4-30-2011

A Nonlinear Mixture Autoregressive Model For Speaker Verification

Sundararajan Srinivasan

A NONLINEAR MIXTURE AUTOREGRESSIVE MODEL FOR SPEAKER VERIFICATION

By

Sundararajan Srinivasan

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

April 2011

A NONLINEAR MIXTURE AUTOREGRESSIVE MODEL FOR SPEAKER VERIFICATION

By

Sundararajan Srinivasan

Approved:

Saurabh Prasad
Assistant Research Professor
GeoSystems Research Institute
(Major Professor)

James E. Fowler
Professor and Graduate Program
Coordinator of Electrical and Computer
Engineering
(Committee Member)

Julie Baca.
Research Associate
Center for Advanced Vehicular Systems
(Committee Member)

Len T. Miller
Professor of Mathematics
(Committee Member)

Lori Bruce
Associate Dean of College of Engineering

Name: Sundararajan Srinivasan

Date of Degree: April 29, 2011

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Saurabh Prasad

Title of Study: A NONLINEAR MIXTURE AUTOREGRESSIVE MODEL FOR SPEAKER VERIFICATION

Pages in Study: 106

Candidate for Degree of Doctor of Philosophy

In this work, we apply a nonlinear mixture autoregressive (MixAR) model to supplant the Gaussian mixture model for speaker verification. MixAR is a statistical model that is a probabilistically weighted combination of components, each of which is an autoregressive filter in addition to a mean. The probabilistic mixing and the data-dependent weights are responsible for the nonlinear nature of the model. Our experiments with synthetic as well as real speech data from standard speech corpora show that MixAR model outperforms GMM, especially under unseen noisy conditions. Moreover, MixAR did not require delta features and used 2.5x fewer parameters to achieve comparable or better performance as that of GMM using static as well as delta features. Also, MixAR suffered less from over-fitting issues than GMM when training data was sparse. However, MixAR performance deteriorated more quickly than that of GMM when evaluation data duration was reduced. This could pose limitations on the required minimum amount of evaluation data when using MixAR model for speaker verification.

ACKNOWLEDGEMENTS

I thank Dr. Joe Picone for having guided me through all parts of this dissertation work. I have learnt a great deal from him about speech technology, about conducting research, about software engineering principles, and also about making pizza the true Italian way! I agree that my major advisor, Dr. C. S. Ramalingam, during my Master's, was entirely correct in his assessment: "Joe? He is a dependable guy."

I also thank Dr. Lazarou for his support and involvement in group meetings all these years. Several other people have made my ISIP years enjoyable; thanks in particular to Tao and Daniel in this regard.

A huge thanks to my friend, Santosh, for having fed and accompanied me in almost every adventure.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS ii

LIST OF TABLES vi

LIST OF FIGURES viii

CHAPTER

I. INTRODUCTION 1

1.1 Speaker Recognition Problem 1
1.2 Speaker Verification Problem 5
1.3 Speech Signal Features – The Frontend 7
1.4 Statistical Speaker Modeling Methods 9
1.4.1 Non-parametric models 10
1.4.2 Parametric Models 11
1.5 Speaker Verification – A Concise History 13
1.6 Survey of Speech Feature and Model Performance Comparisons 15
1.7 Speaker Recognition State-of-the-art Performances on Standard Databases 16
1.7.1 On TIMIT Database 17
1.7.2 On NTIMIT Database 18
1.7.3 On NIST Speaker Recognition Evaluation (SREs) Databases 18
1.7.4 On YOHO Database 20
1.8 Dissertation Organization 20

II. NONLINEARITIES IN THE SPEECH SIGNAL – A BRIEF SURVEY 22

2.1 Introduction 22
2.2 Nonlinear Dynamical Invariants for Broad-Phone Classification Speech 24
2.2.1 Lyapunov Exponents 25
2.2.2 Fractal Dimension 26

2.2.3 Kolmogorov-Sinai Entropy 27
2.3 Nonlinear Invariants for Broad-Phone Classification of Speech 27
2.3.1 Broad-Phone Classification Results Using Invariants 29
2.4 Survey of Application of Nonlinear Invariants in Speech Recognition 32
2.5 Drawbacks of Using Nonlinear Invariant Features in Speech Processing 34
2.6 Survey of Other Nonlinear Approaches for Speaker Recognition 35
2.7 Other novel nonlinear approaches in speech systems 37

III. THE GUASSIAN MIXTURE MODEL 39

3.1 Basic Definition 39
3.2 GMM Parameter Estimation using the EM algorithm 41
3.3 Statistical Modeling of Speech Using GMMs 44
3.4 Training Speaker GMMs using Adaptation 45
3.5 Normalization Techniques for Handset/Channel Compensation 46
3.6 Limitations of GMMs in Speech Processing 48

IV. THE MIXTURE AUTOREGRESSIVE MODEL – A NONLINEAR APPROACH 50

4.1 Why Use Nonlinear Models for Speech? 50
4.2 MixAR: Basic Definition and Properties 50
4.3 Modeling Nonlinearities Using MixAR 52
4.4 Comparison of MixAR to Other Models 55
4.5 MixAR Parameter Estimation using the EM algorithm 56
4.6 Efficiency Considerations for Faster MixAR Training 62
4.7 Example MixAR Parameter Estimation to Verify Implementation 65

V. PRELIMINARY EXPERIMENTS USING SYNTHETIC DATA 67

5.1 Two-Way Classification with Scalar Synthetic Data 67
5.2 Two-Way Classification with Vector Synthetic Data 69
5.3 Two-way Classification with Speech-like Data 70
5.4 Phone Classification Experiments Using a Sustained Phone Database 72
5.5 Speaker Verification Experiments Using Synthetic Data 74

VI. SPEAKER VERIFICATION EXPERIMENTS USING SPEECH DATA 77

6.1 Speaker Verification Experiments with NIST 2001 Database 78
6.2 A note on MixAR order 80
6.3 Speaker Verification Experiments with TIMIT Database 81

6.4 Statistical Significance in Speaker Verification Systems 82
6.5 Speaker Verification Experiments with TIMIT under Noisy Conditions 84
6.6 Speaker Verification Experiments Across Channel Conditions – TIMIT vs. NTIMIT Database 88
6.7 Speaker Verification Performance and Training Data Duration..........89
6.8 Speaker Verification Performance and Evaluation Data Duration 93

VII. CONCLUSION AND FUTURE SCOPE 95

7.1 Conclusion 95
7.2 Future Scope 96

REFERENCES 99

LIST OF TABLES

TABLE Page

5.1 Classification (% error) results for scalar synthetic data (the numbers of parameters are shown in parentheses). 69

5.2 Classification (% error) results for vector synthetic data (the numbers of parameters are shown in parentheses). 71

5.3 Classification Error Rate (%) with 12 speech MFCC-like synthetic features for GMM and MixAR (Number of parameters in each case is in parenthesis)....72

5.4 Sustained phone classification (% accuracy) results with MAR and GMM using 13 MFCC features (Number of parameters in each case is in parenthesis) 73

5.5 Sustained phone classification (% accuracy) results with MAR and GMM using static+Δ+ΔΔ MFCC features. (Number of parameters in each case is in parenthesis)....73

6.1 Speaker recognition EER with NIST-2001 for MixAR and GMM for different feature combinations. 78

6.2 Speaker recognition EER with NIST for MixAR and GMM as a function of #mix. (the numbers of parameters are shown in parentheses)....80

6.3 Speaker recognition EER with NIST for MixAR orders 1 and 2....81

6.4 Speaker Verification Performance (EER) on TIMIT core set (the numbers of parameters are shown in parentheses). 81

6.5 Speaker Verification Performance (EER) for different noise conditions 87

6.6 Speaker Verification Performance EER with TIMIT and NTIMIT....89

6.7 Speaker Verification Performance (EER) as a function of training data duration. (Evaluation utterance durations varied mostly between 20-40s)....91

6.8 Speaker Verification Performance (EER) as a function of evaluation data duration. (Training utterance durations varied mostly around 120s.) 92

LIST OF FIGURES

FIGURE Page

1.1 An overview of the speaker recognition problem 2

1.2 Illustrating use of DET curves and EER for speaker verification........................7

1.3 Overview of extraction method of speech MFCC features (reprinted with permission from Institute of Signal and Information Processing) 9

2.1 Correlation Dimension (a through c), Kolmogorov Entropy (d through f),.......30

2.2 KL Divergence Measure across various phonemes, using the three dynamical invariants. 32

3.1 An overview of the GMM approach. 40

3.2 EM performance as a function of iteration for a 8-mixture GMM model of speaker data from “4516” of NIST 2001 database. 43

4.1 An overview of MixAR modeling approach...52

4.2 Sierspinski Triangle fractal geometry generated using MixAR model.............53

4.3 Example of a MixAR generated signal and its power spectrum; the broad-band nature of power spectrum indicates that the signal is chaotic.........54

4.4 EM Convergence as a function of iterations on a synthetically synthesized signal..60

4.5 Performance of (Generalized) EM using secant method, as a function of iterations for a 8-mixture MixAR model of speaker data from “4516” of NIST-2001 database. 62

5.1 Speaker Verification DET curves for MixAR-generated nonlinear data with 5B car noise..76

6.1 Speaker verification DET curves with NIST 79

6.2 Speaker Verification DET curves for GMM and MixAR models on TIMIT database 82

6.3 DET curves for GMM and MixAR models on noisy TIMIT test with additive car noise at different SNRs..85

6.4 DET curves for GMM and MixAR models on noisy TIMIT test with additive white noise at different SNRs..86

6.5 DET curves for GMM and MixAR models on noisy TIMIT test with additive babble noise at different SNRs..87

6.6 DET curves for GMM and MixAR models on TIMIT and NTIMIT databases..88

6.7 Speaker Verification Performance as a function of training utterance duration..90

6.8 Speaker Verification Performance as a function of evaluation utterance duration..93

CHAPTER I

INTRODUCTION

In this chapter, the problem of speaker verification is introduced. The different flavors of the speaker recognition speech signal features used to represent speaker information are briefly explained and some popular approaches to the speaker modeling problem found in the literature are surveyed. Also a brief historic survey of speaker recognition systems is provided. This is followed by a synthesis of results found in literature assessing the performance of various modeling and feature extraction approaches and a section outlining performances with state-of-the-art speaker recognition systems with standard databases. This sets the tone for the following chapters.

1.1 Speaker Recognition Problem

Speech processing technology has advanced incredibly over the past 30 years. Developments in speech coding have led to a remarkable reduction in bandwidth required for transmitting and storing digital speech. Speech recognition systems are fast becoming quicker, easier, and preferred alternatives to typing text in cell phones. A large number of human agents in customer support services are increasingly replaced by automated interactive voice response systems. Secured entry into office buildings using voice biometric is sure to become commonplace in the near future. The use of voice authentication login in laptop computers is already quite prevalent. There are so many more applications – some serving only fanciful needs, while others becoming indispensable in daily activities – that are made possible by advancements in the field of speech research. And overview of the speaker verification problem is shown below.

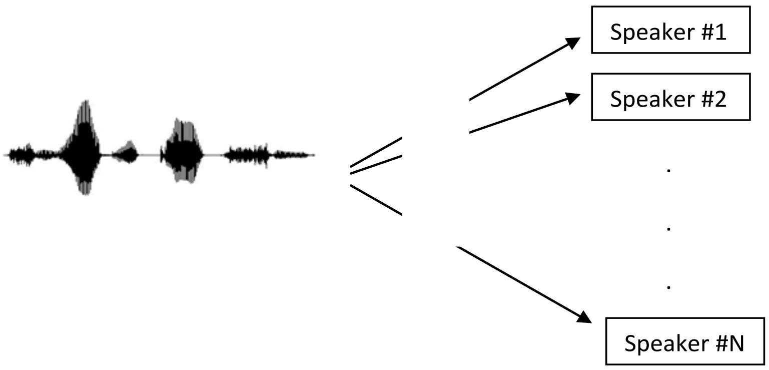


Figure 1.1 An overview of the speaker recognition problem

The goal of the work described in this document is to improve the current technology for speaker recognition, and in particular, speaker verification. Speaker recognition problem deals with deciphering or verifying the identity of a person using speech utterances [1][2][3]. Most speaker recognition systems are applied in security-related applications; for e.g., authenticated access to offices and computers. There are also other applications that do not have much to do with security; for e.g. detecting or identifying different talkers in a conference meeting, in combination with speech recognition to achieve speech-to-text processing along with speaker labels for conversations.

There are different flavors of speaker recognition tasks depending on the application where the technology is employed [2]. First, there is speaker verification vs. speaker identification. In a speaker verification setup, there are several registered speakers in the database, and associated with each speaker is a trained model. During the evaluation phase, a person, X, claims an identity and speaks a preselected or randomized phrase into a microphone. The objective of speaker verification system is to determine if the speech signal is sufficiently close to the stored model associated with the claimed identity. In speaker identification, the goal is to identify a person from a known list using speech. Most speaker recognition applications related to authentication involve speaker verification.

Since input to a speaker verification system can also involve speakers who have never been seen before during enrollment, this is considered an open-set problem. It is assumed the input speech to a speaker identification system is uttered by a speaker from an already known enrollment list, thus it is considered a closed-set problem. Due to this, speaker verification is thought to be a more difficult problem. Speaker identification picks out one speaker from a list, and is hence prone to more errors and the performance worsens as the number of enrolled speakers increases. Since speaker verification is only a binary problem, its performance does not suffer this degradation with additional speakers.

Another way of classifying speaker recognition systems is by the kind of lexicon used for evaluating speaker identities [2][4][5]. In some cases, the speaker can be requested to utter from a known prompt during evaluation. Typically these are from a limited vocabulary lexicon, e.g., digits, and are already seen during enrollment of the speaker. Such a system is called text-dependent speaker recognition. More often, there is no control over what the speaker is uttering, and the utterances are typically form a large vocabulary and are not seen during training. This is called text-independent speaker recognition. Obviously, text-independent speaker recognition is a much harder problem to solve.

Yet another way to distinguish speaker recognition systems is by the number of speakers expected to be present in the utterances [2]. Traditional recognition systems assume that the utterance was from only one speaker whose identity needs to be found or verified. This is called a single-speaker recognition system. More recently, there is much research in identifying the different speakers present in a conference meeting recording and also to segment the different speaker utterances. This multi-speaker detection problem may not always involve speaker recognition since changes in speaker can also be detected using audio events without building explicit speaker models. Nevertheless, it is common to use speaker recognition techniques in such an application. Much research is being devoted to this problem under the name of speaker diarization.

Another feature of some speaker recognition systems that are deployed in widely varying channels or use microphones with very varied characteristics is the normalization technique [5]. Normalization can be performed either at the speech feature level (where it is called feature normalization or compensation), or it can also be performed at the modeling level, (where it is called score normalization). More information on these can be found in Chapter 3.

The goal of this work is to improve the technology for text-independent single speaker verification. Speech and speaker recognition problems are complementary viewpoints of the same problem. Given a speech signal, the goal of speaker-independent speech recognition is to decipher the transcription underlying the utterance, irrespective of the identity of the speaker [1]. On the other hand, text-independent speaker recognition problem attempts to find or confirm the identity of the speaker without regard to what was spoken [5]. It is therefore not surprising that several of the techniques for feature extraction and modeling are common to both the problems.

1.2 Speaker Verification Problem

It was mentioned before that the term speaker recognition encompasses speaker identification and speaker verification. When the goal is to find the identity of an unknown speaker the problem is called speaker identification. Very often in practice, the goal in speaker recognition is to accept or reject the identity claim made by a speaker. This is called speaker verification. This is widely used in a variety of applications ranging from secured access and surveillance to multimodal verification.

In a speaker verification setup, there are several registered speakers in the database, and associated with each speaker is a trained model. During the evaluation phase, a person, X, claims an identity and speaks a preselected or randomized phrase into a microphone. The objective of speaker verification system is to determine if the speech signal is sufficiently close to the stored model associated with the claimed identity.

Speaker verification, like other problems involving binary outcomes ("accept" or "reject") is plagued by two kinds of errors: false alarms and misses. When an imposter is accepted the error is called false alarm, while a true speaker getting rejected constitutes a miss. In all speaker verification problems, a threshold value determining the operating point is set. By varying this threshold we can decrease one error at the expense of an increase in the other. A graph depicting this relationship with the false alarm probability on the x-axis and the miss probability on the y-axis is called a Detection Error Tradeoff (DET) curve [6]. This is very similar to the receiver operating characteristics (ROC) curve popularly used in designing communications receivers [7] where the parameters probability of missed detection and probability of false alarms play analogous roles in constructing the curve. One model is said to be better than another if its DET curve lies closer to the origin than that of the other. In practice, this may not happen consistently for all values on the x-axis (or equivalently, the y-axis). This makes comparison between models difficult. Moreover, it is more convenient to use a single measure of performance to compare models. For these two reasons, scalar performance measures are more commonly used when stating and comparing speaker verification performance. One measure that applies equal importance to the reduction of both kinds of errors – miss and false alarm – is the Equal Error Rate (EER) [6]. This is the point at which the line having a slope of 1 and passing through the origin intersects the DET curve. At this point, the miss probability equals the false alarm probability, and hence the name. Since this work is aimed at a generic speaker verification application, there is no reason to weigh one kind of error more than the other. Hence, EER is used as the performance measure of a model for speaker verification. However, there are applications where it is important to assign more weight to reducing one error than the other; for example, in security applications, it is important that false acceptances do not occur.

The next figure illustrates hypothetical DET curves for two models for speaker verification. From the above discussion it is clear that Model 2 does better than Model 1 because the DET curve is closer to the origin and also EER is lower for Model 2 compared to Model 1.

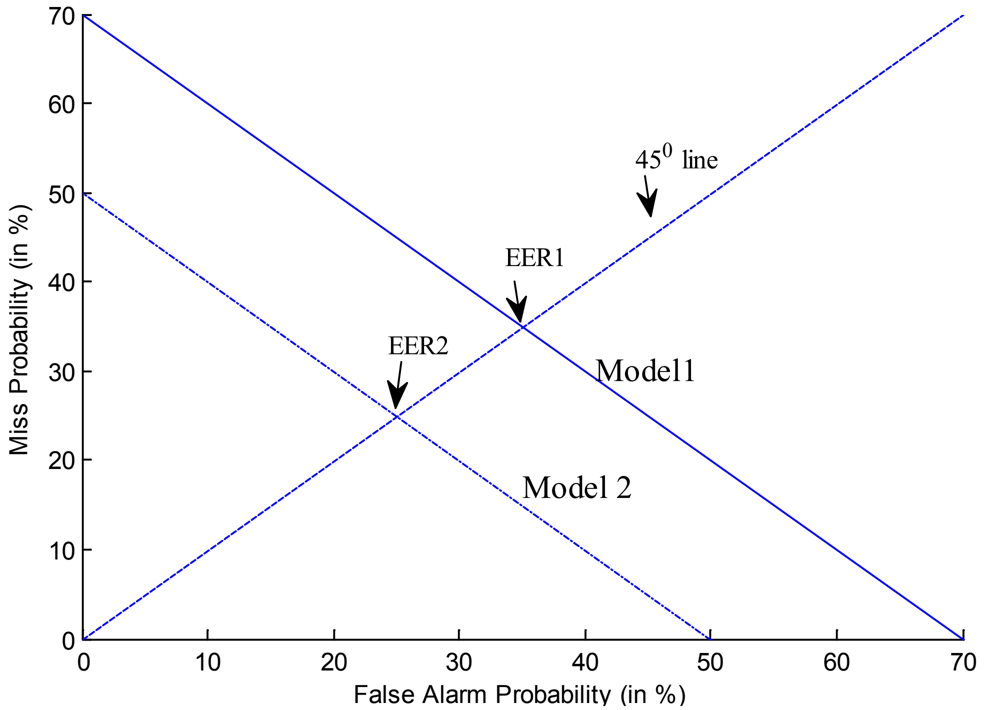


Figure 1.2 Illustrating use of DET curves and EER for speaker verification

1.3 Speech Signal Features – The Frontend

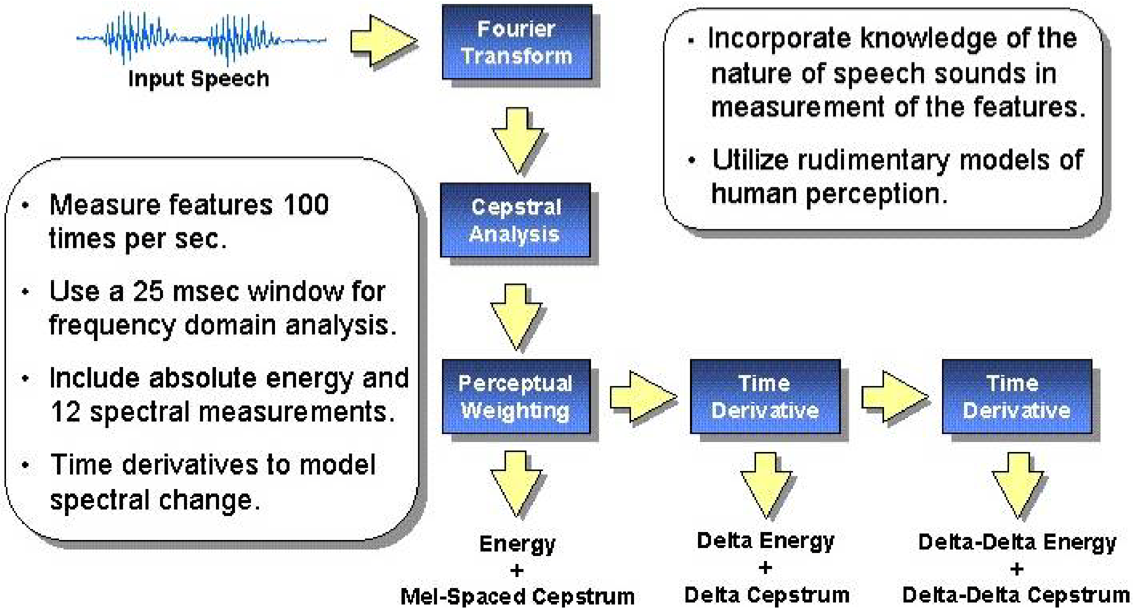
It is evident from the discussion above that speaker verification is a pattern classification problem with data from each speaker's utterance forming a distinct class. Like any pattern classification problem, a set of features to represent the characteristics of each speaker adequately, and also succinctly, is necessary for successful speaker classification [8]. Features used for speaker verification fall into two classes: those that contain low-level information and others that contain high-level features [2][5]. Low level features include such information as the spectral content or the zero-crossings in the signal. Typically these are extracted over short time-frames, and are considered peripheral to brain's perception of speech. High-level features utilize much larger time-frames, and usually involve features like words and phrases that capture idiosyncratic characteristics in a voice, and are considered to be central to the perception of speech by the human brain.

Low-level features have been the most popular and commonly used ones in speaker recognition. This is mainly because of ease of extraction of these features from a speech signal, but also perhaps because even these are quite powerful in providing useful information for speaker recognition. The current work also uses only low-level features called Mel-Frequency Cepstral Coefficients (MFCCs) to be described next. The reason that high-level features are harder to extract is because extraction of such features like words and phrases from the speech signal requires the use of preprocessing with a speech recognizer. However, in recent years there has been increasing interest in the use of some high-level features, not least because of our improved ability to perform speech recognition much more quickly and with much more accuracy than ever before.

For speech as well as speaker recognition tasks, the most popular low-level features are the Mel-Frequency Cepstral Coefficients (MFCCs), sometimes also referred to as cepstral features [1][5] First, the speech signal is windowed into frames of 10 ms duration with an analysis window duration of 25 ms. One optional step at this point is the use of a pre-emphasis filter that amplifies the higher frequencies relative to the lower ones to compensate for the spectral tilt introduced by most microphones. Yet another option is to preprocess this signal to detect and remove silence segments which anyway do not hold any information about the speaker identity.

After the preprocessing steps, the spectral energies from the samples in each frame are found. The spectral bins are logarithmically spaced, with triangular windows of increasing size with increasing frequencies. This is called the Mel-scale frequency spectrum. Then the discrete cosine transform (DCT) of log-spectral energies are computed to yield what is known as the cepstrum. The first few (typically 12-15) cepstral coefficients contain almost all the relevant information in the speech signal and only these are used in modeling and representing the speaker characteristics.

A flowchart of the MFCC feature extraction procedure is provided in the figure below.



Input Speech
Fourier Transform
Cepstral Analysis
Perceptual Weighting
Time Derivative
Time Derivative
Energy + Mel-Spaced Cepstrum
Delta Energy + Delta Cepstrum
Delta-Delta Energy + Delta-Delta Cepstrum
• Incorporate knowledge of the nature of speech sounds in measurement of the features.
• Utilize rudimentary models of human perception.
• Measure features 100 times per sec.
• Use a 25 msec window for frequency domain analysis.
• Include absolute energy and 12 spectral measurements.
• Time derivatives to model spectral change.

Figure 1.3 Overview of extraction method of speech MFCC features (reprinted with permission from Institute of Signal and Information Processing)

It is not important here to delve into all the details of the algorithm used to compute these features, but we can simply note that the MFCCs are physically motivated based on auditory perception properties of the human ear. MFCCs have been the most successful features in speech and speaker recognition applications.

1.4 Statistical Speaker Modeling Methods

The goal of statistical modeling in speaker recognition is to accurately and efficiently represent the probability distribution of speaker features so that even similar sounding speakers can be distinguished and can be done so with as few parameters and as little computational requirement as possible.

There are several statistical models that have been proposed in the literature, and they can be broadly divided into two classes: non-parametric and parametric models.

1.4.1 Non-parametric models

These models are especially useful when the enrollment data is well-matched to the test data (but they do not generalize well to mismatched conditions) [2][4][9]. A good example of this is the method of templates. In an application that implements text-dependent speaker recognition system, several repetitions of the same or a small number of password phrases are recorded as templates during enrollment. During testing, a simple template matching scheme can be used to find the speaker that has the template closest to the enrollment phrase utterance.

Another non-parametric model is the nearest neighbor method using Parzen windows. In this approach, the probability density function of enrollment feature data from every speaker is represented non-parametrically using Parzen windows, Test data are assigned to the class with the maximum local density [8].

Yet another non-parametric representation of speaker data closely related to the nearest neighbor outlined above, uses vector quantization (VQ) [2]. In this approach, the space of features from a speaker is divided into multidimensional Voronoi regions, and each region is represented by its centroid stored into a vector codebook. A test vector is associated with the speaker that has the closest vector in his/her codebook.

In general, non-parametric models have the drawback of not generalizing well and hence are not very popular.

1.4.2 Parametric Models

The more commonly applied class of statistical models in speaker recognition, because of their robustness, is that of parametric models [2][4]. In this class, models are represented by a condensed set of parameters that typically represent a smoothened distribution of the enrollment data. This smoothening is mainly responsible for the greater capacity of generalization, and hence, robustness of the models.

One such approach, the Gaussian Mixture Modeling (GMM), is the most popular approach to speaker modeling [5][11]. Almost all the work on speaker verification utilizes GMMs entirely or in the form of hybrids with other modeling techniques. Due to the central nature of this model to this dissertation work, a separate chapter, Chapter 3 has been devoted to this model and its application in speech processing systems.

One enhancement that is employed in many GMM-based systems but not considered in the current work is the adapted GMM approach. Since the enrollment data for each speaker in many systems is quite limited, there may not be sufficient data to reliably estimate large GMM models. One solution to this problem is to train a common universal model with large number of mixtures using a large amount of speech data from another source. Then, using the limited data from each enrollment speaker, we can adapt intelligently only selected mean vectors in the universal GMM model to form the speaker model. Also, scoring of speaker models can be made efficient by considering only these adapted means during evaluation. Typically, maximum a-posteriori (MAP) method is used for adapting universal models to individual speaker models. Also, instead of using ML techniques for training GMM that optimize parameters for best statistical representation of speakers independently, discriminative techniques can be used to train GMM models that maximize the separation between the models. Such discriminative techniques work by concentrating more on modeling differences between speaker characteristics and suppressing characteristics that are common to speakers. This helps improve speaker recognition performance appreciably.

The most commonly used statistical model for speech recognition – a Hidden Markov Model (HMM) – has also been applied to speaker modeling. In this model, states represent quasi-stationary segments of speech typically using GMM modeling, and passage between quasi-stationary segments are represented using a state transition probability matrix. Though HMMs are found to be unarguably useful for speech recognition, they do not appear to be any more useful than GMMs for speaker recognition.

Several discriminative kernel-based approaches have also been applied in speech processing systems [12][13][14] . More recently, discriminatory modeling techniques are finding increasing application in speaker recognition [15][16][17]. One such particular approach, Support Vector Machines (SVMs) is particularly suited to 2-class problems, and hence, applicable directly in speaker verification. In this, data from the enrollment speaker are taken to be positive samples and the data from impostors as negative samples. These are used to identify optimal hyper-planes for classification by the choice of a few selected sample points called support vectors on the margin between the two classes in a kernel-transformed space. Test data are transformed into a high-dimensional space using the kernel and compared with the support vectors for accepting or rejecting the speaker's identity claim.

However, the GMM still remains an important baseline model because its performance has been studied across a variety of databases. The work presented here attempts to address the drawbacks of GMM modeling by using a nonlinear mixture autoregressive model for speaker modeling.

1.5 Speaker Verification – A Concise History

Speech processing systems started in an analog world through such techniques like bands of analog filters analyzing frequency content of the signal. In-roads in digital signal processing leading to efficient algorithms for computation (eg. Fast Fourier Transform) and efficient digital signal representation techniques coupled with increasing hardware computing speeds accommodated more complex algorithms required for speech processing. In 1976, a prototype speaker recognition system was built by Texas Instruments [18]. Later, National Institute of Standards and Technology (NIST) developed a speech group to promote speech processing research [19]. To this day, NIST speaker recognition evaluation workshops provide a convenient platform for the speaker recognition community to develop and advance on earlier algorithms.

Automatic speech and speaker recognition systems have co-evolved since their inception. Almost every development in new speech features or modeling approach in one field has been applied to the other. Improvements in technology yielded the now-familiar LPC and MFCC features while vector quantization (VQ), GMMs, HMMs, and SVMs represent the commonalities in the modeling paradigm [1][2]. The differing viewpoints of the speaker-independent speech recognition systems and the text-independent speaker recognition systems have naturally led to fine-tuning the commonly used features and modeling approaches. For instance, while GMMs in speech recognition are trained to represent different speakers uttering a similar sound, those in speaker recognition systems are trained to represent different sounds from the same speaker.

One early modeling approach to speaker recognition was a simple time-averaging of speech features to represent the speaker utterance [20]. These were computationally simple but the performance was poor. The subsequent improvements like VQ, GMM and SVM improved performance but used much more computations. Now, surprisingly there has been a resurgence of interest in representing speaker information in the form of vectors called supervectors. In fact, most of the models like GMMs and SVMs can be thought of as supervectors [20][21]. (Such reemergence of old abandoned theories and models appear to be common feature across all sciences; ideas indeed die hard).

Another development that started after mid-1990s is the research on robust speaker recognition systems based on score normalization. While the feature based compensation techniques were borrowed from speech recognition work, these score normalization were specific to speaker recognition systems. More information on this subject can be found in Chapter 3.

Interestingly, in more recent times, there has been an increasing convergence of the two systems: speech recognition systems on cell-phones tend to rely on speaker-specific models adapted from speaker-independent models; also some systems segregate their models based on some speaker clusters (e.g., male vs female) and use the appropriate model based on the detected speaker cluster during recognition phase. On the other hand, speaker recognition systems have started exploring the use of higher level features like words and phrases that could be distinctive of the speakers. This requires a speech recognition pass to extract the higher-level features. Another situation where both the recognition systems come together is related to automatic speaker diarization of a conference meeting – identify and tag the different speaker segments in a meeting, and also recognize the utterances spoken by each speaker.

1.6 Survey of Speech Feature and Model Performance Comparisons

It was mentioned earlier that Mel-Frequency Cepstral Coefficients (MFCCs) were among the most popular features for speaker recognition (as well as speech recognition) tasks. However, even within this feature set, there are different implementations that could potentially lead to different performances. One study [22] reports the differences in performance between various MFCCs as well as a related Human Factor Cepstral Coefficients (HFCC) features built over a probabilistic neural network (PNN) for a text-independent speaker recognition system on the 2001 NIST SRE evaluation data [19]. It concluded that there was no significant difference in performance between the different implementations, though some variation in EER (13.65%-15.41%) was noted. The same research group also published a study [23] with different wavelet features in comparison with MFCCs on 2001 NIST SRE evaluation data. It was reported that EER with MFCCs was 17.54% while that using wavelet coefficients ranged between 16.12% and 19.61% depending on the type of wavelet feature used. Using compensation techniques as well as subsets of the feature vector, the EER was reduced to less than 14%. These results show that even within low-level features, differences in the choice of feature set as well as its implementation can affect the performance of speaker recognition systems to some extent making comparisons between different systems difficult.

Comparisons of MFCC features with a related set of features derived from linear prediction called linear prediction cepstral features (LPCCs) have shown that MFCCs are superior to LPCCs for speaker recognition [24]. However, [25] reports little difference in the speaker identification performance of these two feature sets, while channel compensation techniques was found to have much higher impact on performance.

To evaluate the efficacy of combining information in the higher-level features for speaker recognition, [25] used eight different systems for speaker verification and both the individual and the fusion performances were studied on 2006 NIST SRE evaluation data. When using the systems individually, it was found that frame-level cepstral-based systems performed better than when using higher-level features alone; while a SVM system (EER: 5.07) performed slightly worse than a GMM system (EER: 4.75), the performance of the hybrid Gaussian supervector SVM was higher (EER: 4.15). Also, fusing higher-level features increased the performance considerably; with the 3-best system, EER reduced to 2.86 and using all eight systems, EER reduced to 2.59. This shows that higher-level features can significantly improve speaker recognition performance.

1.7 Speaker Recognition State-of-the-art Performances on Standard Databases

For over the past two decades, several databases have become standard for evaluating speaker recognition systems on. It is not possible to discuss all the best performing speaker recognition systems here for two reasons: first, there are a wide variety of databases available for evaluating speaker recognition with each one having its own range of relevance to practical applications. For instance, TIMIT [87] database is typically used when data for each speaker is very limited and is typically used for initial test of new models and ideas. NTIMIT [94] is similar but with data passed through telephone bandwidth channels. On the other hand, NIST speaker recognition evaluation (NIST SRE) databases provide a larger scope [19]. These data are created and made available every few years and are typically evaluated on more mature technology. Typically, data from older years are used for training initial UBM models. Newer data contain data that often reflect the emergence of novel applications like speaker diarization. YOHO Speaker Verification corpus is another popular database distributed by Linguistic Data Consortium (LDC) specifically for the purpose of evaluating speaker verification systems [27]. Second, even on the same database, different assumptions about the application can lead to considerably different performances. For instance, a researcher with prior knowledge of the application might consider it necessary to include channel compensation techniques during feature extraction or modeling, though the performance on the evaluated database could sometimes suffer appreciably by performing compensation. In addition, several systems perform a silence removal step that also could affect the realized performance considerably. Thus, when comparing performances of two systems, it is not only necessary to consider the major differences in the modeling and feature extraction approaches, but it is also important to take minor differences like channel compensation and silence removal into account. This difficulty in comparison due to differing conditions in various systems was previously mentioned in the previous section and also noted in [35]. However, the published baselines are useful in providing rough estimates on the order of magnitudes we can expect with evaluations on a database. With this caveat, we outline below the performances obtained by some of the popular speaker recognition systems.

1.7.1 On TIMIT Database

TIMIT corpus [87] is a popular database for experiments related to phonetic classification task because of the phonetically balanced design of the database. However, this is also popular for speaker recognition evaluations. Several baselines have been published for various models and they differ widely in their performances. Reynolds work [11] on GMM using models adapted from a Universal Background Model (UBM) showed an EER performance of 0.24% on the core set of 168 speakers while [28] obtained an EER of 0.65% using the same modeling approach. In another report [29], researchers found that EER using adapted GMM approach was lower at 4.45% compared to the 2.52% for their discriminative approach using a maximum-model distance (MMD) approach. It is to be noted that there were variations in the way the speaker data were split for enrollment and testing between these published baselines.

1.7.2 On NTIMIT Database

NTIMIT corpus [94] is TIMIT data filtered through telephone bandwidth channels. Thus unsurprisingly, published baselines on this database are much worse than those for TIMIT corpora [11] reported an EER of 7.19% while [28] reported an EER of 12.41% using the adapted GMM approach with the core test data from 168 speakers. Another work [30] dividing the NTIMIT data differently used a word spotting technique over hidden Markov models and demonstrated an average error rate ((false acceptance + false rejection) / 2) of 16.96%. One research on channel compensation techniques came up with their best estimate of 19.6% EER for the 168 core-speaker set [34] with training done only on the TIMIT database.

1.7.3 On NIST Speaker Recognition Evaluation (SREs) Databases

Of all the standard databases used for speaker recognition evaluations, NIST SREs are perhaps the most comprehensive and powerful. Since 1996 NIST has been instrumental in establishing state-of-the-art feature extraction and modeling approaches [19]. New data and benchmarks are released almost every year and to some extent also reflect the expanding scope and evolution of the speaker recognition systems.

Rules for evaluating data and reporting results are outlined in the evaluation plan documents for each year, and the results are summarized. In addition, this is followed by yearly workshops that allow researchers to meet and discuss their findings.

It is not possible to outline the results from all the year's data here. In this work, we only use the NIST SRE 2001 development data for evaluation. This was chosen since we already had this database available with us and the development data is smaller than the evaluation data, enabling us to quickly estimate the performance of our model compared to published baselines on the evaluation data. Hence, in the following we outline only high points of published results on this data.

Several baselines have been published for 2001 NIST SRE evaluation data. In [31], the baseline GMM gave an EER of 20.6% with 32 mixtures, and a best GMM performance of 18.8% with 128 mixtures. Using a tokenization approach an improved performance was obtained (EER around 13%). [23] compared different wavelet features with MFCCs. It was reported that EER with MFCCs was 17.54% while that using wavelet coefficients ranged between 16.12% and 19.61% depending on the type of wavelet feature used. Using compensation techniques as well as subsets of the feature vector, EER was reduced to less than 14%. In [32], the EER was reduced from the baseline 8.64% to 8.04% by jointly modeling prosodic and spectral features using an ergodic HMM (EHMM).

Another feature of the 2001 NIST SRE (like the other NIST SREs) is that there are several tasks other than the standard speaker recognition (referred to as the one-speaker detection task basic evaluation corpus) described in the evaluation plan. The other relevant task is the extended data task, which consists of much larger amounts of data for training and testing. Experiments with this data usually perform significantly better than with the standard 1-speaker detection task. [33] reports an EER of 0.7% using low-level acoustic features on the 2001 NIST SRE extended task using 8 conversations, and this was further reduced to 0.5% by fusing higher-level prosodic feature information.

1.7.4 On YOHO Database

With YOHO database [27], using GMMs adapted from UBMs, speaker verification performances in terms of equal error rate (EER) varied between ~2% [28] to 0.51% [11]. Earlier reported results on this database on various models varied between 1.42% [35][36], to 0.5% [35][37]. Also, in [38], a mean EER of HMMs 0.473, while a discriminative technique called mixture decomposition discrimination (MDD) yielded an average EER of 1.239% and a hybrid system yielded 0.255%. It is to be noted that wide variation in performances is not only due to variation in modeling and speech features used, but also due to other differences in conditions used in the evaluation procedure.

In this work, we work with TIMIT, NTIMIT and the 2001 NIST SRE development data in addition to some synthetic data.

1.8 Dissertation Organization

The structure of this dissertation is outlined below. The current chapter dealt with an introduction to the problem of speaker verification, the standard features used to represent the speech signal information, and also briefly surveyed the speaker modeling schemes found in literature.

Chapter II examines the evidence from primary literature for the presence of nonlinearities in the speech signal and its consequences to speech processing. After delving into details of the work related to nonlinear dynamical invariants, it also addresses the problems associated with the use of these invariants as features to represent the nonlinear information in the speech signal. Other approaches to nonlinear modeling for speaker verification are discussed. Finally, some novel ideas for speech processing contributed by mathematicians are outlined.

Chapter III is devoted to discussing the basics of Gaussian Mixture Modeling (GMM) and its application to speaker recognition. It is argued here that GMMs are incapable of modeling nonlinear evolution information in the features and that the use of differential features is only a linear approximation to the actual nonlinear dynamics.

Chapter IV motivates the need for a nonlinear model in speech processing. The mixture autoregressive (MixAR) model is borrowed from statistical literature as a novel nonlinear statistical model in speech processing. Comparisons of the MixAR model to the GMM model and other autoregressive models in speech literature are made. The problem of parameter estimation is discussed. Also, efficient implementation techniques are explored.

Chapter V discusses the set of preliminary experiments that were run to evaluate the performance of MixAR in relation to GMM models. After two classification experiments on synthetic data, speaker verification experiments on both synthetic and standard speech data are presented.

Chapter VI discusses the core set of speaker verification experiments with real speech data. Experiments addressing various issues including noise performance, channel variations, statistical significance, and duration effects are discussed in this chapter. These experiments are conducted on a variety of standard databases – TIMIT, NTIMIT, 2001 and NIST SRE development database.

Chapter VII presents the conclusion and future scope of this dissertation work.

CHAPTER II

NONLINEARITIES IN THE SPEECH SIGNAL – A BRIEF SURVEY

In this chapter, the nonlinear nature of speech signals and its consequences in speech and speaker recognition are examined. First, the concept of nonlinear dynamical invariants is introduced. Then the application of these invariants to speech processing is examined. The drawbacks of using nonlinear dynamical invariants as features in speech processing is exposed, motivating alternative approaches to representing the nonlinear dynamic information contained in speech. Some unconventional and little studied nonlinear methods are also briefly surveyed.

2.1 Introduction

Speech recognition systems today still exploit the linear acoustics model of speech production, and rely on traditional measures of the spectrum based on Fourier transforms [1]. Though applications of machine learning to speech recognition have made great strides in recent years, high performance speech recognition systems are still sensitive to mismatches in training and evaluation conditions, or dramatic changes in the acoustic environments in which they operate. We refer to this as the robustness problem – can a speech recognition system achieve high performance on noisy data that has not been observed during training? Our goal in this work is to produce new features for speech recognition that do not rely on traditional measures of the first and second order moments of the signal.

Dynamical systems can be represented by state-space models, where the states of the system evolve in accordance with a deterministic evolution function, and the measurement function maps the states to the observables. The path traced by the system's states as they evolve over time is referred to as a *trajectory*. An *attractor* is defined as the set of points in the state space that are accumulated in the limit as $t \to \infty$. *Invariants* of a system's attractor are measures that quantify the topological or geometrical properties of the attractor, and are invariant under smooth transformations of the space. These smooth transformations include coordinate transformations such as Phase Space Reconstruction of the observed time series [18].

These invariants are a natural choice for characterizing the system that generated the observable. These measures have been previously studied in the context of analysis and synthesis research [18][40] and more recently in the context of speech recognition [41]. In this section, we review algorithms to extract these invariants using a pilot database consisting of elongated pronunciations of a small set of phones, and study discriminability in a feature space comprised of these invariants.

Lyapunov exponents [42] associated with a trajectory provide a measure of the average rates of convergence and divergence of nearby trajectories. Fractal dimension [43] is a measure that quantifies the number of degrees of freedom and the extent of self-similarity in the attractor's structure. Kolmogorov entropy [43] defined over a state-space, measures the rate of information loss or gain over the trajectory. These measures search for a signature of chaos in the observed time series. Since these measures quantify the structure of the underlying nonlinear dynamical system, they are prime candidates for feature extraction of a signal with strong nonlinearities.

One long-term goal in this research was to model phones using dynamical systems, where the state-space that generated the observed acoustic sound corresponds to a unique configuration of the articulators and the driving process. Since each configuration corresponds to a unique attractor in the phase space, it is expected that the invariants extracted from different phones will mirror differences in the corresponding attractors.

Lyapunov exponents [41] have been employed as features in a phonetic recognition system, and studied in combination with conventional cepstral features. In this work, we extend the analysis to three standard invariants of a dynamical system. The motivation behind studying such invariants from a signal processing perspective is to capture the relevant nonlinear dynamical information from the time series – something that is ignored in conventional spectral-based analysis.

2.2 Nonlinear Dynamical Invariants for Broad-Phone Classification Speech

To characterize the structure of the underlying strange attractor from an observed time series, it is necessary to reconstruct a phase space from the time series. This reconstructed phase space captures the structure of the original system's attractor (the true state-space that generated the observable). The process of reconstructing the system's attractor is commonly referred to as embedding.

The simplest method to embed scalar data is the method of delays. In this method, the pseudo phase-space is reconstructed from a scalar time series, by using delayed copies of the original time series as components of the RPS. It involves sliding a window of length *m* through the data to form a series of vectors, stacked row-wise in the matrix.

Each row of this matrix is a point in the reconstructed phase-space. Letting x_j represent the time series, the reconstructed phase space (RPS) is represented as:

$$X = \begin{pmatrix} x_0 & x_\tau & \cdots & x_{(m-1)\tau} \\ x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & & \vdots & \end{pmatrix} \qquad \text{Eq. 2.1}$$

where, m is the embedding dimension and τ is the embedding delay

Taken's theorem [42] provides a suitable value for the embedding dimension, m. The first minima of the auto-mutual information versus delay plot of the time series is a safe choice for embedding delay [42].

2.2.1 Lyapunov Exponents

The analysis of separation in time of two trajectories with infinitesimally close initial points is measured by Lyapunov exponents [42]. For a system whose evolution function is defined by a function f, we need to analyze:

$$\Delta x(t) \approx \Delta x(0) \frac{d}{dx} (f^N) x(0) \qquad \text{Eq. 2.2}$$

To quantify this separation, we assume that the rate of growth (or decay) of the separation between the trajectories is exponential in time. Hence we define the exponents, λ_i as:

$$\lambda_i = \lim_{n \to \infty} \frac{1}{n} \ln(\mathrm{eig}_i \prod_{p=0}^{n} J(p)) \qquad \text{Eq. 2.3}$$

where, J is the Jacobian of the system as the point p moves around the attractor. These exponents are invariant characteristics of the system and are called Lyapunov exponents, and are calculating by applying the above equation to points on the reconstructed attractor. The exponents read from a reconstructed attractor measure the rate of separation of nearby trajectories averaged over the entire attractor.

2.2.2 Fractal Dimension

Fractals are objects which are self-similar at various resolutions. Self-similarity in a geometrical structure is a strong signature of a fractal object. Correlation dimension [43] is a popular choice for numerically estimating the fractal dimension of the attractor. The power-law relation between the correlation integral of an attractor and the neighborhood radius of the analysis hyper-sphere can be used to provide an estimate of the fractal dimension:

$$D = \lim_{N \to \infty} \lim_{\varepsilon \to 0} \frac{\partial \ln C(\varepsilon)}{\partial \ln \varepsilon} \qquad \text{Eq. 2.4}$$

where $C(\varepsilon)$, the correlation integral is defined as:

$$C(\varepsilon) = \frac{2}{N*(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \Theta(\varepsilon - \left\| \vec{x}_i - \vec{x}_j \right\|) \qquad \text{Eq. 2.5}$$

where $\vec{x}$ is a point on the attractor (which has N such points). The correlation integral is essentially a measure of the number of points within a neighborhood of radius ε, averaged over the entire attractor. To avoid temporal correlations in the time series from producing an underestimated dimension, we use Theiler's correction for estimating the correlation integral [43].

2.2.3 Kolmogorov-Sinai Entropy

Entropy is a well known measure used to quantify the amount of disorder in a system. It has also been associated with the amount of information stored in general probability distributions.

Numerically, the Kolmogorov entropy can be estimated as the second order Renyi entropy (K_2) and can be related to the correlation integral of the reconstructed attractor [43] as:

$$C_d(\varepsilon) \sim \lim_{\substack{\varepsilon \to 0 \\ d \to \infty}} \varepsilon^D \exp(-\tau\, d\, K_2) \qquad \text{Eq. 2.6}$$

where D is the fractal dimension of the system's attractor, d is the embedding dimension and τ is the time-delay used for attractor reconstruction. This leads to the relation:

$$K_2 \sim \frac{1}{\tau} \lim_{\substack{\varepsilon \to 0 \\ d \to \infty}} \ln \frac{C_d(\varepsilon)}{C_{d+1}(\varepsilon)} \qquad \text{Eq. 2.7}$$

In a practical situation, the values of ε and d are restricted by the resolution of the attractor and the length of the time series.

2.3 Nonlinear Invariants for Broad-Phone Classification of Speech

In this work, the three nonlinear dynamical invariants described above were extracted by reconstructing the underlying attractor from the observed acoustic utterances. We collected artificially elongated pronunciations of several vowels and consonants from 4 male and 3 female speakers. Each speaker produced sustained sounds (4 seconds long) for three vowels (/aa/, /ae/, /eh/), two nasals (/m/, /n/) and three fricatives (/f/, /sh/, /z/). The data was sampled at 22,050 Hz. For this preliminary study, (c we wanted to avoid artifacts introduced by coarticulation.

The acoustic data from each phoneme was embedded into a reconstructed phase space using time delay embedding with a delay of 10 samples. This delay was selected as the first local minimum of the auto-mutual information vs. delay curve averaged across all phones.

(f

The choice of an embedding dimension of 5 was made after observing the plots of the Lyapunov spectra vs. embedding dimension over a range of embedding dimensions, and noting that the estimates of the Lyapunov spectra converge at an embedding dimension of 5 for most phones, as shown in Figure 2.1.

To estimate the Lyapunov spectra from speech data, the algorithm described (i (1 in [42] was used. Experimentally optimal values were found to be (by varying the parameters and choosing the value at which we obtain convergence of the largest Lyapunov exponent) 30 for number of nearest neighbors, 5 for the evolution step size, and 15 for the number of sub-groups of neighbors. A more detailed explanation of these parameters can be found in [40]. For estimates of Kolmogorov entropy, an embedding dimension of 15 was used. It is clear from the figure that, for reliable entropy estimates, a high embedding dimension must be used.

As a measure of discrimination information between two statistical models representing dynamical information, the Kullback-Leibler (KL) divergence measure was chosen [3]. Invariants were measured for each phoneme using a sliding window, and an accumulated statistical model over each such utterance was built. The discrimination information between a pair of models $p_i(\vec{x})$ and $p_j(\vec{x})$ is given by:

$$J(i,j) = \int_{\vec{x}} p_i(\vec{x}) \ln \frac{p_i(\vec{x})}{p_j(\vec{x})} d\vec{x} \ + \int_{\vec{x}} p_j(\vec{x}) \ln \frac{p_j(\vec{x})}{p_i(\vec{x})} d\vec{x} \qquad \text{Eq. 2.8}$$

J(i,j) provides a symmetric divergence measure between two probability distributions *i* and *j*, from an information theoretic perspective. *J* was used as the metric for quantifying the amount of discrimination information across dynamical invariants extracted from different broad phonetic classes.

2.3.1 Broad-Phone Classification Results Using Invariants

Figure 2.1 shows the three dynamical invariants extracted from various phones using a variety of analysis parameters. For these experiments, a window size of 1,500 samples was chosen. For the set of plots (a) through (c) in Fig 4, the value of the neighborhood radius (epsilon) was varied and the variation in estimated fractal dimension with this parameter was studied. We observe a clear scaling region (where the dimension estimate is unaffected by variations in the neighborhood radius) for vowels and nasals (at epsilon ~ 0.75). Such a scaling region is not present in dimension estimates from fricatives. Also note that the estimate of fractal dimension for vowels and nasals is not sensitive to variations in embedding dimension from 5 through 8. However, the dimension estimate for fricatives increases consistently with an increase in the embedding dimension.

A similar trend is observed for plots (d) through (f), representing the Kolmogorov entropy estimates as a function of the embedding dimension. Once again, vowels and nasals have entropy estimates that stabilize at an embedding dimension of approximately 15. The entropy estimates for fricatives increase consistently with the embedding dimension. This behavior, along with the variation in dimension estimates with embedding dimension, reaffirms the conventional belief that unvoiced fricatives can be

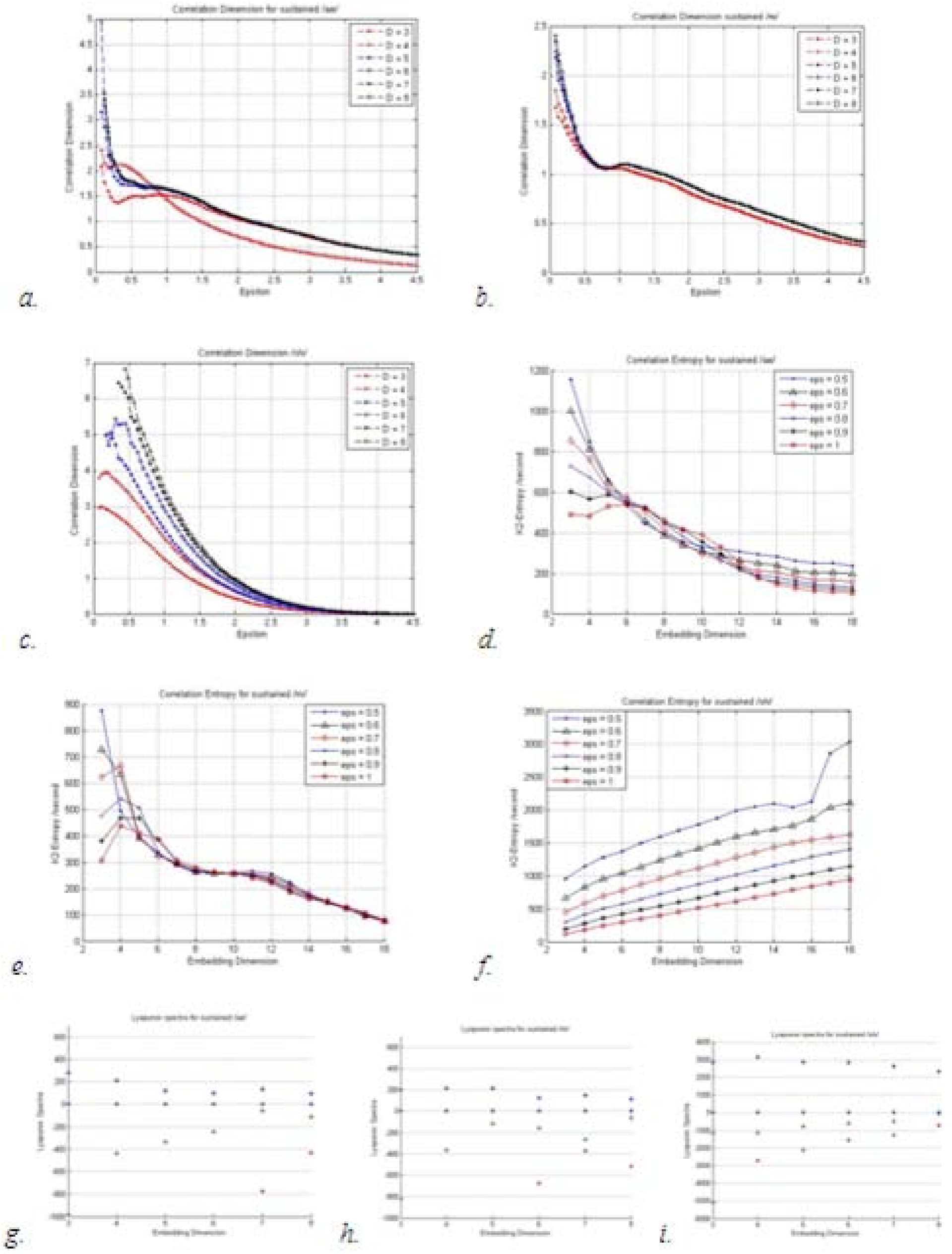


Figure 2.1 Correlation Dimension (a through c), Kolmogorov Entropy (d through f),

Lyapunov Spectra (g through i) estimates for a vowel, a nasal and a fricativemodeled using the combination of a noisy source and linear constant coefficient digital filter. If a time series were generated from an IID stochastic process, an increase in the embedding dimension adds to the randomness in the reconstructed phase space of this series, and hence leads to consistently increasing estimates of fractal dimension and attractor entropy. In [41], estimates of Lyapunov exponents could not be validated for fricatives, which is consistent with our observations using fractal dimension and Kolmogorov entropy estimates.

Plots (g) through (i) depict the Lyapunov spectra as a function of various embedding dimensions. Note that the positive exponent converges to a stable value at an embedding dimension of 5. Another technique for estimating the appropriate embedding dimension from a time series is the method of false nearest neighbors [43].

Figure 2.2 below depicts the KL-divergence measure between phone models formed using the nonlinear dynamical invariants as features. Equation 8 has a closed form expression for normal distributions with different mean vectors and covariance matrices, which is what we used for estimating these divergence measures. A sliding window of length 36 ms to extract the invariants was used. The plots in this figure indicate the separation between statistical models generated using correlation entropy, Lyapunov exponents and correlation dimension extracted from utterances of all seven speakers. Note that the discrimination information of these features is high between vowels and fricatives and nasals and fricatives. The separation between nasals and vowel sounds is small.

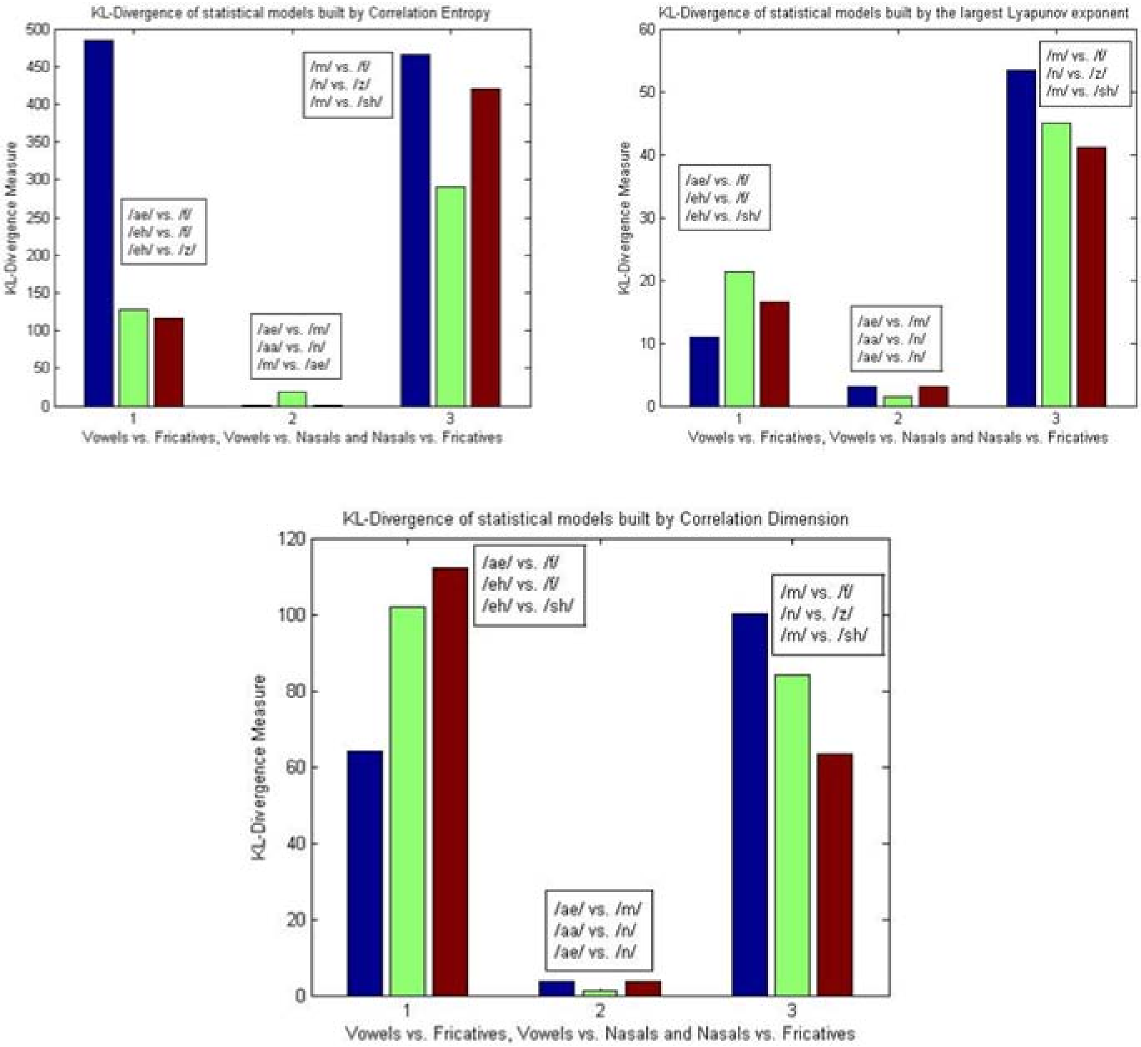


Figure 2.2 KL Divergence Measure across various phonemes, using the three dynamical invariants.

To conclude this subsection, it was shown that the 8 phonetic sounds are pairwise separable using the three nonlinear dynamic invariants alone. These results show promise in the potential use of these invariants for speech recognition applications.

2.4 Survey of Application of Nonlinear Invariants in Speech Recognition

Prior to the 1960s, speech production in the vocal tract was considered a passive linear process. Later, Teager [44] showed that nonlinear mechanisms underlie the speech production process. However, until fairly recently, most speech modeling was based on linear representation of signals – particularly Linear Prediction and its variants [45][46]. This was primarily because of the ease of dealing with linear models, and also the limited computational power available at that time.

Over the past decade or so, there has been a resurgence of interest in accounting for the nonlinear nature of speech signals. Recent work suggests that speech signals have nonlinearities that could contain relevant information in speech processing [41][47][56] The majority of this type of research relies on extracting novel speech features known as nonlinear dynamic invariants, and then using these nonlinear features along with conventional features in conventional pattern recognition or machine learning systems. The goal of these approaches is to supplement nonlinear dynamic information that the conventional feature set does not possess.

Nonlinear dynamic invariants quantify the degree of nonlinearity in a signal. These do not vary with transformations of the signal as long as they are smooth and invertible (such transformations are called diffeomorphisms and are one of the branches of study in topology) [52]. Hence, these coefficients are called invariants. The three most commonly used nonlinear invariants are Lyapunov exponents, fractal dimension, and correlation entropy [43][51][55]. Lyapunov exponents characterize the rate of divergence between nearby trajectories in the phase-space of the signal, while correlation entropy quantifies the rate of information gain or loss. Both these signify the sensitivity of the system to initial conditions. Fractal dimensions capture the geometry of self-similar systems. All three signify the presence or absence of chaos in the system dynamics and can aid in detecting presence of nonlinearity in a signal.

The most common application of nonlinear invariants in speech and speaker recognition systems is to consider them as features and concatenate them with conventional features like MFCCs. In [48] it was shown that the additional information in the nonlinear invariant features extracted from speech could be beneficial in a phone classification task. May [47][48] demonstrated improvements when adding invariant features with MFCCs on a continuous speech recognition task involving noise-free recording conditions, but found that the combined performance worsened when noise was present. For speaker identification, Petry, *et al* [56] showed an improvement in identification accuracy of about 1% relative by adding nonlinear invariant features to cepstral features. This improvement is only marginal considering that the database was composed of only isolated digits.

2.5 Drawbacks of Using Nonlinear Invariant Features in Speech Processing

Judging by the results of the representative examples cited above, it is clear that adding nonlinear invariants as features has not improved the robustness of speech and speaker recognition technologies in harsh or mismatched environments. This failure can be attributed to two reasons. First, it is difficult to estimate invariants reliably from speech. In addition to the extensive tuning required by the parameter estimation algorithms, there is also the problem of a requirement of large durations of the acoustic event [52]. This gravely undermines the applicability of invariant features for a short-time stationary signal like speech. Even if it was somehow possible to estimate the invariants accurately, there is the second and more fundamental problem that invariants only quantify the degree of nonlinearity and do not characterize the nature of the dynamics completely.

This lack of success of nonlinear invariants in improving robustness in speech processing does not imply that nonlinear information is not present or that it is not so useful in speech. Rather, the current evidence from nonlinear invariants provides an almost unequivocal support to the presence of nonlinear dynamics in speech, and we should explore other ways of exploiting this information to advantage. The current work attempts to do this at the modeling level.

2.6 Survey of Other Nonlinear Approaches for Speaker Recognition

Teager and Teager [44] studied the speech signal in the context of modulations of the speech airflow and turbulence. Several physical measures were presented in that work that shows the presence of turbulence of airflow associated with speech. Since that work, along with the advancements in computational power, processing of speech has seen a large number of novel nonlinear modeling approaches for speaker recognition. However, these have been dominated by the more popular nonlinear invariant features approach. With the increasing realization of the limitations of invariant features it is expected that more efforts would be applied on other nonlinear approaches in the future.

Some of the approaches to capture nonlinear information in speech, are motivated by the physical mechanism of the human speech production. In [58] and [59], the speaker verification performance was improved from an EER of 8.9% using only cepstral features to an EER of 7.3% by also including features derived from low-level cochlear models and high-level modulation features. These features were modeled using the GMM-UBM approach on the 2003 NISR SRE one-speaker evaluation database. Work in [60] modeled the glottal flow with the traditional source-filter model with a nonlinear interaction that could also account for the formant modulation noted in the spectrograms. Using features derived from this model, the baseline MFCC performance was improved by as much as 20% relative for smaller datasets.

Some researchers base their nonlinear component in neural networks. One such work [61], considers a Neural Predictive Coding (NPC) as an extension to the classical linear predictive coding (LPC). Features derived from NPC were found to be superior to the traditional ones like LPC and MFCC for a speaker identification task.

By considering a smaller subset of TIMIT database, it was demonstrated in [62] that the use of residual energy after LPCC feature extraction reduces the speaker identification error rate from 6.31% to 3.68%. If nonlinear predictive neural net was used in the place of LPCC, the performance further improved showing an identification error rate of 2.63%.

Another recent interesting work dynamically controls the phoneme class information that is used in speaker recognition using minimum mutual information measures in a nonlinear optimization technique [63]. Speaker recognition experiments indicate that this method increases correct identification rate by 18% relative to a conventional baseline system. In this case, it not the actual nonlinear dynamics in speech that is relevant here but the applicability of standard nonlinear modeling tools like mutual information and nonlinear optimization.

A different class of nonlinear models is Empirical Mode Decomposition (EMD) [64][65][66]. In this method, the signal is decomposed into components called Intrinsic Mode Functions (IMFs) that have zero mean between their maximum and minimum envelopes. From the instantaneous frequency (IF) and energies of all the IMFs, a time-frequency representation can be obtained that can bypass the resolution time and frequency limitations of traditional linear Fourier and wavelet transforms. EMD approach is related to the AM-FM decomposition of the signal into its amplitude envelope and IF components [65]. In [65], it was noted that the combination of AM-FM decomposition features with MFCCs provided a 10% relative improvement in EER performance compared to that using MFCC only on 2001 NIST SRE evaluation data. Speaker identification experiments have also demonstrated improvements with the EMD approach (correct identification rate 95.1%) over the traditional GMM based approach (correct identification rate 92.5%) [66]. It remains to be seen if these results hold up in the presence of noise and for different databases.

2.7 Other novel nonlinear approaches in speech systems

In addition to the traditional three nonlinear invariants, there are several new nonlinear processing methodologies that are alien to current speech processing systems.

At the level of using novel nonlinear speech features, several interesting ones are yet to be explored. Almost all prior work has been in the realm of the three invariants mentioned before. There has been a great deal of mathematical research in this field but there has been little dialogue between the mathematicians and speech scientists about such novelties. Most of the contributions of the fascinating branch of topology to speech have been through the work of mathematicians trying to find data that would fit their topological features and models.

One such work uses the idea of creating topological voiceprints for speaker identification [67]. In this method, the spectrum from the speech signal is extracted. Then the spectrum is used to compute a set of topological indices like relative rotation rates (RRRs). The advantage of such indices is that these are quantified to rational numbers and hence could potentially be robust to noise.

A more recent work involves a branch of mathematical topology called persistent homology [68]. At a fundamental and abstract level, homology aims to classify spaces into equivalence classes according to their topological structure. If one space can be mapped one-one onto another through an invertible smooth transformation the two spaces are considered equivalent [52][69]. For the speech time series, first the researchers used time embedding to create the reconstructed phase space (RPS). Then the point clouds in the space were used to build a manifold using homological complexes. To deal with noisy data, a technique called persistent homology was used. This way the underlying manifold structure of speech could be extracted even under noisy conditions. How well such methods perform for speech and speaker recognition remains to be studied.

At the modeling level, there are several nonlinear signal models like MAR [78], and MixAR [77] that have not been explored for speech applications before and these form the central theme of this dissertation work.

CHAPTER III

THE GUASSIAN MIXTURE MODEL

In the introductory chapter, it was stated that the majority of speaker recognition systems utilize Gaussian Mixture Models (GMMs) either entirely or as part of a hybrid model. In this chapter, the concept of a GMM is introduced. The motivation for these models and why such models currently form the basis of current speech and speaker recognition systems is explained. Next, the main drawbacks of using GMMs are exposed, with particular emphasis placed on its inefficiency at modeling nonlinearly evolving feature streams. Recent findings on the presence of significant nonlinearities in speech signals were summarized in Chapter 2, and provided motivation for the use of nonlinear time-series models for better speech representation – the subject of the next chapter.

3.1 Basic Definition

A random variable ***x*** drawn from a Gaussian Mixture Model has a probability density function defined by [1][5]:

$$p(x) = \sum_{i=1}^{m} W_i N(x \mid \mu_i, \Sigma_i) \qquad \text{Eq. 3.1}$$

Here, *m* is called the order of the GMM and the Gaussian distribution $\mathrm{N}(\mathrm{x} \mid \mu_\mathrm{i}, \sigma_\mathrm{i})$ with mean vector μ_i and covariance matrix Σ_i is defined by:

$$N(x \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{m/2} |\Sigma_i|^{1/2}} \exp(-\frac{1}{2}(x - \mu_i)' \\ (\Sigma_i)^{-1}(x - \mu_i)) \qquad \text{Eq. 3.2}$$

It is apparent from this definition that a GMM can be thought of as a linear functional decomposition using Gaussian distribution functions as basis functions. This interpretation immediately leads us to ask if any function can be represented using a GMM. For our current purposes, it is sufficient to state that almost any probability distribution arising in the real world can be approximated with a GMM to any desired degree of accuracy provided the order *m* is large enough. An overview of GMM model is given in the following figure.

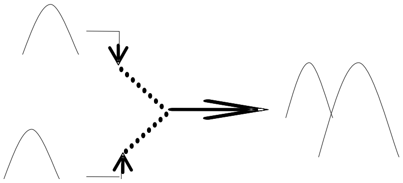


Figure 3.1 An overview of the GMM approach.

Another equivalent way of representing a GMM is the following:

$$x = \begin{cases} \mu_1 + \varepsilon_1 & w.p. \quad W_1 \\ \mu_2 + \varepsilon_2 & w.p. \quad W_2 \\ & \vdots \\ \mu_m + \varepsilon_m & w.p. \quad W_m \end{cases} \qquad \text{Eq. 3.3}$$

where ε_1 is a normal random variable with mean 0 and covariance Σ_i, and *w.p.* denotes “with probability”. According to this interpretation of GMM, a sample is drawn from any one of the *m* possible distinct modes (or single Gaussian densities), and the probability that mode *i* is chosen is determined by its weight W_i.

Such multimodal distributions are extremely useful in speech processing. For example, we can decompose a population of speakers based on the fundamental frequency of their voice into three unimodal Gaussians -- one to represent male, another to represent females and a third to represent children. The overall distribution is the weighted sum of the three unimodal distributions.

3.2 GMM Parameter Estimation using the EM algorithm

Maximum-likelihood (ML) is a well-known approach to estimated parameters of a model [8]. As the name suggests, it aims to find the set of parameter values that maximizes the likelihood of the model for the given training data. A GMM parameter set consists of weights w_l of m mixtures, Gaussian means μ_l for each mixture component, and Gaussian covariance Σ_l for each mixture component:

$$\theta_l = \{\mu_l, \Sigma_l, w_l\}\ l = 1, \ldots, m\,. \qquad \text{Eq. 3.4}$$

To estimate these parameters, we first need an initial guess for these parameters and then we iterate with EM to successively refine the estimates [69][71]. One reliable way to get a good initial estimate for a GMM with m mixtures is to first train mixtures with m/2 mixtures and then perturb the means by a small value to obtain 2 components from each one. To find the parameters for a single mixture GMM, we can simply assign a weight of 1 to the single component, and assign the global mean covariance to the mean and covariance vector of the single component.

These initial parameters can be then refined recursively using an E-step [8]:

$$\gamma_l[n] = \frac{w_l p_l(x[n] \mid \theta)}{\sum_{k=1}^{m} w_k \, p_k(x[n] \mid \theta)} \qquad \text{Eq. 3.4}$$

where

$$p_l(x[n] \mid \theta) \propto \frac{1}{\sigma_l} e^{\frac{-1}{2\sigma_l}(x[n]-\mu_l)^2} . \qquad \text{Eq. 3.5}$$

is the probability a sample was generated from component *l* at time instant *n*. The corresponding M-step is given by the following equations:

$$\hat{w}_l = \frac{\sum_{n=1}^{N} \gamma_l[n]}{\sum_{n=1}^{N} \sum_{l=1}^{m} \gamma_l[n]} \qquad \text{Eq. 3.6}$$

$$\hat{\mu}_l = \frac{\sum_{n=p+1}^{N} \gamma_l[n] \, x[n]}{\sum_{n=p+1}^{N} \gamma_l[n]} \qquad \text{Eq. 3.7}$$

$$\hat{\Sigma} = \frac{\sum_{n=p+1}^{N} \gamma_l[n] \left(x[n]-\hat{\mu}_l\right)^T \left(x[n]-\hat{\mu}_l\right)}{\sum_{n=p+1}^{N} \gamma_l[n]} \qquad \text{Eq. 3.8}$$

The GMM estimation algorithm upto M mixtures can be summarized as follows:

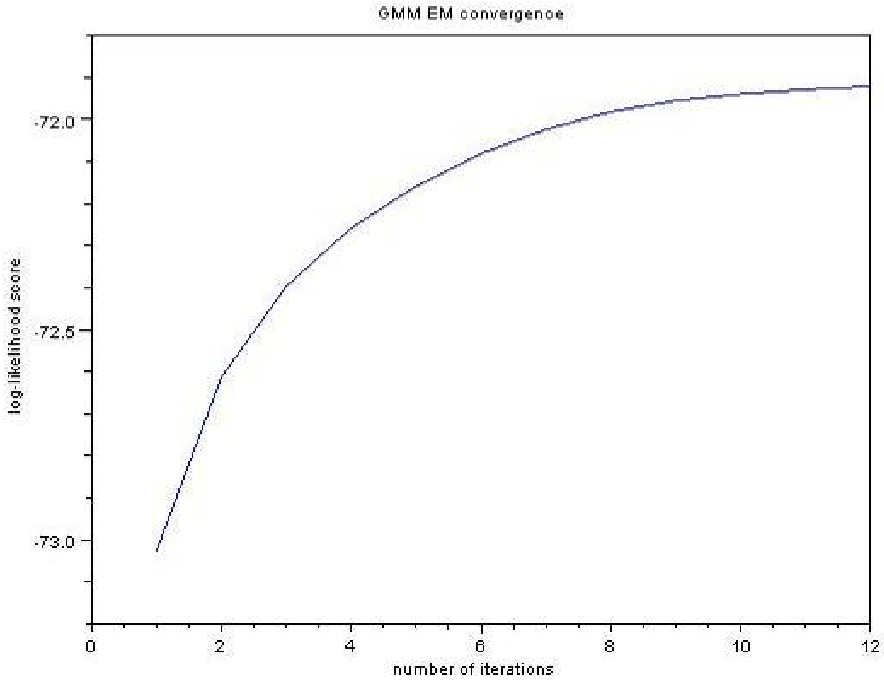


Figure 3.2 EM performance as a function of iteration for a 8-mixture GMM model of speaker data from “4516” of NIST 2001 database.

1) Initialize GMM model for number of mixtures m=1 setting weight to 1, mean to global mean of data, and covariance to global covariance of data.
2) While m < M
3) Perturb mean vectors to obtain twice the number of components, halve the weights for the perturbed components and carry over the covariance matrix as it is to the perturbed components.
4) E –step: calculate expection of data for the current model parameters
5) M-step: update model parameters
6) Repeat steps 4 & 5 until convergence.

One example to illustrate the convergence of the above algorithm for GMM parameter estimation is shown in Fig. 3.2. This is for an 8 mixture GMM training using date from a speaker – *4516* - of NIST-2001 development database [84]. From this we can see that we achieve reasonably quick convergence – in only about 4 iterations.

3.3 Statistical Modeling of Speech Using GMMs

GMM has been the primary statistical representation for speech signals for over two decades. In speech recognition, GMMs are predominantly used within the framework of Hidden Markov Models (HMMs) to model the probabilities of state observations [1]. The success of this model is the result of the short-term stationarity of speech signals. Each state in an HMM can be made to represent the distribution during stationary segments of speech for a given phoneme reasonably well using a GMM. The transitions between these stationary segments can be represented by state transition probabilities. In this paradigm, the individual Gaussian modes of the GMMs can be understood to represent the idiosyncratic ways in which a particular sound can be generated by one or more speakers.

In speaker recognition, it is established that a 1-state HMM, or equivalently, a GMM, would suffice [5]. Here, each Gaussian mode in the GMM represents a different broad class of sounds produced by the speaker; for example, one Gaussian distribution can model the vowel sounds, another the fricatives, and so on. Since the same phoneme varies across a population of speakers due to a number of linguistic and physiological phenomena, it is expected that their respective GMMs will be dissimilar. This idea can be used to recognize the speakers correctly.

GMMs are widely used in speech processing systems for several reasons [5][11]. First, it is somewhat straightforward to estimate the parameters of a GMM from training data. The popular iterative Expectation-Maximization approach guarantees convergence to the maximum likelihood estimate of the parameters [71]. In practice, this convergence is achieved fairly quickly. Both Expectation (E) and Maximization (M) steps have closed-form expressions computed every iteration. Moreover, it is sufficient to approximate full covariance matrices for each Gaussian mixture component with a diagonal covariance matrix, thus greatly decreasing the computation requirements.

3.4 Training Speaker GMMs using Adaptation

Building a good speaker model requires large amounts of data during training. This might be impractical in several scenarios. Recently, GMM adaptation techniques have become very popular to alleviate this problem [5][11].

In this method, first one GMM model – the Universal Background Model (UBM) - representing general speech characteristics is trained using data from a large pool of speakers. From the UBM model, the individual speaker models are trained using an adaptation technique - typically techniques similar to Maximum A-Posteriori (MAP) are employed. In this technique, first step is same as the EM-step for GMM training. Sufficient statistics (weights, means, and covariances) are computed and accumulated. Then, a weighted combination of old and new sufficient statistics is used to generate the new parameter estimate.

The scoring technique still relies on the likelihood ratio technique for verification. However, using all the Guassians for scoring can be computationally prohibitive. Fortunately, typically only a relatively few Gaussians are modified significantly from the

UBM to obtain a speaker model. Hence, the process of scoring can be greatly speeded up with only a marginal or no difference in recognition performance by considering only the top few dominant Gaussians for each speaker.

There are two main advantages of the UBM approach; first, it can train reasonably reliable models even with little speaker data. Second, because the individual speaker models are adapted from a common UBM, the differences between speakers are made more prominent and hence there is better discriminability leading to better recognition performance. However, the current work does not consider adaptation techniques for training.

3.5 Normalization Techniques for Handset/Channel Compensation

It is well-known that handset and channel variations affect the characteristic of speaker distributions significantly leading to performance degradation. To deal with this problem, normalization techniques have been proposed that compensate the effects induces by different channels and handsets. These methods can be broadly divided into feature normalization and score normalization techniques, depending on the level at which the compensation is applied [2].

Feature normalization techniques are simpler of the two classes. Two feature compensation techniques are commonly used. Cepstral Mean Subtraction (CMS) is perhaps the simplest and is based on the observation that different handsets introduce biases to the individual MFCC components [1][72].Thus by simply computing the mean of features from each utterance and then subtracting it removing the bias from the features, we can undo some of the channel-induced effects in the speech signal. This can also be extended to the variance components. RASTA (Relative Spectral) is another popular feature normalization technique with a similar idea, but it uses a bandpass filter instead to remove the slow-varying channel effects from feature stream. These two belong to unsupervised methods since they do not explicitly use any knowledge of channel conditions.

Another class of feature normalization is a supervised method called Feature Mapping (FM) [73][4]. In this method, the features from different channels are transformed into a channel-independent feature space. This method assumes knowledge of the type of channel the speech data came from, and this is typically detected first by simple likelihood techniques.

However, it was discovered that speaker recognition systems are very sensitive to channel mismatch between train and test conditions despite the application of feature compensation techniques described above [74] To alleviate this problem, normalization is done at the likelihood score space separately or in conjunction with feature compensation techniques.

A general score normalization technique is of the form:

$$\hat{s} = \frac{s - \mu_I}{\sigma_I} \qquad \text{Eq. 3.7}$$

where the 'hat' denotes corrected score and the subscript denotes imposter-based calculations for mean and variance correction.

A simple technique called HNORM (Handset Normalization) is used to normalize the scores from different handsets [76]. In this technique, biases and scales of the likelihood scores are estimated using development data for each handset type. During evaluation, first, the type of handset is detected using a maximum likelihood classifier technique. Then the likelihood score for the speaker models are appropriately scaled and bias is removed according to the handset detected.

A similar technique is also used for channel normalization, where it is called ZNORM (Zero-Normalization). [74] This normalization score is dependent on the target speaker – for each target speaker the imposter corrections to the mean and variance are computed offline during enrollment phase and can be easily applied during evaluation. Yet another score normalization called TNORM (Test Normalization) computes normalization parameters based on the test utterance by comparing it to the stored impostor models [74][75]. Hence this has to be both computed and applied during evaluation phase. Using such techniques, there has been much progress in our ability to deal with channel and handset variations. However, the current work does not deal with score normalization. We only use cepstral mean subtraction on the MFCC features.

3.6 Limitations of GMMs in Speech Processing

Application of GMMs in speech processing is not without its drawbacks. GMMs can only represent static distributions and hence cannot represent a random process that is evolving over time. In applications where we model MFCC feature streams representing speech data, the dynamic information in their time-evolution is lost. The most commonly used strategy to circumvent this limitation is to append delta (first time-derivative, or velocity) and delta-delta (second time-derivative, or acceleration) coefficients of MFCCs to the absolute or static MFCCs [1][5].

However, this has two main drawbacks. The first drawback involves redundancy – there is obviously statistical dependence between absolute, static, and acceleration coefficients, but building GMMs over the complete concatenated vector does not take this redundancy into account. Hence, we tend to use more parameters than might be necessary. The second more serious drawback, which is the focus of this dissertation, is the implicit assumption of linearity in the MFCC dynamics. The derivatives of the cepstral features are only a linear approximation of the actual dynamics of the static features. However, as we saw in Chapter II, the speech signal contains significant nonlinear information, and using only derivative features to represent speech MFCC dynamics with GMM modeling is tantamount to discarding any nonlinear information present in the signal.

An obvious fix to this problem is to add features that can represent the nonlinear dynamic information. But as was seen in the previous chapter, this approach is fraught with difficulties. The primary goal of this dissertation is to approach the information representation problem at the modeling level, thereby accounting for the nonlinear dynamics of speech in the base model and minimizing the dimensionality of the feature space.

CHAPTER IV

THE MIXTURE AUTOREGRESSIVE MODEL – A NONLINEAR APPROACH

In this chapter, a nonlinear model called the mixture autoregressive model (MixAR) is introduced. First, the basic definition and a few relevant properties of the model are stated. Connections and comparisons of MixAR are made to GMMs as well as to other autoregressive models found in the speech literature. The problem of parameter estimation is discussed in a framework of maximum likelihood estimation using the popular Expectation-Maximization (EM) approach.

4.1 Why Use Nonlinear Models for Speech?

It is evident from discussions in Chapter II that there are significant nonlinearities in the speech signal, that including nonlinear information can improve robustness of speech systems, and that nonlinear dynamic invariants are ineffective at capturing this information for short-term stationary speech signals. It then follows that we should explore capturing nonlinear information at the modeling level. In Chapter III we explained that the popular GMM approach can at best model linear dynamics using static and differential features. This motivates a search for a model that can capture the nonlinear information in speech from its MFCCs.

4.2 MixAR: Basic Definition and Properties

A mixture autoregressive process (MixAR) of order p with m components, $X=\{x[n]\}$, is defined as [77][78]:

$$x[n]=\begin{cases} a_{1,0}+\sum_{i=1}^{p} a_{1,i}x[n-i]+\varepsilon_1[n] & w.p. \quad W_1(x[n-1]) \\ a_{2,0}+\sum_{i=1}^{p} a_{2,i}x[n-i]+\varepsilon_2[n] & w.p. \quad W_2(x[n-1]) \\ \vdots & \\ a_{m,0}+\sum_{i=1}^{p} a_{m,i}x[n-i]+\varepsilon_m[n] & w.p. \quad W_m(x[n-1]) \end{cases}$$

Eq. 4.1

where ε_i is a zero-mean Gaussian random process with a variance of σ_j^2, "w.p." denotes "with probability" and the gating weights, W_i sum to 1 and these weights are defined by the following equation:

$$W_i(x)=\frac{e^{w_i+g_i x}}{\sum_{j=1}^{m} e^{w_j+g_j x}}$$

Eq. 4.2

The linear prediction coefficients, $\{a_i\}$, represent the dynamic model, where $a_{i,0}$ are the component means, while $\{w_i ,g_i\}$ are called gating coefficients. It is apparent that an m-mixture MixAR process is the weighted sum of m Gaussian autoregressive processes, with the time-dependent weights depending on previous data and the gating coefficients.

One convenient way of viewing this model is as a process in which each data sample at any one point in time is generated from one of the component AR mixture processes chosen randomly according to its weight W_i. An overview of a 2-component MixAR model is illustrated in the following figure.

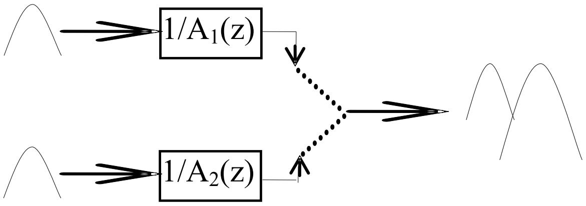


Figure 4.1 An overview of MixAR modeling approach

4.3 Modeling Nonlinearities Using MixAR

One property of MixAR that is of particular relevance here is the ability of MixAR to model nonlinearity in time series [77][78] Though the individual component AR processes are linear, the probabilistic mixing of these AR processes constitutes a nonlinear model. Even when the mixture weights are fixed, the model reduces to MAR, which is still nonlinear. It is to be noted here that even though GMM also employs probabilistic mixing of components, because of the static nature of the components - i.e., each component is a single value of a random variable and not a random process – it cannot model nonlinear dynamics in the data. The addition of a gating system layer for weight generation increases the flexibility of the model even further, allowing us to model distributions as a function of past data even better.

Even simple MAR dynamics can lead to chaotic patterns in data. For example, we can generate fractals out of seemingly trivial MAR models. The famous Sierpinski triangle fractal can be generated out of the following MAR model using only three components and even using only fixed weights:

$$X[n] = \begin{Bmatrix} X[n-1] + A_1 & w.p. W_1 \\ X[n-1] + A_2 & w.p. W_2 \\ X[n-1] + A_2 & w.p. W_3 \end{Bmatrix} \qquad \text{Eq. 4.3}$$

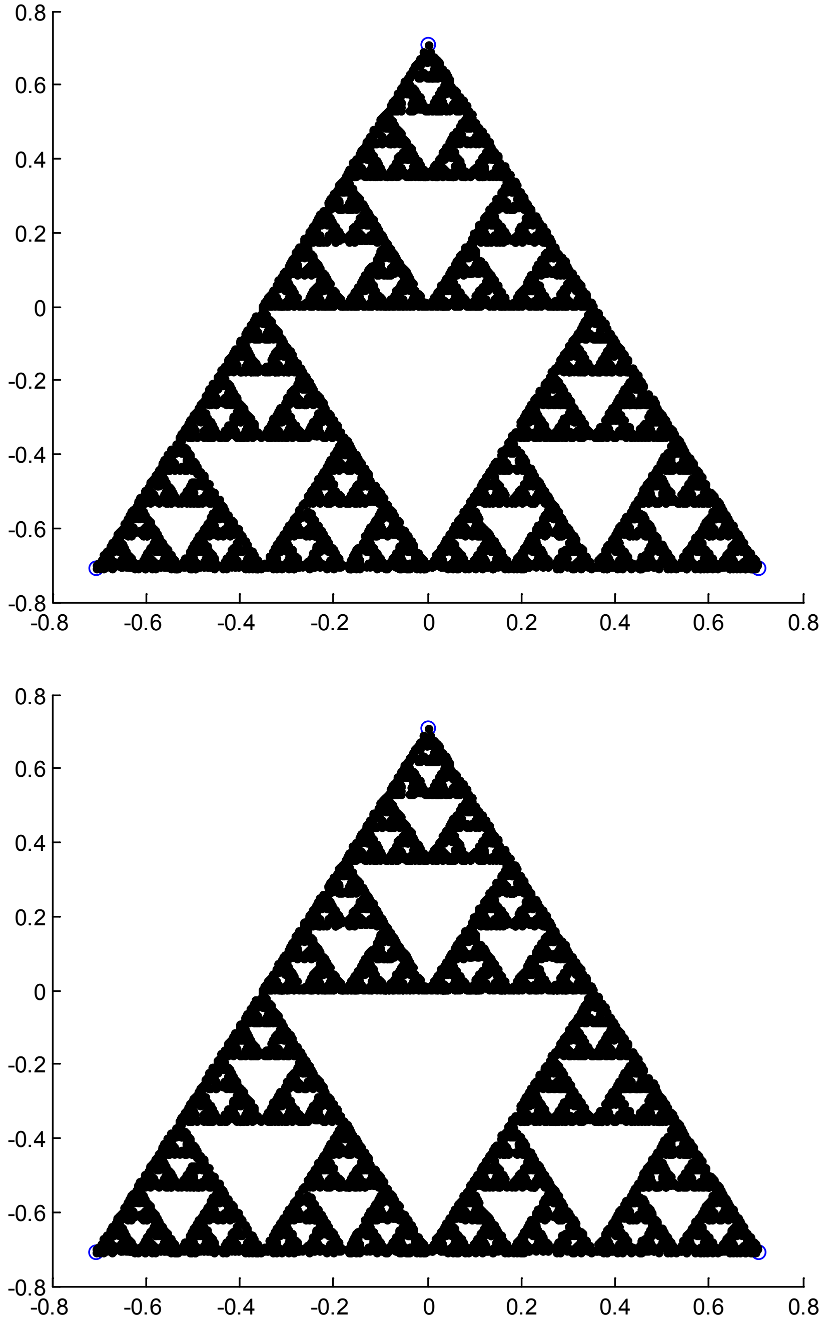


Figure 4.2 Sierspinski Triangle fractal geometry generated using MixAR model

where X is the trajectory of the data points on the 2-D plane, and A_i s are three fixed points on the plane. An example Sierpinski triangle generated from such a model is shown in the figure above.

One indication of a chaotic signal is the bandwidth of the power-spectrum. Most natural signals that are not chaotic exhibit a low-bandwidth in their power spectrum, while chaotic signals have a large bandwidth power spectrum resembling that of a stochastic system. For example, 50,000 samples were generated according to the following MixAR model:

$$X[n] = \begin{Bmatrix} 0.5X[n-1]+0.0 & w.p.\,0.33 \\ 0.2X[n-1]-1.0 & w.p.\,0.33 \\ 0.3X[n-1]+1.0 & w.p.\,0.33 \end{Bmatrix} \qquad \text{Eq. 4.4}$$

A snapshot of the signal and the associated power spectrum are depicted in the figure below.

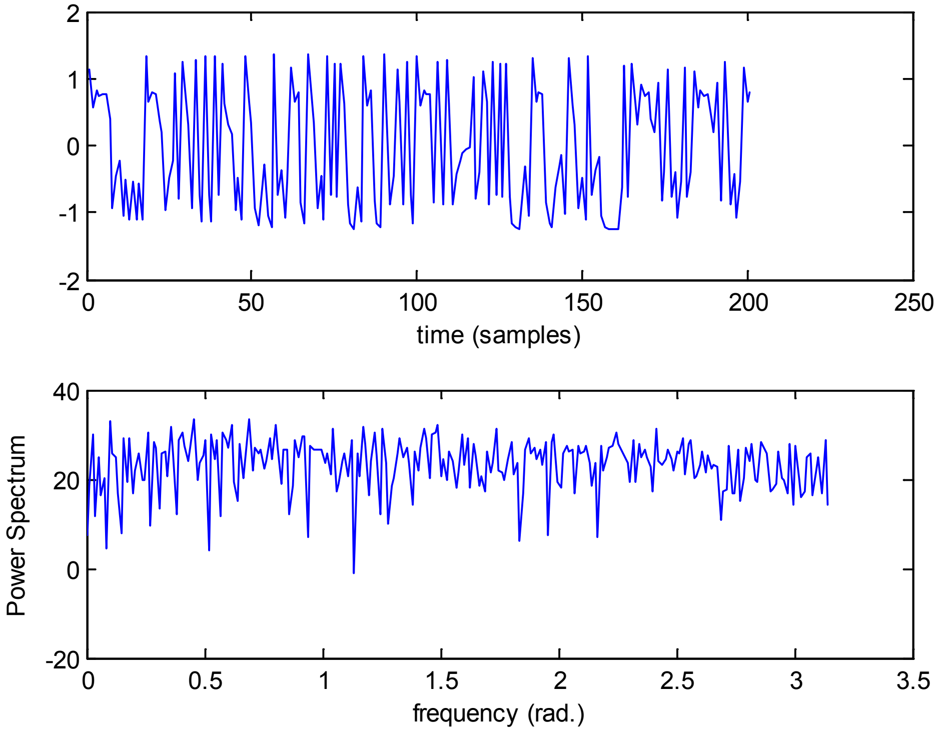


Figure 4.3 Example of a MixAR generated signal and its power spectrum; the broad-band nature of power spectrum indicates that the signal is chaotic

From the power spectrum it can be seen that MixAR signal are broadband signals, and hence possibly chaotic.

4.4 Comparison of MixAR to Other Models

It is easy to find parallels between the MixAR and GMM models. In particular, MixAR can be viewed as a generalization of GMM that models each component as a sum of the output of an autoregressive filter with a specified mean, and with mixture weights determined by a gating system similar to a mixture of experts. It should be noted that with the component orders and g_i set to zero, MixAR, reduces to the familiar GMM. This similarity between the two makes it straightforward to replace GMM with MixAR for speaker recognition.

In a GMM, the distribution remains invariant to the past samples due to the static nature of the model. For MixAR, the conditional distribution given past data varies with time. This model is capable of modeling both the conditional means and variances. Thus, MixAR can model time series that evolve nonlinearly. This property becomes important in speech processing in the light of recent work on nonlinear processing of speech, the subject matter of Chapter II.

Some other properties of MixAR, including a mathematically rigorous proof of the ability of MixARs to arbitrarily closely model stochastic processes are derived in [77]. Note that in the original formulation, both the gate and prediction orders were constrained to be equal. In this work, we restrict our use of MixAR order to one to avoid difficulties during parameter estimation.

Previous work on mixture autoregressive modeling for speech has been in the context of hidden Markov models for speech recognition [83] . One of the earliest applications of autoregressive HMMs (AR-HMMs) considered an autoregressive filter to model state observations in a 5-state HMM for speaker verification [80]. A more recent investigation of AR-HMMs [81] used a switching autoregressive process to capture signal correlations during state transitions. Results on speech recognition showed that at best their model was only comparable to an MFCC-based HMM using a GMM observation model. Another model considered speech features as a GMM white noise process filtered through an autoregressive signal for speaker identification [82].

A more sophisticated model introduced in considers a mixture of autoregressive filters [78] (MAR) for the observation model. Our earlier work [79] considered this model for phone classification. MixAR is a generalization of MAR, where the mixture weights are allowed to be time-varying and data-dependent. Applications of models somewhat related to MixAR were under the context of mixture of experts for time-series prediction [77]. In this work, we apply the MixAR model to feature vectors in a speaker recognition task.

4.5 MixAR Parameter Estimation using the EM algorithm

Similar to the well-known training procedure for GMM, maximum likelihood estimates for MixAR prediction and variance parameters can be calculated using the Expectation-Maximization (EM) algorithm [71][69][77][78]. Given the order, *p*, the parameter set for each of the *m* components of a MAR model consists of *p*+1 predictor coefficients (including the mean), the error variance, and mixing weight:

$$\theta_l = \{a_{l,0}, a_{l,1}, \ldots, a_{l,p}, \sigma_l, w_l, g_l\}\ l = 1, \ldots, m . \qquad \text{Eq. 4.5}$$

To estimate these parameters, we first need an initial guess for these parameters and then we iterate with EM to successively refine the estimates. An initialization strategy that we found to work reasonably well was to first train a GMM with the same number of mixtures and then set each component of the MixAR to have the same mean, variance, and weight as the GMM model. We initialize the predictor coefficients and the data-dependency gating coefficients, $\{A_i\}$ of MixAR to zero.

These initial parameters can be then refined recursively using an E-step [77]:

$$\gamma_l[n] = \frac{W_l p_l(x[n] \mid \theta)}{\sum_{k=1}^{m} W_k \, p_k(x[n] \mid \theta)} \qquad \text{Eq. 4.6}$$

where

$$p_l(x[n] \mid \theta) \propto \frac{1}{\sigma_l} e^{\frac{-1}{2\sigma_l}(x[n] - a_{l,0} - \sum_{i=1}^{m} a_{l,i}\, x[n-i])^2} . \qquad \text{Eq. 4.7}$$

is the probability a sample was generated from component l at time instant n. The corresponding M-step is given by:

$$\hat{A}_l = R_l^{-1} r_l \qquad \text{Eq. 4.8}$$

where

$$R_l = \sum_{n=p+1}^{N} \gamma_l[n] X_{n-1} X_{n-1}^T \qquad \text{Eq. 4.9}$$

$$r_l = \sum_{n=p+1}^{N} \gamma_l[n] X_{n-1} x[n] \qquad \text{Eq.4.10}$$

$$X_{n-1} = \begin{bmatrix} 1 \\ x[n-1] \\ x[n-2] \\ \vdots \\ x[n-p] \end{bmatrix}. \qquad \text{Eq. 4.11}$$

Refer to comments on estimation of predictor coefficients and variances for MixAR and MAR in [77][78]for further details.

However, a complication arises with respect to the estimation of gating coefficients for MixAR. There is no closed-form solution for these, and hence a Newton gradient-ascent approach must be used:

$$\hat{w}_l = w_l + \beta \frac{\Delta Q}{\Delta w_l} \qquad \text{Eq. 4.12}$$

$$\hat{g}_l = g_l + \beta \frac{\Delta Q}{\Delta g_l} \qquad \text{Eq. 4.13}$$

where Q denotes the log-likelihood of the MixAR model for the training data. β and Δ are design parameters to be chosen empirically. The expression for computing Q is:

$$
\begin{aligned}
Q(\theta) &= \sum_{n=1}^{N}\sum_{l=1}^{m}\gamma_l[n]\log(W_l[n]) \\
&+ \sum_{n=1}^{N}\sum_{l=1}^{m}\gamma_l[n]\log(p_l(x[n]\,|\,\theta)
\end{aligned}
\qquad \text{Eq. 4.14}
$$

Due to this complication in the updates for the gate coefficients, the training procedure outlined above is not in the realm of strict EM algorithm but falls under a class of algorithms called as generalized EM algorithms (GEM) [84]. For both EM and GEM algorithms, the E- step is similar. However, while an EM algorithm actually maximizes the expectation during each M-step, a GEM algorithm only guarantees that parameters that increase the model likelihood for the data is increased but does not guarantee that his is maximized at each M-step. This could mean that a GEM algorithm could take more number of iterations for training than an EM algorithm for the same or a comparable problem.

Drawing parallels with the choice of adaptation factor μ in adaptive filter theory, we can envisage that quick and smooth convergence of the GEM algorithm can be achieved by starting with a relatively high value for β and then reducing this value with successive iterations. In our experiments, we found that fixing Δ = 0.01 and running 10 iterations each with β = 0.9, β = 0.5, and β = 0.2 in succession provided a reasonably smooth and quick convergence. One example to illustrate the convergence of the EM algorithm for MixAR with algorithmic values set as mentioned above is shown in Fig. 4.4 below.

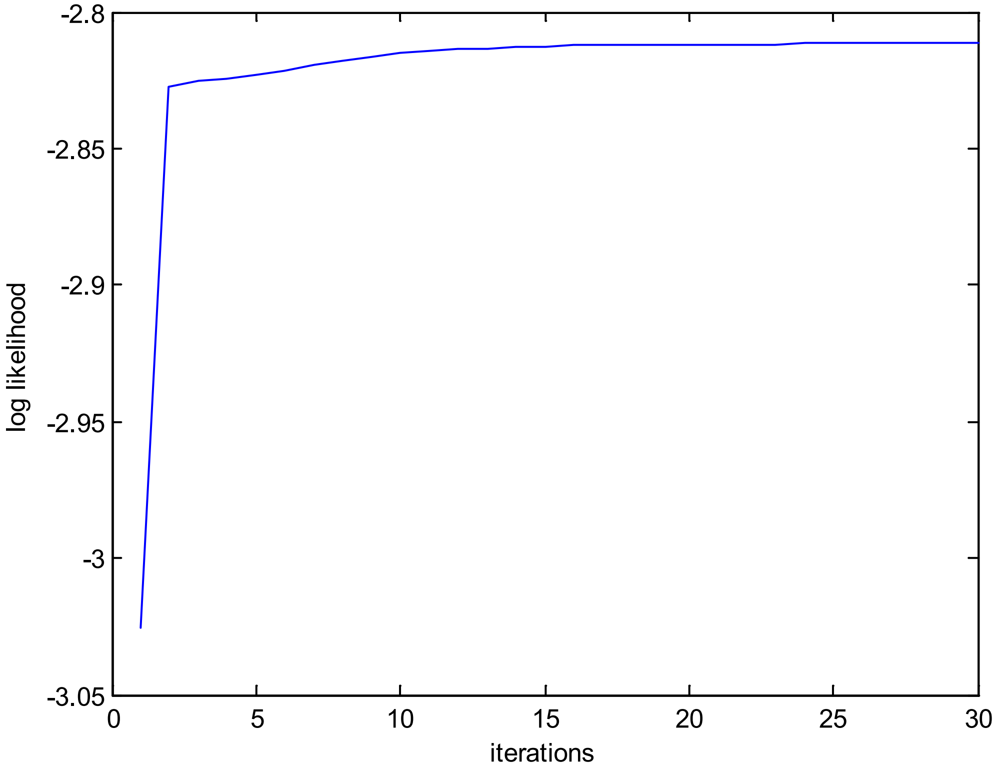


Figure 4.4 EM Convergence as a function of iterations on a synthetically synthesized signal

This is for an 8 mixture MixAR training on a synthetic signal generated from a MixAR model of speaker *4516* of NIST-2001 development database [86]. From this we can see that we achieve reasonably quick convergence with these algorithmic parameter values. However, such convergence is not guaranteed in general and this poses a problem to the application of this model for real-life signals.

Fortunately, we can do better than guessing an appropriate value for beta. We can use the secant method for root-finding and maximization [85][86]. In general, to find the maxima using Newton's method, the iteration is:

$$\hat{x} = x + \frac{f^{'}(x)}{f^{''}(x)} \qquad \text{Eq. 4.15}$$

In the secant method, the double derivative in the denominator is estimated numerically using the secant at the point. Thus, we estimate the scaling factor β as the inverse of double derivative of the log-likelihood w.r.t. the gate parameters:

$$\beta = 1 / \frac{\Delta^2 Q}{\Delta^2 w_l} \qquad \text{Eq. 4.16}$$

During implementation, this scheme amounts to finding for each gate coefficient w_l, the value of Q at three different points, Q(w_l), Q(w_l +Δ), Q(w_l - Δ), and then using the following update equation:

$$\hat{w}_l = w_l + \frac{Q(w_l + \Delta) - Q(w_l - \Delta)}{Q(w_l + \Delta) + Q(w_l - \Delta) - 2Q(w_l)} \qquad \text{Eq. 4.17}$$

Similarly, the update equation for gate coefficients g_l is:

$$\hat{g}_l = g_l + \frac{Q(g_l + \Delta) - Q(g_l - \Delta)}{Q(g_l + \Delta) + Q(g_l - \Delta) - 2Q(g_l)} \qquad \text{Eq. 4.18}$$

Using this method, we obtain convergence curve shown in the following figure for the same data from speaker *4516* of NIST-2001 database [86] used for the previous method. We find that this method is more reliable and quick - three GEM iterations were sufficient. We use this method in future experiments.

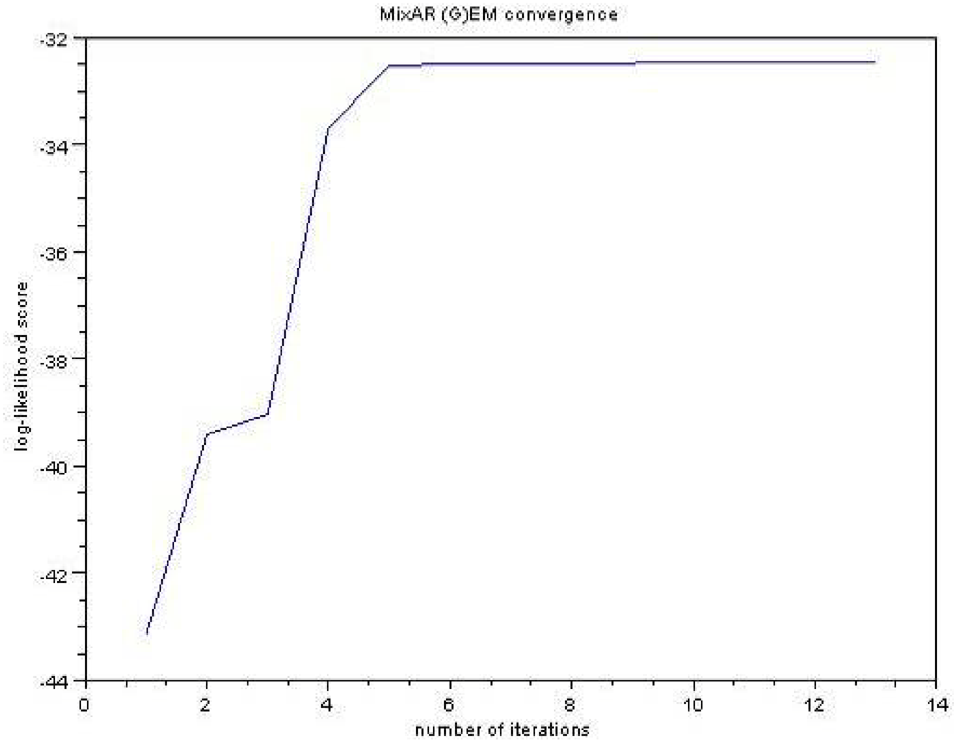


Figure 4.5 Performance of (Generalized) EM using secant method, as a function of iterations for a 8-mixture MixAR model of speaker data from “4516” of NIST-2001 database.

4.6 Efficiency Considerations for Faster MixAR Training

It is clear from the discussion on properties of MixAR that it can be thought of as a generalization of GMM, and hence, would require more computations than a corresponding GMM. Furthermore, this need for excess computations is accentuated during training by the steepest-descent approach that could necessitate more number of iterations than for a comparable GMM. Thus, it would be advantageous if critical computations can be done efficiently or if repeating computations can be avoided.

One particular place where speed-up can be achieved is by considering the statistics that are needed to be computed for M-update in equations 24 and 25. Using the following notation:

$$A_l = \begin{bmatrix} -a_{l,0} \\ -a_{l,1} \\ \vdots \\ -a_{l,p} \end{bmatrix}.$$

Eq. 4.19

and using the definition of X_{n-1} from eqn. 15, we can rewrite equation 25 as:

$$\hat{\sigma}_l^2 = \frac{\sum\limits_{n=p+1}^{N} \gamma_l[n]\left(x[n] - \hat{a}_{l,0} - \sum\limits_{i=1}^{m} \hat{a}_{l,i}\, x[n-i]\right)^2}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

$$= \frac{\sum\limits_{n=p+1}^{N} \gamma_l[n](x[n] + A_l^T X_{n-1})^2}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

Eq. 4.20

By expanding the squared term, this can be further reduced to the following:

$$\hat{\sigma}_l^2 = \frac{\sum\limits_{n=p+1}^{N} \gamma_l[n]\left(x^2[n] + A_l^T X_{n-1} X_{n-1}^T A_l + 2A_l^T X_{n-1}\right)}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

Eq. 4.21

Distributing the sum in the numerator over the individual terms, we get:

$$\hat{\sigma}_l^2 = \frac{\left\{\begin{array}{c} \sum\limits_{n=p+1}^{N} \left(\gamma_l[n]x^2[n]\right) + \sum\limits_{n=p+1}^{N} \left(\gamma_l[n]A_l^T X_{n-1} X_{n-1}^T A_l\right) \\ + \sum\limits_{n=p+1}^{N} \left(2\gamma_l[n]A_l^T x[n] X_{n-1}\right) \end{array}\right\}}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

Eq. 4.22

Since, A_l is independent of n, we can separate them out of each summation term in which it occurs. Thus, we obtain:

$$\hat{\sigma}_l^2 = \frac{\left\{\begin{array}{c} \sum\limits_{n=p+1}^{N} \left(\gamma_l[n]x^2[n]\right) + A_l^T \left(\sum\limits_{n=p+1}^{N} \gamma_l[n] X_{n-1} X_{n-1}^T \right) A_l \\ + 2A_l^T \sum\limits_{n=p+1}^{N} \left(\gamma_l[n] X_{n-1} x[n]\right) \end{array}\right\}}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

Eq. 4.23

Finally, using notations in eqn. 13 and 14, we can rewrite the above eqn. in the following form:

$$\hat{\sigma}_l^2 = \frac{\sum\limits_{n=p+1}^{N} \left(\gamma_l[n]x^2[n]\right) + A_l^T R_l A_l + 2A_l^T r_l}{\sum\limits_{n=p+1}^{N} \gamma_l[n]}$$

Eq. 4.24

We already accumulate R_l and r_l statistics in equation 26 and 27, respectively. Hence, we can reduce the computations for M-update at each iteration by reusing these statistics for computing $\hat{\sigma}_l^2$.

4.7 Example MixAR Parameter Estimation to Verify Implementation

Since the implementation of MixAR parameter estimation is somewhat more complicated due to the GEM approach compared to the EM approach for GMM, a test case was studied as a sanity check to verify the correctness of the implementation.

A MixAR model of order two was trained using speech data for speaker *4516* of NIST 2001 development database. This model, Θ_1, had the following values for its parameters:

$$\Theta_1 : \begin{cases} p = 1, m = 2, \\ a_{1,0} = 0.821974, a_{1,1} = 0.551031, \\ a_{2,0} = 0.94095, a_{2,0} = 0.269543, \\ g_{1,0} = 0.861345, g_{1,1} = \text{-}0.692634, \\ g_{2,0} = 0.891252\ \ g_{2,0} = \text{-}0.696943 \\ \sigma_1 = 3.92727, \sigma_2 = 3.70133 \end{cases} . \qquad \text{Eq. 4.25}$$

A synthetic signal of 20,000 samples was generated according to this model. Then a model was trained on this data using the GEM algorithm for MixAR parameter estimation described above, and the parameter values compared to the original model. The training procedure after 30 GEM iterations on the data yielded the following model Θ_2:

$$\Theta_2 : \begin{cases} p = 1, m = 2, \\ a_{1,0} = 0.814398, a_{1,1} = 0.596219, \\ a_{2,0} = 0.925743, a_{2,0} = 0.190086, \\ g_{1,0} = 0.798216, g_{1,1} = -0.688911, \\ g_{2,0} = -0.822302, g_{2,0} = 0.696971 \\ \sigma_1 = 3.94217, \sigma_2 = 3.69397 \end{cases} \qquad \text{Eq. 4.26}$$

Inspection of the model parameters Θ_1 and Θ_2 clearly indicates that we can estimate the MixAR model reasonably accurately with our implementation. The difference between the two parameter values can be quantified using a normalized squared error between the true and estimated parameter value as $||\Theta_1 - \Theta_2||_2/||\Theta_1||_2$=0.05%, where $||.||_2$ denotes the 2-norm or the Euclidean norm and computed by considering the parameter values as a vector. This error is small enough that we can consider our parameter estimation algorithm and the implementation reliable enough for our purposes.

CHAPTER V

PRELIMINARY EXPERIMENTS USING SYNTHETIC DATA

This chapter describes the experiments that were run to study the performance of MixAR model in relation to that of GMM. To better understand the efficacy of the MixAR model, first its performances on two pattern classification tasks are evaluated. The first task represents generic data with known nonlinearities – both a scalar case and a vector case are tested. The second task is a simple classification task with data for the two classes synthesized from models trained on true speaker data. Next a speaker verification experiment with synthetic data that simulate combinations of noise and nonlinearity was conducted.

Throughout, the performance of both MixAR and GMM models are studied in relation to their model parameter complexities. For all speaker verification experiments, results are reported either in the form of a detection-error-tradeoff (DET) curve, or equal error rate (EER), both of which are standard for this purpose [6].

Finally, all training and evaluation were conducted using ISIP's Production System, a public-domain speech recognition system [88].

5.1 Two-Way Classification with Scalar Synthetic Data

The MAR-HMM approach, like GMM-HMMs, can perform classification using a maximum likelihood approach. The log likelihood of data given a set of MAR-HMM model parameters is used to score each model and the class with the maximum score is chosen. A two-class classification problem was designed where data are randomly generated randomly according to the following MAR parameters. Parameters used for model 1 are:

$$\Theta_1 : \begin{cases} p = 1, m = 2, w_1 = 0.4, w_2 = 0.6, \\ a_{1,0} = -1, a_{1,1} = 0.2, a_{2,0} = 1, a_{2,0} = 0.2,. \\ \sigma_1 = 0.25, \sigma_2 = 0.2 \end{cases}$$

Eq. 5.1

To generate data from class, the model parameters were:

$$\Theta_2 : \begin{cases} p = 0, m = 4, \\ w_1 = 0.2, w_2 = 0.2, w_3 = 0.3, w_4 = 0.3 \\ a_{1,0} = -1.03, a_{2,0} = -0.86, \\ a_{3,0} = 1.13, a_{4,0} = 0.98 \\ \sigma_1 = 0.3263, \sigma_2 = 0.2906, \\ \sigma_3 = 0.2598, \sigma_4 = 0.2894 \end{cases} .$$

Eq. 5.2

For this example we chose the parameters for class 2 such that the marginal distribution is about the same as that of the first class, but it lacked the dependence on past samples unlike class 1. Hence the data for class 2 follows only a GMM distribution. This was done to demonstrate a case where GMM would be unable to achieve good classification due to its ability to capture the dynamics in the model.

The results of these experiments, along with the number of parameters for each model, are shown in Table 5.1. In addition to listing accuracy, the numbers of parameters for each model are shown. Since in this case we knew that the distribution can have a maximum of 4 modes, we use only 2- and 4-mixture models. It can be observed that MAR, with just 2 components and 8 parameters can achieve 100% classification accuracy using only static features. The GMM approach using only static features is unable to do much better than a random guess strategy since the two classes have similar static marginal distribution. This demonstrates ability of MAR model to learn dynamic information.

Table 5.1 Classification (% error) results for scalar synthetic data (the numbers of parameters are shown in parentheses).

# mix.	GMM Static	MixAR Static	GMM Static+Δ	MixAR Static+Δ
2	**47.5 (6)**	**100.0 (8)**	**82.5 (14)**	**100.0 (20)**
4	**52.5 (12)**	**100.0 (16)**	**85.0 (28)**	**100.0 (40)**

With the inclusion of delta coefficients, the GMM performance increases significantly, but even in this case it achieves only 85% accuracy with 28 parameters. Though delta features capture some amount of dynamic information in the features, it is still only a linear approximation, and we cannot capture their nonlinear evolution with just GMMs. From the above, it is clear that at least some dynamic information is better modeled using MAR – and hence MixAR.

5.2 Two-Way Classification with Vector Synthetic Data

A simple 2-way classification experiment was designed to study the performance of MixAR and GMM. Two-dimensional data for the first class was generated using a linear dynamic system:

$$\begin{gathered} x(n-1) = A \times x(n-1) + B \times E(n) \\ A = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}; B = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \end{gathered} \tag{Eq. 5.3}$$

Data for the second class was generated using the simple nonlinear equation:

$$x(n-1) = A \times \text{sign}(x(n-1)) + B \times E(n)$$
$$A = \begin{bmatrix} 0.5\,0.0 \\ 0.0\,0.5 \end{bmatrix}; B = \begin{bmatrix} 1.0\,0.0 \\ 0.0\,1.0 \end{bmatrix} \qquad \text{Eq. 5.4}$$

In both cases, ***E*** denotes an uncorrelated 2-D Gaussian (normal) random variable with a zero mean and unit variance.

For each class, the training data consisted of a sequence of 10,000 vectors, and evaluation data consisted of 100 segments of 200 feature vectors each (the log-likelihood of the entire segment was used to assign a segment to a class). The classification error results are stated in Table 5.2. Clearly, when using only static features, MixAR does much better than GMM if nonlinearities are present. The use of dynamic features enhances GMM performance considerably but still falls far short of MixAR's performance.

5.3 Two-way Classification with Speech-like Data

In order to evaluate how well MixAR does as compared to GMM for speech-like signals, two speakers from the 2001 NIST SRE Corpus [86] were selected. A 3-state HMM with 4 Gaussian mixtures per state and a MixAR model with 4 mixtures were trained over 12 static MFCC coefficients for each speaker. For each class (e.g, a speaker), two speech-like signals of 40,000 vectors were generated – a linear speech-like signal ($\boldsymbol{X_1}$) was synthesized from the HMM model, and a nonlinear speech-like signal ($\boldsymbol{X_2}$) was generated from the MixAR model.

To simulate a range of signals with varying degrees of nonlinearity, the two signals were mixed with a mixing coefficient alpha:

$$X_{\alpha} = (1-\alpha)X_1 + \alpha X_2 \qquad \text{Eq. 5.5}$$

The first 20,000 vectors from each $\boldsymbol{X_\alpha}$ were used as a training set while the remaining vectors were split into 200 segments of 100 vectors each for evaluation. The results are shown in Table5.3.

Table 5.2 Classification (% error) results for vector synthetic data (the numbers of parameters are shown in parentheses).

# mix.	**GMM Static**	**MixAR Static**	**GMM Static+Δ**	**MixAR Static+Δ**
2	**36.0 (12)**	**6.5 (20)**	**10.0 (24)**	**5.5 (40)**
4	**35.5 (24)**	**6.0(40)**	**11.5 (48)**	**4.5 (80)**

From the table we can see that when the amount of nonlinearity is insignificant, GMM performs as well as MixAR. However, as the amount of nonlinearity in the signal increases, MixAR performs significantly better with just static features as compared to GMM with static+Δ features. This clearly demonstrates the superiority of MixAR when dynamics in the data are nonlinear.

Table 5.3 Classification Error Rate (%) with 12 speech MFCC-like synthetic features for GMM and MixAR (Number of parameters in each case is in parenthesis)

α	**GMM-8mix. Static+Δ**	**MixAR 4-mix. Static**
0.0*	**1.5 (288)**	**1.5 (240)**
0.25	**3.25 (576)**	**3.5 (240)**
0.50	**10.25 (576)**	**6.25 (240)**
0.75	**24.75 (576)**	**9.75 (240)**
1.0	**26.75 (576)**	**13.75 (240)**

*: *For this case, GMM performed better with only static features, and this value is stated here.*

5.4 Phone Classification Experiments Using a Sustained Phone Database

While the primary objective of this dissertation work is the application of MixAR modeling approach to speaker verification, we also ran preliminary speech classification experiments with MAR model using sustained phone database.

To test the efficacy of MAR in speech modeling, we made 16 kHz recordings of three distinct phones – “aa” (vowel), “m” (nasal), and “sh” (sibilant). For each phone and for each speaker, 35 recordings were made to serve as training database, while another 15 were reserved for testing. Silence was removed so that we could focus on the ability of the approach to model speech.

From these sound files, the static feature database comprising of 13 MFCC coefficients (including energy) was extracted for each frame 10 ms in duration using a window duration of 25 ms. In addition, we also created a database containing 39-dimensional features by concatenating Δ and ΔΔ MFCC features.

First we evaluated the performances of 2-, 4-, 8-, and 16-mixture GMM and MAR with the 13-dimensional static MFCC features. The results are shown in Table 4. From this, it can be seen that for equal number of parameters MAR outperforms GMM significantly. For instance, MAR achieves a phone classification accuracy of 94.4% with only 320 parameters while a GMM system using 432 parameters could achieve only 93.3%. This clearly shows that MAR can exploit the dynamical information to better model the evolution of MFCCs for phones that GMM is unable to model.

To determine whether MAR is more effective at exploiting dynamics than what GMM can achieve using dynamic features, we also perform another experiment with 39-dimensional features containing both static as well as velocity and acceleration coefficients. The results are tabulated in Table 5. In this case, the results are not conclusive. While MAR shows an accuracy rate of 97.8% with 472 parameters, GMM-HMM attains only 96.7% accuracy with 632 parameters.

Table 5.4 Sustained phone classification (% accuracy) results with MAR and GMM using 13 MFCC features (Number of parameters in each case is in parenthesis)

#mixtures	GMM	MixAR
2	77.8 (54)	83.3 (80)
4	86.7 (108)	90.0 (160)
8	91.1 (216)	94.4 (320)
16	93.3 (432)	95.6 (640)

Table 5.5 Sustained phone classification (% accuracy) results with MAR and GMM using static+Δ+ΔΔ MFCC features. (Number of parameters in each case is in parenthesis)

#mixtures	GMM	MixAR
2	92.2 (158)	94.4 (236)
4	94.4 (316)	97.8 (472)
8	96.7 (632)	97.8 (944)
16	100 (1264)	98.9 (1888)

However, the performance of MAR saturates with an increase in the number of parameters. For example, MAR at 1888 parameters achieves only 98.9% accuracy while GMM achieves 100% with 1264 parameters. We suspect that this could be due to the fact that our parameter estimation and likelihood computation procedures assume that the features are independent. It is well-known that the static MFCC features are uncorrelated (at least, theoretically), but obviously the delta features are correlated with the static ones. While this should also cause problems for GMM, the problem is more acutely so for MAR because in this case, unlike GMMs, we employ the past history explicitly.

These results were presented in [79]. While we have used only MAR in these experiments due to the ease of training procedure compared with that form MixAR, we could also extend this work with MixAR. These results indicate the feasibility of replacing GMMs with MixAR in a HMM framework for continuous speech recognition. We do not discuss the problem of speech recognition further in this work.

5.5 Speaker Verification Experiments Using Synthetic Data

Now that it has been demonstrated that MixAR performs better than GMMs as a pattern classifier for signals that have significant nonlinearities in them, the next step is to find whether this holds true even for speaker verification. It is instructive to study the performance of MixAR and GMM when noise is present in addition to nonlinearity. To have control both on the presence of nonlinearity and noise, synthetic data is again used.

Since our goal is to study speaker verification, we used the development database in the 1-speaker detection task of the 2001 NIST SRE Corpus [86]. This database is a standard for demonstrating speaker verification performance. The development database is small enough to make it manageable and yet large enough to provide a reliable estimate of the performance. All 60 speakers in the training set were used. Each training utterance was about 2 minutes long. Static (13 MFCCs), delta (26 MFCCs) and delta-delta (39 MFCCs) features were extracted.

Two kinds of clean data were synthesized. For the first type, a 10-state HMM with 4-Gaussians per state was trained for each utterance for each MFCC. For the second type, a 32-mixture MixAR model of prediction order 1 was trained for each utterance and for each MFCC. For each of the models trained, new training data of about 30,000 frames per speaker and evaluation data of 20 utterances with about 200 frames for each utterance per speaker were generated according to that model.

Similarly, two kinds of noisy data were generated. For this purpose, first the clean training utterances from the development data were corrupted with car noise from [91] to have an SNR of 5 dB using the FANT software [89]. This was also the methodology followed when the TIDIGIT database was corrupted to generate the AURORA database [90]. The remainder of the steps to yield the two types of noisy data is exactly the same as those for the clean case.

It is to be noted here that the motivation for generation of these two types of data and under noise conditions is to simulate 4 different test conditions: clean and linear, clean and nonlinear, noisy and linear, and, noisy and nonlinear.

Using the synthesized training data, both GMMs and prediction order-1 MixARs are trained for each speaker under each condition. Then the corresponding synthesized evaluation data are used for evaluating speaker verification performance.

For the clean case, there was little difference in performance between GMM and MixAR. For evaluation data containing 5 dB noise, again there was not much variation in performance between GMM and MixAR for HMM-generated data. However, for the data generated from the nonlinear MixAR model and with the addition of noise, MixAR model showed a significant improvement in performance using far fewer parameters. This is evident from the DET plot in Figure 5.1.

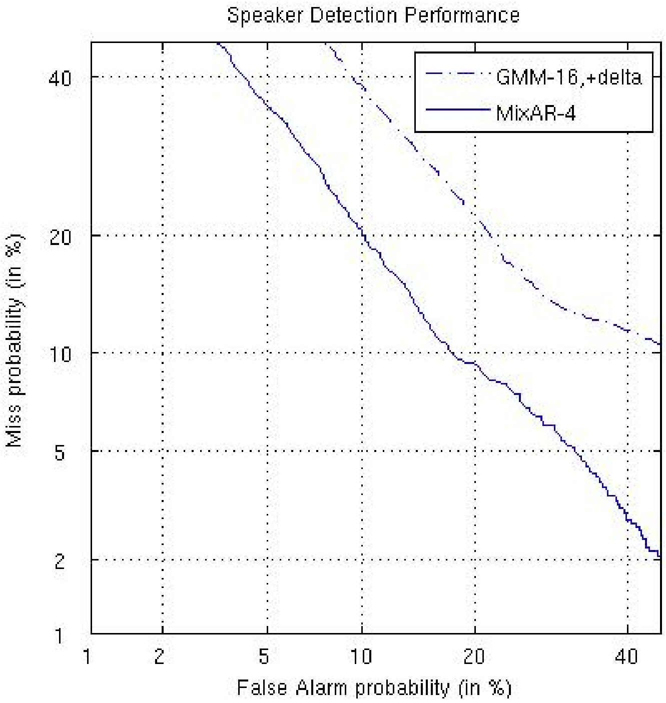


Figure 5.1 Speaker Verification DET curves for MixAR-generated nonlinear data with 5B car noise

These results provide support to the hypothesis that when there are significant nonlinearities in the signal, using this information makes the nonlinear model much more robust to the presence of noise.

CHAPTER VI

SPEAKER VERIFICATION EXPERIMENTS USING SPEECH DATA

Results from preliminary experiments have thus far supported the view that MixAR can achieve better speaker verification performance with fewer parameters than what GMM can achieve. The superior performance is perhaps because MixAR uses nonlinear dynamic information in speech that GMM cannot and the fewer parameters used points to better efficiency and less redundancy in MixAR compared to the GMM representation.

In this chapter we strive to demonstrate that these trends also hold for real speech data. First, using NIST 2001 database, speaker verification performance is studied as a function of the feature set (i.e, static only vs. static+deltas) and also as a function of number of mixtures. Then experimental details and performance on another popular database – TIMIT [87]– are also discussed.

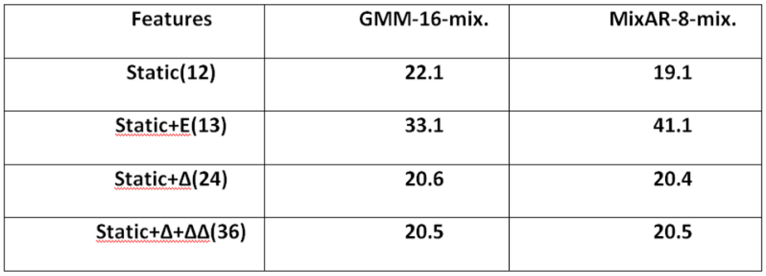
For a successful demonstration of MixAR over GMM for speaker verification, it is necessary to study the performance in noise more extensively. Here, noise of three types and at three different SNRs is added to TIMIT core test utterances and the performances of GMM and MixAR compared. In addition, it is also important to study the variation in performance with varying training and evaluation data lengths. This is done using NIST 2001 database.

6.1 Speaker Verification Experiments with NIST 2001 Database

We applied the MixAR model to the 1-speaker detection task in the 2001 NIST SRE Corpus [86] . Only the development database was used. All 60 speakers were used for training and all 78 utterances were used for evaluation. Each training utterance was about 2 minutes long, while the test utterances were of varying length not exceeding 60 seconds. Static (13 MFCCs), velocity (13 Δ-MFCCs), and acceleration (13 ΔΔ-MFCCs) features were extracted.

First performance is evaluated with and without delta features and energy for a fixed number of mixtures. The results are tabulated in Table 6.1. For GMM, substantial improvement is obtained using the delta features and marginal improvements were obtained using delta-delta features. For MixAR, the use of any delta features provides no measurable improvements. This clearly indicates that MixAR can extract all necessary information from only the static features.

Table 6.1 Speaker recognition EER with NIST-2001 for MixAR and GMM for different feature combinations.



Features	GMM-16-mix.	MixAR-8-mix.
Static(12)	22.1	19.1
Static+E(13)	33.1	41.1
Static+Δ(24)	20.6	20.4
Static+Δ+ΔΔ(36)	20.5	20.5

MixAR and GMM performance was then evaluated as a function of the number mixtures. The EER results are shown in Table 6.2, and the corresponding detection error trade-off (DET) curves are shown in the following figure.

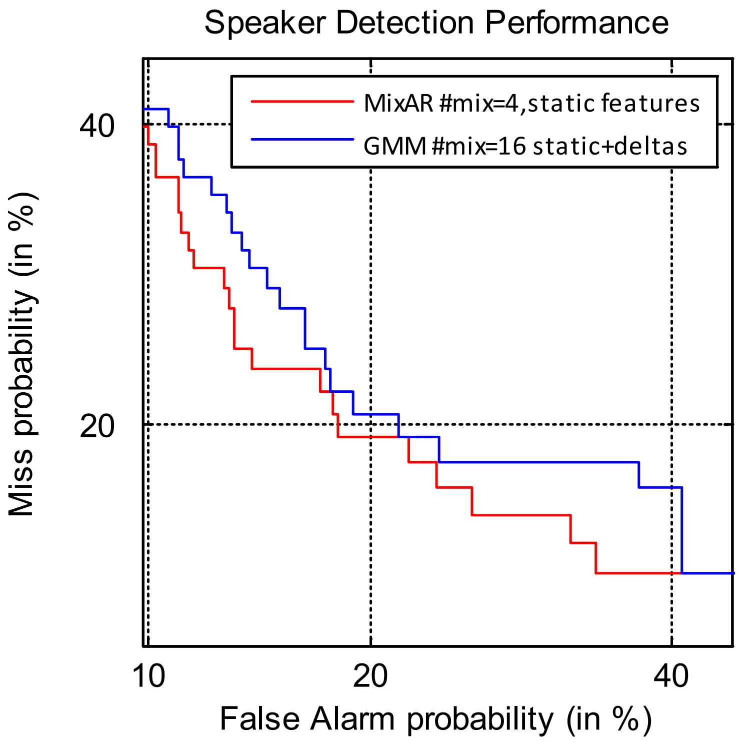


Figure 6.1 Speaker verification DET curves with NIST

Also indicated in parenthesis in the table is the number of parameters for each case. From this table it is clear that MixAR can achieve about the same performance using almost 4x fewer parameters than GMM. This reduction in the number of parameters points to the efficiency of MixAR in capturing the dynamic information.

Moreover, even when considering the best case scenario for GMM with a large number of parameters (8 mixtures with static as well as velocity and acceleration coefficients), there is a 10.6% relative reduction in EER with MixAR. These results appear to strongly indicate that there is nonlinear evolution information in speech features that the GMM model cannot capture using linear derivatives alone and that MixAR can effectively employ this information for achieving better speaker recognition.

Table 6.2 Speaker recognition EER with NIST for MixAR and GMM as a function of #mix. (the numbers of parameters are shown in parentheses).

# mix.	GMM Static+Δ+ΔΔ	MixAR Static
2	23.1(216)	24.1(120)
4	21.7(432)	19.2(240)
8	20.5(864)	19.1(480)
16	20.5(1728)	19.2(960)

6.2 A note on MixAR order

It was stated in Chapter IV that MixAR requires GEM approach to training with no closed form solution for evaluating the M-step for gate coefficients. There we use secant-based Newton gradient ascent approach to update the gate coefficients. Due to this, increasing gate order to a value higher than 1, becomes more computationally expensive while also potentially leading to instabilities in the estimated coefficients. Therefore, we do not consider order greater than 1 for the MixAR model.

However, we ran one experiment with NIST-2001 development database – the same as the one in previous section – with order 2. The performance comparison for the two orders is depicted in Table 6.3. The EER of 19.2 for order 2 is slightly (though perhaps not statistically significantly) lower than that for order 1 providing evidence that increasing orders for MixAR might not be helpful for speaker recognition. This supports fixing the MixAR model order to 1 for all subsequent experiments.

Table 6.3 Speaker recognition EER with NIST for MixAR orders 1 and 2

# mix.	MixAR Order #1	MixAR Order 2
8	19.1	19.2

6.3 Speaker Verification Experiments with TIMIT Database

Next, a real-world speaker verification performance of MixAR in comparison to GMM is studied using the standard TIMIT database [87]. The core test set in this database consists of 168 speakers, with 10 utterances each. For each speaker, 5 utterances were used for training while the remaining 5 were used for evaluation. Static (13 MFCCs), velocity (13 Δ-MFCCs), and acceleration (13 ΔΔ-MFCCs) features were extracted from each utterance. GMM and MixAR models with 4-, 8-, and 16-mixtures were trained for each speaker using training data and then speaker verification evaluation was conducted.

The EER results are shown in Table 6.4. Also indicated in parenthesis is the number of parameters for each case. The corresponding DET plots are depicted in Fig. 6.2. From this table and the figure, it is clear that the MixAR model, using fewer parameters, outperforms GMM.

Table 6.4 Speaker Verification Performance (EER) on TIMIT core set (the numbers of parameters are shown in parentheses).

# mix.	GMM Static+Δ+ΔΔ	MixAR Static
4	3.6 (432)	3.0 (240)
8	2.4 (864)	1.8 (480)
16	2.4 (1728)	1.7 (960)

In addition, only static MFCC features are sufficient for modeling speakers with MixAR. These results are similar to those reported in the previous section for the NIST 2001 database. This lends further support to the hypothesis that MixAR utilizes nonlinear information in speech to represent speaker characteristics better than what GMM modeling can achieve.

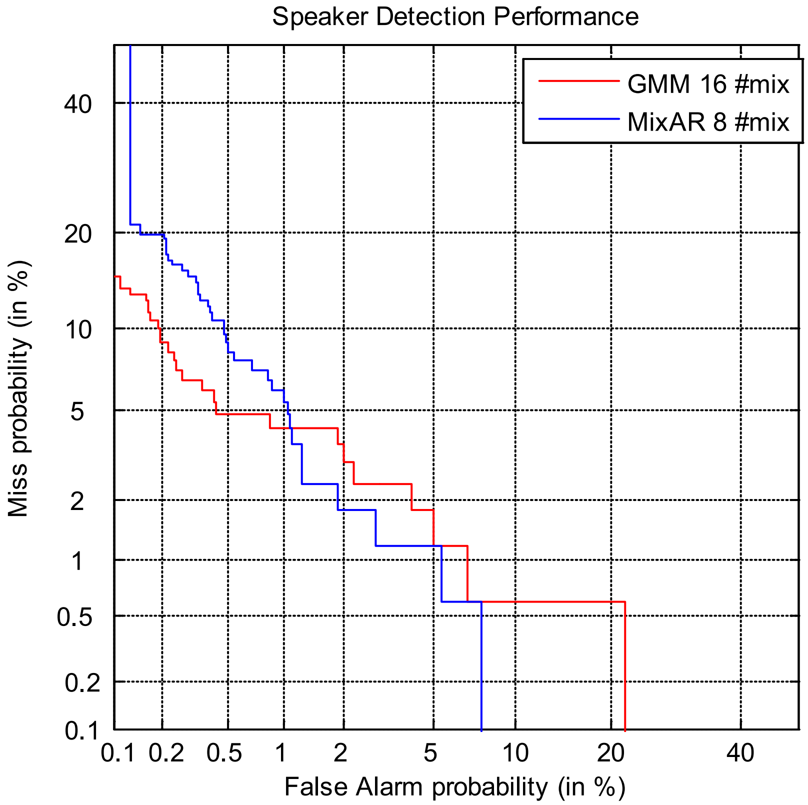


Figure 6.2 Speaker Verification DET curves for GMM and MixAR models on TIMIT database

6.4 Statistical Significance in Speaker Verification Systems

Before we proceed on to more comprehensive set of experiments, we briefly consider the problem of evaluating if the speaker verification results that we obtain are statistically significant. This is actually a much more difficult problem than it appears to be. In [92], the authors derived a statistical hypothesis testing for speaker verification based on the z-test statistic. In this, the idea is to first find false alarm rate (FAR), and false rejection rate (FRR) or equivalently miss probability, at the operating point. With, *NI* and *NC* being the number of imposter and correct speaker trials respectively, the confidence about the EER can be estimated as EER ± σ $Z_{\alpha/2}$ with:

$$\sigma = \sqrt{\frac{EER}{4}\left(\frac{1}{NI}+\frac{1}{NC}\right)} \qquad \text{Eq. 6.1}$$

and $Z_{\alpha/2}$ is the value of the standardized normal variable that bounds a confidence interval of α. For a 90% confidence interval, $Z_{\alpha/2}$=1.645.

For the TIMIT experiment that we discussed in previous section, this amounts to 1.8±0.23% and 2.4±0.43% for MixAR and GMM respectively.

Moreover, using a similar idea a test was derived to find if there is a statistically significant difference in verification performance between two models [92]. To achieve this, we compute the following z-statistic:

$$z = \frac{|\,EER1 - EER2\,|}{\sqrt{\begin{pmatrix} \frac{EER1\,(1-EER1)}{4} + \\ \frac{EER2\,(1-EER2)}{4} \end{pmatrix} \times \quad \left(\frac{1}{NI}+\frac{1}{NC}\right)}} \qquad \text{Eq. 6.2}$$

We can then estimate the degree of confidence that there is a significant difference between the two models using this statistic. Again, for the TIMIT of the previous section, this amounts to z=1.7104, which would correspond to a confidence interval of at most 91.28%.

We can continue evaluating the confidence of our results using similar analyses for other cases too. However, this has only limited usefulness. For instance, in the above formulation the length of evaluation utterances are not taken into account. Yet, it is known that if we increase the length of the evaluation utterances the reliability of the results should improve.

Up to now, there is no single easy way of evaluating the statistical significance of speaker verification results that can be found in literature. Instead, as is usually done in speaker verification work, we resort to using standard speech databases so that the speech processing community can compare other models if necessary on the same data. For this reason, we do not discuss the significance of results for other cases further. Difficulties and pitfalls faced when evaluating statistical significance of speaker verification results are summarized in [93].

6.5 Speaker Verification Experiments with TIMIT under Noisy Conditions

In the previous experiment it was shown that MixAR does better than GMMs for speaker verification on the TIMIT database. The TIMIT data was collected under noise-free recording conditions. To study the how well MixAR performs compared to GMM under noisy conditions thoroughly, a suite of experiments must be conducted. Similar to the way AURORA database was generated from TIDIGITS in [90] for studying noise performance, several noise conditions will be simulated by adding synthesized noise from different noise sources and at different SNRs. several noise conditions were simulated with TIMIT database [87] by adding synthesized noise from three different noise sources:

white, car, and babble. Three SNR levels were used: 10, 5 and 0 dB (in addition to the clean set). The core test partition of the database containing 168 speakers was used. The three types of noise sources were chosen to represent the most commonly occurring types of noise. Also, the four noise levels represent varying degrees of degradation that can occur in the real world. Speech processing systems tend to perform well when the SNR is above 10 dB, so we are primarily interested in studying situations where performance degrades severely.

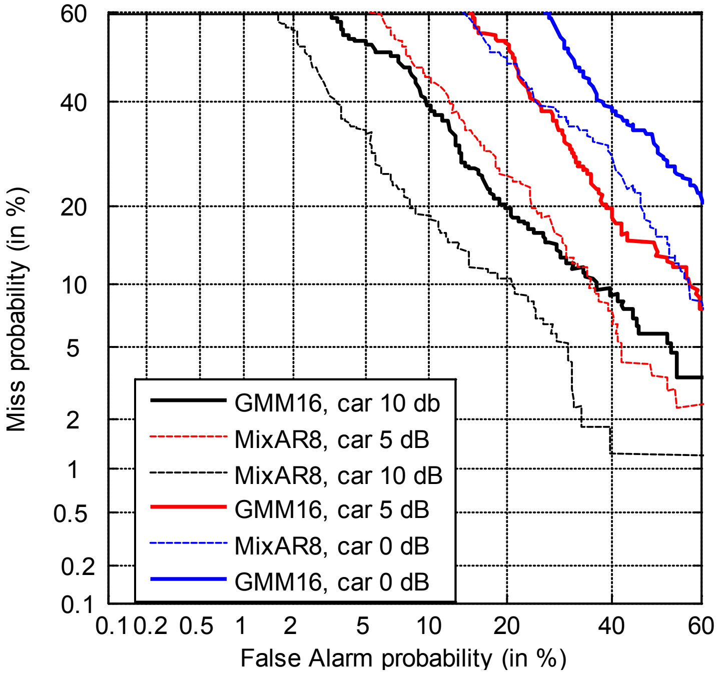


Figure 6.3 DET curves for GMM and MixAR models on noisy TIMIT test with additive car noise at different SNRs

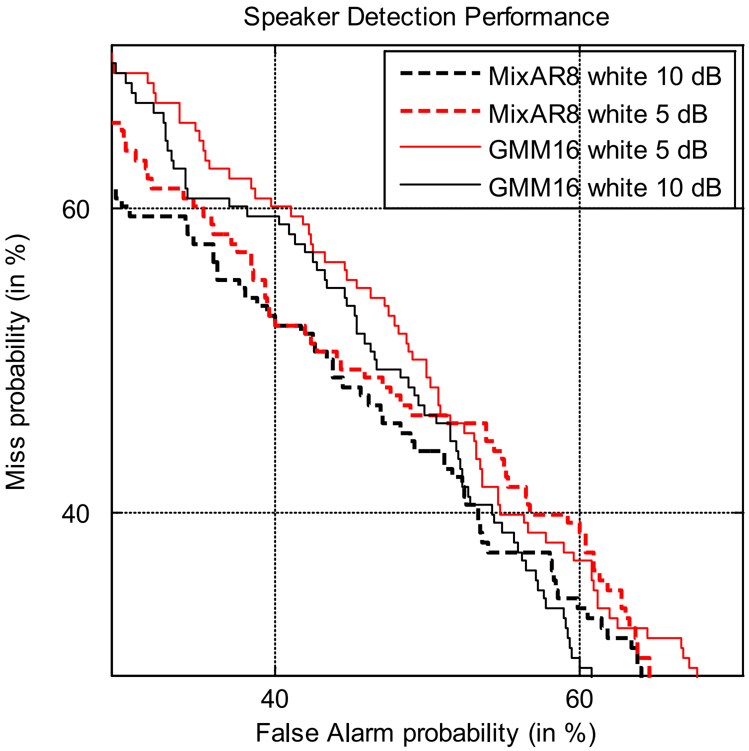


Figure 6.4 DET curves for GMM and MixAR models on noisy TIMIT test with additive white noise at different SNRs

The matrix of experimental results is shown in Table 6.5. The corresponding DET plots are shown in Figs. 6.3, 6.4 and 6.5. From these it is clear that while unseen noise conditions degrades performance for both models, MixAR performs relatively better than GMM and also uses 2.5 times fewer parameters. One trend that holds for both MixAR and GMM is that performance becomes poorer with lower SNRs for the same type of noise, and for a fixed SNR, performance in white noise is poorer than that in babble noise, which itself is poorer than that in car noise. It should be noted here that part of the reason the results in this table show significant degradation with even modest amounts of noise is because the training data is clean while only the test data is corrupted. Hence, these results reflect the potential performance of GMM and MixAR under unseen noise conditions.

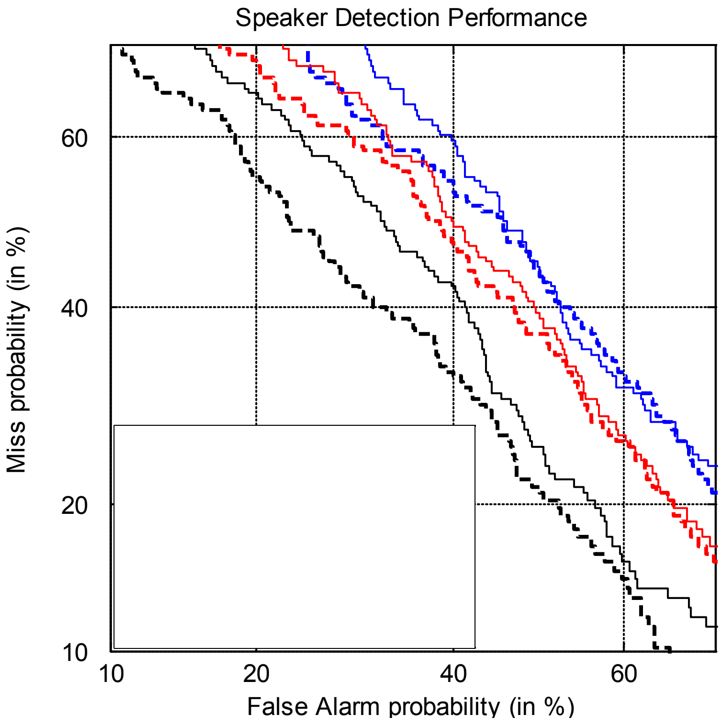


Figure 6.5 DET curves for GMM and MixAR models on noisy TIMIT test with additive babble noise at different SNRs

Table 6.5 Speaker Verification Performance (EER) for different noise conditions

	SNR (dB)	Car Noise	White Noise	Babble Noise
GMM* (1728)	Clean	2.4		
	10 dB	19.7	48.7	40.6
	5 dB	31.2	50.0	44.7
	0 dB	39.3	49.8	48.2
MixAR (480)	Clean	1.8		
	10 dB	13.7	47.0	36.9
	5 dB	23.2	47.6	42.8
	0dB	33.9	48.5	47.6

6.6 Speaker Verification Experiments Across Channel Conditions – TIMIT vs. NTIMIT Database

Channel variation is another problem that afflicts the performance of speech systems. NTIMIT is a database that was created by transmitting TIMIT utterances over different telephone channels [93]. Thus, NTIMIT simulates different channel conditions. We studied the speaker verification performance of NTIMIT core test set of 168 speakers by splitting the data for each speaker into 8 utterances for training and the remaining two utterances for evaluation.

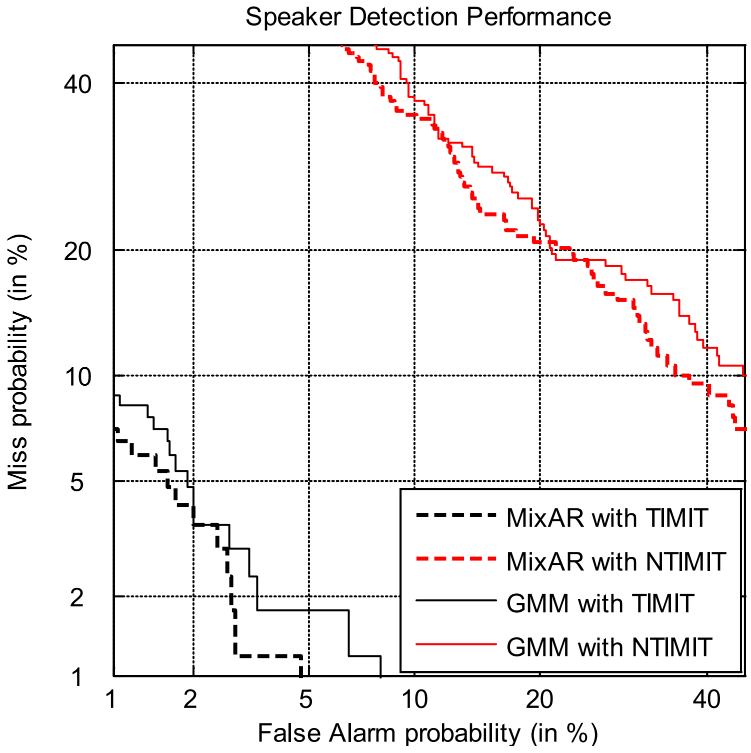


Figure 6.6 DET curves for GMM and MixAR models on TIMIT and NTIMIT databases

Table 6.6 Speaker Verification Performance EER with TIMIT and NTIMIT

Database	GMM (1728) Static+Δ+ΔΔ MFCCs	MixAR (480) Static MFCCs Only
TIMIT	2.4	1.8
NTIMIT	21.0	20.9

The DET performance curves for the 8-mixture MixAR using only static MFCCs (with 480 parameters) and for the 16-mixture GMM (with 1168 parameters) using both static and Δs is shown in Fig. 6.6. The corresponding EERs are shown in Table 6.6. From this it is clear that MixAR using 2.5 times fewer parameters achieves the same or higher level of performance as a GMM.

6.7 Speaker Verification Performance and Training Data Duration

Even if MixAR could do better under the conditions we have tested so far, it is possible that MixAR requires more training data than GMM for reliable parameter estimation. This could be a particular concern considering that MixAR attempts to learn nonlinear dynamic information, and nonlinear dynamics are notoriously difficult to characterize from short lengths of data. For example, it is known that estimates of Lyapunov exponents can be unreliable when the length of data is short [52]. One particular concern with insufficient training data is the problem of over-fitting. Lesser the training data, higher the chances that the model captures the inessential variation in the training data and hence that the model does not generalize well enough. It is therefore necessary to study performance as a function of the amount of training data.

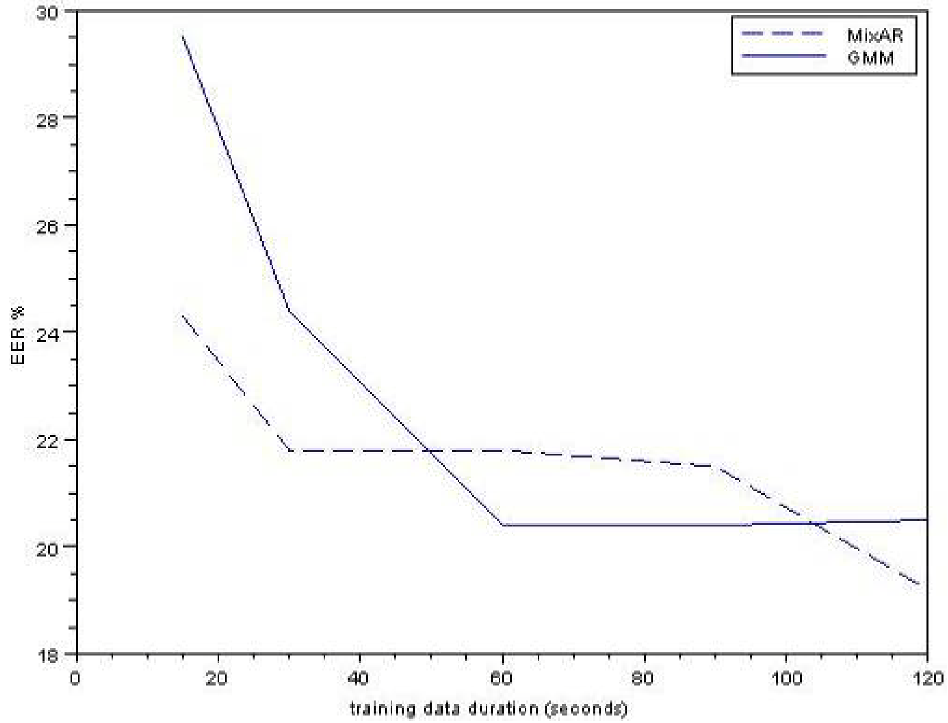


Figure 6.7 Speaker Verification Performance as a function of training utterance duration

Towards this end, we conducted experiments with varying training utterance durations keeping the evaluation utterance duration a constant. Utterances corresponding to five durations – about 120, 90, 60, 30 and 15s - were extracted from training data for each of the 60 speakers from the training part of NIST 2001 development database. All evaluation data for the 78 speakers with durations ranging mostly between 20 and 40 s were used. NIST-2001 database is particularly suited here because the training data is clean and the evaluation data is corrupted by different kinds of noise. This means that models that are over-trained will perform especially poor on the evaluation utterances.

Table 6.7 Speaker Verification Performance (EER) as a function of training data duration. (Evaluation utterance durations varied mostly between 20-40s)

	Training Utterance Duration	**EER**
GMM (864)	**120***	**20.5**
	90	**20.4**
	60	**20.4**
	30	**24.4**
	15	**29.5**
MixAR (480)	**120***	**19.2**
	90	**21.5**
	60	**21.8**
	30	**21.8**
	15	**24.3**

**Training data varied around 120s.*

Again 8-mixture MixAR models were used. The number of mixtures for GMM was reduced from 16 to 8 to alleviate the problem of over-fitting. The results of the experiment are reported in Table 6.7 and also graphed in Figure 6.7. From the table and figure, it is clear that MixAR does not suffer from over-fitting – at least, not any more than GMM does.

Table 6.8 Speaker Verification Performance (EER) as a function of evaluation data duration. (Training utterance durations varied mostly around 120s.)

	Evaluation Utterance Duration	EER
GMM (864)	30*	20.5
	15	21.8
	10	21.5
	5	24.4
	3	26.9
MixAR (480)	30*	19.2
	15	23.4
	10	23.1
	5	25.6
	3	25.6

**Evaluation data varied between 20-40s.*

There is a 43.9% increase in EER for GMM when the training utterance duration reduces from about 120s to 15 s. On the other hand, the corresponding increase in EER for MixAR is only 26.56%. Thus, this experiment leads us to conclude that MixAR can handle shorter training data durations better than GMM.

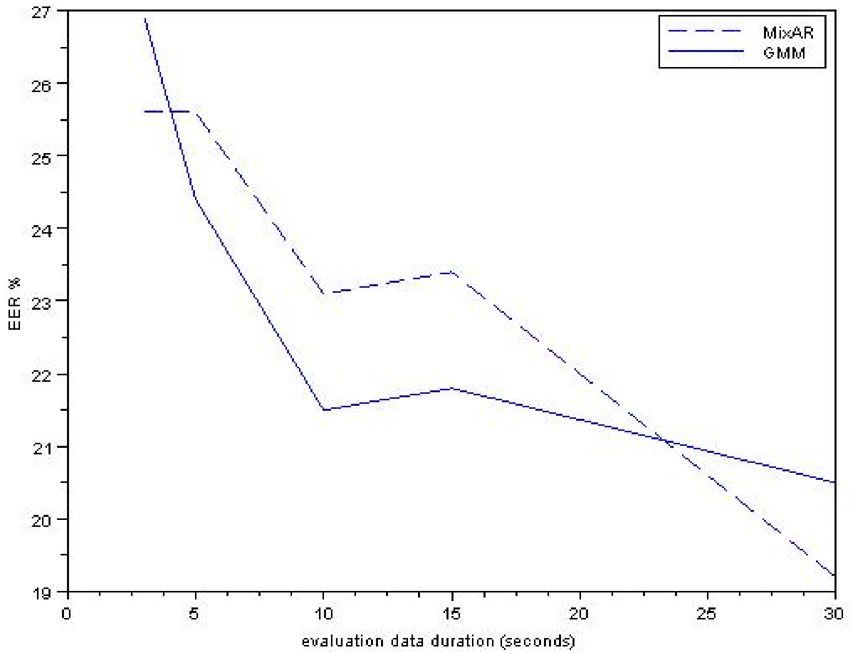


Figure 6.8 Speaker Verification Performance as a function of evaluation utterance duration.

6.8 Speaker Verification Performance and Evaluation Data Duration

It is reasonable to expect that increasing evaluation utterance durations improves speaker verification performance at least up to an extent. For short evaluation utterances, the speaker identity becomes more ambiguous. Thus, it is vital to evaluate the performance of models as a function of evaluation utterance durations.

For this purpose, evaluation utterances of five different durations – about 30, 15, 10, 5, and 3 s – were extracted from each of the 78 test utterances in NIST-2001 development database. All training data from all 60 speakers were used. The results of this experiment are reported in Table 6.8 and also graphed in Fig. 6.8. From this table, it is clear that performance mostly degrades as evaluation duration is reduced.

For GMM, there is an increase in EER of 31.2% as the evaluation duration reduces from about 30s to 3s. The corresponding reduction for MixAR is 33.3%. Thus MixAR appears to get slightly more sensitive than GMM as the evaluation data duration is reduced. This indicates that for very short utterances, GMM could perform better than MixAR.

CHAPTER VII

CONCLUSION AND FUTURE SCOPE

This chapter concludes the dissertation work and also offers some future extensions to the work conducted.

7.1 Conclusion

It is now well established that the dynamics of the speech signal are nonlinear and that this information is useful in speech and speaker recognition especially under noisy conditions. However, majority of current speech systems still utilize only linear processing techniques. Even among those few systems that take this nonlinearity into account, a large fraction use nonlinear dynamic invariants as additional features, which quantify only the degree of nonlinearity in the signal and are too crude to actually capture the signal dynamics. Other approaches are needed to alleviate this problem and to utilize the nonlinearity in speech signals for better performance.

Speaker verification is one facet of speaker recognition where the aim is to ascertain from the speech signal whether the claimed identity of the speaker is true or false. This has applications ranging from secured access and authentication to speaker segmentation in a conference meeting. Majority of the current speaker recognition systems employ Gaussian Mixture Model (GMM) approach to represent the statistics of speech Mel-Cepstral Coefficients (MFCCs). While this approach has been reasonably successful for speaker recognition task, this is fundamentally a linear approach and the nonlinear dynamics in speech are not utilized. To alleviate this problem, more powerful nonlinear modeling approaches or speech features that encapsulate the nonlinear information are required.

In this work, we applied a nonlinear mixture autoregressive (MixAR) model to supplant the Gaussian mixture model for speaker verification. MixAR is a statistical model that is a probabilistically weighted combination of components, each of which is an autoregressive filter in addition to a mean. The probabilistic mixing and the data-dependent weights are responsible for the nonlinear nature of the model. Our experiments with synthetic as well as real speech data show that MixAR model outperforms GMM especially under unseen noisy conditions, presumably due to nonlinear dynamical information contained in the speech features that GMM cannot model but MixAR captures. Moreover, in all cases we tested, MixAR did not require delta features and used 2.5x fewer parameters to achieve comparable or better performance as that of GMM using static as well as delta features. This we hypothesize is due to the in-built dynamic modeling capability of MixAR.

MixAR suffered less from over-fitting issues than GMM when training data was sparse. However, MixAR performance deteriorated quicker than GMM when evaluation data duration was reduced. This could pose limitations on the amount of evaluation data when using MixAR model for speaker verification.

7.2 Future Scope

While this work has shown that MixAR modeling holds much promise for speaker verification especially in mismatched conditions, several issues still need to be investigated. In the following, some of the major open issues and possible extensions are outlined.

Computational complexity is an important component to be studied to ensure that recognition is carried out in real time. It is of little utility in speaker recognition if a new model achieves near 100% accuracy, but takes days to verify the identity of a speaker from an utterance. In the present context, it is necessary to compare the computational requirements of MixAR and GMM. Training is mostly done offline and it does not matter much if this takes more time, but it is especially important that the evaluation has as little overhead as possible.

While we have used an ML-based approach for training both GMM and MixAR speaker models, several GMM systems now use an adapted GMM approach to train speaker models from a universal background model (UBM) [5][11]. To achieve this for MixAR, an adaptation approach similar to that of GMM needs to be developed. In our future work, we will work on deriving an adaptation approach for MixAR and compare it to the popular adapted mean approach for GMMs. We believe that MixAR will continue to outperform GMM under noisy conditions while using fewer parameters.

Discriminative approaches to classification are gaining importance in speech and speaker recognition. Already there exist speaker recognition systems based on GMMs that rely on discriminative training for estimating maximally-separated speaker models [14][16][17]. The key here is to define what separation between two models means, and to come up with an appropriate procedure to train models to maximize this distance for any two speakers. It would be interesting to design a discriminative approach to MixAR modeling for speaker verification and to note whether this improves over the ML approach to training models.

Due to the obvious parallels between MixAR and GMM, we can also look at some extensions of GMMs and check if these could also be made to MixAR. Onse such idea is the use of GMM supervector with a SVM classifier and form a hybrid generative-discriminative model (e. g. [4][15]).

Another very important extension would be to generalize the applicability of MixAR to other speech processing tasks such as speech recognition. We have already demonstrated the feasibility of MAR using a simple phone classification task [79]. The next step in the model analysis pipeline would be to demonstrate this first with simple isolated digit recognition tasks, and then extend this to large-vocabulary continuous speech recognition tasks.

REFERENCES

[1] X. Huang, A. Acero, and H. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, 2001.

[2] A. Rosenberg, F. Bimbot, S. Parthasarathy, "Overview of Speaker Recognition," pp. 725–741, book chapter in: Y. H. J. Benesty (editor), *Handbook of Speech Processing*, Springer, Berlin, Germany, 2008.

[3] J. P. Campbell, "Speaker Recognition: A Tutorial," *Proceedings of IEEE*, vol. 85, no. 9, pp. 1437-1462, Sept. 1997.

[4] H. Li, and T. Kinnuenen, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communications*, vol. 52, no. 1, pp. 12-40, January 2010.

[5] D. A. Reynolds, and W. M. Campbell, "Text-Independent Speaker Recognition," pp. 763–781, book chapter in: Y. H. J. Benesty (editor), *Handbook of Speech Processing*, Springer, Berlin, Germany, 2008.

[6] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in the Assessment of Detection Task Performance," *Proceedings of the uropean Conference on Speech Communication and Technology*, pp. 1895-1898, 1997.

[7] H.Stark, and J. W. Woods, Probability and Random Processes with Application to Signal Processing, 3rd edition, NJ, Prentice Hall, USA, October 2001.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd edition, Wiley-Interscience, USA, October 2000.

[9] S. Kuroiwa , Y. Umeda , S. Tsuge , and F. Ren, "Nonparametric Speaker Recognition Method using Earth Mover's Distance," *IEICE - Transactions on Information and Systems*, vol. E89-D, no. 3, pp. 1074-1081, March 2006.

[10] F. K. Soong, E. Rosenburg, B. Juang, and L. Rabiner, "A Vector Quantization Approach to Speaker Recognition," *AT&T Technical Journal*, vol. 66, pp. 14-26. 1987.

[11] D. A. Reynolds, “Speaker Identification and Verification using Gaussian Mixture Speaker Models”, *Speech Communication*, vol. 17, no. 1-2, pp. 91-108, 1995.

[12] M. Forsyth, “Discriminating Observation Probability HMM for Speaker Verification,” Speech Communication, vol. 17,nos. 1-2, pp. 117-129, August 1995.

[13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, “Support Vector Machines using GMM Supervectors for Speaker Verification,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[14] V. Wan and S. Renals, "Speaker Verification using Sequence Discriminant Support Vector Machines," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 203-210, March 2005.

[15] M. Liu, Y. Xie, Z. Yao, and B. Dai, “A New Hybrid GMM/SVM for Speaker Verification,” *Proceedings of The 18th International Conference on Pattern Recognition*, vol. 4, pp. 314-317, Hong Kong, China, August 2006.

[16] Q. Y. Hong and S. Kwong, “A Discriminative Training Approach for Text-Independent Speaker Recognition,” *Signal Processing*, vol. 85 no. 7, pp. 1449-1463, July 2005.

[17] C. Ma, and E. Chang, “Comparison of Discriminative Training Methods for Speaker Verification,” *Proceedings of the ICASSP 2003*, pp 192-195, 2003.

[18] W. Haberman, and A. Fejfar, “Automatic ID of Personnel through Speaker and Signature Verification: System Verification and Testing,” *Carnahan Conference on Crime Countermeasures*, University of Kentucky, USA, May 1976.

[19] “Speaker Recognition Evaluation (SRE)”, National Institute of Standards and Technology (NIST), online at *http://www.nist.gov/itl/iad/mig/sre.cfm*.

[20] J. Markel, B. Oshika, and A. H. Gray, “Long-Term Ffeature Averaging for Speaker Recognition,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 330-337, August 1977.

[21] T. Kinnunen, V. Hautamaki, and P. Franti,. “On the Use of Long-Term Average Spectrum in Automatic Speaker Recognition,”.*5th Int. Symposium on Chinese Spoken Language Processing*, Singapore, pp. 559-567, December 2006.

[22] T. Ganchev, M. Siafarikas, and N. Fakotakis, “Evaluation of speech parameterization methods for speaker recognition,” *Proc. Acoustics-2006*, Crete, Greece, pp. 105-110, September 2006.

[23] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task," *Proc. 10th International Conference on Speech and Computers (SPECOM) 2005*, Patras, Greece, vol. 1, pp. 191-194, October 2005.

[24] W. P. Ping, and P. H. Xia, "The Application of Fusion technology for Speaker Recognition," *International Journal of Computer Science and Network Security*, vol. 17, no. 12, December 2007.

[25] D. A. Reynolds, "Experimental Evaluation of Features for Robust Speaker Identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, October 1994.

[26] E. Shriberg, "Higher-Level Features in Speaker Recognition," *Speaker Classification I, Lecture Notes in Artificial Intelligence*, Springer, Berlin, pp. 241-259, 2007.

[27] "YOHO Speaker Verification Corpus ," online at *http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC94S16*, Linguistic Data Consortium, Philadelphia, 1994.

[28] B. R. Wildermoth, and K. K. Paliwal, "GMM-based Speaker Verification on Readily Available Databases," *Proc. Micro. Elec. Eng. Research Conference*, Brisbane, Australia, November 2003.

[29] Q. Y. Hong, and S. Kwong, "A Discriminative Training Approach for Text-Independent Speaker Recognition," *Signal Processing*, vol. 85, no. 7, pp. 1449-1463, July 2005.

[30] N. Kakotakis, K. Georgila, and A. Tsopanoglou, "A Continuous Text Independent Speaker Recognition System Based on Vowel Spotting," *Proc. Eurospeech-97*, Rhodes, Greece, pp. 2347-2350, September 1997.

[31] B. Ma, D. Zhu, R. Tong, and H. Li, "Speaker Cluster based GMM Tokenization for Speaker Recognition," *Proc. Interspeech-2006*, Pittsburgh, USA, pp. 505-508, September 2006.

[32] Y. F. Liao, W. C. Chang, Z. Y. Xie, D. Y. Zeng, and Y. T. Juang, "Joint Prosodic and Spectral Modeling for Robust Speaker Verification," *Proc. Speech Prosody 2008*, Campinas, Brazil, pp. 143-146, May 2008.

[33] A. G. Adami, "Modeling Prosodic Differences for Speaker Recognition," *Speech Communications*, vol. 49, no. 4, pp. 277-291, April 2007.

[34] R. B. Reid, "Speaker Verification in the Presence of Channel Mismatch using Gaussian Mixture Models," *Master's thesis*, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1997.

[35] J. Campbell, "Testing with The YOHO CD-ROM Voice Verification Corpus," *Proc. ICASSP*, Detroit, pp. 341-344, May 1995.

[36] A. Higgins, L. Bahler, and J. Porter. "Speaker Verification Using Randomized Phrase Prompting" *Digital Signal Processing*, vol. 1, no. 2, pp. 89-106, 1991.

[37] A. Higgins, L. Bahler, G. Vensko, J. Porter, and D. Vermilyea. *YOHO Speaker Authentication Final Report.* ITT Aerospace/Communications Division, 1992.

[38] R. A. Sukkar, M. B. Gandhi, and A. R. Setlur, “Speaker verification using mixture decomposition discrimination,” *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 292–299, May 2000.

[39] A. Kumar, and S. K. Mullick, “Nonlinear Dynamical Analysis of Speech,” *Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 615-629, July 1996.

[40] M. Banbrook, *Nonlinear Analysis of Speech From a Synthesis Perspective*, Ph.D. Thesis, The University of Edinburgh, Edinburgh, UK, 1996.

[41] I. Kokkinos, and P. Maragos, “Nonlinear Speech Analysis using Models for Chaotic Systems,” *IEEE Transactions on Speech and Audio Processing*, pp. 1098- 1109, Nov. 2005.

[42] J. P. Eckmann, and D. Ruelle, “Ergodic Theory of Chaos and Strange Attractors,” *Reviews of Modern Physics*, vol. 57, pp. 617-656, July 1985.

[43] H. Kantz, and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, UK, 2003.

[44] H. M. Teager and S. M. Teager, “Evidence for Nonlinear Production Mechanisms in the Vocal Tract,” *NATO Advanced Study Institute on Speech Production and Speech Modeling*, Bonas, France, pp. 241-261, July 1989.

[45] J. Markel, and A. Gray, *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

[46] R. W. Schafer, L. R. Rabiner, “Digital Representations of Speech Signals,” *Proceeding of the IEEE*, vol. 63, no. 4, April 1975.

[47] D. May, *Nonlinear Dynamic Invariants For Continuous Speech Recognition*, M.S. Thesis, Department of Electrical and Computer Engineering, Mississippi State University, USA, May 2008.

[48] S. Prasad, S. Srinivasan, M. Pannuri, G. Lazarou and J. Picone, "Nonlinear Dynamical Invariants for Speech Recognition," *Proceedings of the International Conference on Spoken Language Processing*, pp. 2518-2521, Pittsburgh, Pennsylvania, USA, September 2006.

[49] P. Maragos, A. G. Dimakis and I. Kokkinos, "Some Advances in Nonlinear Speech Modeling using Modulations, Fractals, and Chaos," *Proceedings of the IEEE International Conference on Digital Signal Processing*, pp. 325-332, Santorini, Greece, July 2002.

[50] M. Banbrook, S. McLaughlin, "Is Speech Chaotic?: Invariant Geometrical Measures for Speech Data," *IEE Colloquium on Exploiting Chaos in Signal Processin*g, Digest No. 1994/193, pp. 8/1-8/10, London, U.K., June 1994.

[51] P. Maragos, "Fractal Aspects of Speech Signals: Dimension and Interpolation," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 417-420, Toronto, Canada, May 1991.

[52] J.R.Munkres, *Topology*, 2nd edition, Prentice Hall, NJ, USA, 1999.

[53] R. Gilmore and M. Lefranc, *The Topology of Chaos: Alice in Stretch and Squeezeland*, Wiley, New York, USA, July 2002.

[54] M. Banbrook, G. Ushaw, and S. McLaughland, "How to Extract Lyapunov Exponents from Short and Noisy Time Series," *IEEE Transactions on Signal Processing*, vol. 45, no. 5, pp. 1378-1382, May 1997.

[55] H. F. V. Boshoff and M. Grotepass, "The Fractal Dimension of Fricative Speech Sounds," *Proceedings of the South African Symposium on Communication and Signal Processing*, pp. 12-16, Pretoria, South Africa, August 1991.

[56] A. C. Lindgren, M. T. Johnson and J. Povinelli, "Speech Recognition using Reconstructed Phase Space Features," *Proceedings of the International Conference on Acoustics, Speech and Signal Processin*g, vol. 1, pp. I-60-63, Hong Kong, China, April 2003.

[57] A. Petry, D. Augusto, and C. Barone, "Speaker Identification using Nonlinear Dynamical Features," *Chaos, Solitons & Fractals*, vol. 13, no. 2, pp. 221-231, February 2000.

[58] T. F. Quatieri, "Nonlinear Auditory Modeling as a Basis for Speaker Recognition," *Final Report*, MIT Lincoln Laboratory, May 2002.

[59] T. F. Quatieri, N. Maliska, and D. Sturim, "Auditory Signal Processing as a Basis for Speaker Recognition," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, NY, USA, pp. 111-114, October 2003.

[60] M. D. Plumpe, T. F. Quatieri, and D. A. Reynolds, “Modeling of the Glottal Flow Derivative Waveform with Application to Speaker Identification,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 569-586, September 1999.

[61] M. Chetouani, M. Faundez-Zanuy, B. Gas, and J. L. Zarader, “Non-linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition,” *Lecture Notes in Computer Science*, vol. 3445/2005, pp. 344-350, 2005.

[62] M. F. Zanuy, and D. R. Porcheron, “Speaker Recognition using Residual Signal of Linear and Nonlinear Prediction Models,” *Proc. ICSLP 1998*, Sydney, Australia, vol. 2, pp. 121-124, November 1998.

[63] B. J. Lee, J. Y. Choi, and H. G. Kang, “Phonetically optimized speaker modeling for robust speaker recognition,” *Journal of Acoustical Society of America*, vol. 126, no. 3, pp. 100-106, September 2009.

[64] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proc. Royal Soiety of London Series A,* vol. 454, no. 1971, pp. 903-995, March 1998.

[65] E. Ambikairajah, “Emerging Features for Speaker Recognition,” *Proc. 6th International Conference on Communications & Signal Processing-2007*, pp. 1-7, February 2008.

[66] Y. Liu, H. Yang, and H. Zhou, “Speaker Identification based on EMD,” *Proc. Network Infrastructure and Digital Content-2009,* Beijing, China, pp. 808-812, December 2009.

[67] B. G. Mindlin, M. A. Trevisan, and, M. C. Eguia, *Topological Voiceprints for Speaker Identification*, Patent PCT/US2004/027193, August 2004.

[68] K.A.Brown and K.P.Knudson, “Nonlinear Statistics of Human Speech Data,” *International Journal of Bifurcation and Chaos*, vol. 19, no. 7, pp. 2307-2319, July 2009.

[69] K. Janich, *Topology*, Springer-Verlag, New York, USA, January 1984.

[70] A. Dempster, N. Laird, and D. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1-38, February 1977.

[71] J. Blimes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation Problem for Gaussian Mixture and Hidden Markov Models,” *Technical Report TR-97-021*, University of California, Berkeley, May 1997.

[72] P. P. Boda, J. M. Veth, L. W. J. Boves, “Channel Normalisation by Using Rasta Filtering and the Dynamic Cepstrum for Automatic Speech Recognition over the Phone,” *Proc. of the Workshop on Auditory Basis of Speech Perception*, Keele, UK, pp. 317-320, July 1996.

[73] D. Reynolds, “Channel Robust Speaker Verification via Feature Mapping,” *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, vol. 2, pp. 53–56, April 2003.

[74] G. Gravier and G. Chollet, “Comparison of Normalization Techniques for Speaker Verification,” *Proc. Speaker Recognition and its Commercial and Forensic Applications*, Avignon, France, pp. 97-100, April 1998.a

[75] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas,.”Score Normalization for Text-Independent Speaker Verification Systems,” *Digital Signal Processing*, vol. 10, pp.42-54, January 2000.

[76] L. Heck, and M. Weintraub, “Handset-Dependent Background Models for Robust Text-Independent Speaker Recognition,” *Proc. International Conference. on Acoustics, Speech, and Signal Processing*, Munich, Germany, pp. 1071–1074, April 1997.

[77] M. Zeevi, R. Meir, and R. Adler, “Nonlinear Models for Time Series using Mixtures of Autoregressive Models”, *Technical Report*, Technion University, Israel, available online at: *http://ie.technion.ac.il/~radler/mixar.pdf*, October 2000.

[78] C. S. Wong, and W. K. Li, “On a Mixture Autoregressive Model,” *Journal of the Royal Statistical Society*, vol. 62, no. 1, pp. 95-115, February 2000.

[79] S. Srinivasan, T. Ma, D. May, G. Lazarou and J. Picone, "Nonlinear Mixture Autoregressive Hidden Markov Models for Speech Recognition," *Proceedings of the International Conference on Spoken Language Processing*, pp. 960-963, Brisbane, Australia, September 2008.

[80] B. H. Juang, and L. R. Rabiner, “Mixture Autoregressive Hidden Markov Models for Speech Signals,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 6, pp. 1404-1413, December 1985.

[81] Y. Ephraim, and W. J. Roberts, “Revisiting Autoregressive Hidden Markov Modeling of Speech Signals,” *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 166-169, February 2005.

[82] M. E. Ayadi, *Autoregressive Models for Text Independent Speaker Identification in Noisy Environments*, PhD Thesis, Department of Electrical and Computer Engineering, University of Waterloo, Canada, September 2008.

[83] A. X. Carvalho and M. A Tanner, “Modeling Nonlinearities with Mixtures-of-Experts of Time Series Models,” *International Journal of Mathematics and Mathematical Sciences*, vol. 2006, no. 9, pp. 1-22, May 2006.

[84] G. J. McLachlan, and T. Krishnan, *The EM algorithm and extensions*, 2nd edition, Wiley-Interscience, NJ, USA, 2008.

[85] J. E. Dennis, and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial Mathematics, Philadelphia, PA, USA, January 1996.

[86] National Institute of Standards and Technology, “The 2001 NIST Speaker Recognition Evaluation,” *http://www.nist.gov/speech/tests/spk/2001*, 2001.

[87] “TIMIT Acoustic-Phonetic Continuous Speech Corpus,” *http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1*, Linguistic Data Consortium, Philadelphia, 1993.

[88] J. Picone, “ISIP Foundation Classes,” *http://www.isip.piconepress.com/projects/speech/software/downloads/production*, Department of Electrical and Computer Engineering, Temple University, Philadelphia, Pennsylvania, USA, January 2010.

[89] Filtering and Noise-adding Tool (FaNT), available online at *http://dnt.kr.hsnr.de/download.html.*

[90] D. Pearce and H. Hirsch, The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions, *ICSLP 2000*, pp. 29-32, October 2000.

[91] Noise data from Signal Processing Information Base (SPIB) of Rice University, available online at *http://spib.rice.edu/spib/select_noise.html.*

[92] S. Bengio, and J. Mariethoz, “A stastical significance test for person authentication,” *Proceedings of The Speaker and Language Recognition Odyssey*, Toledo, Spain, pp. 237-244, June 2004.

[93] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, “The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspective,” *Speech Communication*, Vol. 31, pp. 225-254, June 2000.

[94] “NTIMIT Speech *Corpus*,” online at *http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S2*, Linguistic Data Consortium, Philadelphia, 1993.