Summer 2006

Gaussian Mixture Models and Neural Networks for Automatic Speaker Identification

Usha Gayatri Chalkapally
Old Dominion University

Recommended Citation
Chalkapally, Usha G.. "Gaussian Mixture Models and Neural Networks for Automatic Speaker Identification" (2006). Master of Science (MS), Thesis, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/9b7q-mw34
https://digitalcommons.odu.edu/ece_etds/304

GAUSSIAN MIXTURE MODELS AND NEURAL NETWORKS FOR AUTOMATIC SPEAKER IDENTIFICATION

by

Usha Gayatri Chalkapally
B.E (E.C.E) June 30, 2004, VCE, Osmania University, India.

A Thesis submitted to the faculty of
Old Dominion University in the Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
AUGUST 2006

Approved by:

Dr. Stephen A. Zahorian (Director)

Dr. Vijayan K. Asari (Member)

Dr. Oscar R. González (Member)

ABSTRACT

GAUSSIAN MIXTURE MODELS AND NEURAL NETWORKS FOR AUTOMATIC SPEAKER IDENTIFICATION

Usha Gayatri Chalkapally
Old Dominion University, August 2006
Director: Dr. Stephen A. Zahorian

Automatic Speaker Recognition is the process of automatically recognizing who is speaking on the basis of individual information contained in speech signals. This technique of Automatic Speaker Recognition makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

In this thesis, the techniques of Gaussian Mixture Models and Neural Networks for Automatic Speaker Identification are presented. Algorithms for Speaker Identification using Gaussian Mixture Models were developed, using both full covariance matrices and diagonal covariance matrices and were tested on the NTIMIT and SPIDRE databases. Experiments were also conducted using the existing neural network – Binary-Pair Partitioned on the NTIMIT and SPIDRE databases.

GMMs are trained with the maximum likelihood approach to give good models for each speaker. In contrast, NNs are trained to discriminate between speakers but form no explicit models for each speaker. It is conjectured that an appropriate combination of maximum likelihood and discriminative training, using both GMMs and NNs should give a better recognition rate than either method alone. Thus, fusion approaches to combine the Gaussian Mixture Models and neural networks for Speaker ID are proposed and tested for various cases. Comparisons of the best results obtained with each method were made for all the cases that were evaluated.

From the comparison of the results obtained with the GMMs alone, NNs alone, and combined GMM/NN, it was observed that the results obtained with GMM alone yielded the best recognition accuracy. The best accuracy using the NTIMIT database was 89.2% on the test data, and the best accuracy using the SPIDRE database was 86.7% on the test data, both results obtained using GMMs alone.

This Thesis is dedicated to my parents Kishan and Sreedevi Chalkapally, my sisters Uma and Gautami and my brother Sai.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr. Stephen A. Zahorian for his support, advice and motivation over the past two years. Without his constant guidance, it would not have been possible to complete this thesis.

I would like to thank Dr. Vijayan K. Asari and Dr. Oscar R. González for agreeing to be on my committee and for putting in their valuable time.

I would like to thank the members of the Speech Communications Lab for helping me with my research and also all my friends for being there for me always.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Speech is one of the most natural ways for humans to interact. When it comes to computers it is no different. If a computer application could be controlled solely by way of voice commands then new opportunities would be presented. Even though the idea of using speech as an input mechanism for a computer is not new, there is still a lot of research going on in this field. In fact, human-computer interaction using speech is still largely unrealized.

A field closely related to automatic speech recognition is automatic speaker recognition. Automatic speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information contained in speech signals. It can be divided into Speaker Identification and Speaker Verification. Speaker Identification determines which registered speaker provides a given utterance from amongst a set of known speakers. Speaker Verification accepts or rejects the claimed identity of a speaker based on an arbitrary utterance from a pool of unlimited speakers.

This technique of automatic speaker recognition makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers. Automatic speaker recognition technology is expected to create new services that will make our daily lives more convenient. Another important application of speaker recognition technology is for forensic purposes.

There are two basic types of Speaker Recognition methods. They are listed below:

1.1 Text- Dependent Speaker Recognition

Text-dependent speaker recognition seeks to associate an unknown speaker with a member from a registered population, given a textual transcription of the phrases uttered by the speaker. Typically, speaker-dependent word or sub word models are built for each speaker. Given a labeled utterance from an unknown speaker, the system makes its speaker recognition decision based on the likelihood scores of the appropriate speaker dependent models.

Text-dependent methods are usually based on template-matching techniques. In this approach, the input utterance is represented by a sequence of feature vectors, generally short-term spectral feature vectors. The time axes of the input utterance and each reference template or reference model of the registered speakers are aligned using a dynamic time warping (DTW) algorithm, and the degree of similarity between them, accumulated from the beginning to the end of the utterance, is calculated.

A Hidden Markov Model (HMM) can efficiently model statistical variation in spectral features. Therefore, HMM-based methods have been used as extensions of the DTW-based methods and have achieved significantly better recognition accuracies.

1.2 Text -Independent Speaker Recognition

On the other hand, for text-independent domains, the speech of a speaker is largely unconstrained and often cannot feasibly be constrained with even the lexical content of highly variable utterances. Because speaker cooperation is not necessary, systems designed for the text-independent domain are considered more flexible.

One of the most successful text-independent recognition methods is based on the vector quantization (VQ). In this method, VQ codebooks consisting of a small number of representative feature vectors are used as an efficient means of characterizing speaker-specific features. A speaker-specific codebook is generated by clustering the training feature vectors of each speaker. In the recognition stage, an input utterance is vector-quantized using the codebook of each reference speaker and the VQ distortion accumulated over the entire input utterance is used to make the recognition decision.

Temporal variation in speech signal parameters over the long term can be represented by stochastic Markovian transitions between states. Therefore, methods using an ergodic HMM, where all possible transitions between states are allowed, have been proposed. Speech segments are classified into one of the broad phonetic categories corresponding to the HMM states. After the classification, appropriate features are selected.

In the training phase, reference templates are generated and verification thresholds are computed for each phonetic category. In the verification phase, after the phonetic categorization, a comparison with the reference template for each particular category provides a verification score for that category. The final verification score is a weighted linear combination of the scores from each category.

The "standard" HMM method has been extended to the richer class of mixture autoregressive (AR) HMMs. In these models, the states are described as a linear combination (mixture) of AR sources. It can be shown that mixture models are equivalent to a larger HMM with simple states, with additional constraints on the possible transitions between states.

It has been shown that a continuous ergodic HMM method is far superior to a discrete ergodic HMM method and that a continuous ergodic HMM method is as robust as a VQ-based method when enough training data is available. However, when little data is available, the VQ-based method is more robust than a continuous HMM method [28].

A method using statistical dynamic features has recently been proposed. In this method, a multivariate auto-regression (MAR) model is applied to the time series of Cepstral vectors and used to characterize speakers. It was reported that identification and verification rates were almost the same as those obtained by an HMM-based method.

All technologies for speaker recognition, identification and verification, text independent and text-dependent, have advantages and disadvantages and require different treatments and techniques. The choice of which technology to use is application-specific.

At the highest level, all speaker recognition systems contain two main modules-- *the training module and the testing module*, shown in Fig. 1.1 and Fig. 1.2 respectively. The training phase is the process that extracts features from a small amount of data from the voice signal that can later be used to represent each speaker. The testing phase involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers.

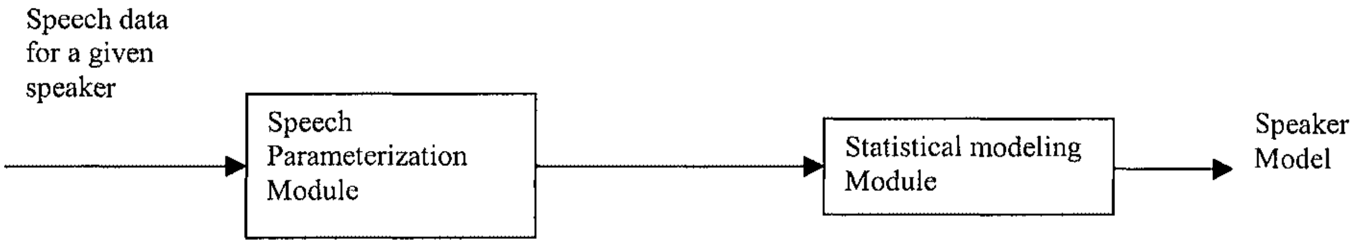
Speech data for a given speaker
Speech Parameterization Module
Statistical modeling Module
Speaker Model

Fig. 1.1 Modular Representation of the Training Phase of a Speaker Identification System [25].

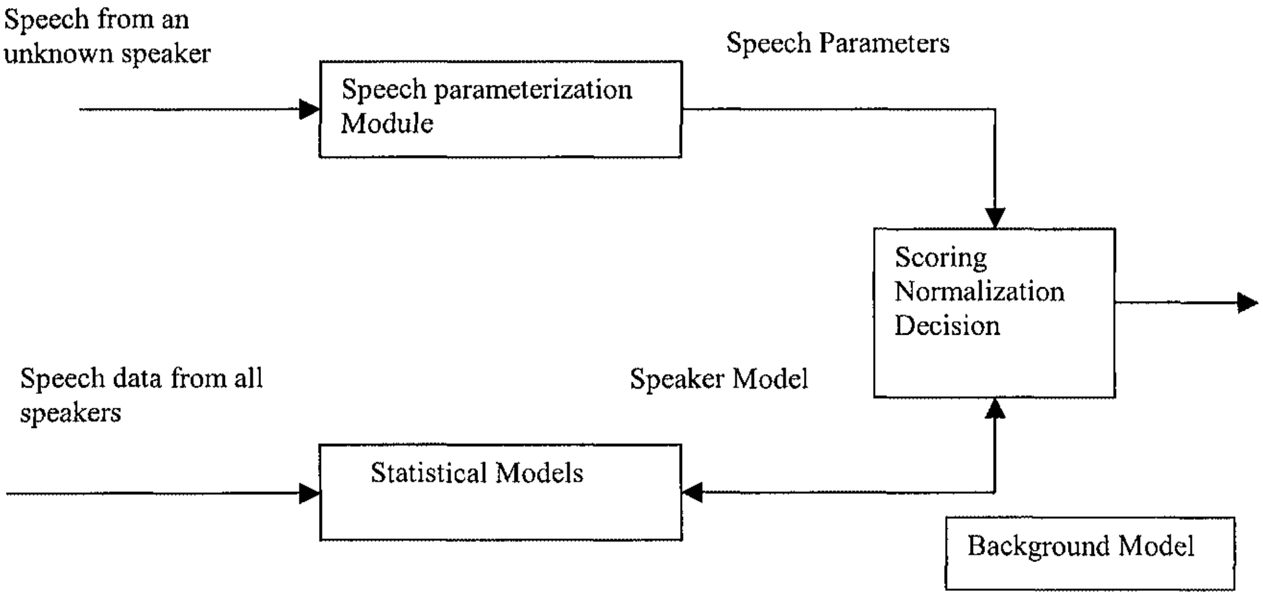

Fig. 1.2 Modular Representation of the Test Phase of a Speaker Identification System [25].

1.3 Outline of Work

The objective of this thesis is to develop a Gaussian Mixture Model (GMM) based Speaker Identification system, compare the results obtained by this model with the existing Binary-pair Partitioned Neural (BPP) network based Speaker Identification system and then investigate methods to combine the models. Since the GMM is trained with maximum likelihood methods, with the model for each speaker obtained using only data from that speaker, whereas the BPP is discriminatively trained, it is hypothesized that a combination of the two methods will be superior to either method alone.

Chapter 2 contains a description of the Gaussian training and testing procedure, in theoretical terms. This chapter also includes the description of the binary-pair partitioned neural network method and also the results from the literature using each of the methods.

Chapter 3 describes the Gaussian Mixture Modeling in more detail and also the process for transforming data with Gaussian Mixture Mdels and then training the neural networks using the Gaussian Mixture Model's outputs as new features.

Chapter 4 gives the description of all the experiments performed and the results obtained using both the methods and also the combined method.

Chapter 5 gives a conclusion to the thesis summarizing the method used and the key results obtained and it suggests some other areas where these ideas can be employed and the future work that is to be done.

CHAPTER II

GAUSSIAN MIXTURE MODELS AND NEURAL NETWORKS FOR SPEAKER IDENTIFICATION

Gaussian Mixture Models have proven to be a very powerful tool to characterize and distinguish acoustic information. This model is the most common approach used for speaker verification and speaker identification systems. In such a system, a Gaussian Mixture Model is obtained for each speaker or a group of speakers. The model characterizes the speaker or the group of speakers based on a set of parameters.

A Gaussian Mixture Model is a probabilistic model defined as a weighted sum of Gaussian density functions with different means and covariances. The model, referred to as Λ, can be represented as a sum of Gaussian densities with means μ_i, covariances Σ_i, and weights w_i, where i=1, 2..., M, with M the number of mixture components, i.e. the number of Gaussian density functions. The model for one dimension is illustrated with the following equation:

$$p(x/\Lambda)=\sum_{i=1}^{M} w_i G(x/\mu_i,\sigma_i) \qquad \textbf{(2.1)}$$

where $G(x/\mu_i,\sigma_i)$ represents the Gaussian density function for the i^{th} mixture component and is given by the equation,

$$G(x/\mu_i,\sigma_i)=\frac{1}{\sqrt{2\pi\sigma_i}}e^{-(x-\mu_i)^2 / 2\sigma_i^2} \qquad \textbf{(2.2)}$$

and the weights are constrained by the equation

$\sum_{i=1}^{M} w_i = 1$ i.e. the sum of the weights is 1.

The figure below shows an example of a single variable Gaussian Mixture Model with two mixture components, i.e. two Gaussian density functions:

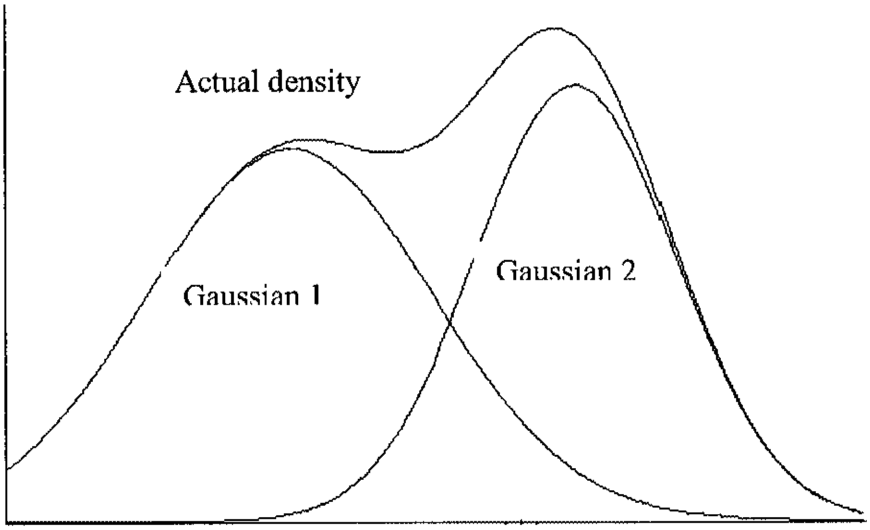

Fig. 2.1. Single Variable Gaussian Mixture Model [8].

The figure above is for single variable Gaussian densities. However, for speaker recognition, the Gaussian densities are generally multivariate, i.e., there are a number of variables which are the features for each speaker's acoustic information. The equations for such a multivariate Gaussian density are given as:

$$G(\mathbf{x}/\mu,\Sigma)=\frac{1}{|2\pi\Sigma|^{1/2}}e^{(-1/2(\mathbf{x}-\mu)^{T}\Sigma^{-1}(\mathbf{x}-\mu))} \quad \textbf{(2.3)}$$

where **x** represents the variable vector, μ is the mean vector and Σ is the covariance matrix. A two-dimensional multivariate Gaussian density is depicted in Figure 2.2.

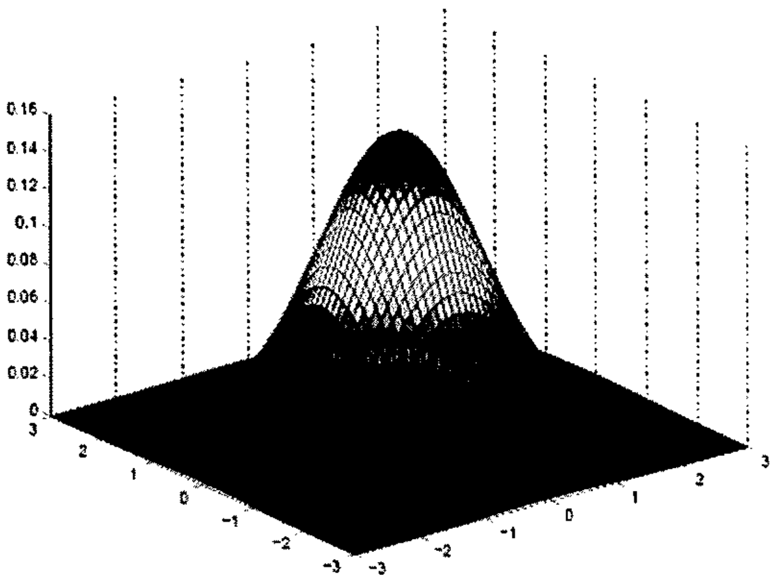

Fig. 2.2. Two-Dimension Gaussian Density Function [8].

2.1 Gaussian Mixture Models for Speaker Identification

Each speaker or a group of speakers will have a model representing its characteristics. Each model consists of the following parameters: M weights w_i, M mean vectors μ_i, and M covariance matrices Σ_i. Such models are estimated for each of the N speakers to be identified and are stored in a database. The training phase of the speaker ID system consists of estimating these parameters. This training phase is explained in more detail below.

In the testing phase, the likelihood values of the test speaker's spectral features with respect to each speaker model are computed. The speaker model that results in the maximum of these likelihood values is considered to be from the unknown speaker.

2.1.1 Gaussian Mixture Model Estimation

EM Algorithm: Expectation Maximization Algorithm

The EM (Expectation Maximization) Algorithm is a general iterative algorithm for maximum likelihood estimation where some unknown parameters are estimated given measurement data. The algorithm is initialized with random values for the parameters that are to be estimated. Then the algorithm iteratively alternates between two steps: Expectation and Maximization. In the Expectation-step, the expected likelihood for the data is computed where the expectation is taken with respect to the computed conditional distribution of the unknown parameters, given the current settings of parameters and the observed data. In the Maximization-step, the parameters are re-estimated by maximizing the likelihood function. Given a new generation of parameter values, the E-step and another M-step can be repeated. This process continues until the likelihood converges, i.e., reaching a local maxima. Thus, the aim of the EM algorithm is to find the parameters θ^n, such that $Q(\theta^{n+1}) > Q(\theta^n)$, where $Q(\theta^n)$ represents the expected value of the complete data log-likelihood function in terms of the previous parameters, θ^n represents the parameters at the end of the nth iteration and $Q(\theta^{n+1})$ represents the likelihood at the end of the (n+1)th iteration. Note that the EM algorithm applied to GMMs can be conveniently developed by positing that some of the data is missing ('incomplete data') even though in fact none of the data is missing ('complete data'). The EM algorithm with these assumptions about data is as follows:

1. Initialize θ^0 randomly or heuristically according to any prior knowledge about the optimal parameter values.

2. Iteratively alternate between the steps Expectation and Maximization until an optimum estimate of the parameters is obtained. The Expectation and Maximization steps respectively are:
   (a) Expectation Step: Compute the function $Q(\theta;\theta'')$
   (b) Maximization Step: Re-estimate the parameter values θ'' by maximizing the value of the function $Q(\theta;\theta'')$, i.e., find the values of θ'' for which the function Q has the maximum value.
3. Stop when the Likelihood value converges.

EM algorithm in detail applied to Gaussian Mixture Models:

The abstract discussion of the previous section is now related specifically to the problem of estimation of Gaussian Mixture Models. Note that θ from the previous discussion refers to the Gaussian model parameters (means μ_i, covariances Σ_i, and weights w_i), and Q is the likelihood of a given set of feature vectors, given θ.

First, the probabilistic model is assumed as

$$p(x \mid \Theta) = \sum_{i=1}^{M} \alpha_i p_i(x \mid \theta_i) \qquad \textbf{(2.4)}$$

where the parameters are $\Theta = ((\alpha_1,......,\alpha_M, \theta_1,......,\theta_M)$, such that $\sum_{i=1}^{M} \alpha_i = 1$ and each p_i is a Gaussian density function parameterized by θ_i (means μ_i, covariances Σ_i). There are M component densities mixed together with M mixing coefficients α_i.

The data log-likelihood expression for this density from the data X is given by:

$$\log(L(\Theta \mid X)) = \log \prod_{i-1}^{N} p(x_i \mid \theta) = \sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} \alpha_i p_j(x_i \mid \theta_j) \right) \qquad \textbf{(2.5)}$$

It is difficult to optimize this expression because it contains the log of the sum. The solution for this is to consider X as incomplete and posit the existence of unobserved data items $Y = \{y_i\}_{i=1}^{N}$ whose values indicate which component density generated each data item. This helps in simplifying the likelihood expression significantly. i.e. it is assumed that $y_i \in 1,...,M$ for each I, and $y_i = k$ if the i$^{\text{th}}$ sample was generated by the k$^{\text{th}}$ mixture component. If the data values of Y are known, the likelihood expression takes the form:

$$\log(L(\Theta \mid X,Y)) = \log(P(X,Y \mid \Theta)) = \sum_{i=1}^{N} \log(P(x_i \mid y_i)P(y)) = \sum_{i=1}^{N} \log(\alpha_{y_i} p_{y_i}(x_i \mid \theta_{y_i}))$$

(2.6)

which, given a particular form of the component densities, can be optimized using a variety of techniques.

The next problem is that the values of Y are unknown. It is assumed to be a random vector. An expression for the distribution of the unobserved data is first derived. The initial parameters for the mixture density are assumed as $\Theta^g = (\alpha_1^g,...,\alpha_M^g,\theta_1^g,...,\theta_M^g)$. Given Θ^g, $p_j(x_i \mid \theta_j^g)$ for each i and j is computed. The mixing parameters α_j can be thought of as prior probabilities of each mixture component, i.e, $\alpha_j = p$ (component j). Using Baye's rule, $p(y_i \mid x_i, \Theta^g)$ can be determined as shown below:

$$p(y_i \mid x_i, \Theta^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i \mid \theta_{y_i}^g)}{p(x_i \mid \Theta^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i \mid \theta_{y_i}^g)}{\sum_{k=1}^{M} \alpha_k^g p_k(x_i \mid \theta_k^g)}$$

(2.7)

$$p(y \mid X, \Theta^g) = \prod_{i=1}^{N} p(y_i \mid x_i, \Theta^g)$$

where Y=$(y_1,...,y_N)$ is an instance of the unobserved data independently drawn. Then the likelihood equation takes the form:

$$Q(\Theta,\Theta^g)=\sum_{y\in \mathrm{Y}}\log(L(\Theta\mid X,Y))p(Y\mid X,\Theta^g)$$
$$Q(\Theta,\Theta^g)=\sum_{y\in \mathrm{Y}}\sum_{i=1}^{N}\log(\alpha_{y_i}p_{y_i}(x_i\mid\theta_{y_i}))\prod_{j=1}^{N}p(y_j\mid x_j,\Theta^g) \quad \textbf{(2.8)}$$

This can be simplified as

$$Q(\Theta,\Theta^g)=\sum_{l=1}^{M}\sum_{i=1}^{N}\log(\alpha_l p_l(x_i\mid\theta_l))p(l\mid x_i,\Theta^g) \quad \textbf{(2.9)}$$

The expression above is to be maximized. To do that, the term containing α_l and the term containing θ_l are maximized independently. The logarithm term helps us in doing that. Solving all the equations would result in the following equations to estimate the new parameters from the old ones.

$$\alpha_l^{new}=\frac{1}{N}\sum_{i=1}^{N}p(l\mid x_i,\Theta^g)$$
$$\mu_l^{new}=\frac{\sum_{i=1}^{N}x_i p(l\mid x_i,\Theta^g)}{\sum_{i=1}^{N}p(l\mid x_i,\Theta^g)} \quad \textbf{(2.10)}$$
$$\Sigma_l^{new}=\frac{\sum_{i=1}^{N}p(l\mid x_i,\Theta^g)(x_i-\mu_l^{new})(x_i-\mu_l^{new})^T}{\sum_{i=1}^{N}p(l\mid x_i,\Theta^g)}$$

The above equations perform both the expectation and the maximization step simultaneously. The algorithm proceeds by using newly derived parameters as the guess for the next iteration [1].

To better illustrate the concepts described above, an example is presented which uses the EM algorithm to derive a simple mixture model. In the example, the means of two mixture components are to be estimated from the given data samples (Figure 2.3) without knowing which mixture component the data sample belongs to. Figure 2.4 shows the Likelihood function. Recall that the likelihood refers to the likelihood of parameter values, given the data.

EM proceeds as follows in this example. In the Expectation-step, an assignment is computed that assigns a posterior probability to each possible association of each individual sample. In the current example, there are 2 mixtures and 6 samples, so the computed probabilities can be represented in a 2 x 6 table. Given these probabilities, EM computes a tight lower bound to the true likelihood function of Figure 2.4. The bound is constructed such that it touches the likelihood function at the current estimate, and it is only close to the true likelihood in the neighborhood of this estimate. The bound along with its corresponding

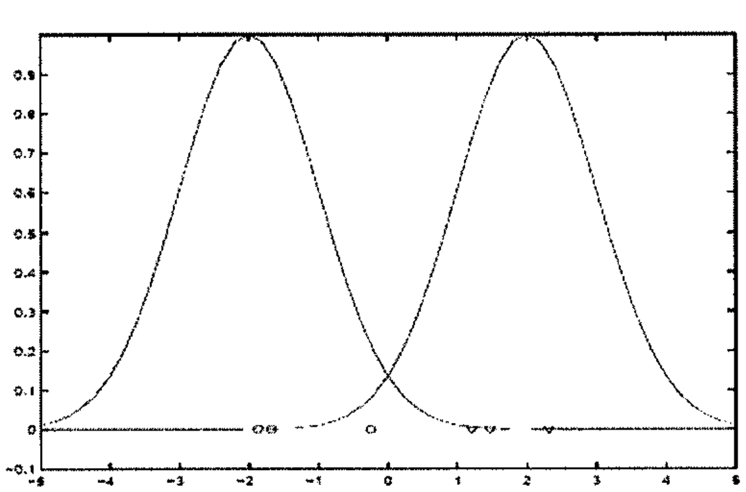

Fig. 2.3. EM Example. The data consists of three samples drawn from each of two mixture components, shown above as circles and triangles. The means of the mixture components are -2 and 2, respectively [8].

probability table is computed in each iteration, as shown in Figure 2.5. In this case, EM was run for 5 iterations. In the Maximization step, the lower bound is maximized (shown by a black asterisk in the figure), and the corresponding new estimate (θ_1, θ_2) is guaranteed to lie closer to the location of the nearest local maximum of the likelihood. Each next bound is an increasingly better approximation to the mode of the likelihood, until at convergence the bound touches the likelihood at the local maximum, and progress can no longer be made. This is shown in the last panel of Figure 2.5.

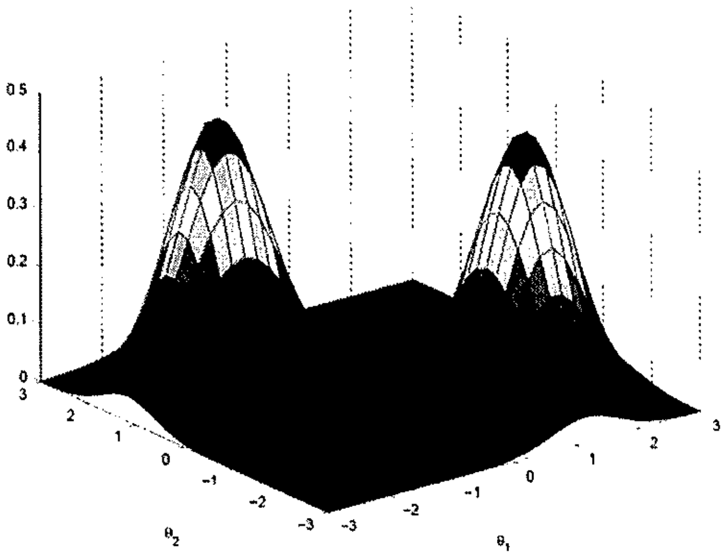

Fig. 2.4. The True Likelihood Function of the Two-Component Means, given the data [8].

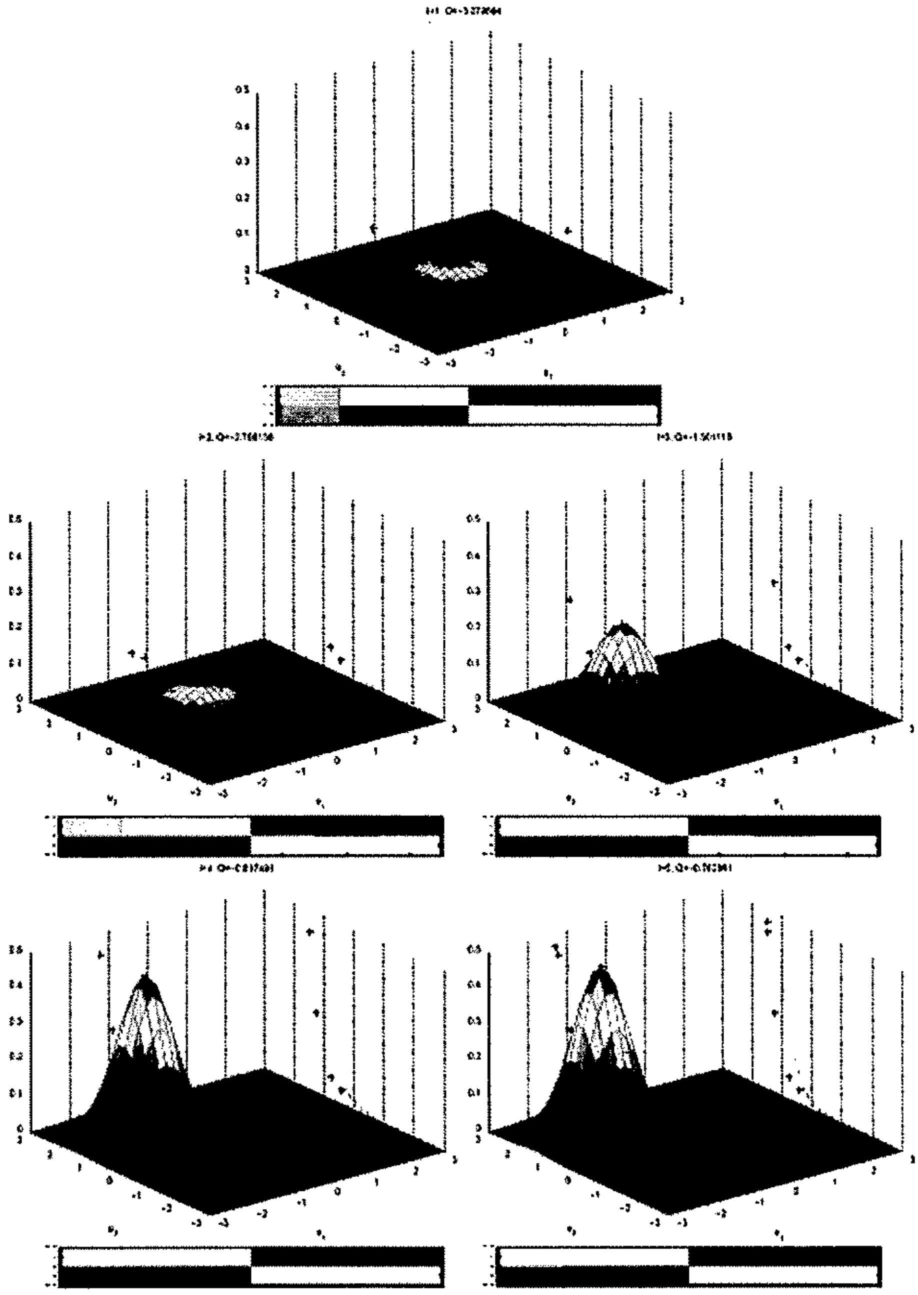

Fig. 2.5. An Illustration of the EM Algorithm for estimating the means of a Two-Component GMM from 6 data points. After 5 iterations, the EM converges to one of the maxima shown [8].

The following graph depicted in Fig. 2.6 illustrates the typical performance of a Gaussian Mixture Model based Speaker Identification system as the number of mixture components is varied. These results were obtained from the NTIMIT database with 102 speakers, using 8 sentences for training and 2 sentences for testing. In the graph, Series1 refers to the case of 25 features and 2 sentences used for testing, Series 2 refers to the case of 25 features and 1 sentence for testing, and Series 3 refers to 20 features and 2 sentences used for testing.

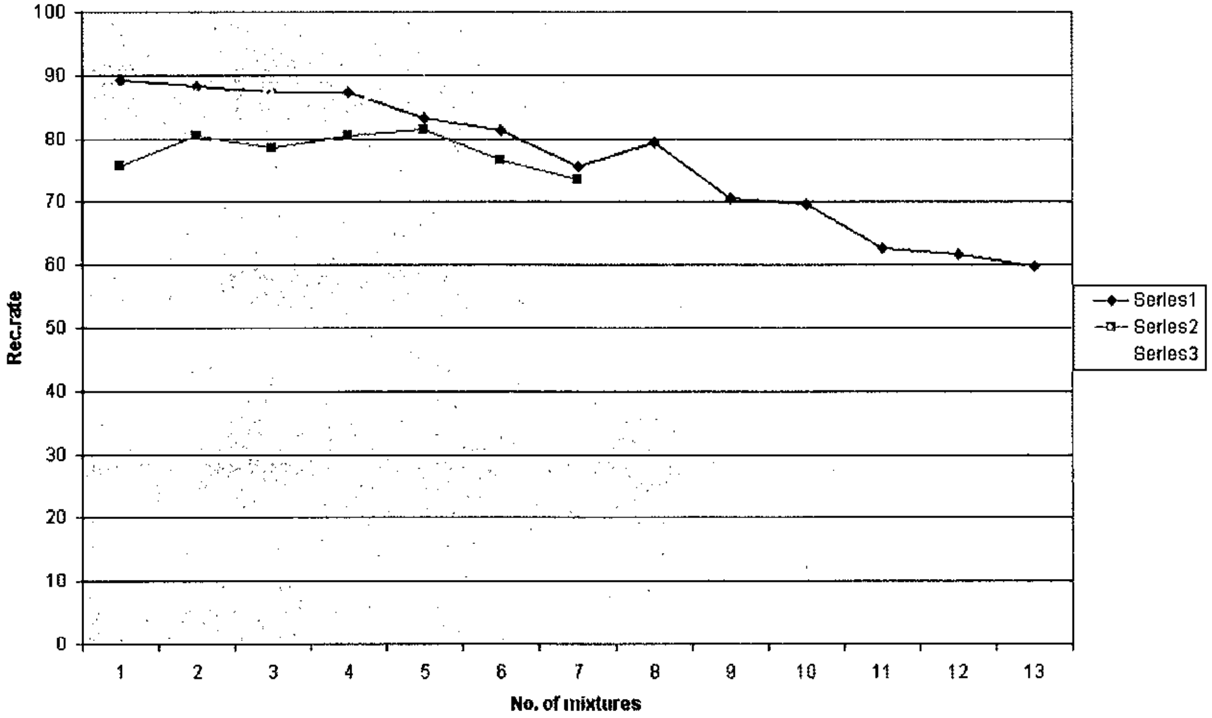

Fig. 2.6. Sample Speaker ID Recognition Results using GMMs.

From the above graph, we can conclude that the recognition rate of the Gaussian Mixture Model Based Speaker Identification System is almost 90% for the best of the cases. Chapter 4 gives results for a variety of cases using two public domain databases.

2.2 Neural Networks for Automatic Speaker ID

Automatic speaker identification is essentially a statistical pattern classification problem involving decisions over M categories. Here, a brief review of methods of data partitioning and advantages of binary pair partitioning are provided along with a discussion of the added feasibility of such a partitioning approach for speaker identification. Neural Networks have been shown to work exceptionally well for small but relatively difficult classification tasks – especially speaker identification. For some cases, it has been shown that a neural network classifier performs significantly better than a maximum likelihood classifier for ASI.

Nevertheless, Neural Network Classifiers (NNC) have problems. For example, it is known that the amount of training data and number of iterations of training have a significant impact on the performance of an NNC. This training/performance issue of the NNC manifests itself particularly when dealing with a relatively larger number of categories (or speakers for the present case). Experiments conducted in the past have revealed that the training time required to train a single neural classifier to perform an M-way classification task is roughly quadratic in M. However, this training time may be reduced by partitioning the classification task, as a series of Y way decisions ($Y \geq 2$). It has further been shown that partitioned neural network classifiers require less training data compared to a single large network. Fig. 2.7 shows the block diagram for a typical Group Partitioned Speaker ID system.

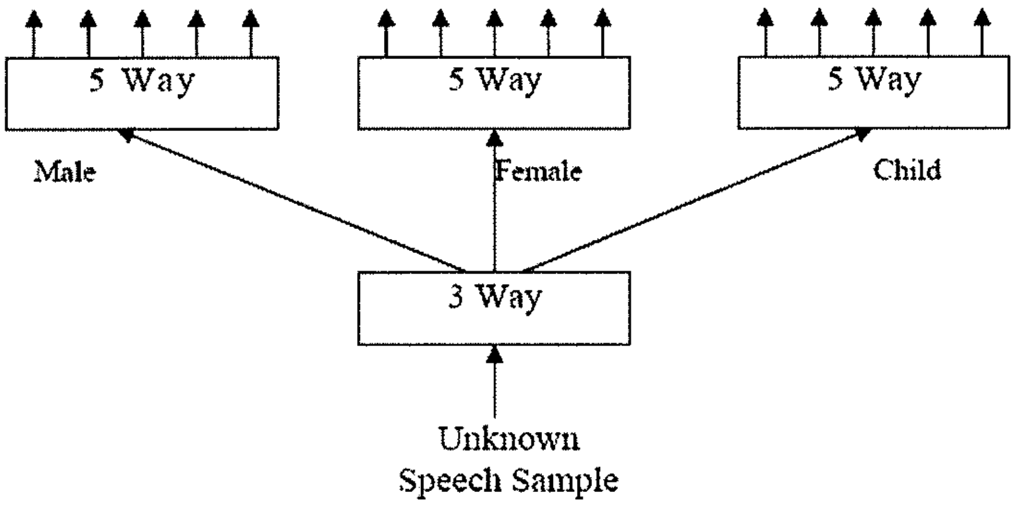

Fig. 2.7. Group Partitioning for Automatic Speaker Identification.

Two of the most distinct forms of partitioning of a classification task are group partitioning and pair-wise or binary pair partitioning (BPP). Group partitioning has been extensively exploited in several previous and current research studies. Nevertheless, it is worthwhile to mention that group partitioning can be successfully applied when the entire data can be broadly categorized into more than one category – for example, for humans (men, women, and children) or for medication (antihistamines, analgesic, etc.). Each broad category may or may not contain sub categories. On the other hand, binary-pair partitioning is more suitable when applied to different subsets belonging to a single category (Males male1, male2, male3...etc). Figure 2.8 presents a graphical depiction of an example for classification layers when group partitioning is employed.

A special case of group partitioning is binary-group partitioning – for which M-1 two-way classifiers are used to achieve M-way classification. The classification progress follows a tree path wherein each branch leads to only one of two possible alternative remaining categories. The performance advantage with such a partitioning is that as long as no errors are made at the preceding levels of the decision tree, no sub classifier needs to make a decision for an input sample it was not trained to classify. Nevertheless, this great benefit is severely affected by the need for a "good" partitioning of the categories for each sub-classifier.

2.2.1 Binary-Pair Partitioned Neural Networks

Binary-Pair Partitioning is a special type of classifier. It uses M*(M-1)/2 two-way classifiers to make an M-way decision. Each binary decision is made between a pair of categories or speakers. Thus, there are M-1 decisions relevant to each user in set M. For classification these decisions are combined to produce an overall decision. This may be visualized as a square matrix form, which shows elements corresponding to pairs of unique categories that can be formed. In this form, only the elements above the principal diagonal are relevant. The speaker pairs below the principal diagonal can be separated using the classifiers for pairs above the diagonal. For example, a classifier that can separate categories 1, 2 will also be able to separate categories 2, 1, which leads to the number of classifiers needed given by the expression above. This leads to a sharp growth in the number of classifiers as M increases. A square matrix representation of the BPP is shown in Fig. 2.8.

$$\begin{bmatrix} & 1,2 & 1,3 & 1,4 & \cdots & 1,M \\ & & 2,3 & 2,4 & \cdots & 2,M \\ & & & 3,4 & \cdots & 3,M \\ & & & & \ddots & \vdots \\ & & & & & M-1,M \\ & & & & & \end{bmatrix}$$

Fig. 2.8. Square Matrix Representation of BPP.

The greatest benefit that has been derived from BPP partitioned neural networks is exceptional performance, which is apparent when we consider that there is a dedicated classifier that has to differentiate between TWO categories only. Other advantages of BPP partitioning over group partitioning is that categories need not be grouped, which eliminates the need for arranging similar categories together prior to classification. Using a conventional single network for an M-way decision requires training time exponentially proportional to the number of users (M). Using the BPP network can reduce the training time to as little as log_2 (*M*). Implementation of the BPP network requires two steps in the classification process:

1. "Elemental" classifiers are trained to distinguish between every possible pair of speakers in set M (users).

2. Test data is then run through each elemental classifier trained in step 1. The decisions made in this process are combined to produce an overall decision based on all the binary-pairs.

In the identification problem we are faced with a fixed number of possible speakers, M. The identification task is to identify the present speaker as one of the M possible speakers. The verification task includes not only this simple classification, but the system must also be able to reject an infinite set of 'impostor' speakers. Thus, the speaker space consists of a set of acceptable speakers, M (users), and an infinite set of possible speakers, N, including N-M impostors. The verification task includes not only correctly identifying users, but also rejecting impostors. Therefore, the Binary-Pair Partitioning has been adapted for verification [4].

2.3 Combined GMM/NN

Based on the results of Gaussian Mixture Model Based Speaker identification and those of BPP NN based Speaker ID, a combination of both the models can be designed for better results. GMMs are trained with the maximum likelihood approach to give good models for each speaker. In contrast, NNs are trained to discriminate speakers but form no explicit models for each speaker. The hypothesis underlying this work is that an appropriate combination of maximum likelihood and discriminative training, using both GMMs and NNs should give a better recognition rate than the individual ones. However, it is not clear how to combine these two models. A few possibilities are presented in the remainder of this thesis, and experimental evaluations given.

This chapter has a discussion of the concepts of Gaussain Mixture Models, Neural Networks and the possibilities their combination of both for speaker identification. The next chapter explains the algorithms used for each method in more detail.

CHAPTER III

ALGORITHMS FOR GMM SPEAKER ID AND COMBINED GMM/NN SPEAKER ID SYSTEM

In this Chapter, the algorithms used for implementing a speaker ID system with Gaussian Mixture Models and a combined GMM/Neural Network model are given in detail. The first step in both of these methods is to extract the features from acoustic (typically, *.wav) files. For experimentation, we used "standard" widely distributed databases because such databases allow comparison of experimental results with other researchers, using other algorithms for feature extraction and/or recognition.

The databases used were NTIMIT (Nynex-Texas Instruments and MIT) consisting of 630 speakers, with each speaker reading 10 sentences and the SPIDRE database consisting of 45 speakers, each speaker having 4 conversations. In both cases, the speech was recorded over telephone lines.

3.1 Feature Extraction

The first step of speaker identification is feature extraction. For the work reported in this thesis, the extraction of the features was done using the ODU Speech lab program 'tfrontm', which computes Discrete Cosine Transform Coefficients from the waveform files. This front end feature extraction process results in feature files. The feature files were written with one file for each speaker, with each sentence considered as one token and with 25 variables (DCTC coefficients) for each frame, and the number of frames depending on the length of the sentence. Thus, for example, with 10 sec. of speech data from each speaker, there would be 1000 frames of data from each speaker.

Using this feature extraction process, each speaker has a feature file representing his/her acoustic information. Typically, each acoustic file is parameterized using:

- 25 DCTCs per frame (Features CC1~ CC25)
- 40 ms frames, spaced 10ms apart
- Kaiser windowing with β = 0/6
- A frequency warping factor of 0.25
- A frequency range of 300 ~ 4000 Hz

More details of the feature extraction process and feature file formats can be found in [3].

The low energy frames were also removed; removing all frames with CC1 values less than a certain threshold. From other work [24], for the case of the SPIDRE database, approximately, 90% of the total frames were removed saving only the top 10% (in terms of signal level) frames. For the case of NTIMIT, it was found empirically that the best choice was to remove approximately the lowest energy 30% of frames. Finally, a scaled file for each speaker was created, containing all the sentences for that speaker (10 for NTIMIT and 4 for SPIDRE).

These files are used by the GMM and NN algorithms for the speaker ID system.

3.2 Gaussian Mixture Model Algorithm for Speaker ID

As was discussed in chapter 2, a Gaussian Mixture Model can be mathematically represented as

$$p(x/\Lambda)=\sum_{i=1}^{M} w_i G(x/\mu_i,\Sigma_i) \qquad \textbf{(3.1)}$$

where $G(x,\mu_i,\Sigma_i)$ represents the Gaussian density function for the i^{th} mixture component and is given by the equation,

$$G(\mathbf{x}/\mu,\Sigma)=\frac{1}{|2\pi\Sigma|^{1/2}}e^{(-1/2(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu))} \quad \textbf{(3.2)}$$

and the weights are constrained by the equation

$$\sum_{i=1}^{M} w_i = 1$$ i.e. the sum of the weights is 1.

This algorithm generates a Gaussian Mixture Model for each speaker. The Model inputs are:

1. Number of mixture components
2. Number of data samples
3. Input Feature file
4. Number of iterations for the EM algorithm
5. Threshold for the log likelihood value

The outputs are these parameters:

1. Weights for each mixture
2. Mean vector for each mixture
3. Covariance matrix for each mixture

An overview of the algorithm is as follows:

1. The input feature file is opened for the data and the file is read according to the specifications given-- i.e. the number of tokens and the number of features to be read from the file.
2. The file with the corresponding specifications is read and the data is stored in a matrix.

3. The Gaussian Mixture Model for each speaker is determined using the EM-Expectation –Maximization algorithm. This can be summarized as follows:

   (i) The dimension of the input data is determined.

   (ii) Initialization for the EM algorithm is done, i.e. the mean vectors, covariance matrices and weights are initialized. Also the initial likelihood values are assumed. Note, as mentioned in chapter two, previous researchers have noted that GMMs for speaker ID generally perform better using many mixtures, but only diagonal covariance matrices.

   (iii) The density function for the set of parameters is determined.

   (iv) The log likelihood value for the pdf is calculated

   (v) If the difference of the current log likelihood value with the previous value is less than the threshold value, the algorithm terminates. Otherwise Steps (iii) through (v) are repeated until the number of iterations is equal to the value specified.

Thus a Gaussian Mixture Model for each speaker is obtained. This is the training phase for the Speaker ID system. Once these models are obtained, the testing has to be done for recognition. This is done by the evaluation algorithm discussed below.

3.3 Evaluation of the Gaussian Mixture Models

This algorithm works as a testing phase for the speaker ID system. Here the input is the test speaker and the output is the likelihood value of the test speaker with each of the speakers in the database. The inputs and outputs for this algorithm are given below:

Inputs:

1. Test speakers feature files
2. The Gaussian Mixture Models for each speaker in the database. (Note that these models are obtained with different sentences than those in the test set.)

Outputs:

1. The likelihood value of each test speaker with respect to each trained speaker model in the database
2. The overall recognition rate of the evaluation system

The algorithm can be summarized as follows:

1. The input feature file of the test speaker is read according to the specifications given, and the data is stored as a matrix.
2. The test data is then compared with the training data from the database, and the log likelihood value of the test data with each speaker is calculated. Note that the calculation is done by first evaluating each frame of data and summing frame scores. The speaker having the maximum likelihood value is identified as the test speaker.
3. Based on the likelihood values the recognition rate of the system is calculated.

This algorithm gives the recognition rate of the speaker ID system based on the likelihood values of the input speaker with respect to each speaker in the database. The speaker having the maximum value of likelihood is chosen as the test speaker.

3.4 Algorithm for Neural Network based speaker ID

Neural networks used in this work have the following architecture:

- Three layers (input, hidden and output)

  Number of input nodes – 24

  Number of hidden nodes – 10

  Number of output nodes – 1

- Feed forward, Memory less
- Sigmoid non-linearity at the nodes $f(x) = \frac{1}{1+e^{-x}}$
- Back-propagation training method

This classifier performs only 2-way decisions. Thus, the classification problem involving M speakers is divided into classifications involving only 2 speakers. All these 2-way classifiers obtained can be considered as elements of a square matrix as shown in Fig. 2.8.

The element <*g, h*> represents the output of a classifier, trained to separate categories g and h only. Each classifier is a neural network trained over categories *g, h | output (g) ~=0 & output (h) ~=1.*Each of the elements in the lower diagonal can be computed as $x_{ij}=1-x_{ji}$ *(binary output).*

The classifier works as follows:

- The unknown parameterized acoustic sequence is passed through all the 2-way classifiers.
- The outputs of all these classifiers are stored at the respective matrix location.
- The row-wise sum of the matrix is computed.
- The row index with the highest sum is used as the index for the recognized speaker.

The neural network is trained to attempt separation of feature frames of one speaker with those of another speaker. In particular, the neural network has an output target of '0' (low) for one speaker and '1' (high) for the other speaker. With this approach and for "clean" speech, typically about 80-95% of frames can be separated for the TIMIT database (using the natural separation level of 0.5 for the neural network output). However, since actual decisions are made by averaging the neural network output over the entire utterance, nearly 100% accuracy is easily obtained for each 2-way classifier.

3.5 GMM combined with Neural Network

The approach for combining Gaussian Mixture Models and Neural Networks for speaker identification is discussed in this section. Two methods have been designed and implemented for the combined GMM/NN speaker identification model.

3.5.1 Method 1- Universal Background Model Approach

The first method employed for the combined GMM/NN model is derived from the Universal Background Model (UBM). In the UBM method, a large GMM is obtained from all the speakers or a group of speakers in the database and this universal model is used as the basis for speaker recognition. This "large" GMM is intended to have enough mixture components to represent any speaker. The UBM approach has recently become the basis for virtually all state-of-the-art Speaker Verification systems [20]. However, it has not been used much for speaker identification.

The first step is to form a single universal Gaussian Mixture Model with a large group of speakers. The UBM is trained with speakers not used for the actual identification, and is intended as a generic model for the universe of speakers. For the speaker verification task, two speech samples are "presented" to the system. The task of the verification system is to determine whether or not the two samples are from the same or different speakers. For the UBM approach, the system determines whether the two samples are more similar to each other or to the UBM.

In this thesis, we use a variant of the UBM approach for speaker identification for the combined GMM/NN method. The specific algorithm is explained below.

Training Phase

1. A group of speakers from the database is used to obtain a universal background Gaussian Mixture Model with many mixtures. This method was only tested using the NTIMIT database and a group of the speakers from the database for creating the model. The parameters obtained, i.e. the weights, means and variances, were stored in a file called **UBM**.
2. The remaining speakers from the database are used to create speaker specific Gaussian Mixture Models. Ideally these models should be created by adapting the UBM models, as mentioned in [20]. However, due to the complexity of this adaptation process, in this work, the speaker specific models were created independently. Specific training tokens were used to create these models. These parameters, i.e. the mean vectors and variances were saved as the speaker specific model. Next, UBM mixture components close to each speaker-specific component were computed. That is, the distances between the means of each mixture in the UBM and each mixture of each speaker specific model are calculated and the mixtures that are close are obtained.

3. a) Using the same training sentences, compute the contribution of log likelihoods from each mixture. Then, determine the P highest values for each frame. These values can be viewed as new features for each speaker.

   b) Evaluate the corresponding mixture log likelihood values for the UBM, using the correspondence table described above. Again obtain the highest P mixture likelihoods. Again, write these values in a feature file with the values obtained as the data for each frame in the feature file.

   Thus, two "feature" files for each are obtained for each speaker. One file is the representation of the speaker with respect to the training GMM for that speaker. The other file represents how the speaker appears when evaluated by the UBM, but with the correspondence linkage as described above.

4. Run the Neural network-BPP algorithm on the above obtained files to train a neural network to discriminate the features for each pair of files mentioned in step 3. Ideally, this network should be able to determine all the UBM features that are different than the speaker-specific GMM features. It was hypothesized that the percentage of actual frames of UBM features "properly" discriminated would be a good indicator for recognizing each speaker.

Test Phase

1. Beginning with the acoustic feature files, use the test sentences for each speaker to create the two sets of files as described above for the training data.
2. Again run the neural network-BPP algorithm on the files obtained in this test phase in the same way as done in the training phase, i.e., take the files obtained from the UBM for training and the files obtained from the speaker specific

models for testing. Obtain the recognition rate. It was anticipated that this rate would be very similar to that obtained in the training phase for that speaker.

3. Compare the recognition rates obtained in both the training and test phases.

The steps described above were implemented and evaluated for the NTIMIT database using 102 speakers (DR2) with 8 sentences for training and 2 sentences for testing. If the recognition rates obtained for the training data and test data for each speaker were very similar, it would have been assumed that the feature files obtained above would be good indicators of speaker identity, and additional NN processing steps would have been implemented to make the final identification. However the recognition rates obtained for training and test data did not match well at all. Therefore, this method was neither fully implemented nor thoroughly tested. It appeared that this particular approach for combining the maximum likelihood and likelihood discriminative estimates was not promising and not worthwhile to investigate in more detail. Nevertheless, it was one method that was implemented and evaluated as a first attempt to combine GMMs and NNs.

3.5.2 Method 2-Speaker Score Averaging

In this method, which is quite similar to methods used in other applications for combining results of two classification methods, the speaker scores calculated from both the methods, i.e. the Gaussian Mixture Model approach and the Neural Network method are averaged. That is for the GMM method, the likelihood scores are computed for each possible speaker. Similarly for the BPP NN, a score is computed for each possible speaker. In both cases, these scores can be (and were) normalized to a 0-1 range. The scores can then be added and averaged to obtain an overall score to identify the most likely speaker. It was hoped that the composite score, based on two quite different methods, would result in higher accuracy than the score based on either method alone.

This process thus consists of the following steps:

1) Run the GMM algorithm on the training data and obtain the speaker specific models for each speaker.
2) Evaluate the test data on the speaker specific models and compute the log likelihood values.
3) From the log likelihood values, obtain the speaker scores for each test speaker. Use an offset and linear scaling to normalize these scores to a 0-1 range.
4) Run the Neural Network-BPP on the same data.
5) Obtain the speaker scores for each test speaker using the Neural Network also. Note that these scores are normalized from the NN to be in a range of 0 to 1.
6) Average the speaker scores for all the speakers obtained from both the methods.
7) The speaker having the highest ranking is identified as the test speaker.

As mentioned above, it was hoped that this score based on averaging would result in higher speaker identification accuracy than the score based on either method alone.

Summary

The various methods used for speaker identification in this work have been discussed in this chapter. The main focus was the Gaussian Mixture Model algorithm. The experiments and the results obtained using these methods are explained in more detail in the next chapter.

CHAPTER IV

EXPERIMENTS AND RESULTS

This Chapter includes all the experiments conducted for this thesis. It begins with a description of the databases used for experimentation. The chapter includes a detailed description of all the experiments conducted. It also includes the results of all the experiments conducted and it gives a comparison between the results wherever applicable.

The experiments for the Automatic Speaker Identification system are based on the following approaches:

1. Neural Networks with Binary-pair Partitioning
2. Gaussian Mixture Models
3. Combination model of Gaussian Mixture Models and Neural Networks using Binary-Pair partitioning

These experiments were conducted on two standard databases, NTIMIT and SPIDRE, as summarized below. These experiments were conducted on standard databases so as to allow effective comparison of the results obtained with those obtained by other researchers and other algorithms.

4.1 NTIMIT Database

The TIMIT database was provided by a joint effort between Texas Instruments and MIT, and was developed primarily to aid acoustic phonetic speech recognition research. NTIMIT is a variant of TIMIT, consisting of the *identical* acoustic files, but re-recorded from telephone lines. The databases consist of 630 speakers from 8 dialect regions (DR 1~8) with 10 sentences per speaker, as listed below and summarized in Table 4.1.

(a) 2 SA sentences – meant to highlight the dialect variance of each speaker,

(b) 3 SI sentences – phonetically diverse selected form existing texts,

(c) 5 SX sentences – phonetically compact and meant to provide good and repeated occurrences of pairs of phones. "SA" consists of 2 (same sentences for all speakers) sentences per speaker which are considered as SRI dialect calibration sentences; "SI" consists of 3 (different) sentences per speaker which are referred to as TI (Texas Instruments) random contextual variants sentences; and "SX" consists of 5 (different) sentences per speaker, which are the MIT phonetically compact sentences. The Dialect Regions and the numbers of speakers in each region are:

Table 4.1. Summary of the TIMIT and NTIMIT Databases

Dialect region	Dialect	Training Speakers	Test speakers
DR1	New England	38	11
DR2	Northern (26 speakers)	76	26
DR3	North Midland (26 speakers)	76	26
DR4	South Midland (32 speakers)	68	32
DR5	Southern (28 speakers)	70	28
DR6	New York City (11 speakers)	35	11
DR7	Western (23 speakers)	77	23
DR8	"Army Brat" (11 speakers)	22	11
Summary	8 Regions	462	168

Experiments in this thesis were mainly conducted with speakers from dialect region 2.

4.2 SPIDRE Database

This corpus is a 2-CD subset of the Switchboard-I collection selected for speaker ID research, with special attention to telephone instrument variation. Combining the two sides of the conversations also permits speaker change detection or speaker monitoring experiments. There are 45 target speakers (27 male, 18 female); four conversations from each target are included, of which two are from the same handset. Since all conversations are two-sided, this results in 180 target recordings. Typically, conversations were about 5 minutes. Thus, there are 180 wave files, four for each of the 45 speakers, saved in uncompressed wave format (8 kHz sampling rate, 16 bits per sample).

An important aspect of the SPIDRE database is that two of the conversations are over the same telephone line, and two of the conversations are over different telephone lines. Thus a total of three telephone lines are used. In all of our experiments, as reported below, three conservations were used for training, and one conversation was used for testing. For the channel matched case, the test conversation is one of the conversations from the set of two recorded over the same phone line. For the channel mismatched case, the test conversation is one of the conversations recorded over a "unique" telephone line. All experiments done in this thesis were on the mismatched case.

4.3 Signal Processing for NTIMIT

For most of the experiments conducted with the NTIMIT database, the 102 speakers from DR2 were used. For most of the experiments, eight sentences were used for training and two sentences were used for testing. Features used were 25 Discrete Cosine Transform Coefficients (DCTC0 to DCTC24) of the log magnitude spectrum for each frame. The feature parameter values for these experiments were (frame length= 30ms, frame spacing = 10ms, Kaiser window, beta of 6, FFT length = 1024, warping factor = 0.45, frequency range of 50 Hz to 3950 Hz, high frequency pre-emphasis with center frequency at 3200 Hz). Typically, about 2400 feature vectors were used for training for each speaker, and about 600 feature vectors were used for testing.

4.4 Signal Processing for SPIDRE

In all experiments 25 Discrete Cosine Transform Coefficients (DCTC0 to DCTC24) were computed for each speech frame as follows. First, a second order high frequency pre-emphasis filter with a broad peak around 3 kHz was applied to the speech signal. The second step was to compute a 1024 point FFT from each 30 ms Kaiser-windowed (coefficient of 5.33) frame of speech data with the window advanced by 10 ms. The following step was to compute the amplitude spectrum, logarithmically scale it, and then frequency warp it with a bilinear function using a coefficient of .25. The next step was to compute the 25-DCTC coefficients as the cosine transform of the scaled magnitude spectrum over the frequency range 25 Hz to 3900 Hz.

The next step in signal processing was removal of low energy frames. DCTC0, computed as described above, was used as the energy measure. For each sentence, the frames were rank ordered in terms of energy. Using two thresholds (expressed as percentages), thresh_low, and thresh_high, the lowest energy and highest energy frames were removed. The motivation for the low energy frame removal is that the lowest energy frames are generally silent or very noisy. The motivation for the high-energy threshold is that the highest energy frames may contain signal distortion due to clipping, or may represent non-speech signals. For example, after listening to many of these conversations, it was observed that many of the largest amplitude portions of each file were due to coughs or laughs.

The next step in signal processing was to perform feature mean and/or variance normalization. For each feature, for each recording, the mean μ, and variance, σ^2 were computed. Then features were rescaled according to

$$y = (x- \mu)/ \sigma \tag{4.1}$$

Such normalization has been shown to remove much of the variability due to channel transfer function, noise, and nonlinearity differences. Another step in signal processing was to again perform mean and variance normalization, but based on the entire database of all speakers. Additionally, the data was typically linearly scaled so that the resultant final standard deviation was .2, or thus an overall range of +-1.0 for each feature, as this range has typically been found to result in best performance for neural network training with bipolar (-1 to +1) sigmoidal nonlinearities.

4.5 Experiment 1: Speaker Identification using Binary-Pair Partitioned Neural Networks

The experiments on the NTIMIT database were conducted using 102 speakers of the total 630 speakers with 8 sentences for training and 2 sentences for testing. For this experiment, 20 DCTC features were used for both training and testing.

The experiments with the SPIDRE database were conducted using all 45 speakers of the database, using 3 sentences for training and 1 sentence for testing. The number of DCTC features used for the SPIDRE experiments was 25. Results for these experiments are shown in Table 4.2.

Table 4.2. Speaker Identification Accuracy for 102 speakers of NTIMIT and 45 speakers of SPIDRE using BPP Neural Networks

Database	Training performance	Test performance
NTIMIT	95.2%	73.0%
SPIDRE	98.2%	80.0%

4.6 Experiment Set 2: Speaker Identification using Gaussian Mixture Models with Full Covariance Matrices

This section outlines various experiments conducted using Gaussian Mixture Models with full covariance matrices and summarizes the results obtained for these experiments. These experiments, as were the ones reported in the previous section, were conducted on the two standard databases NTIMIT and SPIDRE.

4.6.1 Results using the NTIMIT Database

4.6.1.1 Results for 102 Speakers from the NTIMIT Database

For the NTIMIT database, first the experiments were conducted with the 102 speakers of DR2, with 8 training tokens and 2 test tokens. Speaker identification accuracy was examined as the number of mixture components of the GMM was varied from 1 to 25. The GMM was configured as a full covariance matrix. The number of DCTC features was 25, the maximum number tested in the course of this research. The training and test performances were then obtained with the above parameters.

Table 4.3. Speaker Identification Accuracy for 102 Speakers of NTIMIT using a GMM with Full Covariance Matrix and 25 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	89.2%
2	100%	88.2%
3	100%	87.3%
4	100%	87.3%
5	100%	83.3%
6	100%	81.4%
7	100%	75.5%
8	100%	79.4%
9	100%	70.6%
10	100%	69.6%
11	100%	62.7%
12	100%	61.8%
13	100%	59.8%
20	100%	45.0%
25	100%	43.1%

From the table above, it can be observed that the training accuracy is 100% for all the cases as the number of mixtures is increased from 1 to 25. Unlike the training accuracy, the test accuracy was highest for 1 mixture component at 89.2%, and it kept decreasing as the number of mixture components was increased. The number of parameters to be computed for this experiment is M * (1+N+N*N), where M is the number of mixture components and N is the number of features used. Thus, for example, with 10 mixture components and 25 features, the number of parameters that must be estimated by the EM algorithm is 6510.

The experiment was repeated with all the conditions identical, except only 20 DCTCs were used (as opposed to 25 in the previous case), and the number of mixtures was only varied from 1 to 8. This case was considered so as to reduce the number of parameters to be computed and therefore reduce the execution time. Note that the number of parameters to be estimated using the full covariance matrix method (6510 for a typical case as mentioned above) was likely too large for the size of the database (about 1800 training frames.) The results, based on 20 features, are given below in Table 4.4.

Table 4.4. Speaker Identification Accuracy for 102 Speakers of NTIMIT using a GMM with Full Covariance Matrix and 20 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	82.3%
2	100%	87.2%
3	100%	87.2%
5	100%	89.2%
8	100%	78.4%

Even if 20 features were used instead of 25 features for the speaker identification process, the training accuracy remained the same and the test accuracy was also comparable to the one obtained with the experiments using 25 features. The highest accuracy obtained in this case was also 89.2% for 5 mixture components. Therefore, it can be concluded that even if the feature size is reduced, the same accuracy can be obtained using more mixture components. However, the number of parameters to be computed and the time of execution for 5 mixture components with 20 features would be much more than the number of parameters to be computed and the time of execution for 1 mixture component with 25 features.

Then the number of test sentences was changed to 1, keeping the number of training sentences as 8 and the number of DCTC features as 25. This case was to evaluate the case when the amount of test data gets reduced. The results obtained are given below in Table 4.5.

Table 4.5. Speaker Identification Accuracy for 102 Speakers of NTIMIT using a GMM with Full Covariance Matrix and 25 DCTC Features (1 test sentence)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	75.5%
2	100%	80.4%
3	100%	78.4%
4	100%	80.4%
5	100%	81.4%
6	100%	76.5%
7	100%	73.5%

The above table allows us to conclude that the maximum test accuracy that can be obtained when only 1 sentence is used for testing instead of 2 is 81.4% as compared to the 89.2% obtained in the previous cases where 2 sentences were used for testing. The training accuracy is the same as in the previous cases--100%.

The above chart shows that the best case results are obtained when all 25 features are used and 2 test tokens are used. When all the 25 features are used, the best results are obtained for 1 mixture component. While 20 features are used, the best results are obtained for 5 mixture components.

The time required for the execution was quite reasonable. In particular, it took approximately 4 minutes for the code to run for the case of 1 mixture component. Then as the number of mixture components was increased, the time required for the execution increased almost linearly with the number of mixture components, until the number of mixture components was increased to a large number like 25, for which the execution time started increasing rapidly.

4.6.1.2 Results for 630 Speakers from the NTIMIT Database

As the Gaussian Mixture Model approach took far less execution time than the BPP speaker ID system, some experiments were also conducted using all the 630 speakers present in the NTIMIT database and the results were obtained. For the BPP Neural Network to perform this experiment using all the 630 speakers would take more than 1 day, using the type of computers used for the GMM experiments.

For the first experiment, the number of training sentences was 8 and the number of test sentences was 2. 25 DCTC features were used. Experiments were conducted with 1 and 2 mixture components and a full covariance matrix. Results are given in Table 4.6.

Table 4.6. Speaker Identification Accuracy for 630 Speakers of NTIMIT using a GMM with Full Covariance Matrix and 25 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	59.4%
2	100%	58.3%

The number of training and test sentences was changed to 5 training and 5 test sentences, and the same experiment was conducted with 25 DCTC coefficients and varying the number of mixture components. Results are given in Table 4.7.

Table 4.7. Speaker Identification Accuracy for 630 Speakers of NTIMIT using a GMM with Full Covariance Matrix and 25 DCTC Features (5 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	59.2%
2	100%	53.9%

The feature files used above were then scaled using the SCALE program and the experiment was conducted with 8 training sentences and 2 test sentences and 25 DCTC features. The scaling was performed so that the mean of each feature becomes 0, and the standard deviation .2. In this experiment the number of mixtures was varied and the number of features was kept at 25. The best result obtained for this case was 63.01% using 2 mixture components. The results obtained after scaling the feature files are better than the results obtained without scaling.

The following steps were included in the scaling process and the generation of the new set of feature files:

a. energy threshold to remove low energy frames

b. Scaling to have an overall mean of 0.0 and a standard deviation of 0.2.

Table 4.8. Speaker Identification Accuracy for 630 speakers of NTIMIT using a GMM with Full Covariance Matrix and 25 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	61.9%
2	100%	63.0%

4.6.2 Results using the SPIDRE Database

For the SPIDRE database, the experiments were conducted on all the 45 speakers in the database with 3 training tokens and 1 test token by varying the number of mixture components. The number of features used in this experiment was 24 out of the 25 features (removing DCTC0) in the feature file. The results obtained from this experiment are shown below. These are the results for the mismatched headsets group of the SPIDRE database.

Table 4.9. Speaker Identification Accuracy for 45 Speakers of SPIDRE using a GMM with Full Covariance Matrix and 24 DCTC Features (1 test sentence)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	80%
2	100%	77.8%
3	100%	82.2%
4	100%	84.4%
5	100%	84.4%
6	100%	76.5%

From the table shown above, the maximum recognition rate that can be attained using full covariance matrix based Gaussian mixture models on the SPIDRE database is 84.4% corresponding to 4 mixture components.

The recognition rate increases as the number of mixtures is increased to 5 mixture components, but decreases as the number of components is more than 5. This accuracy is higher than the 80% accuracy obtained when the Binary-pair partitioned neural network method is used.

4.7 Experiment Set 3: Speaker Identification using Gaussian Mixture Models with Diagonal Covariance Matrices

The results of the previous section indicate that GMM performance might have been limited due to the very large number of parameters to be estimated and relatively small amount of data. This hypothesis of inadequate data for the number of the parameters to be estimated is also supported by the 100% accuracy obtained with training data, and much lower accuracy obtained with test data.

The Gaussian components act together to model the overall probability density function. However, full covariance matrices are not necessary even if the features are not statistically independent. In particular, a linear combination of diagonal covariance

Gaussians is capable of modeling the correlations between feature vector elements. As noted in a previous chapter, other researchers have noted that speaker ID systems often perform better using only diagonal covariance models.

Considering the discussion in the previous two paragraphs, additional experiments were conducted with diagonal covariance matrix based GMMs, so that the number of parameters to be computed would be fewer than the number required by full covariance matrix based GMMs.

4.7.1 Results using the NTIMIT Database

These experiments were conducted on NTIMIT database with 8 training sentences and 2 test sentences. The results on this set of experiments are given below in Table 4.10.

Table 4.10. Speaker Identification Accuracy for 102 Speakers of NTIMIT using a GMM with Diagonal Covariance Matrix and 25 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
12	100%	82.3%
25	100%	86.3%
40	100%	88.2%
50	100%	82.4%
60	100%	84.3%

Here the number of features was kept constant as 25 and the number of mixture components were varied. From the table, it can be observed that the highest recognition rate is obtained for 40 mixture components. Thus the best result using the diagonal covariance based GMM on the NTIMIT database is 88.2% for 40 mixture components. This is slightly less than the 89.2% obtained with the full covariance GMM. However, the diagonal covariance GMM is potentially advantageous due to the much simpler model and possible reduction in the number of parameters to be computed.

Then, the number of features, i.e. the DCTC coefficients, is changed to 20, and the experiment is carried out in the same way keeping the training sentences as 8 and the test sentences as 2. The number of mixture components is varied and the results are listed below in Table 4.11.

Table 4.11. Speaker Identification Accuracy for 102 speakers of NTIMIT using a GMM with Diagonal Covariance Matrix and 20 DCTC Features (2 test sentences)

No. of mixtures	Training Accuracy	Test Accuracy
1	100%	69.6%
2	99.0%	77.5%
5	100%	76.5%
10	100%	85.3%
20	100%	84.3%
30	100%	87.3%
40	100%	85.3%
50	100%	82.4%
60	100%	82.4%

The above table shows that 87.3% is the highest possible recognition accuracy that can be obtained when diagonal covariance GMMs are used with 20 features. When the same experiment is conducted using full covariance GMMs, the accuracy obtained is 89.2%. Note that the comparable experiment using the BPP-NN method results in 73.0% accuracy.

Overall, the experiments reported thus far in this chapter indicate that the GMM with a full covariance matrix and a small number of mixtures gives results very similar to those obtained with a diagonal covariance matrix and a large number of mixtures. Results obtained with a BPP-NN are much lower. The highest results among the three cases were obtained with the GMM and full covariance matrix.

4.7.2 Results using the SPIDRE Database

Speaker identification experiments with diagonal covariance matrices were also performed on the SPIDRE database of 45 speakers. The number of training sentences used was 3 and the number of test sentences was 1 and all the 24 DCTC features (removing DCTC0) were used. The number of mixture components was varied and the results are given in Table 4.12.

The results obtained using the SPIDRE database with diagonal covariance matrix based GMM were more promising then the results obtained with full covariance matrix based GMM. The best result obtained with the diagonal covariance matrix was 86.7%, as compared to the 84.4% obtained with full covariance GMMs. The number of mixture components used with the diagonal covariance based GMMs (best results with 40 mixture components) is much more than the number of mixture components used with the full covariance based GMMs (best results with 4 mixture components). However, it gives better results and the number of parameters to be computed is slightly fewer (2040 for the best result case) than when full covariance GMMs are used. (2064 for the best case)

Table 4.12. Speaker Identification Accuracy for 45 Speakers of SPIDRE using a GMM with Diagonal Covariance Matrix and 24 DCTC Features (1 test sentence)

No. of mixtures	Training Accuracy	Test Accuracy
5	100%	77.8%
10	100%	84.4%
20	100%	84.4%
30	100%	86.7%
40	100%	86.7%
50	100%	84.4%

4.8 Experiment Set 4: Speaker Identification using Combined GMM/NN with Diagonal Covariance Matrices

The next set of experiments was conducted using the combined GMM and Neural Network approach. In chapter 3, two methods were discussed for the combined GMM/ Neural Network approach. One is the UBM method, and the second one is the likelihood averaging method. As reported in chapter 3, the UBM method did not give good results in the pilot experiments and was therefore dropped. The results obtained by the likelihood averaging method were more promising. The experimental technique for this approach is as follows:

1. The Gaussian Mixture Model algorithm is applied on the speaker feature files, and the parameters of the weights, means and variances are estimated.
2. The test data is evaluated with the above obtained models, the log likelihood values for each speaker with respect to every other speaker are computed, and the speaker score for each speaker are determined and stored.
3. Then the Neural network is used for speaker identification, and the speaker scores are computed for this method.
4. The corresponding scores for each speaker are averaged, and the speaker with the highest total score is chosen as the speaker.

These experiments were conducted using only the SPIDRE database. The results obtained are shown below. The experimental configurations for the GMM and the NN are the same as those reported in previous sections of this chapter. The GMM was implemented with a diagonal covariance only. 24 DCTC features were used. Results obtained are given in Table 4.13.

Table 4.13. Speaker Identification Accuracy for 45 Speakers of SPIDRE using a Combined GMM/NN with Diagonal Covariance Matrix and 24 DCTC Features (1 test sentence)

No. of mixtures for GMM	GMM results	NN results	Combined GMM/NN
6	75%	80%	80%
7	77.8%	80%	80%
9	75.6%	80%	77.7%
30	86.7%	80%	84.4%
40	86.7%	80%	84.4%
60	84.4%	80%	82.2%

The results obtained with this combined GMM/NN approach approximate an average of the results obtained with each method alone. However, since the GMM with a large number of mixtures appears nearly always to be superior to the BPP-NN, the combined approach is actually not advantageous. Thus, although far better than the combined GMM/NN method presented in chapter 3, this approach still does not appear to be worthwhile.

Summary

This chapter gives a detailed explanation of the various experiments conducted and the results obtained with these experiments. It can be observed from the results that the GMM performs far better than the Neural Networks or the combined GMM/NN for speaker ID. The next chapter gives a conclusion of the work done in this thesis and the work that can be done as an extension of this.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In this thesis, the main aim was to investigate a combination of Gaussian Mixture Models and Neural Networks for Speaker Identification system. GMMs are trained with the maximum likelihood approach to give good models for each speaker. In contrast, NNs are trained to discriminate speakers but form no explicit models for each speaker. An appropriate combination of maximum likelihood and discriminative training, using both GMMs and NNs should give a better recognition rate than either method alone.

A comparison of three main algorithms used for speaker identification was also done in this work--the Neural Network Binary Pair-Partitioning method, the Gaussian Mixture Model method using full covariance matrices, and the Gaussian Mixture Model method using diagonal covariance matrices. The general concepts for combining Gaussian Mixture models and Neural Networks for Speaker Identification were discussed. Two methods were developed in some detail and implemented, with only one method yielding reasonable results.

A summary of the results obtained in this work is as follows. The Neural Network- Binary-Pair Partitioning Speaker Identification was performed on the NTIMIT database with 102 speakers. In the database, each speaker produced 10 sentences. For this experiment, 25 DCTC features were computed for each frame of speech, 8 sentences were used for training, and the remaining 2 sentences were used for testing. The recognition rate obtained for this experiment was 95.2% for training and 80.3% for testing. The Neural Network-Binary Pair Partitioning was again used for speaker identification using the SPIDRE database, which has 45 speakers with 4 sentences per speakers. These *.wav files were also converted to feature files with 25 DCTC features for each frame. Out of 4 sentences for each speaker, 3 sentences were used for training and 1 sentence for testing. The best training performance was 98.2% and the best test performance was 80%.

Another series of experiments was performed with the Gaussian Mixture Model method. Various cases were tested with the NTIMIT database. Keeping all the parameters, such as the number of DCTC features, the number of training sentences and the test sentences constant, the number of mixture components was varied. Two variations of the Gaussian Mixture Model were tested. One is the full covariance method and the other is the diagonal covariance method.

The best results obtained when all the 25 DCTC features were used for 102 speakers were 100% for training data and 89.2% for test data. This result was obtained for 1 mixture component used with the full covariance matrix. If 20 DCTC features are used instead of 25 features, and all the other parameters unchanged, the best training performance is 100% and the best test performance is 89.2%, which is obtained when 4 mixture components are used with the full covariance matrix. While using the diagonal covariance matrix for the GMM, the best result obtained was 100% for the training data and 88.2% for the test data when using 25 DCTC features and 8 training and 2 test sentences. This result was obtained with 40 mixture components. Using 20 DCTCs as features, the best training and test results were 100% and 87.2% respectively using 30 mixture components.

Additional tests were performed with all 630 speakers using the Gaussian Mixture Model algorithm only. The Neural Network Method was not tested because of the very long computational time required. Two experiments were performed with the GMM. For the first case, 25 DCTC features were used with 8 training sentences and 2 test sentences. The best test result obtained was 59.3% using one mixture component and a full covariance matrix. The setup for the second case was identical except the number of training and test sentences were each 5. The best test result for this case was 58.24% again with 1 mixture component and the full covariance matrix.

These feature files for all the 630 speakers were then scaled so that each feature, after scaling, had a mean of 0.0 and a standard deviation of 0.2. The GMM speaker identification was then again tested with these scaled feature files. The best test result for this set of files was 63.0% for 2 mixture components. A full covariance matrix, 25 DCTC features, 8 sentences for training and 2 sentences for testing were used.

The SPIDRE database was used for testing the GMM speaker identification again. The best result obtained with 4 mixture components was 84.4%. This was obtained with 25 DCTC features, and 3 training sentences and 1 test sentences. As mentioned, all results were obtained using full covariance matrices.

The experiments were performed again using diagonal covariance matrices and the Gaussian Mixture Models. Results are as follows. Using 102 speakers of NTIMIT with 25 DCTC features and 8 training sentences and 2 test sentences, the best training performance was 100% and the best test performance was 88.2% for 40 mixture components. Using 20 DCTC features instead of 25 and 8 training and 2 test sentences, the best result was 100% for the training data and 87.2% for the test data, again with 40 mixtures. With the SPIDRE database, the best result obtained with 25 DCTC features, 3 training, and 1 test sentences was 86.7%. This result was obtained for the case of 30 mixture components.

The combined Gaussian Mixture Model and Neural Network approach was then tested using the SPIDRE database. Here, the speaker scores obtained with both methods were averaged and the speaker having the highest overall score was identified as the speaker. The best result obtained here was 84.4% for 40 mixture components when 25 DCTC features were used and 3 sentences for training and 1 sentence for testing.

Below is a summary table showing a comparison of all the methods compared in this work and the best results obtained with all the methods in each case. The table shows that the GMM performs far better than the Neural Network BPP method. The GMM system gave a result of 89.2% for the NTIMIT using 102 speakers as compared to the 73.0% of the Neural Network speaker identification system. Additionally, the computational time required was far less for the GMM than the neural network method. So, in any case, the GMM can be said to be performing extremely well. In tests with the SPIDRE database, again, the results obtained with the GMM speaker identification system were comparatively better than the ones obtained with the Neural Network approach.

Here, the diagonal covariance based GMM showed the best accuracy of all the methods used. The diagonal covariance GMM gave an accuracy of 86.7% as compared to the 84.4% of the full covariance GMM and the 80% for the Neural Network Speaker ID. This result is in accordance with the theory that diagonal covariance based GMMs using a large number of mixtures show very good performance [25]. In the table below, the results taken from the literature are also given for comparison. The results for the case of NTIMIT (630 speakers) are from Douglas A. Reynolds' paper on Speaker Identification and Verification using GMMs [27]. In this paper, only 1 sentence is used for testing instead of 2 sentences as used in the experiments conducted in this thesis work. The results for the case of SPIDRE are taken from the previous tests conducted by Ezzaidi and Rouat[29].

Table 5.1. Summary of the Best Results Obtained by Various Methods

Methods used	Neural Network	GMM(full covariance)	GMM(diagonal covariance)	Combined GMM/NN	Results from literature
Cases					
NTIMIT(102)	73.0%	89.2%	88.2%	---------	-------
NTIMIT(630)	--------	63.0%	--------	---------	60.7%
SPIDRE	80.0%	84.4%	86.7%	84.4%	85.0%

Unfortunately, the combined Gaussian Mixture Model and Neural Network approach did not yield good results. Three approaches of combining the GMM and NN were proposed and tested, of which only one method showed reasonable results. However, these results were not comparable to the results of the Gaussian Mixture Model method alone.

Thus, the idea of combining the Maximum Likelihood Classification of the Gaussian Mixture Model and the Discriminative training of the Neural Networks did not work well for the various approaches proposed and tested in this thesis.

Future Work

The work done in this thesis can be extended to Adapted Gaussian Mixture Models suggested by Douglas A. Reynolds [20]. This approach is based on creating a single universal background model for all or a group of speakers in the database and a form of Bayesian adaptation to derive speaker models from the universal background model. The basic idea here is to derive speaker models by updating the well-trained parameters in the

UBM via adaptation. This estimation process is similar to the Expectation-Maximization method. The first step here is the same as for the EM algorithm, where the estimates of the sufficient statistics of the speaker's training data are computed for each mixture in the UBM. However, in the second step, unlike the EM algorithm, the new sufficient statistics are combined with the old sufficient statistics from the UBM mixture parameters using a data dependent mixing coefficient. The data-dependent mixing coefficient is designed so that mixtures with high counts of data from the speaker rely more on the new sufficient statistics for final parameter estimation and mixtures with low counts of data from the speaker rely more on the old sufficient statistics for final parameter estimation. This is an approach where the Gaussian Mixture Models can be improved for better performance of speaker identification systems.

Even though the approaches presented in this thesis for the combination of Gaussian Mixture Models and Neural Networks for Speaker Identification did not yield good results, the idea of generating a single model making use of the maximum likelihood classification characteristic of the Gaussian Mixture Models and the discriminative characteristic of the Neural Network is always possible. A more mathematically rigorous foundation for combining the two approaches should be explored. One possible avenue to investigate is given below.

Probabilistic Neural networks can be used instead of the Binary-Pair Partitioned Neural Networks used in this thesis. A probabilistic neural network uses a supervised training set to develop distribution functions within a pattern layer. These functions are used to estimate the likelihood of an input feature vector being part of a learned category, or class. The learned patterns can also be combined, or weighted, with the a priori probability of each category to determine the most likely class for a given input vector. If the a priori probability of the categories is unknown, then all categories can be assumed to be equally likely and the determination of category is solely based on the closeness of the input feature vector to the distribution function of a class.

The combination of these Probabilistic Neural networks and Gaussian Mixture Models for Speaker Identification shows promising results. [26] Once the *n*-best lists from the PNN and GMM classifiers are obtained, their inner product is to be computed. The best score element of the resulting matrix is searched, and the final result is chosen as that. This approach is similar to the Speaker Score Averaging method of combining the GMMs and Neural Networks for Speaker Identification in this thesis.

REFERENCES

[1] Jeff A. Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,* International Computer Science Institute, Department of Electrical Engineering and Computer Science U.C. Berkeley, April 1998.

[2] Brett Richard Wildermoth. *Text-Independent Speaker Recognition using Source Based Features,* Master of Philosophy, Thesis, Griffith University, Australia, January, 2001.

[3] Ashutosh Mishra. *Automatic Speaker Identification using Reusable and Retrainable Binary–Pair Partitioned Neural Networks,* M.S. Thesis, E.C.E Dept., Old Dominion University, May 2003.

[4] L. Rudasi, Stephen A. Zahorian. *Text-Independent Speaker Identification using Binary-pair Neural Networks, IJCNN- 92*, pp. IV: 679-684.

[5] Detlef Prescher. *A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars,* Institute for Logic, Language and Computation, University of Amsterdam.

[6] S.A.Zahorian and Z.B.Nossair. *A Partitioned Neural Network Approach for Vowel Classification Using Smoothed Time/Frequency Features,* IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 4, pp. 414-425, July 1999.

[7] ChengXiang Zhai. *A Note on the Expectation-Maximization (EM) Algorithm,* Department of Computer Science, University of Illinois at Urbana-Champaign, November 2, 2004.

[8] Frank Dellaert. *The Expectation Maximization Algorithm,* College of Computing, Georgia Institute of Technology, Technical Report number GIT-GVU-02-20, February2002.

[9] Zaki B.Nossair and Stephen A.Zahorian. *A Neural Network Clustering Technique for Text-Independent Speaker Identification,* Department of Electrical and Computer Engineering, Old Dominion University.

[10] Claude A. Norton, III. *Text Independent Speaker Verification using Binary-pair Partitioned Neural Networks,* M.S. Thesis, Old Dominion University, December 1995.

[11] Jinzhong Yang and Rick S. Blum. *Image Fusion Using the Expectation-Maximization Algorithm and a Hidden Markov Model,* ECE Department, Lehigh University.

[12] Douglas A. Reynolds, Larry P. Heck. *Automatic Speaker Recognition-Recent Progress, Current Applications, and Future Trends,* Presented at the AAAS 2000 Meeting Humans, Computers and Speech Symposium 19 February 2000

[13] Douglas A. Reynolds. *An Overview of Automatic Speaker Recognition Technology,* MIT Lincoln Laboratory.

[14] B. Yegnanarayana, K. Sharat Reddy and S. P. Kishore. *Source and System Features for Speaker Recognition using AANN models,* Speech and Vision Laboratory, Department of Computer Science and Engineering, Indian Institute of Technology Madras.

[15] Y. S. Moon, C. C. Leung, K. H. Pun. *Fixed-point GMM-based Speaker Verification over Mobile Embedded System,* Department of Computer Science and Engineering, The Chinese University of Hong Kong.

[16] Stephen A. Zahorian, Peter Silsbee, and Xihong Wang. *Phone Classification with Segmental Features and a Binary- pair Partitioned Neural Network Classifier,* Dept. of Electrical and Computer Engineering, Old Dominion University.

[17] R. Faltlhauser, T.Pfau, G.Ruske. *On-line Speaking Rate Estimation using Gaussian Mixture Models, Institute of Human-Machine-Communication,* Technical University of Munich, Germany.

[18] Peng Yu and Zhigang Cao. *Text-Independent Speaker Identification under Noisy Conditions using Speech Enhancement,* State Key Lab on Microwave and Digital Communications, Department of Electronic Engineering, Tsinghua University, Beijing, 100084.

[19] Ching-Tang Hsieh, Eugene Lai and You-Chuang Wang. *Robust Speaker Identification System Based on Wavelet Transform and Gaussian Mixture Model,* Department of Electrical Engineering, Tamkang University, Taipei, 251 Taiwan.

[20] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. *Speaker Verification Using Adapted Gaussian Mixture Models,* M.I.T. Lincoln Laboratory, 244 Wood St., Lexington, Massachusetts 02420.

[21] Reynolds. D. *The effects of Handset Variability on Speaker Recognition Performance: Experiment on the Switchboard Ccorpus*, In Proc. of IEEE, ICASSP, pages 113–116, May 1996.

[22] Bryan L. Pellom and John H. L. Hansen. *An Efficient Scoring Algorithm for Gaussian Mixture Model Based Speaker Identification.* IEEE Signal Processing Letters vol. 5, no. 11, pp. 281-284, November 1998.

[23] Balakrishnan Narayanaswamy. *Improved Text-Independent Speaker Recognition using Gaussian Mixture Probabilities,* M.S. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, May, 2005.

[24] Mame Sall. Generalization Issues in Pattern Classification Applied to Speaker Identification, M.S. Thesis, E.C.E Dept., Old Dominion University, May 2003.

[24] Frederic Bimbot, Jean Francis Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvian Meignier, Teva Merlin, Javier Ortega-Garcia, Dijana Petrovska Delacretaz, Douglas A. Reynolds. *A Tutorial on Text-Independent Speaker Verification,* EURASIP Journal on Applied Signal Processing 2004:4, 430-451.

[25] Douglas A. Reynolds, Richard C. Rose, *Robust Test Independent Speaker Identification using Gaussian Mixture Models,* IEEE Transactions on Speech and Audio Processing, Vol.3, No.1, January 1995.

[26] Todor Ganchev, Anastasios Tsopanoglou, Nikos Fakotakis, George Kokkinakis, *Probabilistic Neural Networks combined with GMMs for Speaker Recognition over Telephone Channels,* 14th International Conference on Digital Signal Processing (DSP2002), Volume II, pp. 1081-1084, July 1-3, 2002.

[27] Douglas A. Reynolds, *Speaker Identification and Verification using Gaussian Mixture Speaker Models,* MIT Lincoln Laboratory, March 1995.

[28] Text – Independent Speaker Identification provided by Center for Spoken Language Understanding @ OGI, http://cslu.cse.ogi.edu/HLTsurvey/ch1node53.html.

[29] Hassan Ezzaidi and Jean Rouat, *Pitch and MFCC dependent GMM models for speaker identification systems,* IEEE Canadian Conference on Electrical and Computer Engineering, May 2004.

APPENDIX

Matlab software was used for the implementation of all the algorithms in this thesis. The Matlab routines created in this thesis are as follows:

- Gmmspkidtrain – creates GMMs for speakers using the speakers' feature files

  Usage

  [estimate] = gmmspkidtrain (**filename**)

  Inputs:

  1. File containing a list of the feature files of all the speakers in the database

  Outputs:

  1. GMM parameters for each speaker, i.e. the weights, means and covariances for each speaker

  Entries of the **filename:**

  1. Number of Speakers
  2. Token Selection Mode (1 for selecting all tokens and 2 for selecting specific tokens)
  3. First token to be read
  4. Step size of the tokens to be read
  5. Last token to be read
  6. First feature to be read
  7. Last feature to be read
  8. Number of mixture components
  9. List of the feature files to be read

- Gmmspkidtest – performs the evaluation step in the speaker ID process

  Usage

  [evaluation] = gmmspkidtest (estimate, **testfilename**)

  Inputs:

  1. Structure of the variables of the GMM parameters for all the speakers used for training
  2. File containing the list of the features files of all the speakers to be tested

  Outputs:

  1. Likelihood values of each test speaker with respect to all the speakers in the training set

  Entries of the **testfilename:**

  1. Number of Speakers
  2. Token Selection Mode (1 for selecting all tokens and 2 for selecting specific tokens)
  3. First token to be read
  4. Step size of the tokens to be read
  5. Last token to be read
  6. First feature to be read
  7. Last feature to be read
  8. Number of mixture components
  9. List of the feature files to be read

- Gmmb_pdf1 – Computes the likelihood values of the test data with respect to each speaker

  Usage

  [eval1] = gmmb_pdf1 (testdata, estimate)

  Inputs:

  1. Test data that whose likelihoods with respect to the speakers in the database are to be calculated
  2. Structure of the variables of the GMM parameters for all the speakers used for training

  Outputs:

  1. The likelihood values of the test data with respect to the speakers in the database

- Gmm_algo – estimates the GMM parameters for one speaker with full covariance matrix

  Usage:

  estimate = gmm_em (data,c)

  Inputs:

  1. Data whose GMM parameters are to be estimated
  2. Number of mixture components for the GMM

  Outputs:

  1. GMM parameters for the data given as the input

- Gmm_algodiag – estimates the GMM parameters for one speaker with diagonal covariance matrix

  Usage:

  estimate = gmm_em (data,c)

  Inputs:

  3. Data whose GMM parameters are to be estimated
  4. Number of mixture components for the GMM

  Outputs:

  2. GMM parameters for the data given as the input

- Gmmb_covfixer – converts a matrix into a valid covariance matrix

  Usage:

  [nsigma, varargout] = gmmb_covfixer (sigma)

  Inputs:

  1. Matrix which has to be converted into a valid covariance matrix

  Outputs:

  1. Valid covariance matrix

- Gmmcpdf – computes the weighted probability density for a GMM, given the data, means, covariances and the weights

  Usage:

  tulo = gmmcpdf(data, mu, sigma, weight)

  Inputs:

  1. Data whose pdf is to be computed
  2. Mean of the data

3. Covariance of the data

4. Weights of each mixture component

Outputs:

1. Weighted probability density of the inputs

- Scores – calculates the average of the scores of the NN speaker ID program and the GMM speaker ID program

  Usage:

  avg = scores (nnscores, gmmscores)

  Inputs:

  1. speaker scores obtained by the NN program

  2. speaker scores obtained by the GMM program

  Outputs:

  1. average of the scores obtained from both the NN and GMM programs

CURRICULUM VITAE

USHA GAYATRI
CHALKAPALLY

Phone:
Mobile: 757-332-1516
E-mail:
gayatrich1@gmail.com

Objective	Work in the field of digital signal processing and communications.
Education	**[Fall-2004 – Summer 2006] Old Dominion University VA** M.S. (Electrical and Computer Engineering), GPA – 3.71/4.0 **[2000-2004] Vasavi College of Engineering, Hyderabad, INDIA.** B.E (E.C.E) : 74.2%

Work experience

- Teaching Assistant and grader for Linear Systems (ECE-302) at Old Dominion University [Spring-2005]
- Teaching Assistant and grader for Digital Signal Processing (ECE-481/581) at Old Dominion University [Fall-2005]
- Teaching Assistant and grader for Communications Systems(ECE-451/551) at Old Dominion University [Spring-2006]
- Academic assistant at Student support services, Old Dominion University, Virginia [Spring2005 to Summer 2006]