AUTOMATIC SPEAKER IDENTIFICATION

USING REUSABLE AND RETRAINABLE BINARY–PAIR

PARTITIONED NEURAL NETWORKS

by

Ashutosh Mishra
B.E (EE) July 1999, Manipal Institute of Technology, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
May 2003

Approved by:

_____
Stephen. A. Zahorian (Director)

_____
W. Steven Gray (Member)

_____
Oscar R. González (Member)

ABSTRACT

AUTOMATIC SPEAKER IDENTIFICATION
USING REUSABLE AND RETRAINABLE BINARY-PAIR PARTITIONED
NEURAL NETWORKS

Ashutosh Mishra
Old Dominion University
May 2003
Director: Dr. Stephen A. Zahorian

This thesis presents an extension of the work previously done on speaker identification using Binary Pair Partitioned (BPP) neural networks. In the previous work, a separate network was used for each pair of speakers in the speaker population. Although the basic BPP approach did perform well and had a simple underlying algorithm, it had the obvious disadvantage of requiring an extremely large number of networks for speaker identification with large speaker populations. It also requires training of networks proportional to the square of the number of speakers under consideration, leading to a very large number of networks to be trained and correspondingly large training and evaluation times.

In the present work, the concepts of clustered speakers and reusable binary networks are investigated. Systematic methods are explored for using a network originally trained to separate only two specific speakers to also separate other speakers of other speaker pairs. For example, it would seem quite likely that a network trained to separate a particular female speaker from a particular male speaker would also reliably separate many other male speakers from many other female speakers. The focal point of the research is to develop a method for reducing the training time and the number of networks required to achieve a desired performance level. A new method of reducing the network requirement is developed along with another method to improve the accuracy to compensate for the expected loss resulting from the network reduction (compared to the BPP approach). The two methods investigated are-reusable binary-paired partitioned neural networks (RBPP) and retrained and reusable binary-pair partitioned neural networks (RRBPP).

Both the methods explored and described in this thesis work very well for clean (studio quality) speech but do not provide the desired level of performance with bandwidth – limited speech (telephone quality). In this thesis, a detailed description of both the methods and the experimental results is provided.

All experimental results reported are based on either the Texas Instruments Massachusetts Institute of Technology (TIMIT) or Nynex TIMIT (NTIMIT) databases, using 8 sentences (approximately 24 seconds) for training and up to two sentences (approximately 6

seconds for testing). Best results obtained with TIMIT, using 102 speakers, for BPP, RBPP, and RRBPP respectively (for 2 sentences i.e. ~ 6 seconds of test data) are 99.02 %, 99.02 %, 99.02 % of speakers correctly identified. Corresponding recognition rates for NTIMIT, again using 102 speakers, are 84.3%, 75.5% and 77.5%. Using all 630 speakers, the accuracy rates for TIMIT are 99%, 97% and 96%, and the accuracy rates for NTIMIT are ~72 %, 48% and 41 %.

This thesis is humbly dedicated to my parents – Asha & Anjani.

You are the world I know, all the love I need, and the name I worship.

My darling sister – Anubha, who means the life to me…

I thank you for your unconditional acceptance, and everything I am.

- Ashutosh, 2003

ACKNOWLEDGMENT

No work is ever complete, without the due acknowledgement of the contributions and guidance of individuals around us. First and foremost, I thank my advisor Dr. Stephen A. Zahorian, for his acceptance and patience. Without his guidance, this thesis would not have been possible.

I also thank members of the faculty – Dr. W. Steven Gray and Dr. Oscar R. Gonzalez for consenting to be on the thesis advisory committee despite very short notice. This is also the appropriate time to acknowledge other members of the faculty, Dr. Lee A. Belfore, Dr. Vijayan K. Asari, and Dr. Amin Dharamsi for their unbelievable patience with my incessant queries and all the members of the staff for all their assistance and support.

We are surrounded by individuals who make each day more worthwhile and better for us and I would be failing my duty if I do not express my gratitude for all my colleagues and friends in the speech communication lab. Specifically, I would like to thank my special friends, Arturo – for his true academic spirit, and Sai – for just being there.

Finally, I bow my head to Dr. D.K. Pandey, and Mrs. Snehlata Pandey, my uncle and aunt, who have been my primary source of inspiration and support.

I would also like to acknowledge the support received from the National Science Foundation via the grant IRI-9217436 and the project – JW900.

Thank you everyone!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Effective classification and pattern recognition have been the focus of intense research over the last couple of decades, ever since computer power has enabled abstract algorithmic concepts to be implemented in real time. This work is an attempt to provide another useful contribution in this field, in particular an effective, fast and accurate "speaker identification" method.

In addition to the message content present in a human acoustic speech signal, the signal also contains information related to the gender, mood, and identity of the speaker. All of this information is embedded in the signal and the ability to use this information is inherent in human beings. The automation of this process of identification requires quantifiable knowledge of relevant information required and specific algorithms to *use* the information. One approach for using the information for the actual classification task is based on neural networks. A modification of the more typical approach of a single large neural network is to use a system of pair wise networks, one for each speaker pair. In the following sections of this chapter, there is brief introduction to the concept of speaker recognition using binary pair partitioned neural networks.

1.1 Speaker Recognition

Pattern recognition research in speech can be broadly classified in two categories – speech recognition and its complementary process [1] - speaker recognition. Speech recognition relates to methods and the ability to identify *what* has been spoken, *duration* of speech content in a signal, etc. The goal of speaker recognition is to identify the speaker, independently of what the speaker is saying. Speaker recognition can be further classified into two sub categories – (a) speaker verification and (b) speaker identification.

In speech recognition, since the underlying textual message is the target, the variations in a speech signal due to speaker identity are usually regarded as noise. Conversely, for the task of speaker identification, it is the message component that is not important. The distinguishing acoustic cues for different speakers are difficult to separate from those that reflect the identity of the sounds. *Phoneme* is the term used to label the elements that carry the linguistic information. In speaker *recognition*, it is the astute utilization of non-linguistic variability contained in the

---

The journal model used is *IEEE Transactions on Speech and Audio Processing.*

speech signal that is of primary concern. This variability is due to physical differences in the vocal chords and vocal tract. Unfortunately, none of the acoustic cues clearly point *exclusively* towards the *identity* of a particular speaker and this is the primary and biggest challenge that leads to speaker recognition being an extremely demanding task.

The task of automatic speaker recognition leads to either automatic speaker identification (ASI) or automatic speaker verification (ASV). Nevertheless, both these tasks have a significant degree of similarity. Both systems operate upon a closed database of speech reference patterns for known speakers and both systems use similar analysis and decision techniques. For ASV, any un-registered user is considered as an impostor. There is a matching process of traditional classification involved in both – that is, a training phase followed by an evaluation phase. Both the processes require the extraction of distinct acoustic "features" from the speech signal, and it is these features that are used in subsequent processing steps. During the training phase, features extracted from the speech signal are passed through the speaker recognition system in question to form a "model" for each speaker. Once the training phase is completed, a feature set from a speech sample of an unknown speaker (but belonging to the closed database for the case of ASI) is evaluated with respect to each speaker model and a decision is made. For the case of ASV, the unknown speaker presents a speech sample and a claimed identity, as one of the training speakers. The task of the system is to evaluate whether or not the claimed identity is correct; it is important to note that the unknown speaker may not have even been included in the training database – in which case, the ASV system returns a "false" or impostor.

1.1.1 Speaker Identification

A typical speaker identification system consists of prototypes for all users (M speakers). This involves capture of speech samples for each possible speaker that is then used to construct the prototype (features) for each speaker – i.e. the training phase. Thereafter, an unknown sample of speech is presented to the system and the resulting prototype is compared against all the other M prototypes – before a decision can be made. Generally, the comparison takes the form of a distance measure between the prototype and the unknown speech sample. In the case of a speaker system with 'M' speakers, M comparisons are needed.

Three sources of signal variability, which exist in a typical ASI system, are speaker variations, channel variations, and content (as in words in a text description of the speech) variations. Sometimes, it may happen that the speaker may attempt to disguise his/her voice to prevent/attempt correct identification. This is a classic example of speaker variability. Unless speech samples are gathered in a clandestine manner, such subjects can "trick" the system. As

mentioned earlier, the channel of communication is another element that is uncontrolled and causes the variability. Speech signals often need to be transmitted over some form of communication channel from the source to the recording device. Bandwidth limitations and other interference lead to a low signal-to noise ratio, especially when the transmission medium is the standard telephone wire, ultimately resulting in a poor recorded signal quality. Another important aspect that needs to be mentioned at this juncture is that usually there is no control over the content of the spoken speech, giving rise to the need for "text-independent" speaker identification systems.

1.2 Neural Networks

Automatic speaker identification is a statistical classification problem involving two basic issues – feature selection (the form in which data is represented) and the statistical classifier to be used. This research work uses an artificial neural network – more specifically, a *feed forward*, *memory-less multi layer* perceptron for the classifier. The computational element of such a classifier is a "neuron" which essentially forms the weighted sum of its inputs and passes the result through a limiting non-linearity. The weighted sum also includes an offset or bias term. "Feed-forward" implies that information flow occurs only in one direction – input to one or more hidden layer(s), and finally to the output (neurons) nodes. Memory-less implies that the network outputs are only dependent on the current input pattern. The process used to train the weights and the offset is the "error back propagation method". Furthermore, the number of layers in a multi-layer perceptron determines the type of decision regions it is capable of forming in the hyperspace formed by its inputs. A neuron of a single layer network forms a single hyper plane decision boundary. A two layer (consisting of one hidden layer) network is capable of forming convex open or closed decision regions by performing certain Boolean logic operations on the decisions formed by the hidden layer neurons. A three-layer network can form arbitrary, disjoint decision regions whose complexity is limited only by the number of nodes [19]. In each experiment reported in this thesis, the network(s) used were two-layered fully-interconnected, memory-less and feed-forward, with a sigmoid non-linearity.

1.2.1 Partitioned Neural Networks for ASI

Automatic speaker identification is essentially a statistical pattern classification problem involving decisions over M categories. In this chapter, a brief review of methods of data partitioning and advantages of binary pair partitioning will be 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. In a previous study [6], it has been shown that a neural network classifier performs significantly better than a maximum likelihood classifier for ASI. Nevertheless, the 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 the 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 a 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.

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 and referenced in works such as [20], etc. Since the BPP approach is the underlying partitioning approach for this work, it will be the main focus of discussion for the remainder of this chapter. 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 humans →(man, woman,

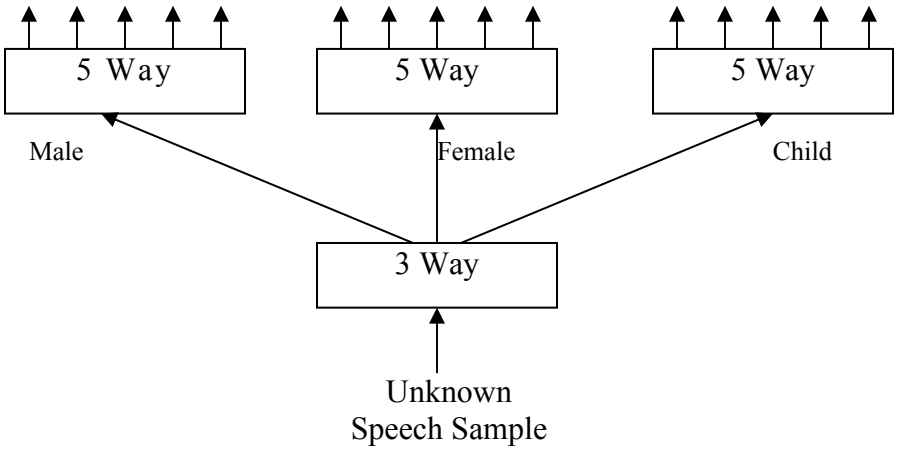

Fig. 1-2-1 Group Partitioning

child) or medication → (antihistamines, analgesic, etc.), 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 1-

2-1 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 on 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.

1.2.2 Binary – Pair Partitioning

This form of partitioning involves using a large number of binary classifiers, with each classifier trained to distinguish between only two categories. Thus with M categories, there is a

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

Fig. 1-2-2 BPP classifier matrix representation

requirement of $\frac{M(M-1)}{2}$ classifiers, which also equals the number of unique pairs that can be formed with a speaker population of M speakers. This may be visualized as a square matrix form (Fig.1-2-2), which shows elements corresponding to pairs of *unique* categories that can be formed. What is important to note in the given form is that 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, as shown in Figure 1-2-3. 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

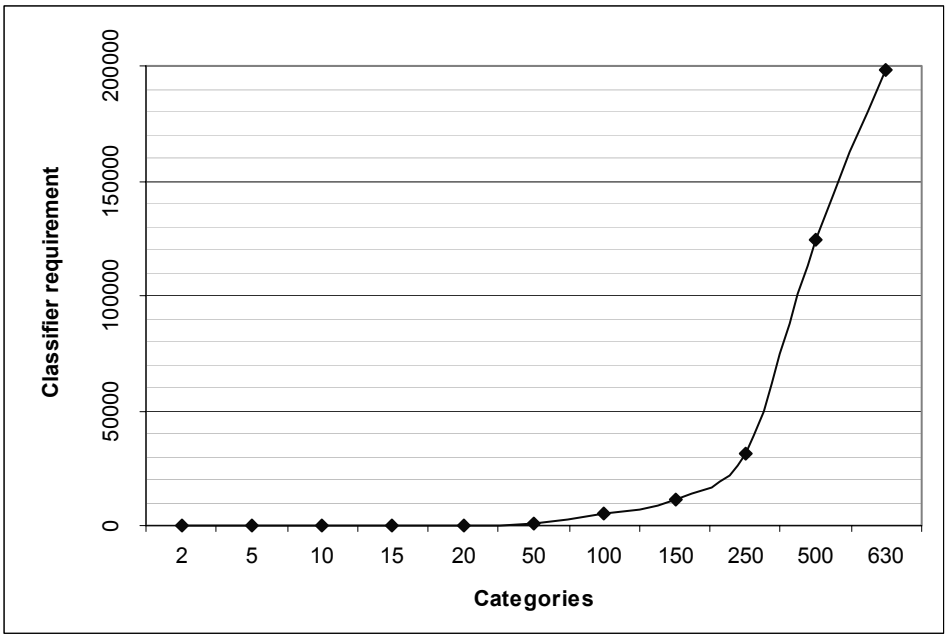

Fig. 1-2-3 Classifier requirement as a function of categories.

BPP partitioning over group partitioning is that categories need not be grouped, which eliminates the need for arranging similar categories together prior to classification. However, there are certain disadvantages with such an approach which, along with the alternative schemes (RBPP and RRBPP), will be discussed in the following chapter.

1.3 Objective of This Research

The primary aim of this research is to explore the concept and possibility of reducing the resources required for effective speaker identification by way of modifying the binary pair partitioned neural networks. As will be detailed in forthcoming chapters, the proposed method does lead to effective and acceptable performance results with a *significant* decrease in the number of networks required, at least for the case of clean speech. Unfortunately, for the more important case of telephone speech, it does not appear that the new methods are nearly as accurate as the BPP method. Nevertheless, this thesis report documents the work that has been done. There is always room for improvement at the conclusion of any research, especially one that deals with data that is as variable as human speech. Some possibilities for improvements and extensions are presented in the final chapter of this work.

CHAPTER II

BACKGROUND REVIEW

2.1 BPP Details

In the previous chapter, it was shown that the BPP method leads to an extremely large number of classifiers that are needed, especially as the number of categories exceeds 200. Both training and evaluation time are increased due to other factors that are directly affected. Part of the problem is that storage space is needed for all the classifiers and access times are involved. Despite the large number of classifiers, the significant training time reduction is still an advantage over using a single large classifier. However, the group partitioning approach might be better yet since the number of classifiers required is significantly lower. This method as implemented in our lab is as follows.

First feature vectors are computed for each speech "frame" and each speaker. Frames are typically 30-40 ms long and spaced 10 ms apart. Thus, with 10 sec. of speech data from each speaker, there would be 1000 frames of data from each speaker. Next a 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 (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.

The BPP evaluation method can be implemented in two basic forms-each providing excellent performance. The demarcating factor is the rate at which the performance deteriorates in the two methods in the event of some classifiers not performing correctly. We first summarize the basic processing steps for evaluating an unknown speaker for the "global soft search" method:

- The unknown speech sample (converted into a series of feature vectors) is passed through all the $\frac{M(M-1)}{2}$ classifiers that have been trained using the speech samples of "M" speakers, for each frame.
- The outputs of each classifier for each frame are averaged over all frames to obtain an overall result for each classifier.

- Each of the classifiers results is considered as a "vote". These $\frac{M(M-1)}{2}$ outputs may be considered as elements of the matrix shown earlier. Note that the blank entries in the matrix are also filled in with complementary results for each classifier i.e. $x_{ij} = 1 - x_{ji}$.
- The sum of the *M* column entries in each row is then used to make the decision.

In this method, the row averages as explained are computed. Thereafter the decision is based on the row index providing the highest sum. The potential drawback with this approach is that it involves all the $\frac{M(M-1)}{2}$ classifiers for each decision, which is definitely a time consuming process.

The second approach for using BPP partitioning may be thought of as a binary tree search. In this approach, a particular classifier is used to evaluate the test data. Based on the output of this classifier, all classifiers involving the "low output" speaker are not considered thereafter. To illustrate this, consider the following:

Test data → *classifier (2, 1)* → ***(low)***. Implies unknown is not speaker 2.

Test data → *classifier (3,1)* →***(high)***. Implies unknown speaker is not 1.

Test data → classifier (4,3)→ ***(low)***. Implies unknown speaker is not 4.

.

.

and so on.

Thus the number of classifiers required for a decision is only $M - 1$, leading to a huge reduction in evaluation time for a large M. Nevertheless, the potential problem with this approach is that it has a severe dependence upon the performance of all the classifiers that *are* considered. If *any* intermediate classifier gives an incorrect output, the end identity will be incorrect. On the other hand, with the global search approach, since it considers all the $\frac{M(M-1)}{2}$ classifiers; the effects of incorrect performance of one or more classifiers are more likely to be offset. All the experiments and results presented in this thesis have been obtained using the "global search" approach.

In the forthcoming chapters, we detail the effects of these factors on the performance rates and a detailed explanation of the reusable neural network algorithm and the re-trainable, re-usable neural networks and their suitability for clean and telephone speech. In the next section, a brief account of some other research in the field of speaker identification is provided.

2.2 Related Research

In recent years, there has been considerable research, in the area of automatic speaker identification and this section deals with the salient aspects of a few such works. They deal with different approaches towards the task of classification and different *forms* of data representation.

2.2.1 Gaussian Mixture Models (GMM)

One of the most important and heavily utilized techniques for classification has been Gaussian Mixture Modeling. It has been shown in several studies to be extremely effective for the task of speaker identification. A GMM as used in several works such as [5] and [15] is defined as a weighted sum of M component densities given by:

$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x})$, where $\vec{x}$ is a D dimensional vector and $b_i(\vec{x}), (i = 1,...M)$ are component densities, $p_i, i = 1...M$ are the mixture weights. Each component density is assumed to have a Gaussian distribution and given by: $b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_i \mid^{1/2}} e^{-\frac{(\vec{x}-\vec{\mu}_i)' \Sigma_i^{-1} (\vec{x}-\vec{\mu}_i)}{2}}$ with vectors μ_i and Σ_i as the mean vector and the covariance matrix respectively and each speaker is represented by a GMM and denoted as a model λ.

One of the fundamental reasons justifying the usage of GMMs is that a speaker's acoustic feature may be represented as Gaussian distribution, which may represent some "general speaker-dependent vocal tract configuration." As used, the spectral shape of the i^{th} acoustic feature is represented by the mean of the corresponding component density and the variations in the feature by the covariance matrix. Now for speaker identification, a group of 'S' speakers are represented as $\lambda_1, \lambda_2 ... \lambda_S$. For a given observation sequence of acoustic features, the aim is to evaluate the maximum *a posteriori* probability about the corresponding speaker model. Finally, the decision rule simplifies to: $\hat{S} = \arg\left(\max_{1 \le k \le S} \left[\sum_{t=1}^{M} \log p(\vec{x}_t \mid \lambda_k) \right] \right)$

It should be noted that all acoustic files are passed through a front end[2] to produce a *D* dimensional feature vector.

While on the topic of GMMs, a different method of their usage has been proposed in [15], which is discussed in the next section.

---

[2] More discussion on this in chapter 4.

2.2.2 Frame Pruning

This procedure proposes the elimination of 'neutral' frames – i.e. those frames, from which no particular identity of any speaker emerges. It is based on the assumption that "the maximum likelihood scores resulting in correct identification are generally higher than the maximum likelihood scores resulting in incorrect identification". What this implies is that with those frames which do not contribute enough information towards the correct identification, then it is *not due to the fact that a particular incorrect model performs better* (that is the information from frames corresponding to incorrect speakers is dominant), *rather it is due to the fact that the model chosen for the speaker is performing badly*. This eventually leads to a selection criterion for frame pruning, which is as follows:

A normalized score in the form of a log likelihood ratio is used in a slightly modified form – that is the minus log likelihood ratio. This is computed for all the frames in a specified segment length and frames with the lowest scores are selected for pruning. The reason why this approach using GMMs has potential is due to the fact that certain frames may have lower minus log likelihood scores for the true speaker than for the "non-target" speakers. This essentially calls for the removal of such *error* frames.

2.2.3 Exploring the Effects of Transmission Channels

In works such as [4] performance loss in speaker identification for telephone speech has been explained. The authors of the study state that the performance drop associated with band limiting and filtering does *not* account for the entire performance drop. The system used for speaker identification is based on Gaussian mixture models (as described in Section 2.3.1). Also, the telephone degradations were simulated and applied to clean wideband speech from the TIMIT database, in order to test the validity of the speaker models and to check if all the factors were being accounted for. Some of these simulated factors were:

- Band limiting - (300~3400 Hz)
- Spectral shaping (filtering): A FIR channel filter with a spectrum matching (in a MSE sense) the sweep tones form the actual NTIMIT telephone lines, was used to filter the clean TIMIT speech.
- Noise addition: To match the SNR of the corresponding NTIMIT sentence, broadband Gaussian noise was added to each TIMIT sentence.

It was noticed, that despite these attempted simulations, the corrupted TIMIT results were still ~16% higher than the corresponding original NTIMIT results – which implies that there are still

certain unaccounted factors in the assumed model. Correspondingly, in their work, the authors propose *nonlinear microphone effects* as these unaccounted factors. One of the most important is the distortion due to *carbon button microphone*. In their studies, they do present evidence of a "phantom formant" – formed at sum and difference points of formant frequencies, when simple static non-linearities were applied to the speech signal. This also implies that "resonance bandwidths can be narrowed/broadened depending on the order of the non linearity". Finally, this work has followed the same test configuration[3] as mentioned in this work, and the NTIMIT performance results over the entire database of 630 speakers obtained by the authors is 60.7% ±1.4% (using one sentence of test speech).

2.2.4 Application of Neural Networks in Speaker Identification

In the past decade, several researchers have exploited neural networks in classification tasks in the area of speaker identification. A case in the point is [18] where, predictive neural networks (PNN)[4], (a non-linear predictive model) are used as the classifiers, with the speaker model being ergodic – allowing transitions to any other state. In this approach, a PNN was assigned to each state. One model was trained for each speaker using the forward-backward algorithm, and identification was based on the model that provided the maximum probability. Investigations of the effect of changing the number of hidden nodes, from 5~40, and models with 1 and 4 states, it was found that the identification accuracy increased with an increase in the hidden nodes, until there were 10 hidden nodes, progressively decreasing thereafter. Also, the performance of the 4-state model was always higher than the 1 state model. It was also concluded, that a non-linear model performed better than a linear model (having the same architecture). Finally, this research did reveal that performance varies as the number of iterations, test data length, number of hidden nodes, etc. However, it is difficult to determine optimal values for these parameters, beforehand – whether PNN or DNNs were used.

In another related and recent work [17], the authors have suggested alternative training strategies for training multi-state models and using temporal alignment for the improvement of the neural predictive models, since "a NN state may abusively generalize even on the data it has never seen, thus obtaining better performance than the correct model." The authors report an increase in identification rates, after the additional temporal alignment, and claiming the performance could be additionally improved, if the neural predictor and the state model corresponding to a speaker model were to be "simultaneously optimized."

---

[3] Refer to chapter 4 for additional details.
[4] Compare to a Discriminative NN, which is trained using all speakers' data.

So far, we have only discussed a few approaches to the task of classification in speaker identification. Nevertheless, one of the vital aspects that need to be addressed now is that of "front end" analysis – or the task of using the acoustic data. The front-end analysis usually consists of three primary steps: normalization, parameterization and feature extraction. All these steps are primarily geared towards information reduction or removal of redundancies, since the objective is preserve the speaker identity, and not the textual content. This leads to the removal of many aspects of the speech data that provide the sense of "naturalness and intelligibility".

Normalization is done to remove or compensate for the variability due to factors that are not particularly dependent upon the speaker dependent acoustics, rather due to factors such as background noise, distance from the microphone, transmission loss, etc. *Parameterization* is the process that involves the major data reduction by way of converting the signal into parameters and features. In the past, several SI systems have employed spectrum-based features – notably the mel-cepstrum. Nevertheless, irrespective of the classification technique employed, it has been shown by [12], that their performance depends strongly upon the front-end analysis.

Features that are selected for use as training data for any classifier should – (a) be able to accurately represent the acoustic signal and (b) should be able to exist in a compact form too. In this section, a description of the feature selection used in this research is described. Keeping in mind the desired properties, *cepstral coefficients* [7] have been used. In all the experiments reported in this thesis, the acoustic signal was sampled at 16 KHz and then analyzed using 40 ms frames. Each frame is spaced 10 ms apart. A Kaiser window was used for spectral estimation and its function is as follows:

$$w_k(n) = I_o(b) \Big/ I_o(a); \; for \; |n| < Q; \qquad [1]$$

$$= 0, \; otherwise.$$

In the above equation, '*a'* has been empirically determined to be 5.33. The value of *'b'* is given by: $b = a\left[1 - \left(\frac{n}{Q}\right)^2\right]^{\frac{1}{2}}$

In the earlier expression, $I_o(x)$ is the modified Bessel function, given by the series expansion of

$$I_o(x) = 1 + \sum_{n=1}^{\infty} \left[\frac{\left(x/2 \right)^n}{(n!)} \right]^2$$

Each frame is represented by a 25-term[5] Discrete Cosine Transform coefficient (DCTC) expansion computed from the log-magnitude spectrum and they are essentially equivalent to the cepstral coefficients. The DCTC expansion has been used because of its ability to represent the magnitude frequency components in a compact form. These coefficients were then scaled, linearly using the equation $X_i^{'} = \frac{X_i - \bar{X}}{5.\sigma_x}$ where X_i is the scaled value and equal to the original coefficient minus the original mean, divided by five standard deviations. This is done to normalize the feature space and provide a better overall performance, having been effectively utilized in the previous works by such as [7].

In the next chapters, we detail the *algorithms* for implementing the re-usable and then retrain-able *and* re-usable neural network for the task of speaker identification. Also, discussed, will be other factors such as criterion for network replacement, re-training data size, and effects of other factors such as training iterations, threshold, and the applicability of these methods to telephone speech.

[5] Determined for maximum performance rates (described in sec. 4.4)

CHAPTER III

THE RBPP AND RRBPP ALGORITHMS

In the previous chapters, the binary pair partition based method was introduced. In this chapter, a detailed description of the algorithm for reusing the networks is provided followed by a description of the method for retraining and reusing the networks.

3.1 Reusable Networks

As mentioned earlier, Binary Pair partitioning leads to the requirement of a neural network (classifier) for every pair of speakers present in the dataset. Thus, it is only natural to question the possibility of *more than one* such pair being separated by a particular classifier. In other words, if a particular classifier (trained over the data from one particular speaker pair) can separate *another* pair to a pre-set level of performance, then all such speaker pairs can be mapped to that particular network. Needless to say, this would lead to a significant reduction in the number of classifiers. This is the essence of the entire concept of reusable and, to be discussed later, re-trained and re-usable networks. It should also be understood that the underlying basis for the assumption that a classifier will be able to separate (to a desired degree of satisfaction), categories different from those it has been trained for is not universally applicable. However, for the case of human speech, even though there is a difference in the identity of the speaker, there may be groups of speakers, all of whom sound very *similar*. For example, as mentioned earlier, a classifier that is trained to separate *one particular male speaker from a particular female speaker* will likely be able to differentiate between *most* other male/female speaker pairs.

In light of the information presented above, once a network is trained to differentiate between two speakers, how do we measure its effectiveness for separating other speaker pairs? In particular, a parameter is needed to determine whether a particular network is "good enough" for other speaker pair(s). For this, we used a simply defined *Threshold value*. For two other speakers to be separable with respect to a certain threshold, one of the speaker averages (over all frames) at the network output must be greater than the threshold, and the other speaker average must be less than *(1- threshold)*. In our case, with the unipolar sigmoid, thresholds tested ranged from 0.55 to 0.75. Lower values of threshold lead to a larger reduction in the number of classifiers required but also lead to a sharp decline in the performance. Higher values lead to a

relatively low network reduction but a relatively low performance loss[6]. Another point that should be noted is that beyond a certain threshold value, both RBPP and RRBPP lead to BPP classification (due to no reduction in networks).

There are multiple factors that affect the performance rate of the neural networks towards the task of speaker identification, and the time taken for them to be trained. Some of the important factors are as follows:

- Quality of speech data (bandwidth)
- Number of training iterations
- Network architecture, number of nodes
- Number of features per frame
- Group data sizes[7]

In the forthcoming sections, we define some commonly used terms and present the algorithms for RBPP and RRBPP methods.

3.2 Definitions

In this section, we define a few ideas and terms used in this research. These will be used throughout the remainder of this thesis without any further explanation.

3.2.1 Speaker Pair Index

As mentioned earlier, in a BPP approach, the number of classifiers required for classification of M categories $= \frac{M(M-1)}{2}$. All these categories are paired and each of these pairs can be considered to form the elements of a matrix as depicted in fig.1-2-2. If we approach all these pairs (present in the upper triangle) *sequentially*, and give them an index, then all the unique pairs will form the following sequence…*(1,2), (1,3). ...(1,M), (2,3),(2,4)....(2,M), ...(M-1, M).* Now if we *index* these pairs sequentially, then, the pair (1,2) has an index '1', (1,M) has an index *M-1,* and so on. In fact, there exists an empirical expression to link a particular pair with the index of the pairs, and that is: $\left(\frac{(a-1)\times(b-1)}{2}\right)+b-1$

In this expression, 'a' is the index of first speaker and b is the second speaker index. Thus, for example, the index of the pair (1, 2) = 2, index of pair (1, M) = M; and (8, M) has an index equal to $_{Int}\left(\frac{7\times(M-1)}{2}\right)+M-1$.

[6] All comparisons are vis-à-vis BPP approach.
[7] Applicable ONLY to the re-trainable and re-usable networks

3.2.2 Neural Network Output

In all the neural networks used for the experimental results presented in this work, the number of input nodes equals the number of features used, the number of hidden nodes equaled 10 and there was one output node. The activation function at each node is the 'sigmoid' of the form:

$f(x) = \frac{1}{1+e^{-x}}$, which has an output form as shown in figure. 3.1. The output varies between 1 and 0, as the input to this function varies between $+\infty \sim -\infty$.

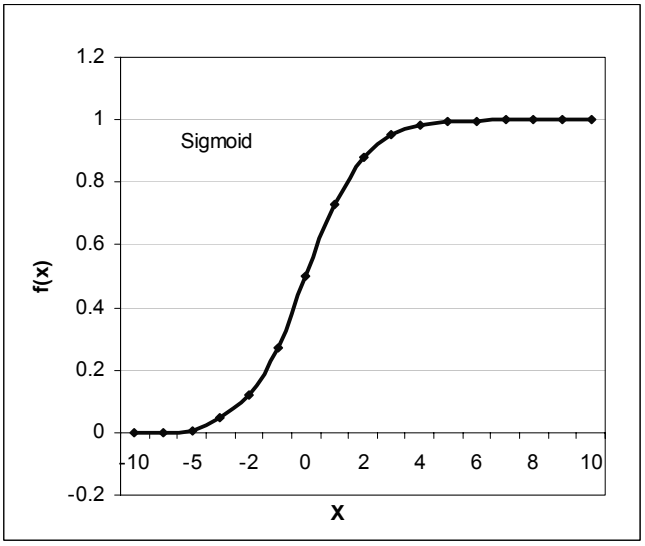

Fig. 3.1 Characteristic of the sigmoid activation function

Since the neural network inputs are dependent upon the number of features, per frame, this implies that in every pass through the network, one frame is processed. To compute the speaker *average*, we sum the outputs for the entire utterance (all the frames for a particular speaker) and divide it by the number of frames.[8]

Now this average output value per speaker is a value (between 0~1). In the next section, a description of the reusable networks is provided along with the algorithm and how the evaluation/recognition performance is affected by the threshold value.

3. 3 Reusable Networks

Referring to figure 1.1, the classification task for *M* speakers, requires $\frac{M(M-1)}{2}$ BPP neural network classifiers. But if we assume that one particular network can be used to classify

[8] A detailed pseudo-code may be found in the appendix.

some other pair too, then the total number of networks is *not* equal to $\frac{M(M-1)}{2}$. Consider the following scenario over 6 speakers, in which the following networks have the following applicability…

Network # 1 → speaker pair **(2, 1)**, (3, 5)

Network # 2 → speaker pair **(3, 1)**, (6, 1)

Remaining networks → Uni-pair.

In the above, bold typeface corresponds to the speaker pair – which is the same as for the BPP case. The second pair is the additional pair, whose BPP network has been replaced by some – existing network. What this implies is that, if this is possible, then total network requirement for the entire closed set of 6 speakers – reduces from $\left(\frac{6\times(6-1)}{2}\right)$=15 networks to 13. This is just an example, and in a real scenario, depending upon the nature of the speech data being classified, the number of networks for the same number of speakers would typically reduce from 15 to perhaps 7~8. In general, a network trained over the speech data from a particular pair index, may also separate other speakers but typically with a lower level of performance. We now provide an algorithm to reuse networks in a systematic way, but based on the aforementioned general principle.

3.3.1 RBPP Algorithm

We first present a pseudo code version of the RBPP algorithm, followed by an explanation in words[9].

<begin>

While (index <= indexfinal; index++);

{{{**Read** <speaker pair data>;

While (training cycles <=total iterations OR *mse<min_error)*

<*Function **train neural net*** ψ>;

While *(index2 <final index);*

{{ While *(~ exist(Ψ_{index2}));* → if no network has been marked for some pair.

[

[9] ψ represents a neural network.

$$out\_1 = \left(\frac{\sum_{frames} \Psi(spkr1\_data)}{frames} \right);$$

$$out\_2 = \left(\frac{\sum_{frames} \Psi(spkr2\_data)}{frames} \right);$$

(If out_1>threshold) ***AND*** *out_2<(1-threshold)*

*{ψ(Index2) =ψ; (*mark current network as applicable for others).

}]

}}

}}}

In words, the above algorithm, starting from the first speaker pair, trains a neural network on the data from this speaker, such that the output corresponding to samples from one speaker are "high" and "low" from the other. Once this network is trained, we evaluate all other speakers with this network. All speakers whose network output is higher than the specified threshold are identified as "high" speakers for this network. All speakers whose network is lower than *(1-threshold)* are labeled as "low." The single network is then considered adequate (with respect to the specified threshold), for separating all the high speakers from all the low speakers. For example, if there are 10 high speakers and 5 low speakers, the network trained on the single speaker pair can, in fact also separate 50 *additional* speaker pairs. Then we flag all such speaker pairs against the current network and move on to train another speaker pair. Any speaker pair that is flagged as being separable by an already trained network does not need to be considered as a potential training pair, or need to be checked for, with respect to new networks that are trained even though there *might* exist new networks that would provide *better* separation than the network flagged against those pairs. This is done to ensure *rapid* convergence and avoid any endless recursion.

The actual algorithm for RBPP has some important additional aspects beyond the basic approach mentioned above. First, in the training process, the threshold is first set at a very low value (typically 0.51) and then gradually raised to the final desired value. This changing threshold is called *threshold_active*. There is also another parameter, labeled as "*thresh_final"*, used to control the training. With these two points in mind, the training begins by flagging speaker pairs separated with respect to *threshold_active*. However, only those speaker pairs

separated by a threshold greater than *thresh_final* are flagged as completely finished. As training progresses, *threshold_active* is recomputed after each network is training, as the average separation factor for separated speaker pairs. Also, after each network is trained, it is used to evaluate all speakers, except for those flagged as completely finished. If a newly trained network better separates an already flagged pair, that speaker pair is re-flagged against the newly trained network. Speaker pairs are selected for training according to whichever speaker pair is currently separated the most poorly. With these modifications, an RBPP system of networks is rapidly determined, albeit with a low threshold and thus poor identification accuracy, for the entire group of speakers. The threshold, total number of networks, and identification accuracy gradually improves as more speaker pairs are trained. It is also possible that some speaker pairs originally used for training will eventually be removed from consideration, since other speaker pairs may eventually be more effective at separating pairs.

3.4 Retraining and Re-using Networks

In the previous section, the concept of reusing the individual classifiers was introduced, in order to reduce the total number of classifiers. In this section, in continuance with that same objective, we look at a method that aims to further increase the "effectiveness" by *retraining* each of these classifiers using the data it can classify "naturally". Recalling the earlier description – a classifier trained on the data of one particular speaker data *usually* has the ability to classify several other speaker pairs too, depending upon the *goodness* measure we consider acceptable that is, it is within the *natural* classification ability of the network. Now, with the goal of *increasing* this ability, we use the following method:

As mentioned above, after each network is trained the network evaluates all other speakers, resulting in a group of "high" speakers and a group of "low" speakers. Then, we combine the speech data of each of these groups, and re-train the classifier over this larger *speech data pair*. Once this network has been retrained, it is again evaluated for its ability to classify speakers over *all* the pairs. All pairs where speakers produce an average output value greater than or equal to *threshold value* (for high) and *1-threshold* (for low) are flagged against the current network.

Pilot experiments based on retraining using all the high and low speakers resulted in poor performance. Instead, performance was enhanced retraining with only the "highest" of the high speakers and lowest of the low speakers (i.e., only data that is well separated by the original network). More discussion of this point is given in the experimental results chapter. However, it

appeared that generally, for the databases used, it was best to retrain with group sizes between 2 and 5 speakers.

Thereafter, there is the issue of *which* speakers to use in each group. To illustrate this point, consider a scenario in which 20 speakers are classified as "high" and 15 speakers are classified as "low" before retraining and for which a maximum of 5 speakers should be incorporated in each group. These speakers are sorted in the decreasing order (for the high output speakers) and increasing order (for the low output speakers), based on network output average value. The five highest-ranking speakers from each list are speakers best separated by the original network, and the ones used for the retraining. The aim of this exercise is to provide the two data sets that provide patterns to cover as many categories while remaining easily distinguishable. The next section illustrates all the above-mentioned steps involved in retraining the reusable networks in a pseudo-code form, followed by some additional description behind the reason for sorting these retraining data sets.

3.4.1 RRBPP Algorithm

<Begin>

While (index <= indexfinal; index++);

{{{

Read <speaker pair data>;

While (training cycles <=total iterations OR *mse<min_error)*

<Function ***train neural net ψ****>*;

While *(index2 <final index); index2=1;*

{{While *(~ exist (*Ψ_{index2}*));* → if no network has been marked for some pair.

[

$$out\_1 = \left(\frac{\sum_{frames} \Psi(spkr1\_data)}{frames} \right);$$

$$out\_2 = \left(\frac{\sum_{frames} \Psi(spkr2\_data)}{frames} \right);$$

If (out_1>threshold) ***AND*** *out_2<(1-threshold)*

{

Group A=out_1; Group B=out_2;

}

]

Call ***function*** *<SORT groups>;*

Speech data1= $\sum_{n=1}^{MaxSize}\sum_{i=1}^{P} frame_i(spkr(groupA_n))$, where p=number of frames for a speaker

Speech data2= $\sum_{n=1}^{MaxSize}\sum_{i=1}^{P} frame_i(spkr(groupB_n))$

Call ***function*** *train neural network* Ψ *;*

While *(~ exist (*Ψ_{index2}*));* → if no network has been marked for some pair.

[

$$out\_1 = \left(\frac{\sum_{frames} \Psi(spkr1\_data)}{frames} \right);$$

$$out\_2 = \left(\frac{\sum_{frames} \Psi(spkr2\_data)}{frames} \right);$$

If (out_1>threshold) ***AND*** *out_2<(1-threshold)*

{

ψ*(index2)=*ψ*; (*mark current network as applicable for others).

}

]

}}

}}}

<End>

In the above algorithm (which is a modification of the one presented in the previous section), after a network has been trained, all other speaker pairs are evaluated using that network. It then takes all the *speakers* that produced a high output, and sorts them, in descending order (based on their output) and forms a group. Similarly, it sorts all the low output speakers in ascending order and another group is formed. Thereafter, using the combined speech data of *"k"* speakers (where "k" is the retraining group size and $1 \le k \le MaxSize$), from each group the

network is *re*-trained. Once the training is done, all speakers are re-evaluated against the current network, and all such pairs that are separated satisfactorily by the network (produce a low output of (1-threshold) and a high greater than *threshold*) are flagged against it. Other additional aspects, present in this algorithm are same as that present in the RBPP algorithm, which are described in sec.3.2.1.

Earlier, it was mentioned that using the entire speech data corresponding to the speakers classified by a particular network as its retraining data, yields lower performance results, leading to a requirement for a limit on the groups sizes. The reason behind this is as follows. Consider a scenario that the threshold value is set to 0.6. This implies that this network would be applicable to any speaker pair (say pair *a)*, wherein one speaker provides an average output value of 0.6 and the other say 0.3. Now if there are some other pairs (say *b, c)*, where the speaker data provides output values of 0.9, 0.11 and 0.85, 0.19 respectively, then while combining the speaker data for retraining, pair *a* would "corrupt" the data set because its speaker data are not as distinguishable as compared to pairs *b* and *c*. This is the reason why we sort the speakers in the manner described above prior to the data combination. Finally, it needs to be mentioned at this point that network weights are *not* re-initialized prior to the re-training cycle, so that the network preserves its initial classification bias and the training error converges faster during the retraining cycle.

In this chapter, we have introduced the methods for using the binary-paired neural networks and defined some key terms, along with the criterion for re-training and the retraining data set selection. We also provided additional reasoning, behind using this approach and in the next chapter we present experimental results, which shall highlight their merits and performance.

CHAPTER IV

EXPERIMENTS

In the previous chapter, we introduced and explained two approaches for re-usage of trained binary neural networks for speaker identification. In one approach (RBPP) the trained networks are re-used with out modification. In the second approach, (RRBPP), each binary network is first retrained on speaker groups before using it to separate additional speaker pairs. In this chapter, several experiments are reported and analyzed. Prior to the actual experiments being reported, the next section documents pre-processing steps involved in the front-end analysis of the acoustic files followed by an explanation of the training and testing setup.

4.1 Database and Feature Extraction Details

As previously mentioned, all experiments were performed using the TIMIT and NTIMIT databases. Although these databases are primarily intended to validate phonetic analysis in automatic speech recognition systems, the large number of speakers (630) also provides a good test bed for evaluation of speaker identification systems. The acoustic files contained in the TIMIT and NTIMIT database are structured identically and are actually the same sentences spoken by the same speakers. However, the NIMIT database was recorded over the telephone lines, thus limiting the bandwidth and adding noise to the signals. Thus the front end processing of each database is very similar, as described below.

Each database consists of 10 sentences (2 SA, 3 SI and 5 SX sentences) from each of the 630 unique speakers. The front end processing was configured to create 630 *scaled* and *parameterized* files, each consisting of all the 10 sentences for each speaker in the order just mentioned. Each frame of each sentence for each speaker was represented using 25 DCTC (cepstral like) features. Unless otherwise mentioned, for neural network training purposes, sentences 1~8 (2 SA, 3 SI and 3 SX) were used. Testing (evaluation) was carried out using the remaining 2 SX sentences per speaker. Finally, all the pilot tests were conducted over a subset of 102 speakers belonging to the dialect region 2 from each database. Other relevant settings used in the front end are listed below.

For the TIMIT database, frames used were 40 ms long, spaced 10ms apart. For each frame, a FFT of 1024 points was computed after Kaiser windowing with a $\beta = 6$, DCTCs were computed using a warping factor of 0.25 over a frequency range of 75~6000 Hz.

For the NTIMIT database speakers, all the above values were the same except for the frequency range, which was 300~3400 Hz. Apart from that, for the NTIMIT speakers, low energy frames, (frames that had CC1 values less than two standard deviations, below the mean) were removed. This removal of low energy frames was not done for TIMIT speakers. The decision to remove the low energy frames for the case of NTIMIT but not for TIMIT was motivated by the idea that low energy frames for NTIMT (but not TIMIT) were likely to be dominated by noise. However, the actual decision was based on the results of pilot experiments, which showed some benefit for the case of NTIMIT but not for TIMIT.

In the following sections (4.2~4.6), we present experimental results with the setup described above. The reason why 102 speakers were chosen to conduct pilot tests was due to the fact that testing the entire database is an extremely time consuming operation. This was not considered feasible, especially when most of these tests were conducted to study the effect of factors such as number of features, training iterations, training data size etc. Using *all* the speakers from a particular dialect region (102 speakers, from dialect region 2) was presumably a more difficult task than using 102 speakers from several dialect regions, since all the speakers in dialect region 2 had the same general accent. Thus the results obtained from these 102 speakers were assumed to be a better indicator of algorithmic accuracy than would have been results obtained from 102 "random" speakers. In most of the experiments reported here, we report performance as a function of speech length. This is useful because, in practice, speaker id systems are expected to work with very limited duration speech segments.

We begin with basic tests using RBPP and RRBPP methods with 102 speakers, testing the effects of some of the parameters that affect performance. Included also are comparisons with the BPP method. Finally, tests are given for the entire 630 speakers, using the “optimal” values of all the factors found from the experiments with the 102 speakers.

4.2 Experiment I - RBPP baseline Performance

In this section, the tables 4.2-a and 4.2-b list the total networks required for various threshold values, for tests involving 102 speakers from TIMIT and NTIMIT database. Also presented are the figures 4.2-a and 4.2-b that depict the identification performance for each database for each threshold.

Table 4-2-a Network requirement with RBPP v/s BPP approach over 102 speakers from TIMIT database speakers

Network requirement		
Threshold	**RBPP**	**BPP**
0.55	120	5151
0.6	472	
0.65	1113	
0.7	1835	
0.75	2651	

Upon examination of Table 4-2-a, and the corresponding identification performance in fig.4-2-a, one of the most obvious aspects noticed is that the biggest jump in performance

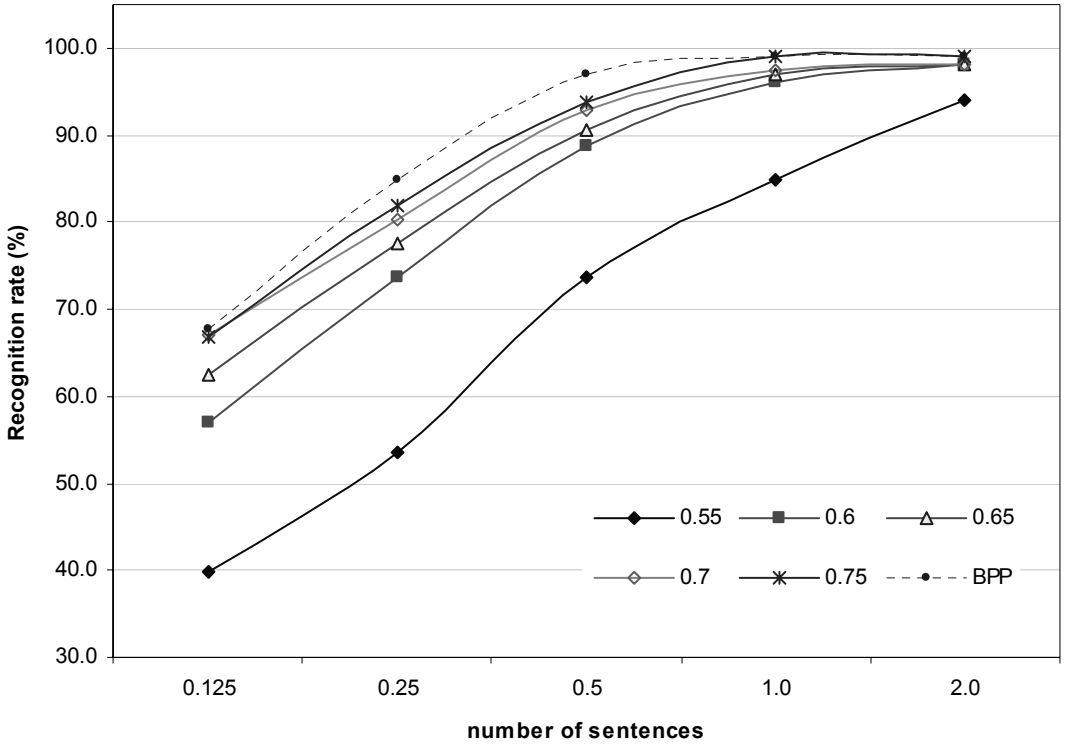

Fig. 4-2-a Performance with 102 speakers from TIMIT database using RBPP & BPP approach. . Note that each curve annotated with a number is for the RBPP case, with the number denoting the RBPP threshold level.

(irrespective of the amount of the speech data being evaluated) occurs with a change in the threshold level from 0.55 ~ 0.6. The number of networks required increases by approximately a factor of 4 when the threshold increases over this range. Such a significant jump can be attributed to the way *threshold* works. A threshold value of 0.55 implies that any network whose output

gives a *speaker average* of 0.55 for one category and 1-0.55 = 0.45 for the other is considered as acceptable for separating those two speakers. During the evaluation phase, when the unknown data appears, since some of the networks only separate some speakers with a low reliability, some errors are likely to be made.

Table 4-2-b Network requirement with RBPP v/s BPP approach over 102 speakers from NTIMIT database

Network requirement		
Threshold	**RBPP**	**BPP**
0.55	252	5151
0.6	1024	
0.625	1476	

The advantage that occurs with the RBPP approach over the BPP is quite clear for threshold values of 0.65 and above. At a threshold of 0.75, the maximum considered, the difference in the recognition rates between the BPP and the RBPP approach is ~ 3%, but the network requirement drops by a factor of ~2. In fact, the recognition rates are almost identical

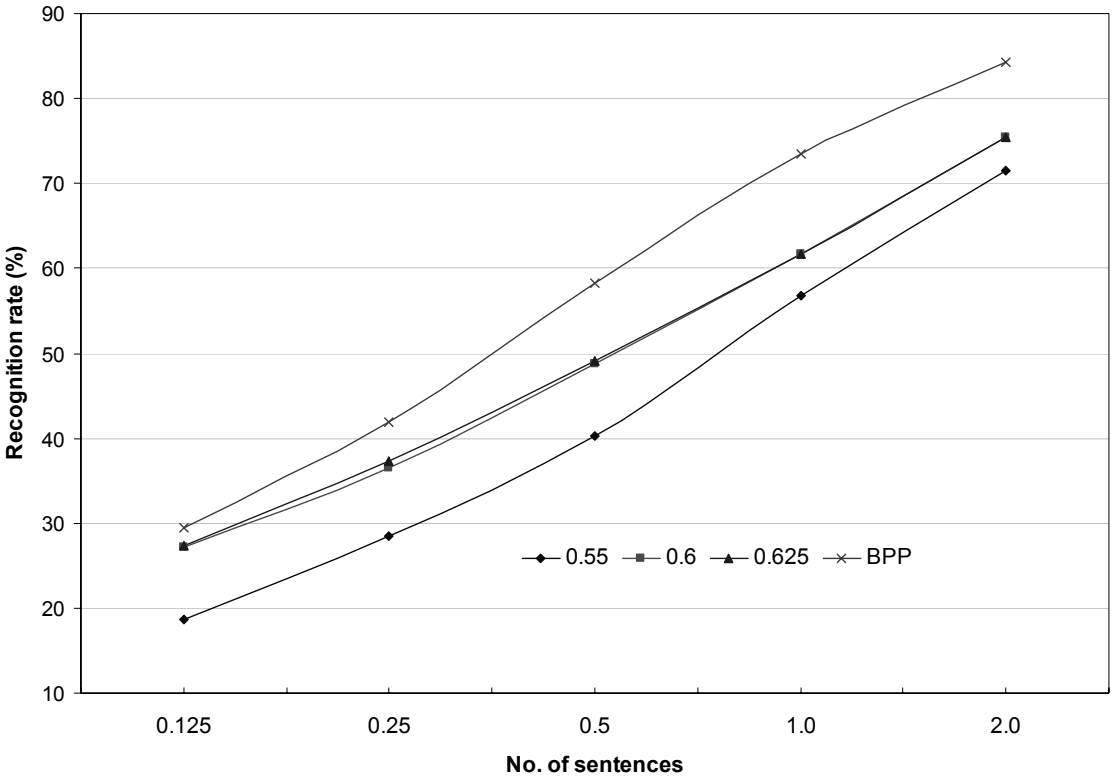

Fig. 4-2-b Performance with 102 speakers from NTIMIT database using RBPP & BPP approach. Note that each curve annotated with a number is for the RBPP case, with the number denoting the RBPP threshold level.

when using speech data greater than or equal to 3 seconds (1 sentence). This scenario however changes quite a bit when we consider the same for the NITMIT database, with the same speakers as illustrated in table 4-2-b and figure, 4-2-b.

In this case, the most obvious point to be noted is the overall, lower identification accuracy – which is expected, given the severe bandwidth limitation and signal degradation in the underlying telephone speech. That said, yet again, we notice that the biggest jump in performance occurs when the threshold changes from 0.55 to 0.6, for the same reason cited above. However, unlike the recognition rates obtained for the TIMIT database, we see that the RBPP performance is not able to match that of the BPP (plateauing at ~75 % compared to 84.3% for BPP). The highest value of threshold considered with .625 (as opposed to .75 for TIMIT) since the bandwidth reduction and added noise made the speakers so difficult to separate that thresholds of greater than .625 resulted in very little network reduction. The number of networks required at a threshold value of 0.55 is larger by a factor of 2.8 (compared to TIMIT). The number of networks still increases by a factor of ~4 when the threshold increases to 0.6.

In the next section, we give experimental results based on re-training of the re-usable neural networks (RRBPP approach).

4.3 Experiment II – RRBPP Baseline Performance

The following table (*table 4.3.a*) lists the network requirement with the RRBPP approach and those required with the RBPP approach over the TIMIT database. This is followed by the identification accuracy curves (*fig. 4.3.a*) for the RRBPP method.

Table 4-3-a Network requirement with RRBPP v/s RBPP and BPP approach over 102 speakers from TIMIT database speakers

Network Requirement			
Threshold	**RRBPP**	**RBPP**	**BPP**
0.55	94	120	5151
0.6	315	472	
0.65	753	1113	
0.7	1367	1835	
0.75	2158	2651	

We see from the above table, that the network requirement for the classification reduces by as much as 33 % at threshold value of 0.6. Nevertheless, the performance drop is negligible as

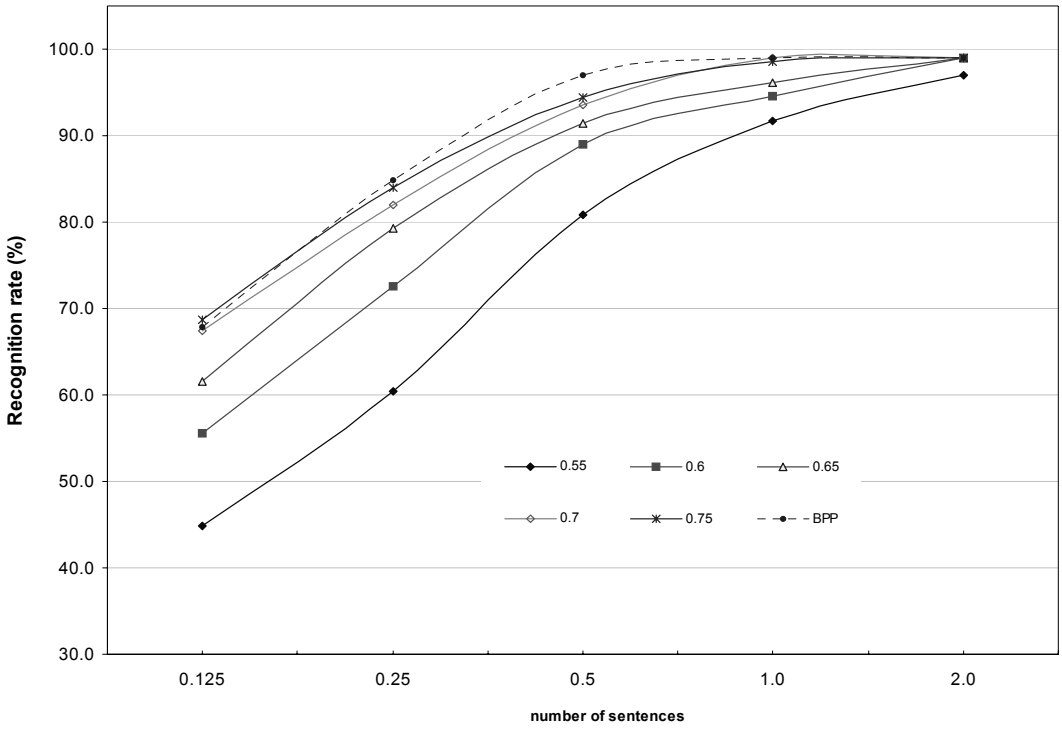

Fig. 4-3-a Performance with 102 speakers from TIMIT database using RRBPP & BPP approach Note that each curve annotated with a number is for the RRBPP case, with the number denoting the RRBPP threshold level.

compared to the BPP method and RBPP approach (less than 1 percent when evaluating using one sentence or more). Furthermore, at lower threshold values, the performance is actually better than the RBPP approach. Apart from this, here again we notice that the biggest jump in performance is between threshold values of 0.55 and 0.6. The performance rate is again almost identical at threshold values of 0.7 and above, with half a sentence (1.5 seconds of speech data) and above. With the TIMIT database, the advantage of the RRBPP approach is clear, however, yet again, this scenario changes quite a bit, for the NTIMIT database (as highlighted in table 4.3.b and figure 4.3.b).

It may be noticed that for NTIMIT, the highest threshold is again set to 0.625. The reason is that beyond 0.625, the performance does not increase appreciably, while the number of networks increases drastically. With RBPP and RRBPP the increase in recognition rates are ~1% and 1.1% respectively (one sentence), while the networks required increase 40% and 48%

respectively. As the population size increases – this network requirement increase makes the value of RBPP and RRBPP questionable. Based on these observations, the maximum threshold for NTIMIT database was set to 0.625 instead of 0.65.

Table 4-3-b Network requirement with RRBPP v/s RBPP and BPP approach over 102 speakers from NTIMIT database

Network Requirement			
Threshold	**RRBPP**	**RBPP**	**BPP**
0.55	175	252	5151
0.6	656	1024	
0.625	1143	1476	

In the case of the NTIMIT database, we notice that the highest network reduction from the RBPP approach occurs at a threshold value of 0.6 (~33 %), but as we shall see, this significant network reduction does not provide a very good performance. The overall performance does seem to be better than that obtained by the RBPP approach (with 2 sentences i.e. 6 seconds of speech or more). Nevertheless, it still is not able to quite match that offered by the BPP approach. It should be borne in mind that the all the experimental results provided so far are over 102 speakers, and

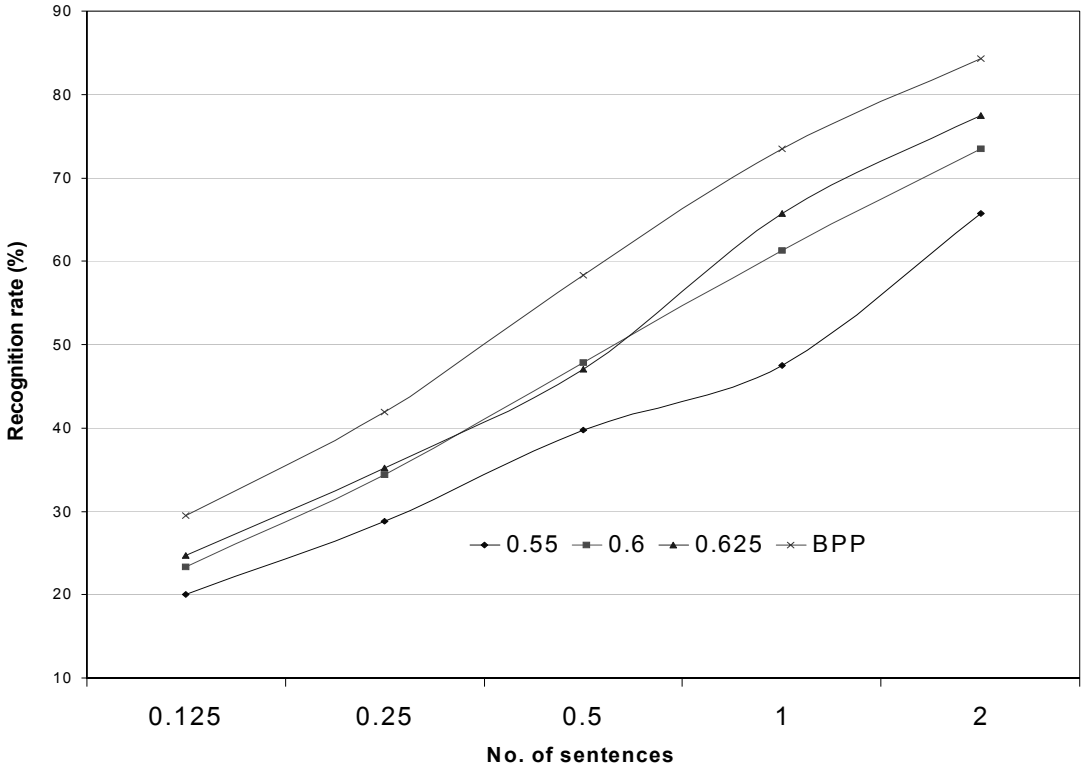

Fig. 4-3-b: Performance with 102 speakers from NTIMIT database using RRBPP & BPP approach. Note that each curve annotated with a number is for the RRBPP case, with the number denoting the RRBPP threshold level.

actually drop further, once we evaluate them over the entire database (630 speakers).

In the next two figures we observe the network requirement variation over the NTIMIT and TIMIT databases for the same speakers using RBPP and RRBPP methods.

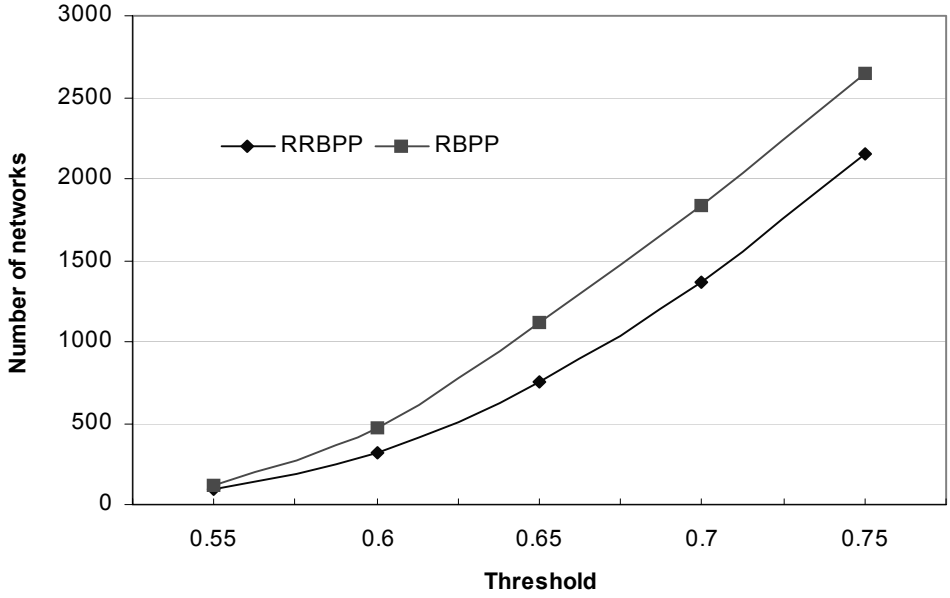

Fig. 4-3-c Network variation for 102 speakers from TIMIT database using RBPP and RRBPP approach.

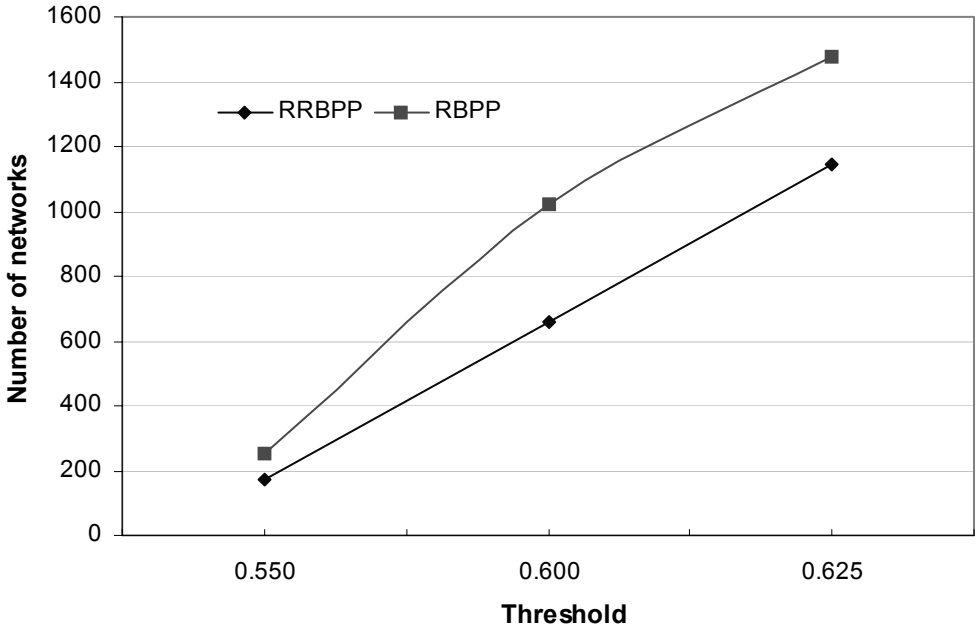

Fig. 4-3-d Network requirement for 102 speakers from NTIMIT database using RRBPP & RBPP approach.

4.4 Experiment III – Number of Features

As mentioned earlier, the number of features (per frame) is also the number of input nodes for each neural network. Tests have shown that changing the number of features changes the performance. Recall that features are one form of representing the vocal tract information present in each frame of the acoustic signal. If fewer features are available for this, the vocal tract is tracked less precisely. Conversely, it would seem that increasing the number of these features, would allow for a finer track of the information in each frame. However, increasing the number of features does not increase performance beyond a certain point, due to the "curse of dimensionality."[10] In fact, contrary to what might be expected, the performance does not even plateau once "feature saturation" takes place. Rather, beyond a certain point, the performance actually starts dropping which suggests some "optimal" value for the number of features that can be used.

With tests conducted with varying number of features, it was seen that performance rates increase with an increase in the number of features, until a value of around 25 and start dropping thereafter. This effect is illustrated in the figure 4.4.

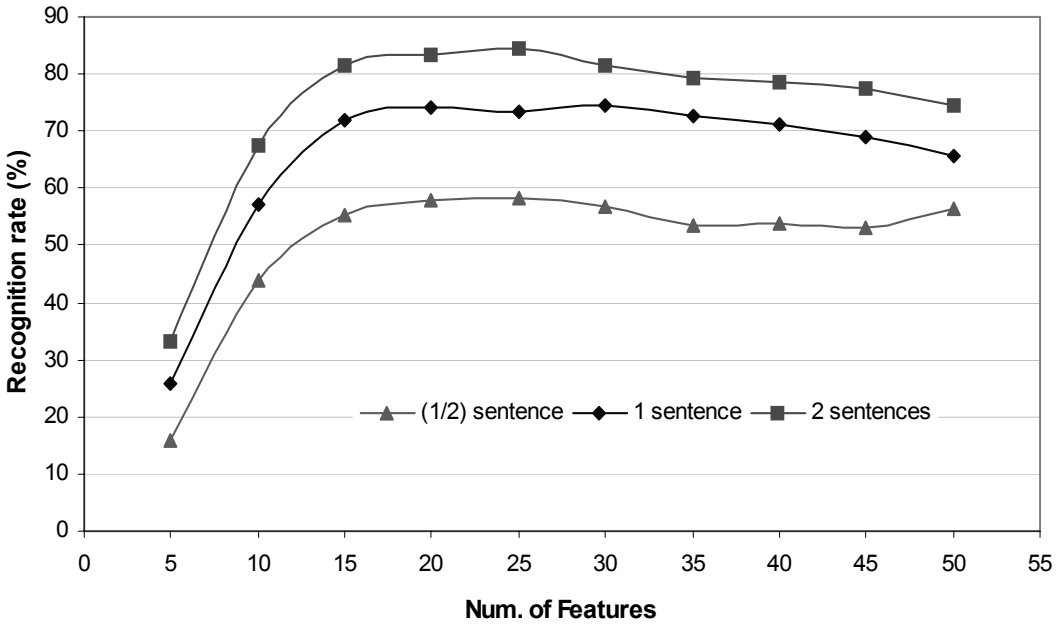

Fig. 4.4 Effect of Number of features on performance using the BPP approach with 102 speakers of DR2 from NTIMIT database

[10] This in turns leads to a requirement of an extremely and unfeasibly large training data set.

4.5 Experiment IV – Training Data Size

As mentioned earlier, the amount of testing data has a direct impact on performance of classifiers. These have already been presented in figures *4.3-a* and *4.3-b*, where we see performance does increase (though non-linearly) with respect to the number of unknown sentences available for evaluation. Correspondingly, the amount of training data also has a direct effect on such performance rates. The obvious trade-off involved is performance v/s the need to collect longer speech segments. Note that longer lengths of training speech do not necessarily require more processing time, since network training is generally fixed according to a certain total number of network updates. However, it is not always feasible to obtain longer length recordings from each speaker. Thus, once performance rates start "flattening" out (as evident – in studio quality speech – fig. 3-2) despite the increase in the training/testing data volume, then the error

Fig. 4.5-a Effect of training data size on performance and 2 Test sentences with BPP approach using 102 speakers from NTIMIT database.

rates that need to be weighed vis-à-vis the inconvenience of more training data. To illustrate the effect of training data volume on performance, we present figures 4.5-a and 4.5-b, based on performance figures obtained by conducting BPP test with 102 speakers from NTIMIT and TIMIT database respectively, with 2,4, 6, 8 sentences used for training and 2 sentence (SX) used for testing in each instance.

As we see, the performance increases by a factor of ~2, when the number of training sentences increases from 2 to 4. It again jumps considerably, when there are 6 training sentences to be used. However the increase between 6 and 8 sentences is not very appreciable, probably due

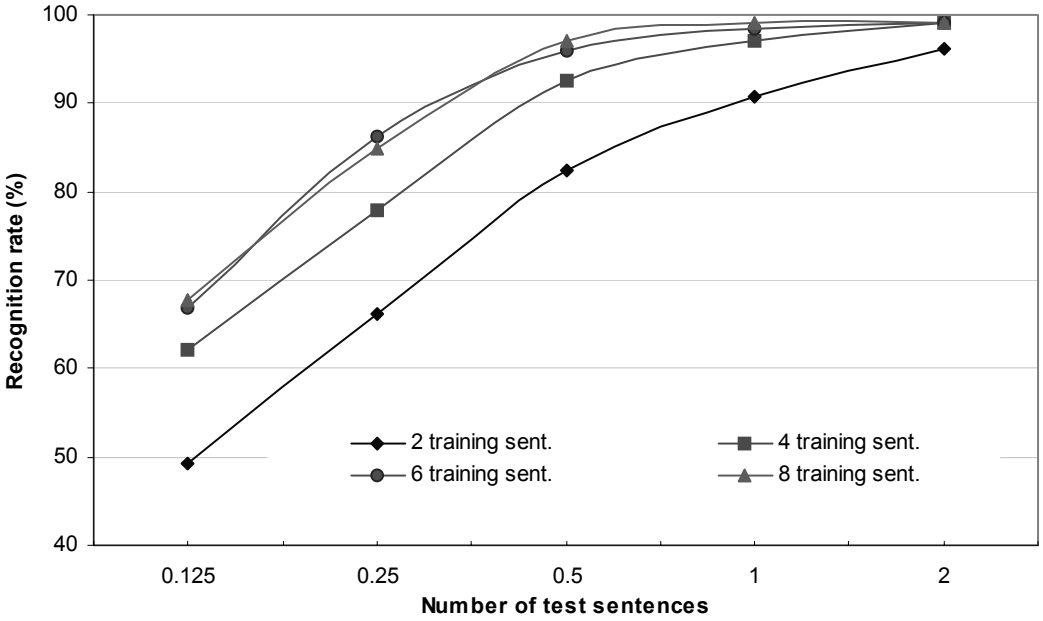

Fig. 4.5-b Effect of training data size on performance with 2 Test sentences with BPP approach using 102 speakers from TIMIT database.

to the fact that the amount of information available in TIMIT features is exhausted.

In case of NTIMIT, there is again a consistent performance increase, with an increase in the number of training sentences, and the final increase in performance when the training sentences change from 6 to 8 indicate that there more training data is likely to be desirable.

4.6 Experiment V – Multiple Choice Criteria (*"N"* Top Choices)

So far, all the results that have been presented have been based on considering only whether or not the highest scoring speaker is correct or not. As mentioned previously, the "scores" are computed by averaging all the neural network outputs from all frames, summing over time, and then combining them to give indications for each speaker. If we change the acceptance criteria to consider a speaker as correct, if the score for that speaker is one of the "top n" scores, then the "recognition" rate increases as a function of *n.* Consider figure (4.6) – which shows "recognition rates considering the *Top N* categories from the NTIMIT database. Only

NTIMIT is being presented, since the dramatic increase in performance is more obvious here. Note that speaker identification typically considers only the top category (Top 1 choice or the

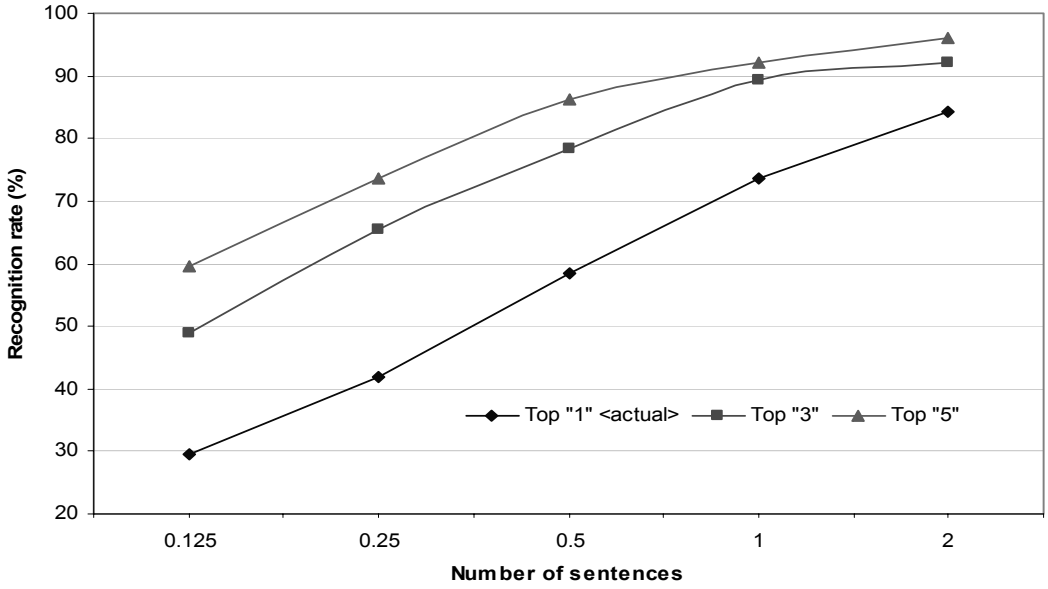

Fig. 4.6 Performance with 102 speakers from NTIMIT database considering top "*N"* categories (for N=1, 3 & 5), using RBPP method at a threshold value of 0.60.

"bottom" curve in fig. 4.6), and as we have already noticed that neither RBPP nor RRBPP methods are able to approach the BPP performance, this performance measure opens up a new approach towards the classification task.[11]

The figure clearly shows that if we consider 3 "best" categories as an acceptable region, the performance rate jumps by a factor of almost ~1.5 and almost doubles if this region is expanded to account for the top 5 categories. Now considering that the total population is 102 categories, these correspond to correct choice being in the top 2.94 % and 4.9% respectively of the population. That said – it is arguable that the speaker identification process should ultimately yield a *single category result* – which leads to the alternative line of thought involving a two-step process that may be considered as follows.

Given a classification task for *"M"* speakers, we can use the *RBPP or RRBPP* approach to narrow down the "possible" speakers to the top *z*% sub-set of the population. Obviously the classifier overhead would be considerably reduced (compared to the BPP approach as already

[11] Provides scope of further investigation.

illustrated). Finally, to arrive at a decision, we employ BPP classifiers for a "*p"* speaker set, where $\frac{Mz}{100} = p$. That is, "top" *p* choices.

This directly implies that the number of BPP classifiers needed would be roughly $\frac{z^2}{10^4}$ times the original number needed for the entire speaker population and this, coupled with the significant reduction that occurs due to the re-usage and/or re-training of network, *may* lead to a feasible solution to the overall classification issue.

4.7 Experiment VI – Effect of Changing the Testing /Training Data

In this section we attempt to study the affects of the methods (BPP, RBPP and RRBPP), when:

- Training data is changed while testing data is constant.
- Testing data is changed, while training data is constant.

In the first part, where training data was variable, we trained the classifiers (in all the three methods) using two training configurations. For *Configuration 1,* 1 SI and 3 SX sentences were used for training, while for configuration *2,* 2 SA and 2 SI sentences were used for training. The testing with both configurations was done using 2 SX sentences (SX 9 and SX 10). Also, the

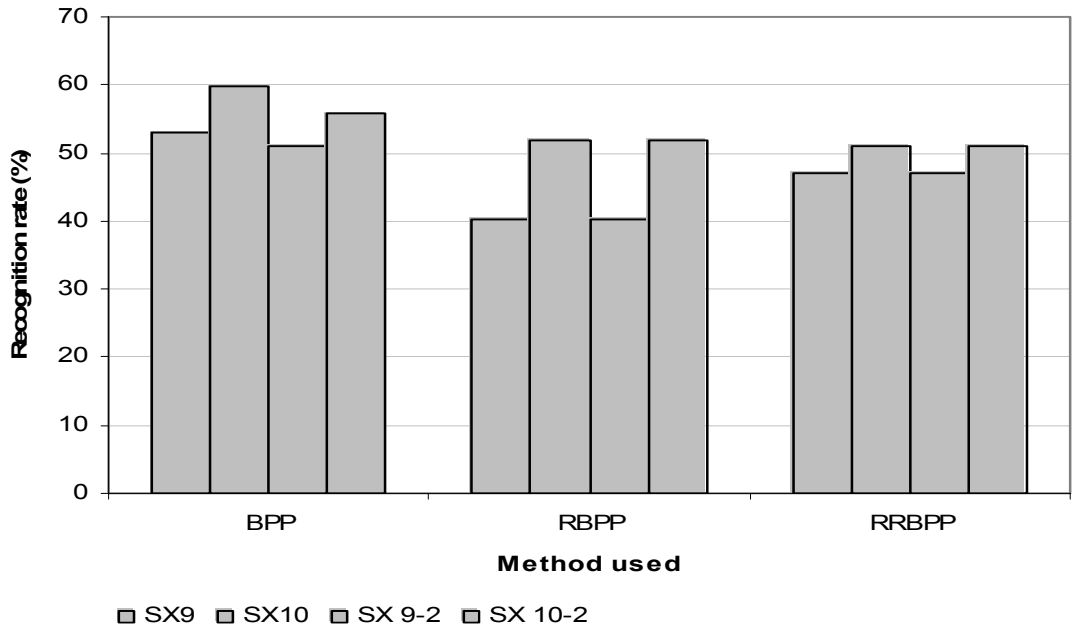

Fig. 4.7-a Effect of different training data, over fixed testing data. Testing done over sentence SX9 and SX10. SX 9-2 and SX 10-2 mean recognition with second configuration.

RBPP and RRBPP were evaluated at a threshold of 0.60. All tests were done using 102 speakers from dialect region 2 of the NTIMIT database. The performance of the three methods for both the configurations is given in fig. 4.7-a.

From the figure (4.7-a), we notice that the recognition rate for each sentence is nearly the same irrespective of the training data used. This seems to hold true for all three methods. Especially for the RBPP and RRBPP methods, the recognition rates are nearly identical for each of the training configurations (there *are* however, some differences in recognition rates as the test sentence changes, and also between RBPP and RRBPP). This is significant, given that the in the second training configuration, there are no SX sentences. What this point towards, is that the effect of training data variability is very minimal with re-usable or re-usable and retraining networks.

In the second test, we studied the effect of changing the training data, while keeping the training data fixed. The training was done over there SX and three SI sentences (6 sentences total), while testing was done using two SX (SX 9, SX 10) and two SA sentences (SA 1, SA2), but testing each sentence individually. Here again, the RBPP and RRBPP were evaluated at a threshold of 0.60 and all these performance values are presented in fig. 4.7-b.

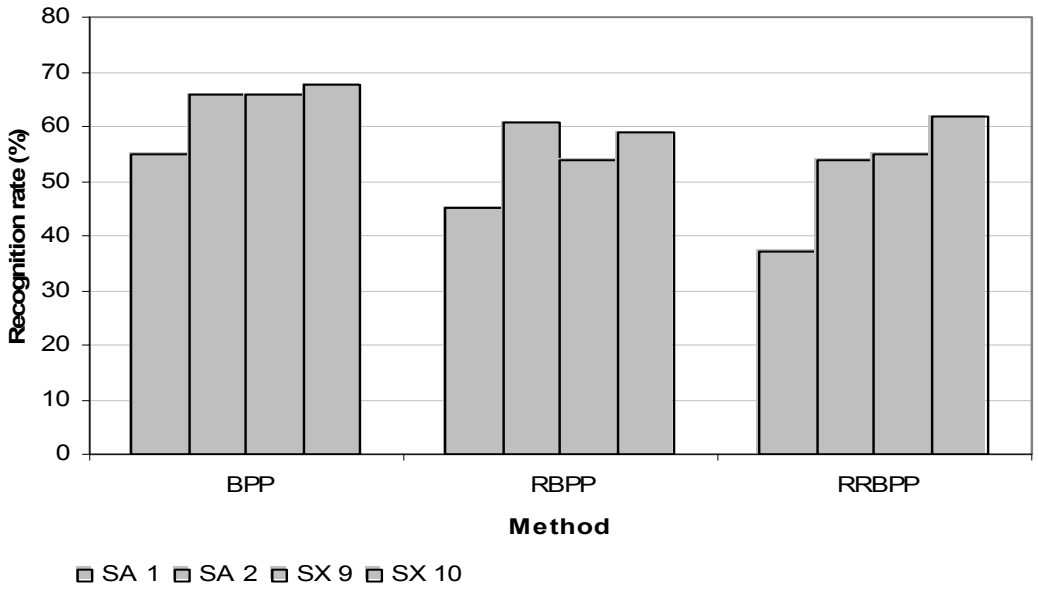

Fig. 4.7-b Effect of different testing data, with fixed training data. Testing was done with SA1, SA2, SX9 and SX10.

From the figure (4.7-b), we can see that with training data fixed, the performance changes as test data changes. In general, for each method, performance is better with the SX sentences than with the SA sentences. Thus it would appear that sentences which are phonetically balanced (SX) are

more effective for speaker identification over telephone lines (compared to those sentences designed to highlight dialect differences i.e. SA sentences). Finally it should be noted that performance results in fig. 4.7-b are higher than that in fig. 4.7-a. This is likely due to the fact that six training sentences were used for the results in fig 4.7-b, versus four training sentences for the results shown in fig 4.7-a.

In summary, the experiments of this section indicate that the phonetic content of training data is not especially important, but as noted both here and in a previous section the total length of training data is very important. For the case of test data, both the phonetic content and the length affect speaker recognition accuracy.

4.8 Experiment VII – Identification Rates with the Entire NTIMIT/TIMIT Database

So far, we have presented results based on 102 speakers corresponding to the dialect region 2, from both TIMIT and NTIMIT database. These pilot tests were conducted to arrive at an "optimal" test configuration. One of the important findings of these tests was that for the TIMIT database, the optimal threshold level is 0.65. Beyond this threshold value, the amount of training time is not justified by the marginal increase in identification rate. For NTIMIT, this threshold value was fixed at 0.6.[12]

Table 4.8-a Networks needed for the entire TIMIT database using BPP, RBPP and RRBPP methods

Thresh	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**BPP**
RBPP	302	2597	12317	32572	68063	198135
RRBPP	280	1971	8334	24721	55875	198135

Table 4.8-b Networks needed for the entire NTIMIT database using BPP, RBPP and RRBPP methods

Threshold	**0.55**	**0.575**	**0.60**	**0.625**	**BPP**
RBPP	3455	12238	24060	44839	198135
RRBPP	2998	9898	10502	30767	198135

[12] Training time for NTIMIT database for a threshold value of 0.65 was almost 30 *days*, using dual 1 GHz, PIII computer, with 512 MB RAM and running Microsoft Windows 2000 Pro.

Identification accuracy is plotted in figures 4.8-a, 4.8-b for TIMIT and NTIMIT respectively, with identical format. The numbers of networks used for various threshold values are also shown

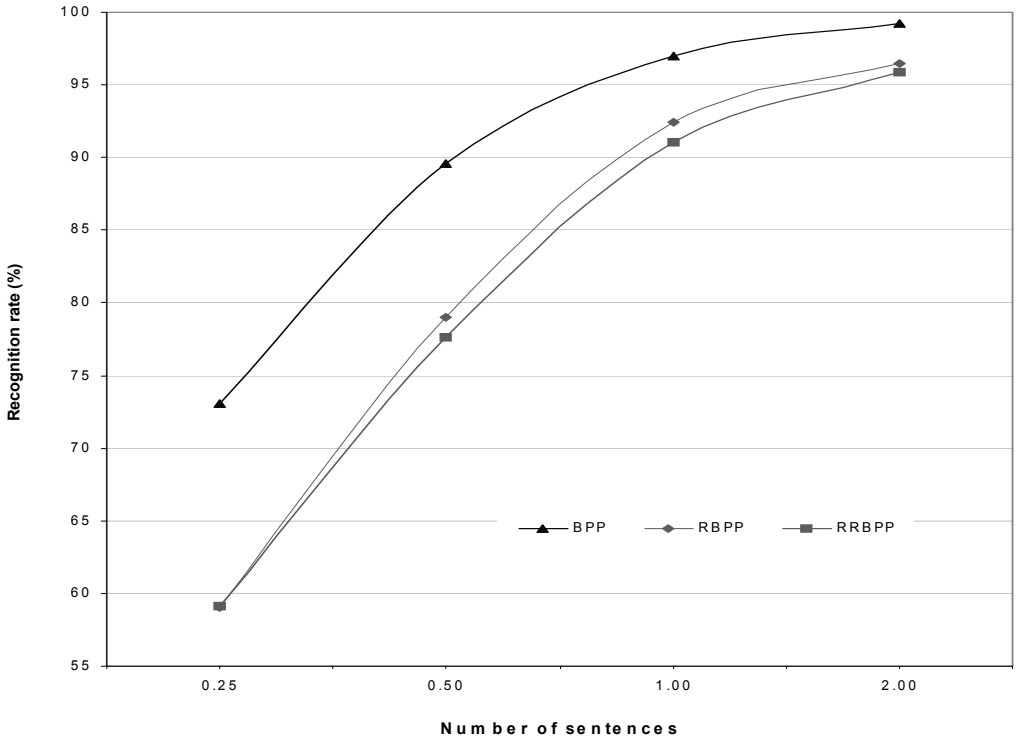

Fig. 4.8-a Accuracy of speaker identification with TIMIT database using RRBPP, RBPP (0.65 threshold) and BPP over all the 630 speakers.

in tables 4.8-a, 4.8-b. The most dramatic difference between this case and the data from TIMIT is that the overall accuracy is severely degraded, as expected. For this case, performance of the RBPP and RRBPP method are approximately equal for the two thresholds depicted. Additionally for each threshold value, many more networks are needed than for the case of TIMIT. Finally, and very importantly from the point of view of the present research, there is much more degradation in performance with both RBPP and RRBPP (versus BPP), than for the case of TIMIT.

Also presented is the effect of considering top "N" choices with TIMIT database over the entire 630 speakers, using the RRBPP approach in fig. 4.8-c. Once again, we can see that this method has a very good identification rate, given the number of classifiers it uses. However, since

the performance of reusable networks is not very satisfactory over the entire 630 speakers, in fig. 4.8-d, we present the effect of considering top "n" choices, with the NTIMIT database, but using

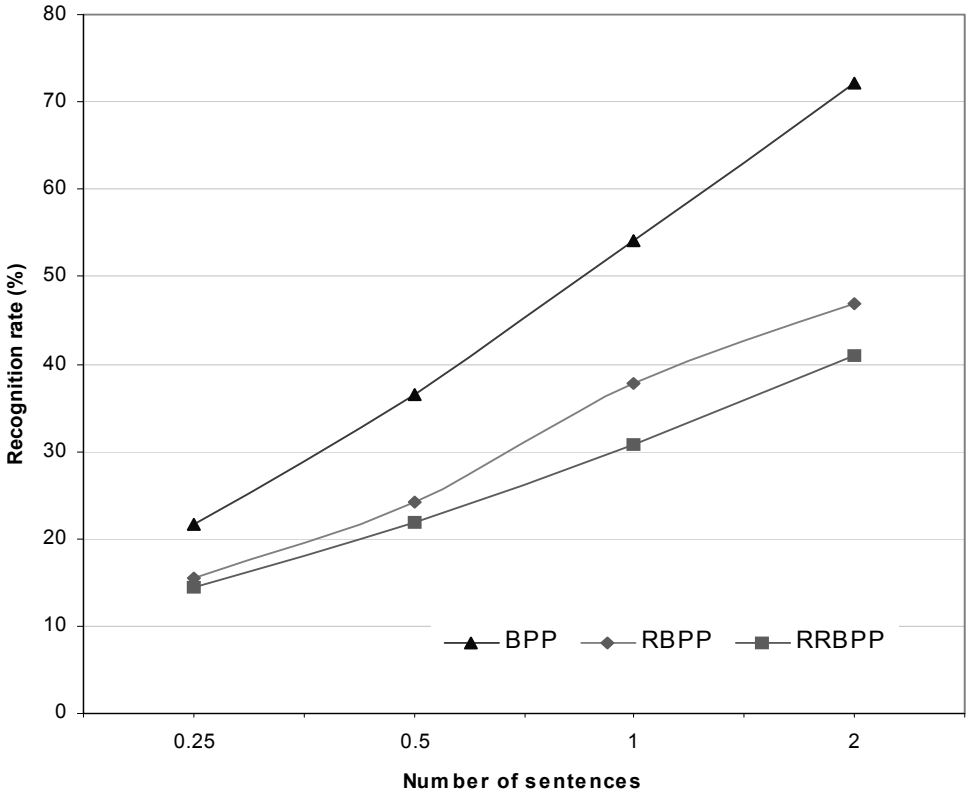

Fig. 4.8-b Accuracy of speaker identification with NTIMIT database using RRBPP, RBPP (0.6 threshold) and BPP over all the 630 speakers.

the BPP approach. One inference that is drawn from this is that despite the fact that the performance of the BPP approach is better compared to RBPP and RRBPP method, it still leaves scope for extensive research, in the form of methods for using better features, etc, since the performance is still some what low (~70 % with top choice and ~92% with top 10 choice).

In this chapter, we presented several experiments, which were conducted over 102 speakers belonging to the dialect region 2 from both NTIMIT and TIMIT databases. One of the primary reason for fixing the training and testing data as mentioned at the onset of this chapter, was to follow the setup mentioned in the work [4]. This allowed us to use their identification results as a valid reference for comparative studies. Apart from that, 102 speakers is a reasonable figure of speakers to work with, especially when conducting these pilot tests, as the amount of computational time involved is less than 1/30 of that required for tests with all 630 speakers.

In the next chapter, we conclude this thesis by summarizing all the work done and presented in the preceding chapters, along with a brief description of a recently developed application for performing *real time* speaker identification using the BPP networks as the underlying recognition engine.

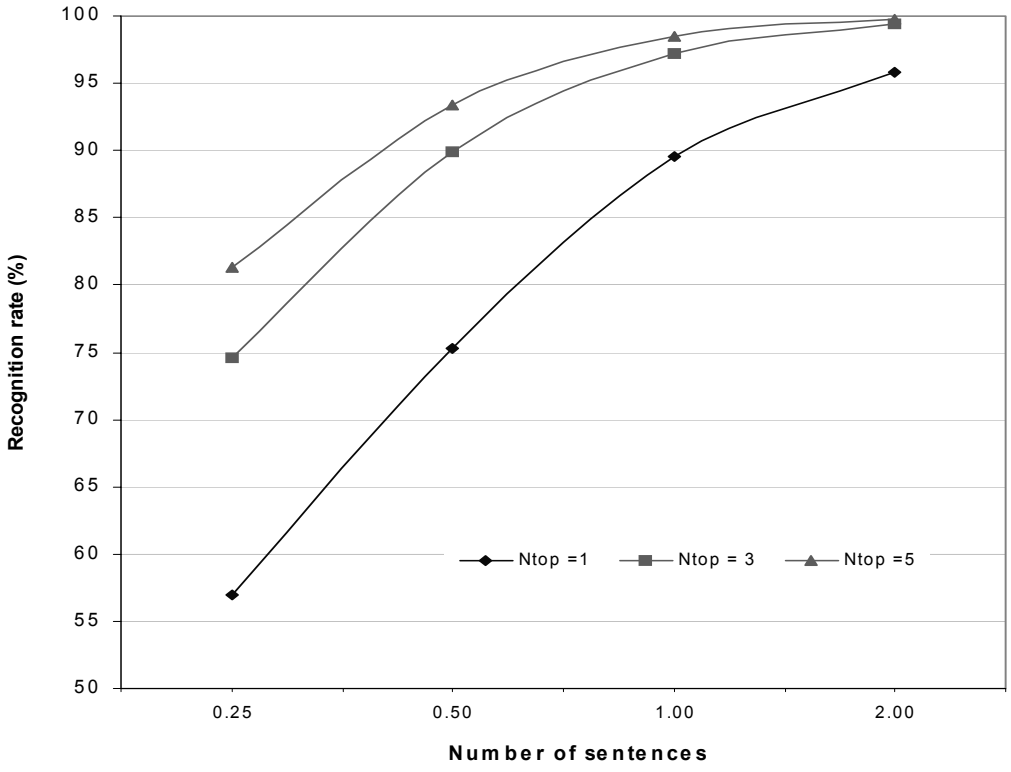

Fig. 4.8-c Effect of considering Top N choices (1≤N≤5) with TIMIT database for RRBPP for threshold of 0.65.

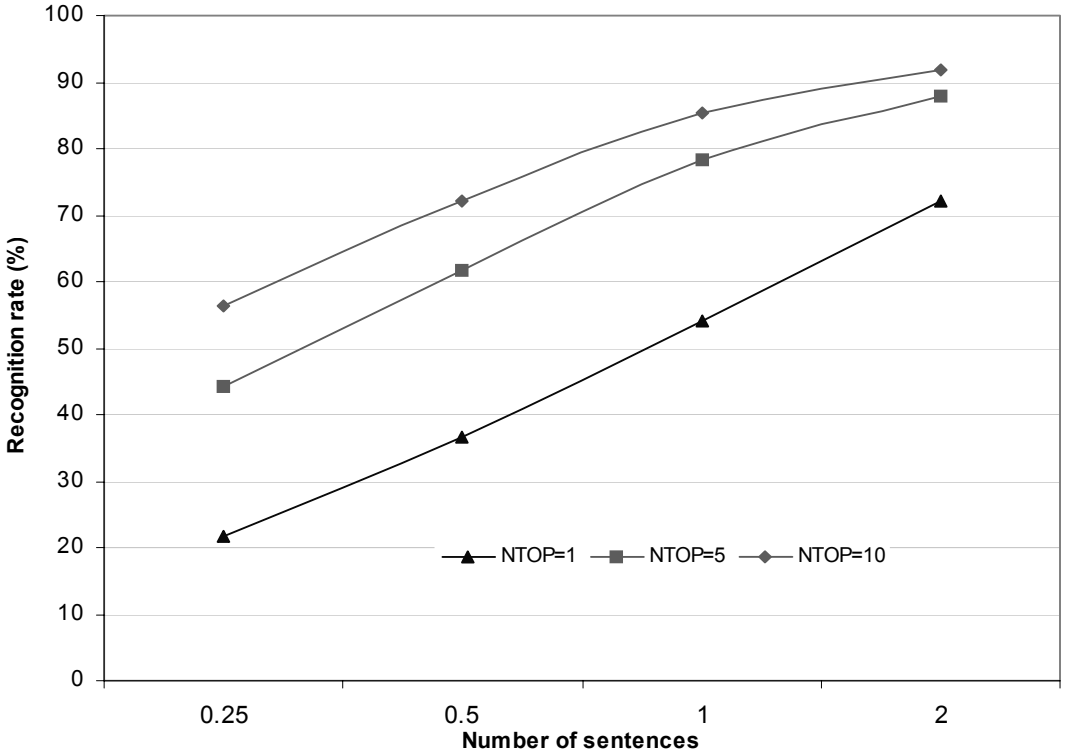

Fig. 4.8-d Effect of considering top N choices (1≤N≤10) with NTIMIT database using the BPP method.

CHAPTER V

CONCLUSIONS AND INFERENCES

There were several key reasons behind this research. One of the most important reasons was that, prior to this work there had been no thorough study addressing the behavior of speaker identification systems over the *entire* NTIMIT and TIMIT databases. Studies had dealt with 200 speakers or less which could primarily be attributed to the fact that all these tests are extremely time consuming and given the computational resources available, it was not particularly feasible until the past few years.

Additionally, the shift towards the requirement for robust speaker identification schemes for telephone speech has warranted an in-depth study of methods for reducing the overall time required for pre/post processing involved using telephone speech, while simultaneously maintaining or improving accuracy. The binary partitioned network scheme, as illustrated in this thesis, has the fundamental drawback of requiring an extremely large number of classifiers for the classification task, which in turn has other implications.

This drawback leads to the study of methods for reducing the number of classifiers required, namely the RBPP and RRBPP methods. Unfortunately, despite providing extremely encouraging results with clean speech, the performance with telephone quality speech is not very acceptable – as compared to that obtained with the BPP method or by other researchers employing other methods and/or models. The best-known *identification performance* with the telephone quality speech with one test sentence (~3 sec.) over the entire 630 speakers is that of 60.7 %. The corresponding result with BPP, RBPP and RRBPP methods are ~54 %, 38% and 31%. The low performance figures for telephone speech have led to several studies, which have concluded that there are degradation factors in telephone speech in addition to the bandwidth limitation and transmission channel noise. We also observe that the phonetic content of training data is not especially important, but the total length of training data is very important. For the case of test data, both the phonetic content and the length affect speaker recognition accuracy. From experiments in section 4.7, we also observe that it is easier to lose the variance due to the dialect in an acoustic signal after transmission over a telephone channel, more than anything else, and this variance provides a good feature space for speaker identification.

We now present a brief description of a real-time speaker identification system, which has been developed using the basic BPP method as its core classification engine. This application is the first attempt at providing a *tangible* application of the research done with BPP networks.

5.1 Real-Time Speaker Identification

This application is the first attempt at applying the methods discussed in this thesis for the purpose of real-time speaker identification. The BPP was selected as the core classification engine because of its superlative performance. However, this does not rule out the feasibility of using the RBPP/RRBPP approach in future implementations, especially since the real time system can be used with "clean" speech. To mention the basics of the real time system, the entire application was written in MATLAB and then compiled to a stand-alone C++ console application (using the MATLAB compiler). The entire structure of this application consists of an overall shell that binds the voice capture module, acoustic signal parameterization module, and the core classifier generation and the complimentary evaluation module.

As mentioned earlier, since speaker id requires a closed database, speakers are required to “enroll” into the database by way of providing a 10 sec. speech sample, and a user ID that is later used to link each name to a number. This ID (which is a two character string and meant to represent the initials of the speaker) is checked for uniqueness before being accepted. Provision has been made for rejection of a speech sample during the enrolment/identification stage, if the acoustic signal has any unwanted effects (present due to excessive speaker volume, microphone gain being too low/high, etc). It should also be noted that once the character id is entered, it is assigned a numerical serial number which is matched against the serial number returned during the identification stage. The application uses this serial number to return the character ID.

Once a speaker has provided the acoustic sample for enrollment, this sample is parameterized and written to a database, followed by an invocation of the binary classifier generator, that takes all N+1 such speaker files (assuming that N speakers have already enrolled), and generates N classifiers corresponding to the speaker pairs comprised of the newly added speaker with the existing N speakers. There are provisions to add other binary classifiers, if any of the needed classifiers are not already trained. During the evaluation/identification phase, any member of the database is expected to provide only a 10 sec. speech sample, which is again parameterized and passed through all the classifiers resulting in a “score” vector for each of the potential speakers. The speaker with the highest score is returned as the one identified, using the ID label originally given to that speaker. Other parameters (number of features, training iterations, etc.) have been kept same.

The performance results have been very encouraging so far and, despite the entire application being in the initial stage, shows a lot of promise. For diagnostic purposes, the application has been programmed to return character Ids corresponding to the top two scores, which also has anecdotal aspects, since the chances are that a high pitched male present in the database, maybe be declared to sound "dangerously" close to normal voiced female and vice-versa!

Another important measure of performance of a real-time system is the time required to return a verdict. So far, the time required to return with a verdict, *after* reading the network weights, is around 2~3 seconds (which is not really expected to increase very significantly with an increase in the speaker population)[13]. Several improvements for this application are in the process of development, namely a Graphical User Interface, improved network I/O, and a better control of the front end parameters. It is also planned to provide an alternative classification core using the RBPP/RRBPP methods, which will reduce the identification time.

5.2 Suggestions for Future Work

As mentioned in section 4.6, a two step approach could be used for speaker classification, using a relatively small number of classifiers to first narrow the speaker search to one of a small number of possible speakers, with a more detailed second step of classification to determine the specific identity of the unknown speaker. This concept could be explored further. The RBPP and RRBPP methods could also be more extensively examined and possibly improved. One possibility for improvement would be to examine the midpoint of the each neural network output for each pair of speakers that each network is intended to separate. The basic assumption in this work was that the midpoint of each network for each speaker pair is 0.5. This assumption was tested and found to hold quite well for the training speakers in the BPP method. However, the assumption was not examined for the cases of the reusable networks. If the network midpoints do vary for the various speaker pairs separated by a single network, this information could be used to improve the speaker classification.

Another aspect that can be explored is an application of the multi-state predictive neural networks. As mentioned in the section 2.3.4, there is a definite possibility of using one set of classifiers to categorize the acoustic segment into a broad category – say *type of utterance.* Thereafter, using a multi-sate model (where *states* represent the past frames and re-usable

[13] Based on a dual PIII ~ 1 GHz, with 512 MB RAM, current population of 17 speakers, and uses 10 sec. of speech, sampled @ 11025 Hz.

networks) we may compare which speaker has the highest probability of producing the next frame for the given *type* of acoustic sample.

References

[1] O'Shaughnessy, Douglas (1986) "Speaker Recognition", IEEE ASSP Magazine, pp. 4 – 17.

[2] Gish, H., and Schmidt, M. (1994), "Text-Independent Speaker Identification," IEEE Signal Processing Magazine, pp. 18-32.

[3] Matsui, T. and Furui, S. (1991), "A Text-Independent Speaker Recognition Method Robust Against Utterance Variations," Proc. ICASSP-91, pp. 377-380.

[4] Reynolds, D. A., Zissman, M. A., Quatieri, T. F., O'Leary, G. C., Carlson, B. A. (1995), "The Effects of Telephone Transmission Degradations on Speaker Recognition Performance," Proc. ICASSP-95, pp. 329-332.

[5] Reynolds, D.A., and Rose, D.A. (1995), "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," IEEE Trans. Speech, Audio Processing, vol. 3, pp. 72-83.

[6] Rudasi, L. and Zahorian, S. A. (1991), "Text-independent Talker Identification with Neural Networks," Proc. ICASSP-91, pp. 389-392.

[7] Rudasi, L. and Zahorian, S. A. (1992), "Text-Independent Speaker Identification using Binary-pair Partitioned Neural Networks," Proc. IJCNN-92, pp. IV: 679-684.

[8] Zahorian, S. A. (1999), "Reusable Binary-Paired Partitioned Neural Networks for Text-Independent Speaker Identification, Proc. ICASSP-99, pp. II: 849- 852

[9] Duda R.O, Hart Peter E. and Stork, David G. "Pattern Classification" 2ed, (Oct. 2000) Wiley Interscience, New York, pp. 282-318.

[10] Rose, R.C, Hofstetter, E.M and Reynolds, D.A. (1994) “Integrated Models of Signal and Background with Application to Speaker Identification in Noise”, IEEE Trans. Speech, Audio Processing, vol.2, no.2, pp. 245 – 257.

[11] Gopalan, K, Anderson, Timothy R. and Cupples, Edward J. (1999) “A Comparison of Speaker Identification Results using Features Based on Cepstrum and Fourier – Bessel Expansion”, IEEE Trans. Speech, Audio Processing, vol.7, no.3, pp. 289 – 294.

[12] Murthy, Hema A., Beaufays, Francoise, Heck, Larry P, Weintraub, Mitchel (1999) IEEE Trans. Speech, Audio Processing, vol.7, no.5, pp. 554 – 568.

[13] Reynolds, D.A (1994) “Experimental Evaluation of Features for Robust Speaker Identification”, IEEE Trans. Speech, Audio Processing, vol.2, no.4, pp. 639 – 643.

[14] Bennani, Y. and Gallinari, P. (1991), "On the Use of TDNN-Extracted Feature Information in Talker Identification," Proc. ICASSP-91, pp. 385-388.

[15] Besacier, L., Bonastre, J.F. (1998) “Frame Pruning for Speaker Recognition”, Ref. 0-7803-4428-6/98 IEEE, pp. 765 – 768.

[16] Proakis, John G., Manolakis, Dimitris G. “Digital Signal Processing 3ed” (Oct. 1995) Prentice Hall, New Jersey.

[17] Artiéres, T., Gallinari, P (1995) “Multi state predictive neural networks for text independent speaker recognition”, LAFORIA UA CNRS 1095 Tour 46-00, Boite 169 Universit. Paris 6, 4 place Jussieu 75252 Paris cedex 05 France. http://www-connex.lip6.fr/articles.html#1995

[18] Hattori H., (1992) “Text Independent Speaker Recognition using neural networks”, ICASSP, vol. 2, pp. 153 – 156.

[19] R. Lippman, (1987) “An introduction to computing with neural nets”, IEEE ASSP Magazine, April, pp. 4-22.

[20] Zaki B. Nossair, (1989) “Dynamic Spectral Shape features as acoustic correlates for stop consonants”, Dept. of Electrical and Computer Engineering, Old Dominion University.

Appendix

A.1 Speaker Average

The following is the pseudo-code for calculation of the speaker average, as calculated for an entire sentence(s) over all the frames.

<***While*** *frames <= total training frames for the speaker*>

$$y_j = \left(\sum_{i=1}^{n} w_{ij} x_i \right) + w_{bj}.1$$; j=1:10 (nodes in hidden layer); n is the # of inputs.=25

$$out_j = \left(\frac{1}{1+e^{-y_j}} \right)$$; Output at hidden node j

$$k_j = \left(\sum_{j=1}^{5} out_j w'_j \right) + w_{out}.1$$; This is the input at the output node.

$$output = \left(\frac{1}{1+e^{-out_j}} \right)$$; Final output for the frame;

Sum = sum + output;

Frame = frame+1;

Return ;>

Average output = sum/frames;

A.2 Speaker Index / Speaker Pair

Recall, that each classifier is trained over *unique* speaker *pairs*, which are considered in an orderly manner. This leads to a unique *index* for each pair, depending upon the index of each speaker. So starting from pair 2, 1 (index →1) we have the speaker pair index up to pair 630,629 (index→198135).

This speaker pair index is computed as:

$$index = \left(\frac{(sp1-1)(sp2-1)}{2} \right) + sp2 - 1$$, Where *sp1* and *sp2* are speaker indexes.

However, to obtain the individual speaker indices, we just iterate through all the pairs, incrementing a counter, till the counter equals the pair index.

A.3 Front End Parameters (NTIMIT)

```
// Basic parameters
Sample_rate:            16000    // Hz          8000  - 22050 Hz
Segment_time:           10000    // ms          50  -   500 ms
Frame_time:             40       // ms          5  -    40 ms
Frame_space:            10       // ms          2  -   100 ms
FFT_length:             1024     // points      64  -  1024 points
Kaiser_Window_beta:     6        // unit less   0  -     6
Num_DCTC:               25       // unit less   8  -    25
DCTC_warp_fact:         0.25     // unit less   0  -     1
BVF_norm_flag:          0        // unit less   0  or    1
Low_freq:               300      // Hz          0  -   300 Hz
High_freq:              3400     // Hz          3000  -  8000 Hz
Prefilt_Center_Freq:    3200     // Hz
Spectral_range:         45       // dB

// Morphological filter parameters
Freq_kernel_before:      25   // Hz            0  -   100 Hz       long
Freq_kernel_after:       50   // Hz            0  -   100 Hz       long
Time_kernel_before:      0    // frames        0  -     5 frames   long
Time_kernel_after:       0    // frames        Currently fixed to 0      long

// Block parameters
Block_length_min:        1   // frames         1  -    20 frames   long
Block_length_max:        1   // frames         1  -    20 frames   long
Block_jump:              1   // frames         1  -    20 frames   long
Num_DCS:                 1   // unit less      1  -     5          long
Time_warp_fact:          0   // unit less      0  -    10          float
BVT_norm_flag:           0   // unit less      0  or    1          long
```

A.4 TIMIT/NTIMIT Database Structure

Both the TIMIT and NTIMIT databases have 630 speakers who are divided into 8 sub-categories (dialect region), and the gender-wise composition of each dialect region is as follows.

Dialect Region (dr)	#Male	#Female	Total
1	31 (63%)	18 (27%)	49 (8%)
2	71 (70%)	31 (30%)	102 (16%)
3	79 (67%)	23 (23%)	102 (16%)
4	69 (69%)	31 (31%)	100 (16%)
5	62 (63%)	36 (37%)	98 (16%)
6	30 (65%)	16 (35%)	46 (7%)

7	74 (74%)	26 (26%)	100 (16%)
8	22 (67%)	11 (33%)	33 (5%)
8	438 (70%)	192 (30%)	630 (100%)

The dialect regions are:

dr1: New England	dr2: Northern	dr3: North Midland	dr4: South Midland
dr5: Southern	dr6: New York City	dr7: Western	dr8: Army Brat

CURRICULUM VITA

ASHUTOSH MISHRA

OBJECTIVE

Active research and academic involvement.

EXPERIENCE

1999–2001 LML – Vespa Kanpur, India

Sr. Engineer (R and D)

- Developed the first (pre-fab) simulator for design and testing of A.C alternators for a range of motorcycles manufactured.

2000–2001 Harcourt Butler Technological Institute Kanpur, India

Visiting faculty (Dept. of Electrical Engineering)

EDUCATION

Fall 2001 ~ present Old Dominion University Norfolk, VA

- M.S (Electrical & Computer Engineering), Grad. – May. 2003
- Best Graduate Teaching Assistant (2002)
- *Ashutosh Mishra, S.A Zahorian - 'Reusable and retrainable binary pair partitioned neural networks for speaker identification', Euro speech, 2003*

1995–1999 Mangalore University Karnataka, India

- B.E., Electrical & Electronics Engineering
- Graduated First Class (Distinction in final year).

INTERESTS

Algorithm Development, Code Optimization, Digital Signal Processing, Automatic Speaker Recognition, Neural Networks, Reading & Writing.

RESEARCH RELATED

- Extensive programming experience in MATLAB, FORTRAN, MS VB/VC.
- Primary OS – MS WINDOWS 9x/2000/XP

FAX (757) 683-4549 • E-MAIL: WIREDASHU@HOTMAIL.COM
1334, W 42ND ST. • NORFOLK, VIRGINIA 23508 • PHONE (757) 515-4804