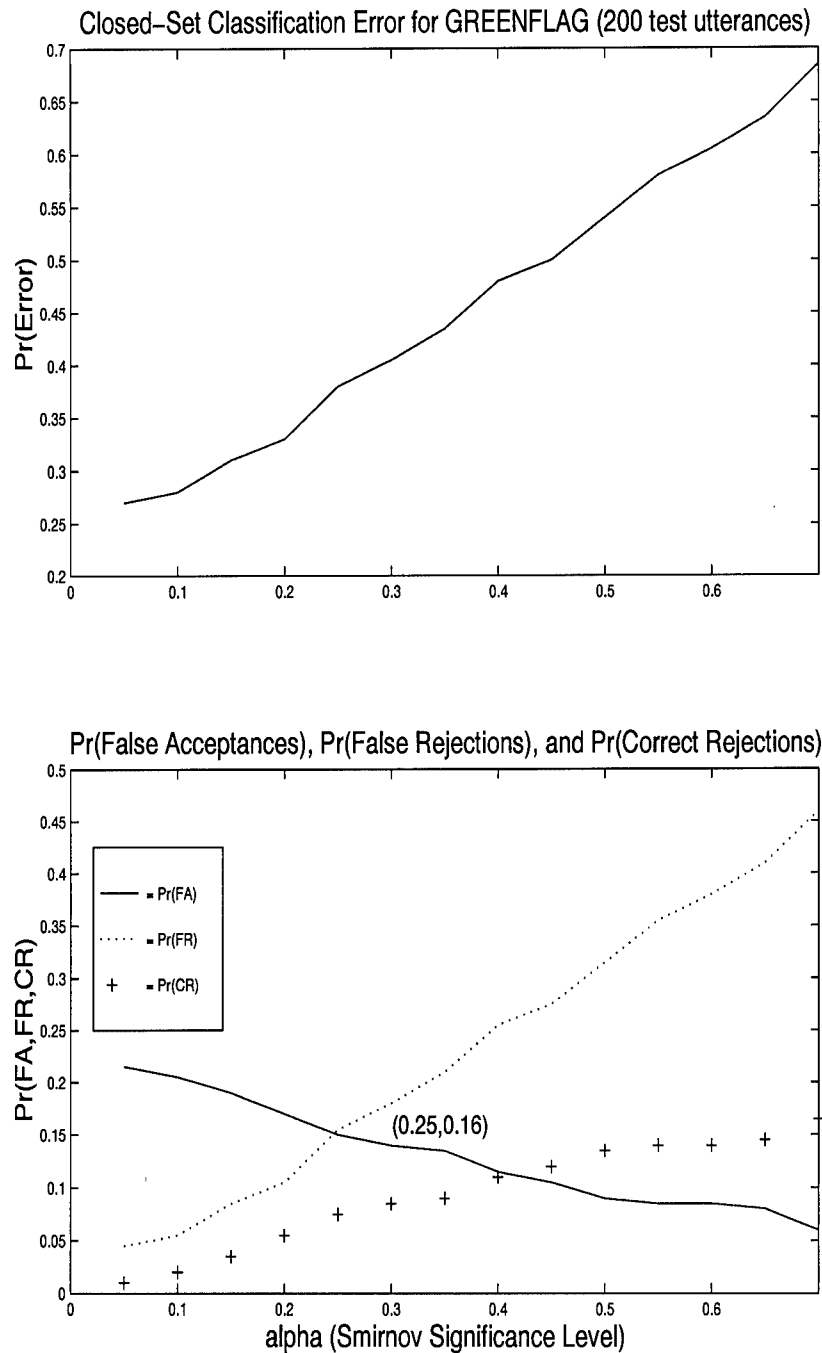
Figure 38. Closed-Set Speaker Recognition Results for GREENFLAG, using the LPREFC feature set, Karhunen-Loève initialization followed by the LBG Algorithm (8 codewords), and the fuzzy class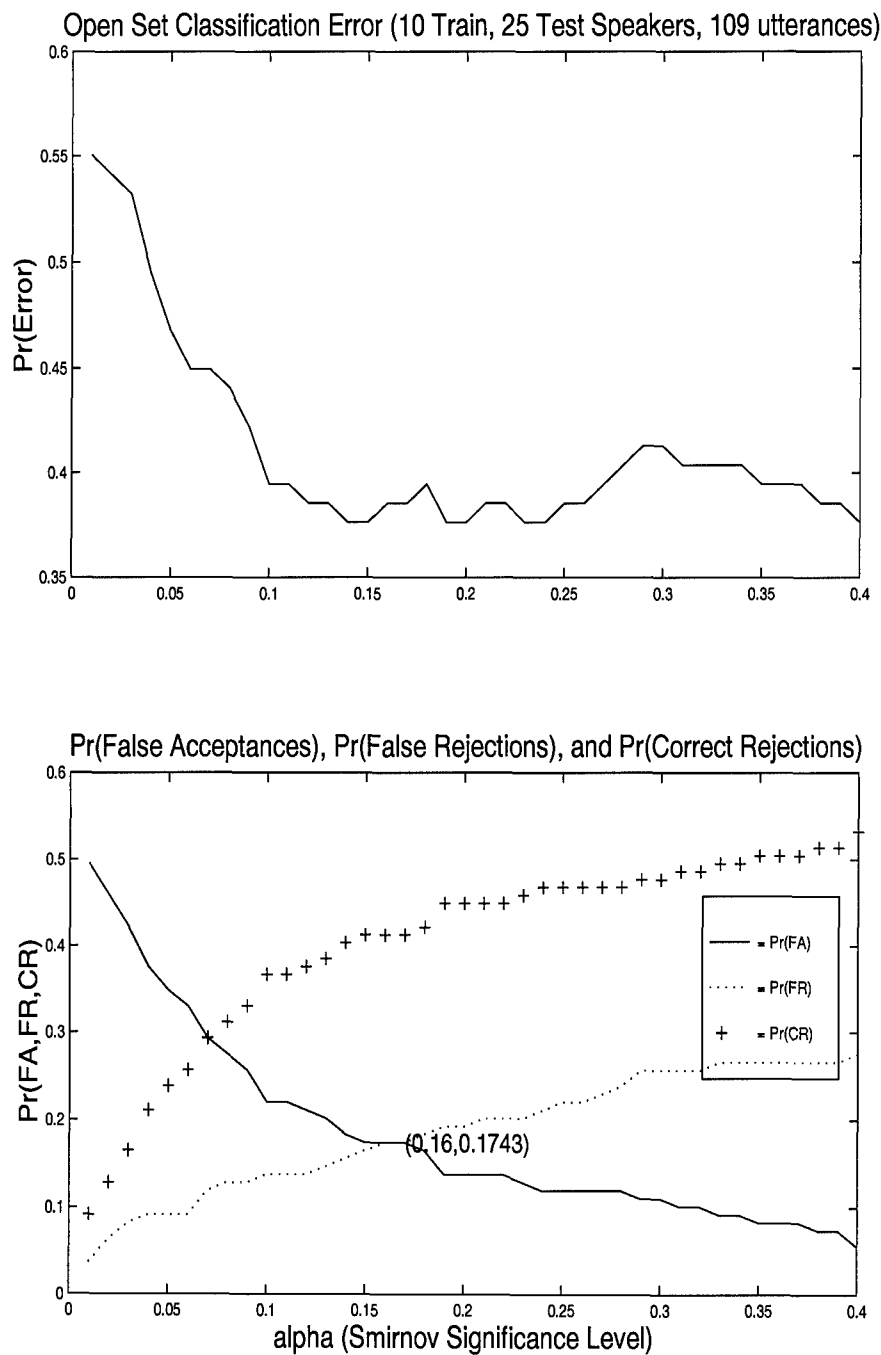ifier followed by the Smirnov Test for Common Distributions. These plots show the results of applying the proposed open-set speaker recognition system in a closed-set mode for all 41 GREENFLAG speakers. As shown, the proposed open-set system can be used to correct closed-set classification errors, which could be useful if false acceptances are intolerable. The trade-off for these correct rejections, however, entails accepting a large number of false rejections.

89

B.5 Conclusion

This appendix provides a complete view of the results of this thesis. First, the results of the feature analysis are provided. Next, the results of testing the proposed open-set speaker recognition system on the TIMIT and GREENFLAG corpora are provided.

Appendix C. Baseline Tests

C.1 Introduction

This appendix provides a comparison of speaker identification methods. The goal here is to substantiate the choice of the classification method applied in the text-independent, open-set speaker recognition system. Since there is no baseline system for the open-set task, this comparison is limited to closed-set speaker identification.

C.2 Systems Considered

Each of the following vector quantization-based systems use codebooks formed with the Karhunen-Loève initialization [14], followed by the LBG Algorithm [35] as described in Section 3.5.1. The following methods were considered in this baseline testing:

- **Minimum Euclidean Distance (MED).** This method serves as the baseline, and is similar to that described by Shore and Burton [61]. Classification of an utterance entails finding the minimum average (over all frames, using the minimum codeword distance for each codebook) Euclidean distance.

- **Maximum Summed Membership Function Values (MSU).** This method calculates a fuzzy membership function value for each frame of an utterance (see Equation 3). Classification of the utterance is based on the maximum summed (over all frames) membership function value. In such a system, the Smirnov Test can then be applied to all of the winning speaker's frames to accomplish the open-set task.

- **By-Frame Majority Voting (BFU).** This method also calculates a fuzzy membership function value for each frame of an utterance; however, it classifies an utterance based on a by-frame majority voting scheme, wherein each frame is classified based on the maximum membership function value. The Smirnov Test can then be applied only to the winning frames to accomplish the open-set task.

Table 12. Results of Baseline Testing. This table shows the results of closed-set speaker identification for each method considered. Ten speakers were used for each test group, and each TIMIT dialect region was tested on 40 utterances, while 53 utterances were used for GREENFLAG. The weighted mean and standard deviation were used reflect the different number test utterances for the dialect regions and GREENFLAG. Based on the weighted mean, the by-frame majority voting method (BFU) performs best.

Test Group	Speaker ID Error Rate		
	MED	MSU	BFU
Dialect Region 1.	0.10	0.12	0.05
Dialect Region 2.	0.07	0.10	0.05
Dialect Region 3.	0.05	0.10	0.05
Dialect Region 4.	0.10	0.22	0.03
Dialect Region 5.	0.05	0.15	0.07
Dialect Region 6.	0.10	0.18	0.12
Dialect Region 7.	0.03	0.07	0.05
Dialect Region 8.	0.18	0.30	0.05
GREENFLAG	0.04	0.07	0.13
weighted mean	0.079	0.143	0.069
standard deviation	0.039	0.064	0.039

C.3 Baseline Tests

Baseline tests consisted of testing the three methods in closed-set speaker identification for both the TIMIT (all eight dialect regions) and the GREENFLAG corpora. Similar to the requirements of Section 1.5, one 2-4 second utterance was used for training. Based on the findings of the feature and codebook analyses (Sections 4.2 and 4.3), the LPCEPSTRA_E feature set and 10 codewords per codebook were used for the TIMIT speakers, while the LPREFC feature set and eight codewords were used for GREENFLAG. Ten speakers were used in each test group, and the number of test utterances were 40 for each of TIMIT's dialect regions and 53 for GREENFLAG.

Table 12 summarizes the results of the baseline testing. As shown in Table 12, the by-frame majority voting method (BFU) performs better (in terms of the weighted mean and standard deviation of the speaker identification error rates) than the other methods considered. Hence, the basic classification method used for the open-set task will be based on the by-frame majority voting method.

C.4 Conclusion

This appendix provides a comparison of the classification method used in the proposed text-independent, open-set system (operating in a closed-set mode) to a minimum averaged Euclidean distance baseline method. The results justify the use of the by-frame majority voting scheme.

Bibliography

1. "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, NIST Speech Disc CD1-1.1." Produced on CD-ROM by the National Institute of Standards and Technology (NIST), October 1990.

2. "NTIMIT Speech Corpus, NIST Speech Discs 10-1.1, 10-2.1." Produced on CD-ROM by the National Institute of Standards and Technology, August 1992.

3. ACB. "Military Technologies Find New Law Enforcement Uses," *Signal, 50*(3):55–57 (November 1995).

4. Afifi, A. A. and S. P. Azen. *Statistical Analysis, A Computer Oriented Approach* (2 Edition). New York, NY 10003: Academic Press, 1979.

5. Assaleh, Khaled T. and Richard J. Mammone. "New LP-Derived Features for Speaker Identification," *IEEE Transactions on Speech and Audio Processing, 2*(4):630–638 (October 1994).

6. Atal, B. S. and S. L. Hanauer. "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *The Journal of the Acoustic Society of America, 50*:637–655 (August 1971).

7. Atal, Bishnu S. "Automatic Recognition of Speakers from Their Voices," *Proceedings of the IEEE, 64*(4):460–475 (April 1976).

8. Basztura, Czeslaw. "Experiments of Automatic Speaker Recognition in Open Sets," *Speech Communication, 10*(2):117–127 (June 1991).

9. Bezdek, James C. *Self-Organization and Clustering Algorithms.* Technical Report, National Science Foundation, 1991.

10. Brown, Kathy L. and E. Bryan George. "CTIMIT: A Speech Corpus for the Cellular Environment with Applications to Automatic Speech Recognition," *Proceedings of the 1995 International Acoustics, Speech, and Signal Processing Conference, I*:105–108 (1995).

11. Colombi, J., et al. "Auditory Model Representation for Speaker Recognition," *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, II*:700–703 (March 1993).

12. Davis, Steven B. and Paul Mermelstein. "Comparison of Parametric Representations for Mono-syllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing, 28*(4):357–366 (August 1980).

13. Deller, John R., et al. *Discrete-Time Processing of Speech Signals.* Englewood Cliffs, NJ: Macmillan Publishing Company, 1993.

14. DeSimio, Martin P., et al. *Karhunen-Loève Based Initialization for Generalized Lloyd Iteration.* Technical Report, WPAFB, OH: Air Force Institute of Technology, November 1995.

15. Doddington, George R. "Speaker Recognition - Identifying People by their Voices," *Proceedings of the IEEE, 73*(11):1651–1664 (November 1985).

16. Dougherty, Edward R. *Probability and Statistics for the Engineering, Computing, and Physical Sciences.* Englewood Cliffs, NJ: Prentice-Hall Inc., 1990.

17. Duda, Richard O. and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

18. Fenstermacher, Laurie and Douglas Smith. "Tactical Speaker Recognition Using Feature and Classifier Fusion," *SPIE: Applications of Artificial Neural Networks, 2243*:34–41 (April 1994).

19. Foley, Donald H. "Considerations of Sample and Feature Size," *IEEE Transactions on Information Theory, IT-18*(5):618–626 (September 1972).

20. Furui, Sadaoki. "Cepstral Analysis Technique for Automatic Speaker Verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-29*(2):254–272 (April 1981).

21. Giannelli, Paul C. "Daubert: Interpreting the Federal Rules of Evidence," *Cardozo Law Review, 15*(6/7):1999–2026 (1994).

22. Gish and Schmidt. "Text-Independent Speaker Identification," *IEEE Signal Processing Magazine*, 18–32 (October 1994).

23. Gregory, Sharon E. "Voice Spectrography Evidence: Approaches to Admissibility," *University of Richmond Law Review, 20*:357–376 (Winter 1986).

24. Hermansky, Hynek, et al. "RASTA-PLP Speech Analysis Technique," *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, I*:121–124 (1993).

25. Hermansky, Hynek, et al. "Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing," *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, II*:83–86 (1993).

26. Juang, Biing-Hwang, et al. "On the Use of Bandpass Liftering in Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35*(7):947–954 (July 1987).

27. Kao, Yu-Hung, et al. "Robustness Study of Free-Text Speaker Identification and Verification," *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing, II*:379–382 (1993).

28. Katsavounidis, Ioannis, et al. "A New Initialization Technique for Generalized Lloyd Iteration," *IEEE Signal Processing Letters, 1*(10):144–146 (October 1994).

29. Keller, James M., et al. "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, SMC-15*(4):580–585 (July/August 1985).

30. Kraniauskas, Peter. "A Plain Man's Guide to the FFT," *IEEE Signal Processing Magazine*, 24–35 (April 1994).

31. Lapin, Lawrence L. *Probability and Statistics for Modern Engineering.* Monterey, CA: Brooks/Cole Engineering Division, 1983.

32. Lee, Chulhee and David A. Landgrebe. "Feature Extraction Based on Decision Boundaries," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(4):388–400 (April 1993).

33. Lee, Kai-Fu. *Automatic Speech Recognition: The Development of the SPHINX System.* Norwell, MA: Kluwer Academic Publishers, 1989.

34. Lilliefors, Hubert W. "On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown," *American Statistical Association Journal*, 399–402 (June 1956).

35. Linde, Y., et al. "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications, COM-28*:84–94 (January 1980).

36. Makhoul, John. "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, *63*:561–580 (1975).

37. Mason, Robert D. *Statistical Techniques in Business and Economics* (5 Edition). Homewood, IL 60430: Richard D. Irwin, Inc., 1982.

38. Matsui, Tomoko and Sadaoki Furui. "Comparison of Text-Independent Speaker Recognition Methods Using Vector-Quantization Distortion and Discrete and Continuous HMMs," *Electronics and Communications in Japan Part 3*, *77*(12):63–70 (1994).

39. Morgan, Nelson, "Connectionist Speech Recognition Using Stochastic Perceptual Principles." Presentation at Ohio State University, May 1995.

40. Nolan, Francis. *The Phonetic Bases of Speaker Recognition*. New York, NY: Cambridge University Press, 1983.

41. Openshaw, J.P., et al. "A Comparison of Composite Features under Degraded Speech in Speaker Recognition," *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing*, *II*:371–374 (1993).

42. Oppenheim, A. V. and R. W. Schafer. *Discrete-time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall International, 1989.

43. O'Shaughnessy, D. "Speaker Recognition," *IEEE ASSP Magazine*, 4–17 (October 1986).

44. Pal, Sankar K. and Dwijesh Dutta Majumder. "Fuzzy Sets and Decisionmaking Approaches in Vowel and Speaker Recognition," *IEEE Transactions in Systems, Man, and Cybernetics*, 625–629 (August 1977).

45. Papoulis, Athanasios. *Probability, Random Variables, and Stochastic Processes*. New York, NY: McGraw-Hill, 1991.

46. Parsons, Thomas. *Voice and Speech Processing*. New York: McGraw-Hill Book Company, 1987.

47. Poritz, Alan B. "Hidden Markov Models: A Guided Tour," *Proceedings of the 1988 International Conference on Acoustics, Speech, and Signal Processing*, *1*:7–13 (April 1988).

48. Rabiner, Lawrence R. and Biing-Hwang Juang. "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, 4–16 (January 1986).

49. Rabiner, Lawrence R. and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: PTR Prentice Hall (Signal Processing Series), 1993.

50. Rabiner, Lawrence R., et al. "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *ASSP-26*(6):575–582 (December 1978).

51. Ramachandran, Ravi P., et al. "A Comparative Study of Robust Linear Predictive Analysis Methods with Applications to Speaker Identification," *IEEE Transactions on Speech and Audio Processing*, *3*(2):117–125 (March 1995).

52. Reynolds, Douglas A. *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*. MS thesis, Georgia Institute of Technology, August 1992.

53. Reynolds, Douglas A. "Large Population Speaker Identification Using Clean and Telephone Speech," *IEEE Signal Processing Letters*, *2*(3):46–48 (March 1995).

54. Reynolds, Douglas A. "Speaker Identification and Verification using Gaussian Mixture Speaker Models," *Speech Communication* (1995). to appear.

55. Reynolds, Douglas A. and Richard C. Rose. "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Transactions on Speech and Audio Processing*, *3*(1):72–83 (January 1995).

56. Ricart, Richard, et al. "Speaker Recognition in Tactical Communications," *Proceedings of the 1994 International Conference on Acoustics, Speech, and Signal Processing, I:*329–332 (1994).

57. Roe, David B. and Jay G. Wilpon. "Whither Speech Recognition: The Next 25 Years," *IEEE Communications Magazine*, 54–62 (1993).

58. Ruck, Dennis W. *Characterization of Multilayer Perceptrons and their Application to Multisensor Automatic Target Detection*. PhD dissertation, Air Force Institute of Technology, WPAFB, OH, December 1990.

59. Ruck, Dennis W., "EENG 620 and EENG 621 Class Notes and Personal Interviews," 1995.

60. Ruck, Dennis W., et al. "Multisensor Fusion Classification with a Multilayer Perceptron." *Proceedings of IEEE/INNS International Joint Conference on Neural Networks.* 863–868. June 1990.

61. Shore, John E. and David K. Burton. "Discrete Utterance Speech Recognition Without Time Alignment," *IEEE Transactions on Information Theory, IT-29*(4):473–491 (July 1983).

62. Soong, Frank K. and Aaron E. Rosenberg. "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing, 36*(6):871–879 (June 1988).

63. Sprent, Peter. *Applied Nonparametric Statistical Methods*. New York, NY: Chapman and Hall, 1989.

64. Szu, Harold H., et al. "Neural Network Adaptive Wavelets for Signal Representation and Classification," *Optical Engineering, 31*(9):1907–1916 (September 1992).

65. Tierney, Joseph. "A Study of LPC Analysis of Speech in Additive Noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-28*(4):389–397 (August 1980).

66. Walter, Sharon. "Final Report for the Speaker Identification Technology Program, Contract Number F30602-93-C-0011." Submitted to the Defense Technical Information Agency (DTIC) for publication., May 1994.

67. Watterson, Clark C., et al. "Experimental Confirmation of an HF Channel Model," *IEEE Transactions on Communication Technology, COM-18*(6):792–803 (December 1970).

68. Wilks, Samuel S. *Mathematical Statistics*. New York, NY: John Wiley and Sons, Inc., 1962.

69. Wolf, J. "Efficient Acoustic Parameters for Speaker Recognition," *Journal of the Acoustical Society of America, 51*(6):2044–2056 (1972).

70. Young, Stephen and Philip Woodland. "HTK: Hidden Markov Model Toolkit V1.4 Reference Manual,". Cambridge University Technology Transfer Company, February 1993.

Vita

Captain Stephen V. Pellissier ~~████████████████~~, the son of Dick and Terry Pellissier. He graduated as valedictorian from Williams High School, in 1983, and attended Northern Arizona University, graduating with a Bachelor of Science Degree in Electrical Engineering with honors, Magna Cum Laude, in May 1988. As a Distinguished Military Graduate of the ROTC program, Stephen was also commissioned in May 1988, as a U.S. Army Signal Corps officer, and later received a Regular Army Commission in 1989. After completing the Signal Officer Basic Course at Ft. Gordon, the Telecommunications Operations Officers Course at the Air Force Institute of Technology (AFIT), and Airborne School in 1989, Stephen served as an electronics engineer with the USACECOM Intelligence Materiel Management Center, Vint Hill Farms Station (VHFS), VA, supporting tactical and strategic signal intelligence and electronic warfare systems. Stephen served in Operations Desert Shield and Desert Storm as the Site Chief for the forward deployed intelligence and electronic warfare (IEW) special repair activities, which provided depot-level maintenance and logistics support to military intelligence (MI) units from the division-level MI battalions to echelons above corps. Upon returning from Southwest Asia, Stephen assumed the duty of Chief, Tactical IEW New Equipment Training Team, in addition to the electronics engineering duties. In December 1991, Stephen took command of the U.S. Army Information Systems Command Vint Hill Company. His company supported VHFS in the areas of local area networking, telecommunications operations, telephone switching, and automation. Following a successful command, in February 1993, Stephen returned to Ft. Gordon to attend the Signal Officer Advanced Course (SOAC). In addition to completing SOAC as an Honor Graduate, Stephen's peers nominated him for the Kilbourne Leadership Award. Stephen reported to the 1st Signal Brigade, Korea, in August 1993 and became Chief, Facility Control Office, a 24 hour watch-desk for all strategic communications supporting U.S. Forces Korea. In September 1994, Stephen returned to AFIT to pursue a Master of Science Degree in Electrical Engineering. Stephen's awards include the Bronze Star Medal, the Meritorious Service Medal, the Army Commendation Medal, with Oak Leaf Cluster, and the Army Achievement Medal. Stephen is an active member of the Armed Forces Communications & Electronics Association, the Institute of Electronic & Electrical Engineers, the Phi Kappa Phi National Honor Society, and the Tau Beta Pi Engineering Honor Society.

Permanent address: ~~████████████████~~
~~████████████████~~

98

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE

Text-Independent, Open-Set Speaker Recognition

5. FUNDING NUMBERS

6. AUTHOR(S)

Stephen V. Pellissier
CPT, USA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
2950 P Street
Wright-Patterson Air Force Base, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/96M-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Communications-Electronics Command
Intelligence and Electronic Warfare Directorate
ATTN: AMSEL-RD-IEW-TAS (Joseph Karakowski)
Fort Monmouth, NJ 07703

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Closed-set speaker recognition systems abound, and the overwhelming majority of research in speaker recognition in the past has been limited to this task. A realistically viable system must be capable of dealing with the open-set task. This effort attacks the open-set task, identifying the best features to use, and proposes the use of a fuzzy classifier followed by hypothesis testing as a model for text-independent, open-set speaker recognition. Using the TIMIT corpus and Rome Laboratory's GREENFLAG tactical communications corpus, this thesis demonstrates that the proposed system succeeded in open-set speaker recognition. Considering the fact that extremely short utterances were used to train the system (compared to other closed-set speaker identification work), this system attained reasonable open-set classification error rates as low as 23% for TIMIT and 26% for GREENFLAG. Feature analysis identified the liftered linear prediction cepstral coefficients with or without the normalized log energy or pitch appended as a robust feature set (based on the 17 feature sets considered), well suited for clean speech and speech degraded by tactical communications channels. Finally, in contrast to previous efforts which have used codebooks consisting of 32–512 codewords, codebook analysis revealed that relatively small codebooks (with as few as 8–10 codewords) are adequate, if not optimal, in terms of classification accuracy and computational complexity for vector quantization-based classification techniques.

14. SUBJECT TERMS

Speaker Recognition, Speaker Identification, Open-Set, Closed-Set, Fuzzy Classification, Vector Quantization, Hypothesis Testing, Speech Features

15. NUMBER OF PAGES

111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	-	Contract	**PR** - Project	
G	-	Grant	**TA** - Task	
PE	-	Program Element	**WU** - Work Unit Accession No.	

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	-	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	-	See authorities.
NASA	-	See Handbook NHB 2200.2.
NTIS	-	Leave blank.

Block 12b. Distribution Code.

DOD	-	Leave blank.
DOE	-	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	-	Leave blank.
NTIS	-	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.