Text-Independent Automatic Speaker Identification Using Partitioned Neural Networks

Laszlo Rudasi
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds
Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation
https://digitalcommons.odu.edu/ece_etds/190

This Dissertation is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.
TEXT-INDEPENDENT AUTOMATIC SPEAKER IDENTIFICATION

USING PARTITIONED NEURAL NETWORKS

by

Laszlo Rudasi

B.S.E.E. May 1982, The University of Akron
M.E.E.E. May 1986, Old Dominion University

A Dissertation submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY
ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 1992

Approved by:

Stephen A. Zahorian (Director)
ABSTRACT

TEXT-INDEPENDENT AUTOMATIC SPEAKER IDENTIFICATION USING PARTITIONED NEURAL NETWORKS

Laszlo Rudasi
Old Dominion University, 1992
Director: Dr. Stephen A. Zahorian

This dissertation introduces a binary partitioned approach to statistical pattern classification which is applied to talker identification using neural networks. In recent years artificial neural networks have been shown to work exceptionally well for small but difficult pattern classification tasks. However, their application to large tasks (i.e., having more than ten to 20 categories) is limited by a dramatic increase in required training time. The time required to train a single network to perform N-way classification is nearly proportional to the exponential of N. In contrast, the binary partitioned approach requires training times on the order of N^2. Besides partitioning, other related issues were investigated such as acoustic feature selection for speaker identification and neural network optimization.

The binary partitioned approach was used to develop an automatic speaker identification system for 120 male and 130 female speakers of a standard speech data base. The system performs with 100% accuracy in a text-independent mode when trained with about nine to 14 seconds of speech and tested with six to eight seconds of speech.
ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Stephen A. Zahorian, for his invaluable time, patients, advice, and support. Without his guidance this dissertation would not have been possible. I would also like to thank the other members of my committee, Dr. John Stoughton, Dr. David Livingston, and Dr. Michael Overstreet, for their generous assistance and time.

Portions of this work were supported by grant IRI-9003226 from the National Science Foundation.

I dedicate this work to my two wonderful children, Alexander and Eleanor. Only their love makes it all worth while.
TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................ vi

LIST OF FIGURES ..................................................................................................... viii

1 INTRODUCTION .................................................................................................. 1
  1.1 Definitions and Applications ....................................................................... 1
  1.2 Overview of Speaker Recognition Studies .......................................... 3
  1.3 Neural Networks .................................................................................. 10
  1.4 Acoustic Feature Computations ................................................................ 13
  1.5 The DARPA TIMIT Database ................................................................ 14
  1.6 Overview of Following Chapters ......................................................... 16

2 PARTITIONING THE SPEAKER IDENTIFICATION TASK ................... 18
  2.1 Introduction .......................................................................................... 18
  2.2 Partitioning the General Classification Task ..................................... 19
    2.2.1 Group Partitioning ................................................................. 19
    2.2.2 Binary-Pair Partitioning (BPP) ............................................. 23
    2.2.3 Data Partitioning ................................................................. 28
    2.2.4 Combining BPP and Group Partitioning ................................ 32
  2.3 Neural Networks for Speaker Identification ......................................... 33
    2.3.1 Single Large Network ............................................................ 33
    2.3.2 Binary-Pair Neural Networks ................................................ 34
  2.4 Experiments .......................................................................................... 35
    2.4.1 Performance Comparison of One Large Network vs. BPP Networks .................................. 38
    2.4.2 Training Time Comparison of One Large Network vs. BPP Networks .................................. 40
    2.4.3 Search Methods for Combining BPP decisions into N-way decisions .................................. 42
2.4.4 Performance as a Function of the Number of Speakers ................................ 44
2.5 Summary .......................................................................................... 47

3 OPTIMIZATION EXPERIMENTS ................................................ 48
3.1 Introduction ................................................................................ 48
3.2 Experiments ............................................................................... 50
  3.2.1 Network Size and Amount of Training ................... 51
  3.2.2 In Search of Overtraining .......................................... 53
  3.2.3 Variable (Feedback) Training ................................ 55
  3.2.4 The Amount of Training Data .................................. 58
  3.2.5 Fixed-text vs. Random-text Training Data ............. 58
  3.2.6 Learning Rate and Amount of Training ................. 62
  3.2.7 Random Initialization ................................................ 62
3.3 Summary ...................................................................................... 67

4 SPEAKER IDENTIFICATION POTENTIAL OF SPEECH DATA BASED ON PHONETIC GROUP AND SIGNAL ENERGY ................................................ 68
4.1 Introduction ................................................................................ 68
  4.1.1 Which speech data carry speaker identity information? .......... 68
  4.1.2 Can performance be improved by selective use of training data? ........ 69
  4.1.3 Given a trained set of BPP networks, can their use be improved? .......... 70
    4.1.3.1 Improving Frame Level Probability Estimates by Incorporating Energy and Phonetic Information .... 71
    4.1.3.2 Combining Frame Level Probability Estimates into Segment Level Decisions .................................. 72
4.2 Experiments ................................................................................ 74
  4.2.1 Intrinsic Potential of Data Based on Phonetic Category ............................................. 75
  4.2.2 With testing fixed, can training be improved? ...... 81
  4.2.3 With training fixed, can testing be improved? ...... 86
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Summary of results from representative speaker recognition studies</td>
<td>6-8</td>
</tr>
<tr>
<td>1-2</td>
<td>Performance comparison of the Maximum Likelihood and Neural Network Classifiers</td>
<td>13</td>
</tr>
<tr>
<td>2-1</td>
<td>An example of how four global search methods might work with a simple three-way classification problem that has been binary-pair partitioned into three sub-classifiers making soft decisions</td>
<td>25</td>
</tr>
<tr>
<td>3-1</td>
<td>Average training data error rates after 1000 training iterations for combinations of training data and randomly initialized neural networks</td>
<td>65</td>
</tr>
<tr>
<td>3-2</td>
<td>The relative effect of training data versus random initialization on average training data error rates as a function of the amount of training</td>
<td>66</td>
</tr>
<tr>
<td>4-1</td>
<td>Percentage of occurrence, binary level talker recognition rate, and average normalized energy of individual 32 ms speech frames of each of the 57 phonemes</td>
<td>76</td>
</tr>
<tr>
<td>4-2</td>
<td>Percentage of occurrence, binary level talker recognition rate, and average normalized energy of individual 32 ms speech frames of each of the five phoneme groups</td>
<td>77</td>
</tr>
<tr>
<td>4-3</td>
<td>The relative potential of phoneme groups for speaker identification with the bias due to uneven distribution of training data removed</td>
<td>80</td>
</tr>
<tr>
<td>4-4</td>
<td>Binary level error rates with one second test speech lengths with various combinations of evaluation methods</td>
<td>89</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4-5</td>
<td>Average polynomial coefficients derived by the Least Mean Squares method and used for probability estimation with method 4</td>
<td>90</td>
</tr>
<tr>
<td>5-1</td>
<td>The relative performance of groups of DCTCs and the degree of frequency warping for speaker identification</td>
<td>108</td>
</tr>
<tr>
<td>5-2</td>
<td>The relative performance of static and dynamic acoustic features for speaker identification</td>
<td>115</td>
</tr>
<tr>
<td>5-3</td>
<td>The relative performance of memoryless, time delay, and recurrent neural network architectures for speaker identification</td>
<td>121</td>
</tr>
<tr>
<td>5-4</td>
<td>The number of undetermined transitions encountered in unwrapping the spectral phase of 32ms (512 point) speech segments as a function of frequency oversampling (FFT size) for two uncertainty thresholds</td>
<td>123</td>
</tr>
<tr>
<td>5-5</td>
<td>Summary of automatic speaker identification attempts using DSTCs computed from unwrapped and tilt normalized spectral phase of the speech signal</td>
<td>128</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>An example of group partitioning</td>
<td>21</td>
</tr>
<tr>
<td>2-2</td>
<td>Speaker identification performance comparison of one large network versus a system of binary-pair networks</td>
<td>39</td>
</tr>
<tr>
<td>2-3</td>
<td>Training time comparison of one large network versus a system of binary-pair networks</td>
<td>41</td>
</tr>
<tr>
<td>2-4</td>
<td>Speaker identification performance comparison of two evaluation methods using binary-pair partitioned neural networks trained with the variable training method</td>
<td>43</td>
</tr>
<tr>
<td>2-5</td>
<td>Speaker identification performance comparison of five evaluation methods using binary-pair partitioned neural networks trained with a fixed number of iterations</td>
<td>45</td>
</tr>
<tr>
<td>2-6</td>
<td>Speaker identification performance of binary-pair neural networks for varying numbers of speakers</td>
<td>46</td>
</tr>
<tr>
<td>3-1</td>
<td>Average binary-pair classifier performance with one second test speech as a function of network size and amount of training</td>
<td>52</td>
</tr>
<tr>
<td>3-2</td>
<td>The performance of six individual binary-pair classifiers as a function of the amount of training showing the effect of overtraining</td>
<td>54</td>
</tr>
<tr>
<td>3-3</td>
<td>Performance comparison of variable training versus fixed training with one second test speech segments</td>
<td>57</td>
</tr>
<tr>
<td>3-4</td>
<td>Binary classifier performance as a function of the number of training sentences and the test speech length</td>
<td>59</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>3-5</td>
<td>Performance comparison of binary classifiers trained with random-text (SI) speech versus fixed-text (SA) speech</td>
<td></td>
</tr>
<tr>
<td>3-6</td>
<td>Binary classifier performance with one second test speech as a function of learning rate and amount of training</td>
<td></td>
</tr>
<tr>
<td>4-1</td>
<td>Speaker identification performance and energy of individual 32 ms speech frames averaged for each phoneme and grouped into five broad phonetic categories (from Table 4-1)</td>
<td></td>
</tr>
<tr>
<td>4-2</td>
<td>The relative performance of the seven training data selection criteria as a function of test speech length</td>
<td></td>
</tr>
<tr>
<td>4-3</td>
<td>One second test speech recognition rates as a function of training data selection criterion and amount of training (varied from 100 to 2,000 iterations)</td>
<td></td>
</tr>
<tr>
<td>4-4</td>
<td>The amount of training speech available for 20 speakers depending on the selection criterion</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td>The effect on speaker identification of augmenting the acoustic feature set with phonetic group information</td>
<td></td>
</tr>
<tr>
<td>5-1</td>
<td>Bark and bilinear frequency warping with three values of alpha (α)</td>
<td></td>
</tr>
<tr>
<td>5-2</td>
<td>Examples of cosine basis functions modified to include bilinear frequency warping with α=0.5</td>
<td></td>
</tr>
<tr>
<td>5-3</td>
<td>The relative worth of the first 15 DCTCs (with α=0.5) to speaker identification with three test speech lengths</td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>Performance comparison of four window lengths for speaker identification</td>
<td></td>
</tr>
<tr>
<td>5-5</td>
<td>Weighting functions (windows) used for computing dynamic features from static features</td>
<td></td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure

5-6  Time-Delay (TDNN) and Recurrent (RNN) neural network architectures ................................................................. 118

5-7  The principle phase component of the DFT of a typical 32 ms speech segment computed from 512 and 8192 point FFTs ..... 124

5-8  Performance comparison of bandlimited (300-3,300 Hz) and non-bandlimited (0-8,000 Hz) speech with 15 and 30 DCTCs computed with $\alpha=0.5$ ................................................................. 130

6-1  Speaker identification performance as a function of the number of speakers and test speech length .......................... 136
CHAPTER ONE

INTRODUCTION

1.1 Definitions and Applications

The speech signal contains various types of information in addition to the spoken text. One is the speaker's identity, which includes gender, age group and accent. The speech signal is also influenced by such factors as the speakers emotional state and any respiratory ailments, such as a cold, that the speaker may have.

The goal of speech processing is to extract information from the speech signal. The main area in speech processing which has received the most attention in the past few decades is automatic speech recognition. Its goal is to allow easier human to machine communication. Another important area in speech processing is automatic speaker recognition. Its goal is the recognition or verification of a talker's identity by a machine. This has several potential applications. One is to help bridge the performance gap between speaker-dependent and speaker-independent automatic speech recognition. Most machine-based speech recognition systems perform significantly better with a small group of similar sounding users whose voices were used to train the system than with a general speaker population. Training the system with the voices of a large variety of speakers
improves performance but does not solve the problem completely. One solution to good speaker-independent performance is to first recognize the individual characteristics of a particular talker and then use the speech recognition system with a parameter set trained only with very similar sounding speakers.

Other potential applications of speaker recognition are in the areas of security and forensics. Some of these include controlling access to confidential information over the telephone, helping to prevent credit card fraud, and helping criminal investigations. Automatic speaker recognition systems may ultimately replace keys and badges to control access to buildings, rooms, and automobiles.

Two distinct speaker recognition tasks have been defined: speaker identification and speaker verification. Speaker verification requires a true or false decision to be made as to the claimed identity of a speaker. A "customer" presents a claimed identity along with a sample of speech and the system must either accept the claim as "legitimate" or reject the speaker as an "imposter." Speaker identification, on the other hand, is as follows. Given that an unknown speaker belongs to a known, closed set of N speakers, the task is to determine which one. If a speaker identification system is presented with a speaker that is not one of the N speakers, then it will select one of the N which is most similar, or "closest" to the unknown speaker.

For the case of speaker identification, accuracy is equal to the percent correct identification. For speaker verification, error rates are typically quoted for equal error rates of "false acceptances" and "false rejections." Another basic issue is whether the system operates in text-dependent or text-independent mode. For the text-dependent case,
decisions are based on fixed text, rather than a random selection. Because of the removal of phonetic variability, text-dependent systems usually achieve higher accuracy than text-independent systems. Another general consideration which has an effect on performance is the amount of speech used for training speaker models. Generally, the more speech data is available for training, the more likely it is that a statistical classifier will generalize to other data. Another major variable affecting the performance of speaker recognition systems is the amount of test speech available.

Speaker recognition is an inherently easier problem to solve than phonetic (unconstrained text) speech recognition because the speaker identity information does not rapidly change as does the phonetic content, and therefore information may be integrated over a longer period of time. On the other hand, the phonetic content of speech is much more pronounced than the speaker specific information, especially with similar sounding speakers. Evidence of this is reported by Kohonen (1988). "The 'neural' phonetic typewriter," a self-organizing feature map trained unsupervised with speech of several people would organize to distinguish phonemes, not the speakers.

1.2 Overview of Speaker Recognition Studies

Over the past several years many studies have been conducted in automatic speaker recognition. There are so many variations in the details of the problem definition and in the data bases used to test speaker identification algorithms that meaningful comparisons are difficult to make. For example many studies are conducted with "laboratory-quality" speech while others are conducted with "telephone-quality" speech.
The amount and type of speech material used for training and testing varies considerably among the studies.

In some early work in speaker recognition, pitch contours and energy were used as the primary features (Atal, 1972). More recently, features which represent the average magnitude spectrum, and thus model vocal tract characteristics, have been found to have better discrimination ability. In particular since the studies by Atal (1974) and Furui (1981), cepstral coefficients have been the primary features used for speaker recognition. Some studies have shown that pitch can be used to supplement the spectral features (Matsui and Furui, 1991). Another basic issue is whether speaker identity is primarily cued by instantaneous or transitional spectral cues. Several studies have investigated this point (for example, Furui, 1981; Soong and Rosenberg, 1988; Tseng et al., 1992) and generally found that the two types of cues complement each other (particularly for noisy or degraded speech), but that the instantaneous cues are more important.

Several algorithms have been developed over the last ten years for text-independent and text-dependent speaker recognition. This discussion is concentrated on those developed for text-independent recognition, the focus of the present study. Roughly speaking these algorithms can be sorted into four groups: (1) nonparametric representations of speaker feature distributions, generally using a vector quantizer (VQ) codebook; (2) parametric representations of speaker characteristics, for example using a mixture of multivariate Gaussian densities, or multivariate linear prediction coefficients; (3) Hidden Markov Model (HMM) techniques, and (4) methods based on neural networks. The VQ technique, (Soong, Rosenberg et al., 1985, 1987, 1988; Matsui et al., 1992),
computes a speaker-specific codebook from a training sample from each speaker. A test speech sample is recognized according to whichever speaker codebook can represent the sample with the least distortion. The parametric probability methods, (Gish, 1990; Tseng et al., 1992; Rose et al., 1990; Montacie et al., 1992), adjust model parameters in the training phase so that the model has the highest likelihood, given the data. In testing, a sample utterance is identified as that speaker whose model has the highest probability of having produced that utterance. Hidden Markov models have been used for speaker identification since 1982 (Poritz, 1982; Tishby, 1991, Tseng et al., 1992). A model can be trained which allows transitions between any two states (ergodic), and with output probabilities which reflect the average speaker-specific spectral characteristics for each state. Thus HMM's inherently model some transitional information, although in practice this information appears to be far less important than the output probabilities (Tishby, 1991; Matsui and Furui, 1992). Neural networks can be trained discriminatively as classifiers, given a sample of speech from each of the speakers to be identified (Oglesby and Mason, 1990; Rudasi and Zahorian, 1991b, 1992). A potential problem with a neural network method is the prohibitively long network training times for large speaker populations, unless the problem is partitioned.

Survey papers by O'Shaughnessy (1986) and Doddington (1985) give good summaries of speaker identification work prior to 1986. Since that time, a small number of journal articles on speaker recognition methods have been published, and numerous conference papers (ten, twelve, and ten in ICASSP-90,-91, and -92, respectively). Table 1-1 lists a brief summary of 17 studies, citing features used, method, data base, and
<table>
<thead>
<tr>
<th>Study</th>
<th>Features</th>
<th>Method</th>
<th>Database</th>
<th>Best Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furui (1981)</td>
<td>10 LPC cepstra + 8 &quot;dynamic&quot; terms</td>
<td>Template matching and DTW</td>
<td>10 customers and 40 impostors, 20 seconds for training, text dependent</td>
<td>Less than 1% verification error</td>
</tr>
<tr>
<td>Atal (1974)</td>
<td>12 LPC cepstra</td>
<td>Non-Euclidean template matching, using DTW</td>
<td>10 female speakers, 3 seconds for training, .6 seconds for testing, text dependent</td>
<td>98% identification</td>
</tr>
<tr>
<td>Rosenberg &amp; Soong (1987)</td>
<td>8 LPC cepstra and 8 &quot;dynamic&quot; terms</td>
<td>VQ codebook</td>
<td>50 male and 50 female speakers, 40 seconds for training and 3.5 seconds for testing semi-text-independent, telephone digits</td>
<td>97.9% identification</td>
</tr>
<tr>
<td>Soong &amp; Rosenberg (1988)</td>
<td>9 LPC cepstra, and 9 &quot;dynamic&quot; features</td>
<td>VQ codebook</td>
<td>5 male and 5 female speakers, 25 seconds for training, .5 second for testing, semi text independent, telephone digits</td>
<td>93% identification</td>
</tr>
<tr>
<td>Gish (1990)</td>
<td>cepstra</td>
<td>Parametric probability density estimation</td>
<td>16 speakers, 90 seconds for training, 60 seconds for testing, long distance telephone data (&quot;King&quot; database).</td>
<td>About 80% identification</td>
</tr>
<tr>
<td>Oglesby and Mason (1990)</td>
<td>10 LPC cepstra</td>
<td>Neural networks</td>
<td>10 speakers, 50 seconds for training, .5 second for testing, semi text independent, digits</td>
<td>92% correct</td>
</tr>
</tbody>
</table>

(continued on following page)

Table 1-1. Summary of results from representative speaker recognition studies (page 1 of 3).
<table>
<thead>
<tr>
<th>Study</th>
<th>Features</th>
<th>Method</th>
<th>Database</th>
<th>Best Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosenberg, et al. (1990)</td>
<td>12 LPC cepstra and delta cepstra</td>
<td>HMM's</td>
<td>10 male and 10 female speakers, 40 seconds for training and 3.5 seconds for testing, semi text-independent, telephone digits</td>
<td>Equal error verification rate of 1%</td>
</tr>
<tr>
<td>Rose &amp; Reynolds (1990)</td>
<td>19 mel cepstra</td>
<td>Parametric probability estimation</td>
<td>8 male and 4 female speakers, 45 seconds for training, 1 second for testing text independent</td>
<td>89% identification</td>
</tr>
<tr>
<td>Rudasi and Zahorian (1991b, 1992)</td>
<td>15-30 Bark-scaled cepstra</td>
<td>Binary-pair neural networks</td>
<td>120 male and 130 female speakers, 10 seconds for training, 8 seconds for testing, from TIMIT, text independent, lab quality</td>
<td>100% identification</td>
</tr>
<tr>
<td>Bennani and Gallinari (1991)</td>
<td>16th order LP model</td>
<td>Time Delay neural network</td>
<td>10 male and 10 female speakers, 10 seconds for training, 10 seconds for testing, text independent</td>
<td>98% identification</td>
</tr>
<tr>
<td>Tishby (1991)</td>
<td>8th order LP</td>
<td>HMM</td>
<td>50 male, 50 female speakers, 50 seconds for training, 5 seconds for testing, semi text independent, telephone digits</td>
<td>1% verification error</td>
</tr>
</tbody>
</table>

Table 1-1 (cont’d). Summary of results from representative speaker recognition studies (page 2 of 3).
<table>
<thead>
<tr>
<th>Study</th>
<th>Features</th>
<th>Method</th>
<th>Database</th>
<th>Best Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hattori (1992)</td>
<td>40 channel auditory model (Seneff)</td>
<td>Predictive neural network</td>
<td>24 female speakers, 10 seconds for training, 6 seconds testing, from TIMIT text independent, lab quality</td>
<td>100% identification</td>
</tr>
<tr>
<td>Matsui and Furui</td>
<td>16 LPC cepstral coefficients</td>
<td>VQ and HMM</td>
<td>23 male and 13 female speakers, about 40 seconds for training and 20 seconds for testing</td>
<td>95% identification with VQ and continuous HMM</td>
</tr>
<tr>
<td>Tseng, et al. (1992)</td>
<td>8 LPC cepstra and 8 delta cepstra</td>
<td>Continuous parametric probabilistic maps</td>
<td>10 male and 10 female speakers, about 40 seconds each for training and testing, semi text independent, telephone digits</td>
<td>97.14% for text-independent case</td>
</tr>
<tr>
<td>Montacie, et al.</td>
<td>cepstral coefficients</td>
<td>Multivariate linear prediction</td>
<td>290 male and 130 female speakers, 10 seconds for training, 8 seconds for testing, from TIMIT, text independent, lab quality</td>
<td>98.4% correct identification</td>
</tr>
<tr>
<td>Kao, et al. (1992)</td>
<td>20 LPC cepstral coefficients</td>
<td>VQ</td>
<td>51 speakers, 90 seconds for training, 30 seconds for testing, long distance telephone data (&quot;King&quot; database)</td>
<td>67.6% correct identification</td>
</tr>
</tbody>
</table>

Table 1-1 (cont'd). Summary of results from representative speaker recognition studies (page 3 of 3).
results. A direct comparison of results is not possible because of differences in the databases and differences in evaluation criteria. The major issue impacting performance is the quality of the data. For example, speech recorded in a single session per speaker in a quiet laboratory, such as the TIMIT database, is much easier to identify than speech obtained using a number of widely varying noisy telephone lines over a period of several months, such as the King data base. Even for telephone data, there are big differences depending on the number and variability of telephone connections used.

The data shown in Table 1–1 is meant to be representative of recent work, but by no means exhaustive. It does appear that the most commonly used features are cepstral coefficients. All the algorithms mentioned above have been used. In many cases, studies have been devoted to refinements of the basic methods and also to comparing methods. For example the paper by Tseng et al. (1992) illustrates an improved a parametric method for modeling probability densities. Matsui and Furui (1992) compare VQ, discrete HMM, and continuous HMM methods, and show that the VQ and continuous HMM are roughly comparable in performance and both are better than the discrete HMM. They also show that continuous HMM performance is much more closely linked to the total number of mixtures, independently of the number of states, again showing that transition probabilities convey relatively little useful information.

Some attempt can be made to compare results with apparently comparable data bases. Several studies have reported results using the TIMIT data base with text independent algorithms, generally using about ten seconds each for training and testing. By far, the two best reported identification results are 100% correct for 47 male speakers.
(Rudasi and Zahorian, 1991b), (which has since been extended to 100% correct for 120 male and 130 female speakers), and 98.4% correct with 290 male and 130 female speakers (Montacie et al, 1992). Several studies have used a telephone digit data base. Generally about 40 to 75 seconds of speech are used for training (80 to 100 digits), and 0.5 to 5 seconds for testing (one to ten digits). A recent benchmark (Tseng, et al., 1992) for text-independent identification is 97.14% using 20 speakers (ten male and ten female) and three digits for testing (about 75 seconds for training, and 1.5 seconds for testing) and 8.01% equal error verification rate. For text-dependent models, the corresponding best identification and equal error verification rates are 98.31% and 3.97%. Note that the digit data base in the text-independent mode is still highly constrained and thus not completely text independent. A difficult database which has been widely used is the "King" text-independent conversational telephone database (Gish, 1990; Higgins, et al., 1991; Kao et al., 1992). A reported benchmark for identification is 67.6% for 51 speakers (using 90 seconds for model training and 30 seconds for testing, with the same recording conditions for training and test conversations) (Kao, et al., 1992).

1.3 Neural Networks

Automatic speaker identification is a statistical pattern recognition or classification problem. As such, it involves two basic issues. One is the feature selection, i.e. the form in which data is to be represented, and the other is the statistical classifier algorithm used.

In this study, the basic classifier used is an artificial neural network, more specifically, a memoryless, feed-forward, multilayer perceptron. Lippmann (1987)
provides a good introduction to this and several other types of neural network architectures. The computational element of this type of classifier is a "neuron" which forms a weighted sum of its inputs and passes the result through a limiting non-linearity. The weighted sum of each neuron also includes an offset, or bias term. As "feed-forward" implies, information in this type of network progresses only in one direction, from the input nodes, through one or more hidden layers of neurons, to the output neurons. Networks in which each neuron computes a weighted sum of all the nodes at the one previous (lower) level are referred to as "fully interconnected." This may be slightly misleading since it could be interpreted to imply direct connections to all lower layers. As the term "memoryless" implies, the network outputs are only dependent on the current input pattern. The neuron non-linearities are the sigmoidal function which limits the output in the range of zero to one. The network weights and offsets are trained by the error backpropagation method introduced by Rumelhart, et al. (1986), and Werbos (1974).

The number of layers of a multilayer 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 hyperplane decision boundary. A two-layer (one hidden layer) network is able to form 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 (Lippmann, 1987).
In each experiment reported in this dissertation, the network or networks used were two layer, fully interconnected, memoryless, feed-forward, with sigmoid nonlinearities. Network weights were initialized with random values uniformly distributed from -0.05 to 0.05. Each was trained with the backprop method with a fixed learning rate (0.1 – 0.3) and fixed momentum term (0.7). The output targets were 0.999 for the node corresponding to the correct category, and 0.001 for the other(s). Training of a networks is performed as follows. Individual speech frames are presented to the network one at a time. Based on the differences between desired outputs and actual outputs a small change is made in each of the network weights and offsets. This is considered one complete training iteration. Pilot studies have shown that this method causes faster convergence and results in better final solutions than the alternative method of accumulating weight changes and updating the weights only after all categories were presented to the network. The training is cycled to the next category after each iteration. Also, after each iteration, the data pointer of the particular category is advanced by 11, 13, or 17 frames to prevent the network from temporarily learning phonetic content, rather than text-independent speaker identity.

A preliminary speaker identification experiment was performed to compare the suitability of the neural network approach to that of the Gaussian Maximum Likelihood classifier. The task was the identification of a five speaker population based on individual speech frames (described in Section 1.4). The results are tabulated in terms of recognition rates for both the training and the testing data for both methods in Table 1–2.
Training data rec. rate (%) | Testing data rec. rate (%)  
--- | ---  
Maximum Likelihood Classifier | 62.20 | 51.19  
Neural Network Classifier | 76.82 | 64.55

**Table 1-2.** Performance comparison of the Maximum Likelihood and Neural Network classifiers.

The performance of the neural network approach is substantially better. This is undoubtably due to the fact that the neural network is not subject to the joint Gaussian distribution assumption as is the Maximum Likelihood Classifier (Lippmann, 1987; Parsons, 1986).

### 1.4 Acoustic Feature Computations

Besides the classifier, the other basic issue involved in the speaker identification problem is the acoustic feature set to be used. In general, for any pattern recognition problem, the feature set used to characterize the data should retain all the relevant information while resulting in as much data reduction as possible by eliminating the non-relevant information. Relevant information is that which is different from one category to another and thus can aid in discriminating among the categories. Much previous work in speaker recognition has been devoted to feature selection as was discussed briefly in Section 1.2.
In this study, the feature set used for encoding the speech signal is a form of cepstral coefficients computed as follows. First, the speech signal is high-frequency preemphasized with transfer function $1 - 0.9z^{-1}$. The speech signal is then windowed using a 32 ms Hamming window, with a 10 ms frame spacing. The magnitude spectrum is computed using a 512 point FFT for each frame. The spectrum is then log amplitude scaled with a minimum floor of $-50$ dB for individual points relative to the peak of the frame. In earlier experiments (Chapters 2, 3, and 4) the spectrum was then frequency warped with a bilinear transform with a coefficient of 0.6. The frequency warping was accomplished using linear interpolation and resampling, as discussed in Section 5.1.1. Fifteen cepstral coefficients were then computed over a frequency range of 150 Hz to 6000 Hz. As a result of experiments performed for Chapter 5, some changes were made to the feature set computations. For the latest experiments, 30 cepstral coefficients were computed from the entire frequency range (0 to 8,000 Hz). The degree of frequency warping was also reduced to 0.25, and the method of frequency warping was changed to the use of modified cosine basis functions, as described in Appendix A. For all experiments, the individual features were normalized for zero mean and standard deviation of one, using statistics computed from the entire training set.

### 1.5 The DARPA TIMIT Database

The speaker identification experiments described in this dissertation were performed with the prototype version of the DARPA TIMIT Acoustic Phonetic Continuous Speech Database, issued in December, 1988. The database consists of the
speech signal and phonetic transcription of ten sentences from each of 420 talkers. The
ten sentences from each talker are as follows: two dialect calibration (SA) sentences,
three random contextual (SI) sentences, and five phonetically balanced (SX) sentences.
The two SA sentences are the same for each of the talkers. The SI and SX sentences
vary from talker to talker. The average sentence is about three seconds long. The speech
data was sampled at 16,000 samples per second with a 16 bit analog to digital converter.
A complete description of the design of the TIMIT database can be found in Fisher et al.
(1986).

The breakdown of the 420 speakers is as follows. 290 are male and 130 are
female. The speakers are assigned to seven dialect regions of the United States: New
England, Northern, North Midwest, Southern, South Midwest, Western, and New York
City. The eighth category is called "army brat", which is a mixture of dialects. Most of
the experiments were performed with the speech of 20 male talkers belonging to the
Northern dialect region. When 47 speakers were used, they were all the male speakers
of the database of that region. When 130 female speakers were used, these were all the
females from all eight dialect regions of the database. The experiments with 120 male
speakers involved all males from the first three dialect regions, i.e. New England (22),
Northern (47), and North Midwest (51).

In most experiments, the SA and SI sentences were used for training and the SX
sentences for testing. Thus it should be noted that of the five training sentences, only the
two SA sentences were the same for all the speakers. The average length of the
combined SA and SI sentences per speaker is 14.5 seconds. The five SX sentences
average 13.2 seconds per speaker. In the majority of experiments, testing was performed with several test speech lengths, ranging from individual (0.032 second) frames to eight seconds. To make thorough use of the limited amount of testing data available for each speaker, the adjacent test speech segments were overlapped by 50 percent. Also, to allow the use of more test segments, the total test speech was wrapped in a circular fashion. Thus, for example, with 13 seconds of total test speech available, three eight-second segments were extracted as follows: 0–8, 4–12, and 8–13 continued with 0–3 seconds.

1.6 Overview of Following Chapters

The most severe limitation of most existing speaker identification systems is that they do not scale up well to large speaker populations. In Chapter 2, methods of partitioning are discussed, one of which allows any N-way classification task to be reduced to two-way classification tasks, each involving only two individual categories, independent of the others.

Chapter 3 addresses the problem how the two-way elemental (partitioned) decisions can best be performed with neural networks. The results of a number of optimization experiments are presented.

In Chapter 4 the effect of phonetic content and short term energy of the speech signal on speaker identification performance is examined. Potential performance improvements by the incorporation of phonetic group and short term energy information are also investigated.
Feature selection for speaker identification is examined in Chapter 5. The cepstral coefficients are optimized and methods of incorporating dynamic information are considered. The use of spectral phase derived features is also explored to some extent.

Finally, in Chapter 6, the significant conclusions are summarized, and potential future improvements are discussed.
CHAPTER TWO

PARTITIONING THE SPEAKER IDENTIFICATION TASK

2.1 Introduction

In this chapter a binary partitioned approach to classification is investigated which is applied to talker identification using neural networks. Neural networks have been shown to work exceptionally well for small but difficult statistical classification tasks. However, their application to large tasks, i.e., having more than ten to 20 categories, is limited by a dramatic increase in required training time. The time required to train a single network to perform N-way classification is nearly proportional to the exponential of N. In contrast, by partitioning the classification problem, the required training time will be reduced to the order of N^2. Experimental evidence also suggests that the partitioned neural network approach requires less training data then the use of a single large network.

Over the past several years many studies have been conducted in automatic speaker identification or verification. However, in the majority of previous studies, the number of speakers has been restricted to a relatively small number such as ten or twenty. Many previously presented classification schemes for speaker identification do not scale
up well to a large number of categories (speakers). In this chapter, a method is presented for partitioning a large classifier using a large number of small classifiers, and the suitability of this method to speaker identification with neural networks for large speaker populations is investigated.

2.2 Partitioning the General Classification Task

Several possible ways to partition a large classification task exist. Generally, they involve applying subclassifiers to an unknown sample, each of which eliminates one or more potential candidates, until the number of contending categories is reduced to one. The two most clearly distinct approaches to partitioning are group partitioning and what may be termed pairwise or binary-pair partitioning (BPP). Of these, the grouping of categories is commonly used. The binary-pair partitioning approach has received far less attention and has not been thoroughly examined before. In this study, this approach is presented as an alternative for a difficult classification problem for which group partitioning alone is not viable.

2.2.1 Group Partitioning

One partitioning approach is to use a tree-like network of classifiers, with each classifier sorting unknown samples into distinct groups. As an unknown sample works its way up the classification tree, the number of categories per group is successively reduced at each branch until a leaf, corresponding to a single category is reached. For example, the task of identifying the 40 or so phonemes of the English language may be
group partitioned as follows. First, the broad phonetic category (vowel, stop consonant, nasal, fricative, etc.) of the unknown sample is determined. After this, the sample is routed to the corresponding subclassifier which determines the category within the particular group. Some of the subclassifiers can be further partitioned. For example, the six stop consonants (B,D,G,P,K,T) are commonly divided into the voiced (B,D,G) and the un-voiced (P,K,T) groups (Nossair, 1989, p.12-15).

Waibel (1989) used group partitioning to tackle the problem of excessive training times required by large time-delay neural networks (TDNNs). He trained a three category (B,D,G) TDNN classifier in three days to achieve 98.3% recognition accuracy. A six category (B,D,G,P,K,T) classifier required 18 days of training to achieve the same recognition rate using the same computer. He also trained another three-way classifier for the remaining three stop consonants (P,K,T) and a two way classifier to distinguish the two groups (voiced and un-voiced). He then compared various methods for combining the three subclassifiers to perform the six way classification. Each method obtained a recognition rate over 98%, while requiring substantially less total training time than the 18 days required by the non-partitioned approach.

Another successful application of group partitioning was presented by Rudasi and Zahorian (1990). A speaker identification system was developed for a speech data base consisting of voice samples from ten adult men, ten adult women, and ten children. As illustrated in Fig. 2–1, the system consists of four independent classifiers, one three-way classifier to determine the group (men, women, or children) that an unknown input belongs to, and three ten-way classifiers to determine identities within the groups.
Figure 2-1. An example of group partitioning.
A special case of group partitioning is binary group partitioning. This approach uses \( N-1 \) two-way classifiers to make \( N \)-way decisions. Each classifier would sort incoming data into one of two groups. If the two groups are close to even at each branch of the tree, then at each step of classification half the remaining possible categories are eliminated and the number of required decisions for an \( N \)-category problem is on the order of \( \log_2 N \). If the tree is uneven, then the required number of decisions may be from 1 to \( N-1 \), depending on the category and the degree of unevenness.

One advantage of group partitioning is that as long as there are no errors made at the beginning levels of the decision tree, no subclassifier needs to make a decision about an input sample that it was not designed (or trained) to classify. The most important limitation of this method, however, is the need to find a "good" partitioning of the categories for each subclassifier. An ideal partitioning is such that members of each group have similarities which are not shared outside the group. If the categories are placed in arbitrary groups, which do not meet this criterion, then the low level classifiers will need to recognize individual members within groups, and thus the group partitioning will not help the task.

Another indirect example of group partitioning was presented by Kohonen (1988). In this case partitioning was not sought but was forced by the difficulty of the problem. The "main feature map" trained on all phonemes could not distinguish three similar sounding nasals (\( m, n, \) and \( \text{ng} \)) and thus lumped them together to be distinguished later by an auxiliary feature map.
2.2.2 Binary-pair Partitioning (BPP)

The approach examined in this chapter is another alternative to partitioning the classification problem. This method involves using a large number of binary classifiers, with each classifier designed (trained) to distinguish between only two categories. With \( N \) categories, there are a total of \( N(N-1)/2 \) pairs of categories. Therefore, a total of \( N(N-1)/2 \) binary classifiers are required, each trained to optimally discriminate between one specific pair of categories. If each of these pairs can be successfully discriminated, the overall classification problem can be solved.

There are two fundamentally different approaches for using these binary classifiers to classify an unknown sample, and each has several variations. They both classify with 100% accuracy as long as all of the binary-pair classifiers perform correctly. However, the performance of the two basic approaches may degrade differently if not all of the binary-pair classifiers work perfectly.

The first, and perhaps the simplest method for classifying an unknown sample may be termed the "global search" method. It requires that the unknown sample be evaluated by all of the binary-pair partitioned classifiers and tallying the "votes" for each of the categories. Using the global search method based on hard decisions, if all the binary classifiers work properly then only the correct category will receive the perfect score of \( N-1 \), while the best possible competing score is one less, or \( N-2 \). A slight variation of this method is suggested by the fact that many statistical classifiers (such as artificial neural networks) make probabilistic (soft) decisions. The method of summing soft
decisions will be referred to as the "global soft decision search" or "global soft search" in the experimental sections and in later chapters.

Several variations of the global soft search method may be defined. Each of these address the potential weakness of the global soft search method, which is that each of the binary-pair subclassifier outputs are weighted equally. It may be advantageous to weight decisions relating to the top contenders more heavily than the other decisions. One possibility is to successively eliminate the weakest category or categories using the global search and each time recomputing the sums from the binary-pair classifiers that are relevant only to the remaining categories. The special case of this, in which only the one weakest category is eliminated each time will be referred to as "Iterative Elimination" in Section 2.4.3. Another variation of the global soft search may be termed the weighted global soft search. With this method, first a soft decision hierarchy among the categories is determined using the regular (un-weighted) soft search and then the sums are recomputed such that the outputs of the binary-pair classifiers which have to do with the top contenders are weighted more than the others. This method will be referred to as the "Weighted Soft Search" in Section 2.4.3.

For a simple example illustrating the four global search methods, see Table 2–1. In this example, a three-way task is binary-pair partitioned and the outputs of each of the three BPP classifier for a specific input sample are shown. The output of each BPP classifier should be considered a conditional probability. For example, the "2–3" classifier output is conditioned on the sample belonging to categories "2" or "3". As this example shows, the four global search methods may not always produce the same result.
25

### Table 2-1

An example of how four global search methods might work with a simple three-way classification problem that has been binary-pair partitioned into three sub-classifiers making soft decisions.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Categories</th>
<th></th>
<th></th>
<th></th>
<th>3-way Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>Binary-pair classifiers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and their outputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-2</td>
<td>0.6</td>
<td>0.4</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-3</td>
<td>0.6</td>
<td>-</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-3</td>
<td>-</td>
<td>0.9</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3-way classifiers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and their sums</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hard</strong></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td><strong>Soft</strong></td>
<td>1.2</td>
<td>1.3</td>
<td>0.5</td>
<td></td>
<td>&quot;2&quot;</td>
</tr>
<tr>
<td><strong>Weakest category (3) eliminated</strong></td>
<td>0.6</td>
<td>0.4</td>
<td>-</td>
<td></td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td><strong>Strongest pair (1-2) weighted by K others by 1</strong></td>
<td>0.6K +0.6</td>
<td>0.4K+ 0.9</td>
<td>0.5</td>
<td>&quot;1&quot; if K&gt;1.5 &quot;2&quot; if K&lt;1.5</td>
<td></td>
</tr>
</tbody>
</table>

Note that the *tree-search based on hard decisions* method would pick category "1".
Note that the example in Table 2-1 is independent of the particular classification task, and need not be speaker identification. Also, the example is independent of the type of sub-classifier that was used to make the binary-pair partitioned (2-way) decisions. The BPP sub-classifiers did not have to be neural networks, but could have been any classifiers that provide probabilistic (soft) decisions.

The other basic approach to performing N-way classification with the binary-pair classifiers is a tree-search based on hard decisions. The advantage of this approach over the global search is that the unknown sample is evaluated by only N-1 of the classifiers, rather than all N-(N-1)/2. Since each binary decision eliminates one category from contention, only one category remains after the N-1 decisions. In general, the only constraint in the order in which the classifiers are applied to an unknown sample is that at each point in time a classifier is chosen which must be related to two categories which have not yet been eliminated at that time. If there are no other constraints placed on the order in which the classifiers are applied, then it is possible for same categories to be selected after having survived only one elimination attempt, while some others may be required to survive N-1. If all the binary-pair classifiers are guaranteed to work properly then this would not make any difference, but otherwise, some categories are likely to be favored over others.

To eliminate this potential disparity, an additional constraint may be made such that each category be required to survive no less then M and no greater than M+1 elimination attempts, where $M \leq \log_2 N < M+1$. This approach may be thought of as a series of simultaneous elimination rounds. First, with N categories all the categories
are paired in $N/2$ pairs, and one of each pair is eliminated by the application of the binary classifier corresponding to the pair. The winners of each round are then paired again in the next round until only one category survives. If the number of contenders at any time is odd, the one "left-out" category automatically advances to the next round. Therefore, the total number of binary decisions made is $N-1$, and the number of these that must made correctly is on the order of $\log_2 N$. These will be termed "relevant" decisions. The rest of the decisions are made between pairs of categories which should both be eliminated later on. These are then termed "non-relevant" decisions. This classification method will be referred to as the "binary tree search" or "tree search based on hard decisions" in the experimental sections as well as later chapters.

There are three advantages of the tree search based on hard decisions method. The first is that to classify an unknown sample only $N-1$ of the BPP network outputs need to be computed rather than all $N-(N-1)/2$. Another is that all the BPP classifiers working correctly is a sufficient but not necessary condition to guarantee the top level ($N$-way) decision to be made correctly. Finally, during use (testing) the networks are evaluated independently of one another and need not be trained the same amount, nor do they even have to be the same type of statistical classifier. This allows the use of variable training, i.e., more iterations for those BPP networks whose categories are more difficult to distinguish, and fewer iterations for those networks whose categories separate more easily.

The advantage of the global soft search method is that it provides a likelihood hierarchy (first, second, third, etc.) of the categories, as well as a relative probability for
each of the candidates. So if the correct category is not chosen as the top choice, it is likely to be chosen as the second, third, etc., which may be useful in many applications. Also, it allows all of the BPP classifiers to be evaluated simultaneously (in parallel).

The major advantage of binary-pair partitioning over group partitioning is that the categories need not be grouped. There is no need to find "natural" partitions (groups). Rather, each classifier can be highly tuned to discriminate between the two members of its particular pair. A potential disadvantage is the requirement for a large number of classifiers. For example with N=100, 4950 binary-pair classifiers are required versus only 99 binary group classifiers.

Pairwise partitioning of categories was first introduced by Widrow, et al. (1962). In this early study, the issues discussed in this dissertation were not addressed. Instead, the pairwise partitioning was utilized to construct two-layer networks of neurons, long before Werbos (1974) and Rumelhart et al. (1986) introduced the training of multi-layer networks with the now well known error back-propagation method. The method introduced by Widrow involved training individual single neuron classifiers to distinguish pairs of categories and then connecting them together with an additional layer of neurons whose weights were designed to implement AND gates.

2.2.3 Data Partitioning

Consider the general task of designing a single two-way classifier. This may be an individual binary-pair partitioned classifier or a binary group classifier. Depending on the type of statistical classifier algorithm to be used, the amount of training data available,
and the actual hardware implementation, it may be advantageous, or even necessary to partition further. Since there are only two categories involved, BPP and group partitioning are not applicable. The further partitioning that may be required may be termed *data partitioning*. With this approach, several independent versions of the two-way classifier are trained, each with only a subset of the total available training data. When classifying an unknown sample, the outputs of the sub-classifiers could then be averaged somehow (perhaps weighted). The circumstances under which this approach should be considered are the availability of (and the requirement to have represented) a large amount of training data and the use of a classifier which trains iteratively by making several passes over the data, such as neural networks or Hidden Markov Model classifiers. Classifiers such as the *Gaussian maximum likelihood classifier*, which requires computing averages and covariances in a deterministic fashion would not benefit from data partitioning in most applications, unless *data weighting* is to be used, as discussed below.

Consider the following example in which data partitioning is being applied. The task is the identification of the gender of a talker from a segment of speech signal. The training data available is the TIMIT data base with 290 male and 130 female speakers, which is over 400 mega-bytes of raw data. Consider that only half the speech from each speaker is to be used for training and that the acoustic feature set representation (15 cepstra every 10 ms) results in another five-to-one data reduction (using 4-byte real numbers); the training data is on the order of 40 mega-bytes. To iteratively train a single classifier with this much data would require either a lot of memory or disk I/O. Also, depending on the training method used, there may be additional difficulties due to the
large amount of data. If the classifier parameters (connection weights in case of neural networks) were updated with each individual frame, then the speakers would not be represented equally by the trained classifier, due to the fact that at any time during training some will not have been "seen" for a long time. On the other hand, if the parameter changes are accumulated until each speaker is seen at least once, then there have to be a lot of calculations performed before an update is made, making the learning very slow.

Some pilot experiments were performed, in which data partitioning was used to solve the difficulties discussed above. In one case, the speakers were divided into groups of ten (29 male and 13 female groups) and a separate sub-classifier was trained to distinguish each male group from each female group, resulting in 377 sub-classifiers. In another case, ten male groups with 29 speakers each and ten female groups with 13 speakers each were formed. In this case only ten sub-classifiers were trained, each with one of the male and one of the female groups. To date, the best performing system determines the gender of speakers with 98.4% accuracy with test speech lengths of one to two seconds. The results of these pilot experiments have not yet been published.

As mentioned above, there is another potential advantage to data partitioning which is not limited to iteratively trained classifiers. This advantage is due to the fact that the outputs of the sub-classifiers may be weighted before a combined (averaged) decision is made. By weighting the output of the sub-classifiers, weighting of the training data may be accomplished without having to retrain the system. By weighting the sub-classifiers (and thereby the training data) some additional information may be
incorporated which may have relevance to the decision to be made. Yet, another reason for data partitioning may be the need to remove the effect of a systematically misleading bias in the training data. For example, consider the problem of two-way automatic language detection in which the task is to distinguish languages A and B. Suppose that in the available training data, each language is represented by male and female speakers. By training a separate A/B classifier for males and females, the gender information of a test speaker, if known, may be made use of by weighting the corresponding (sub)classifier more heavily. Furthermore, consider the possibility that the languages are represented by different percentages of male and female speakers in the training data. In this case it is very helpful to partition the data based on gender to remove the bias, even if the gender of test speakers is not to be used.

Another possible reason for using a data partitioned approach may be the availability of two or more types of data representing each category. Consider for example the problem of acoustic feature selection for speaker recognition. It is a well known fact that with spectral analysis, there is a tradeoff between time resolution and frequency resolution (as quantified by Eq. (5–2) in Chapter 5), depending on the length of time window used, which should depend on the phonetic content of the speech. With stop consonants the window should be short for better time resolution; with vowels, which are relatively unchanging for longer durations, the window should be longer for improved frequency resolution (Nossair, 1989; Jaghargi, 1990). It may be advantageous, therefore, to switch back and forth in time between two window lengths. Rather than mix the two data types, it may be better to also switch back and forth between two independently
trained classifiers, each designed to accomplish the same speaker recognition task, but with the two different types of data.

2.2.4 Combining BPP and Group Partitioning

The relative advantages and disadvantages of BPP versus group partitioning were discussed in the above sections. The main disadvantage of the BPP approach is the large number of sub-classifiers needed with a large number of categories. If any grouping of the categories is possible, then the number of BPP sub-classifiers can be reduced. Consider for example the speaker identification task with N speakers. If BPP is used alone, the number of sub-classifiers needed is N-(N-1)/2. If a subset of P of the N speakers can clearly be identified as belonging to some group, and another subset of Q speakers can be identified as not belonging to that group, then the number of BPP classifiers can be reduced by P\times Q. Any number of groups may exist, some of which may be mutually exclusive and some of which may overlap. For example, the groups "male" and "female" are exclusive. Some others, such as "very old", "has high-pitched voice", and "has oriental accent" may be independent. Some groups may be self-organizing (data driven), so that the commonality of its members is not obvious. Even a "group" containing a single member can be useful, as long as there is at least one other individual that is clearly excluded by it.

The general procedure for using any number of group classifiers to reduce the number of BPP classifiers needed is as follows. Given the training data for N categories and a number of group classifiers, for each of the N categories, a "characteristics string"
is determined using the training data. The length of the string depends on the number of 
groups defined by the group classifiers. Each element of the string represents the result 
of a three-way decision made by the corresponding group classifier regarding the 
membership of the category. The possible outcomes of the decision are: "definitely yes",
"definitely no", and "not certain", represented by "1", "0", and "X", respectively. The 
next step is then to train the BPP classifiers to distinguish the individual pairs of 
categories, but only for category pairs which are not already separated by the group 
classifiers. To determine if they are, their "characteristics strings" are compared. A 
single mismatch between corresponding elements in the two strings means that the 
categories are already separated by at least one of the group classifiers, and therefore the 
BPP classifier is not needed. A mismatch is a "0" and "1" combination.

2.3 Neural Networks for Speaker Identification

2.3.1 Single Large Network

Neural networks have been applied to many classification tasks with great success. 
In several previous studies, and as a control for the present study, a two-layer, feed­
forward, fully-interconnected, memoryless neural network, trained with backpropagation 
was used for speaker identification. The main variables with this method are the number 
of hidden nodes, the amount of training data, the amount of training iterations, the 
learning rate, and the test speech length. Variables which are of lesser importance, or 
which have more or less been optimized in many previous studies, are the number of
layers, the type of non-linearity function used (sigmoid), the method of initializing the weights, and the momentum term (Oglesby and Mason, 1990). The output nodes of the network each correspond to one of the categories (speakers). For each input pattern, the network is trained to have the output corresponding to the correct category high, while keeping the other outputs low. During classification of an unknown sample, the output nodes are accumulated over the number of frames in the sample, and the category with the greatest sum is chosen. Since the neural net is memoryless, the time averaging is external to it. Thus the network makes many independent soft decisions each of which is based on a small segment of speech. This approach contrasts considerably with first computing speech statistics, such as covariance matrices, etc. and then classifying based on long term average properties (Savic and Gupta, 1990 and Ren-hua, Lin-shen, and Fujisaki, 1990).

The main problem with the use of one large network for talker identification is that training time increases exponentially with the number of categories. Thus large problems become unsolvable. Furthermore, the addition of a new talker to an existing system generally requires retraining the entire network. These problems are solved by the use of binary-pair neural networks. This new approach also offers the advantage of modularity, which could be useful in implementation.

2.3.2 Binary-pair Neural Networks

The solution investigated in this chapter to the problem encountered when attempting to use one large network to classify many categories is to replace it with a
large number of much smaller networks. The binary-pair approach is not restricted to neural nets, but seems particularly appropriate for the case of neural nets. As discussed earlier, N-(N-1)/2 small classifiers are trained, each to distinguish between two of the N categories. Each of these small binary (two-way) neural nets are independent of the others. They also train very rapidly because they are independent of the training data of the non-relevant categories.

The only difference between the binary-pair partitioned neural networks and the non-partitioned single large network is that the BPP networks have only two output nodes, and much fewer (about four to ten) hidden nodes. Each of the two outputs corresponds to one of the two categories the particular BPP sub-classifier is intended to distinguish. Note, however, that the BPP sub-classifier could also be a single output neural network, such that the "high" and "low" output levels each represent one of the two categories. This would result in somewhat fewer network connections, although the reduction would not be significant, since most of the connections are at the lower layer. Therefore, the BPP neural networks used in this, as well as future chapters, all have two output nodes.

2.4 Experiments

The experiments were designed with the following four goals in mind:

(1) to compare the performance of the binary-pair partitioned approach to that obtainable with one large network;

(2) to compare the relative training time requirements of the two methods;
(3) to compare the performance of the global soft search and the hard decision binary tree search methods of combining binary-pair partitioned decisions into top-level (N-way) decisions; and

(4) to examine the performance degradation of the binary-pair partitioned approach as categories are added.

The experiments described in this chapter were performed on a subset of the DARPA TIMIT Acoustic Phonetic Continuous Speech Database described in Section 1.4. The 47 male talkers belonging to the Northern dialect region were used. In experiments with less than 47 categories, the talkers were chosen based on the alphabetic ordering of the speaker initials. In each experiment, the two SA and three SI sentences were used for training and the five SX sentences for testing.

The feature set used for encoding the speech signal was the modified form of cepstral coefficients as described in general in Section 1.5. More specifically, the speech signal was high-frequency preemphasized with transfer function 1-0.95z⁻¹. The speech signal was then windowed using a 32 msec Hamming window, with a 10 ms frame spacing. The magnitude spectrum was computed using a 512 point FFT for each frame. The spectrum was then log amplitude scaled and frequency warped with a bilinear transform with a coefficient of 0.6. Fifteen cepstral coefficients were then computed over a frequency range of 150 Hz to 6000 Hz. Using statistics computed from the training data, the cepstral coefficients were normalized for zero mean and a standard deviation of one for each coefficient.
Low energy frames were discarded as input to the speaker identification systems for both training and testing. Low energy frames were defined as those with a normalized zero-order cepstral coefficient (DCTC1) of less than $-1$. The threshold was determined from pilot experiments which indicated that low energy frames were poor predictors of talker identity. This threshold eliminated about 20 percent of the frames including beginning and ending silence.

In each experiment, the network or networks used were two layer, fully interconnected, memoryless, feed-forward, with sigmoid non-linearities. Network weights were initialized with random values uniformly distributed from $-0.05$ to $0.05$. Each was trained with the backprop method with a fixed learning rate (0.1 - 0.3) and fixed momentum term (0.7). The output targets were 0.999 for the node corresponding to the correct category, and 0.001 for the other(s).

For some of the experiments, networks were trained using what may be termed the variable training method. With this method, training of each binary-pair classifier continues only until an empirically determined threshold of performance with the training data is reached. Typically this threshold was 65% to 75% of individual training data frames correctly classified. With this method, some binary-pair networks were trained for far more iterations (up to 20 times) than others. The exact threshold used depended on the number of hidden nodes (i.e. network size). The other method may be termed fixed training in which case all the BPP classifiers are trained the same number of iterations.
2.4.1 Performance Comparison of One Large Network vs. BPP Networks

This experiment was designed to compare the performance of the binary-pair partitioned neural network method to that attainable with a single "large" non-partitioned neural network. Three cases were considered: N=5, N=10, and N=15. Several experimental runs were made for each of the six systems, while varying the following parameters: number of hidden nodes, learning rate, minimum acceptable recognition rate of the individual training data frames, and the number of training iterations between checking the training data performance. For each of the six systems, the result of the best run is shown in Fig. 2–2.

The basic result is that performance is almost the same for the two classification schemes. For short duration segments, and for N=5, the single network classifier is slightly better. But more importantly, for N=10 and N=15, for longer speech segments (which are needed for reliability and are therefore of more interest) the partitioned system performs slightly better. The fact that the single network degrades more with the increase in N tends to suggest that it needs more training data per talker than the partitioned system as N increases.

It is worth noting that the single network benefitted more from the optimization process. The partitioned system is less sensitive to changes in the variables, which further suggests that the small binary nets are less sensitive to overtraining and need less training data. However, for both cases, the results were not overly sensitive to the network parameters. For example, as the number of hidden nodes changed by a factor of two from the "optimum" value, performance degraded only slightly. The numbers of hidden
Figure 2-2. Speaker identification performance comparison of one large network versus a system of binary-pair networks.
nodes used for the data plotted in Fig. 2–2 were ten, 20, and 45 for the single networks with \( N = 5, 10, \) and 15 respectively, and six for the case of all the binary pair networks.

### 2.4.2 Training Time Comparison of One Large Network vs. BPP Networks

This experiment was designed to compare the training time requirement of the single large (non-partitioned) network to that of the binary partitioned system as a function of the number of speakers (\( N \)). Both systems were required to perform the same task, which was to correctly identify \( N \) speakers, each represented by a single four second segment of test speech. Each system was tested frequently during training to determine the time at which each of the \( N \) speakers were first correctly identified. The results, shown in Fig. 2–3 reflect only the training time and not the time required to frequently evaluate each system. The figure clearly shows that training time increases much more rapidly with \( N \) for the large network versus the binary-pair partitioned system of networks.

Each of the large single networks had \( 2^N \) hidden nodes, which was found nearly optimal for performance in the first experiment. In a few side experiments with \( N=10, 20 \) hidden nodes were found close to optimal with respect to training time as well. Two runs were made for each value of \( N \), with learning rates of 0.1 and 0.2. The two results were then averaged. Either of the two learning rates proved to be consistently better than the other.

The training time requirement of the binary partitioned approach was averaged from a much larger number of runs. 190 binary classifiers were trained, (corresponding
Figure 2-3. Training time comparison of one large network versus a system of binary-pair networks.
to N=20). They were each trained to the required performance threshold in 0.47 to 2.35 seconds, with an overall average of 1.05 seconds. The expected value of the training time for a system with any N was, therefore determined to be $1.05 \cdot N \cdot (N-1)/2$ seconds.

The computer used for these experiments is based on a 16 MHz Intel 80386 processor with an 80387 coprocessor. The programs were written in Microsoft Fortran V5.0 running under OS/2.

2.4.3 Search Methods for Combining BPP decisions into N-way decisions

This set of experiments compares some of the basic search methods of using binary pair neural networks for N-way classification (as discussed in Section 2.2.2). Two independent experiments were conducted under different conditions which resulted in somewhat conflicting results.

In the first experiment only two search methods were compared: the global soft search and the binary tree search based on hard decisions. Both methods used the same set of binary-pair classifiers. The binary-pair classifiers were each trained to get a recognition rate of at least 75% on their respective training data sets on individual speech frames. This resulted in training iterations in the range of 1,000 to 20,000, with an average of about 2,500. The experiment was performed with all 47 speakers, although for the 20 speaker evaluations only the relevant subset of the BPP classifiers were used. The results shown in Fig. 2–4 indicate that with the variable training method the binary tree search consistently performs somewhat better than the global soft search method.
Figure 2-4. Speaker identification performance comparison of two evaluation methods using binary-pair partitioned neural networks trained with the variable training method.
In the second experiment, search methods were again compared but under a different set of conditions. This time the BPP neural networks were each trained a fixed number (10,000) of iterations. They were also larger (ten hidden nodes versus six). Also, the minimum energy threshold was removed for both training and testing data, i.e. only the beginning and ending silence was discarded. The relative performance of the hard decision tree search, the global soft search, the iterative elimination and two variations of the weighted soft search were examined. In the first variation the weight of each BPP classifier was computed as the sum of the two non-weighted sums of the two categories the particular classifier was intended to distinguish. In the second variation, the weight was the product of the two relevant sums. These methods will be referred to as "weighted soft search 1 and 2", respectively. The performance of each of the five search methods is plotted in Fig. 2–5 as a function of test speech length. Under this set of circumstances, the global soft search and the two weighted soft search methods perform about equally well and significantly better than the other two methods, including the hard decision tree-search.

2.4.4 Performance as a Function of the Number of Speakers

This experiment shows recognition results for the binary-pair partitioned system as a function of test speech length for N = 5, 10, 20, and 47. The results, shown in Fig. 2–6 were obtained with the binary tree search method, and close to "optimum" values for hidden nodes (six), training thresholds (75%), etc., determined in the experiment of Section 2.4.1. As expected the results degrade as N increases, in the sense that a larger
Figure 2-5. Speaker identification performance comparison of five evaluation methods using binary-pair partitioned neural networks trained with a fixed number of iterations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2-6. Speaker identification performance of binary-pair neural networks for varying numbers of speakers.
length of test speech is required to reach 100% performance. However, even for 47 speakers, 100% performance is reached with eight seconds of test speech data.

2.5 Summary

In this chapter a previously unexplored method for partitioning a large classification problem using $N(N-1)/2$ pairwise or binary-pair classifiers was investigated. The binary-pair classifier approach has been applied to a talker identification problem using neural networks for the binary-pair classifiers. This partitioned approach performs comparably, or even better, than a single large neural network. For large values of $N$ (greater than ten), the partitioned approach requires only a fraction of the training time required for a single large network. For $N = 47$, the training time for the partitioned network would be about two to three orders of magnitude less than for the single large network. Methods for combining binary-pair decisions to form $N$-way decisions were discussed and their performances were experimentally investigated. Each have advantages over the others depending on the constraints of the classification problem.
CHAPTER THREE

OPTIMIZATION EXPERIMENTS

3.1 Introduction

The previous chapter demonstrated the use of binary-pair (2-way) classifiers for making N-way decisions. It was shown that for the case of neural network classifiers, the main advantage of partitioning is a vast reduction in required training time. In this chapter, a number of parameters affecting the performance of memoryless binary-pair partitioned neural network classifiers are examined in more detail. The parameters considered are the following: the network size, the amount of training (number of training iterations), the training data recognition rate threshold that should be reached before training is stopped, the amount and type of training data, the learning rate, and the initialization of network connection weights and neuron offsets.

The literature on neural networks applied to pattern recognition contains the results of many studies conducted in the past about some of the topics considered here. Very few, however, deal with the specific application considered here. Even studies in speaker identification and/or verification (such as Oglesby and Mason, 1990) generally consider classifiers trained with the speech of several speakers. This is the first reported study in
which the specific aim is to optimally separate or distinguish individual pairs of speakers, which is the speaker recognition task distilled to its most essential, basic form.

One of the fundamental considerations in the application of neural networks to pattern recognition is the optimal network size to be used. If the network is too small, it may be unable to learn enough relevant detail about the data to perform well. On the other hand, a very large network trained with a relatively small data set may overtrain and not perform well with other data. It is generally true about statistical classifiers that more training data is required to estimate more parameters. Duda and Hart (1973) termed this phenomenon as the "curse of dimensionality." The optimal neural network size, therefore, may depend on the amount of training data available.

One commonly known heuristic to limit network size based on the amount of training data is as follows. Each neural network weight is considered to be equivalent to about one and a half bits of information. A network should not have more weights then it takes to encode the category of each training sample (Lang et al, 1990). For example, consider a binary classifier trained with data consisting of 200 tokens from each of the two categories. In this case it takes 400 bits of storage to memorize the data set (1 bit per token). Thus the upper limit to the number of network weights should be 400/1.5, or 267. One obvious limitation to this rule is that it does not consider the degree of redundancy that may be present in the data. For example, the amount of data computed from any speech sample could always be "doubled" just by halving the frame spacing. However, this would not double the amount of information extracted from the speech signal because the redundancy may nearly double.
3.2 Experiments

Most of the experiments performed were designed to optimize the performance of binary-pair partitioned neural networks with respect to one or two variables. For most experiments the constant parameters were set at values very close to optimum, with the main restriction in each case being that the training time be not more than a few days for each experiment.

In each of the seven experiments, the following parameters were constant unless specifically stated otherwise. The speaker population consisted of the first 20 male speakers of the Northern dialect region of the DARPA TIMIT data base (as discussed in Section 1.5), which meant that for each experimental run, \(20 \cdot (20-1)/2 = 190\) binary-pair neural networks were trained. Generally, five sentences (two "SA" and three "SI") were used for training and five sentences ("SX") were used for testing. The input data to the networks were the 15 normalized DCT coefficients, as described in Section 1.4. In every case, the low energy training data was discarded (with normalized DCTC1 of less than \(-1\)). However, all, except beginning and ending silence, of the testing data was retained. In each case, two layer fully interconnected memoryless feed-forward networks were used with 15 inputs, about six hidden nodes and two output nodes. The networks were trained with backpropagation with learning rate of 0.2 and momentum term of 0.7. The networks were typically trained ten-thousand iterations, where an iteration consists of one frame from each of the two categories to be presented to the network for training. During one iteration, two changes are made to each weight and offset of the network. Since the length of training data for each speaker is on the order of ten seconds, or about a
thousand frames, ten-thousand iterations make about ten complete passes over the training data. After each training iteration the data pointers are incremented by 11, 13, or 17, in order for the training to move on quickly to different phonetic regions of the training speech. This is to prevent the network learning phonetic content rather than text-independent speaker characteristics. Before training, each of the network weights and offsets were initialized with random, independent values, uniformly distributed from $-0.05$ to $0.05$. The output targets in each case were 0.999 and 0.001 for the output nodes corresponding to the correct and incorrect categories, respectively.

### 3.2.1 Network Size and Amount of Training

This experiment was designed to determine the effect of varying the network size and the amount of training. Five binary-pair partitioned systems of different size networks, with 2, 4, 6, 10, and 20 hidden nodes were each trained 100, 200, 400, 1000, 2000, 4000, 10,000 and 20,000 iterations. The performance of each network size and training amount combination was tested with one second test speech segments. The resulting binary-pair recognition rates are plotted in Fig. 3–1. Each of the 40 recognition rates were computed from about 9970 recognition attempts. (Each of the 190 binary-pair classifiers was tested with the speech of the corresponding two speakers. Each speaker has an average of 13.2 seconds of test speech, which is spliced into about 26 one-second segments by allowing a half second overlap for adjacent segments.) Figure 3–1 shows that the recognition rate is much more a function of the amount of training than of network size. The two hidden node networks perform somewhat poorer than the others.
Figure 3-1. Average binary-pair classifier performance with one second test speech as a function of network size and amount of training.
at all training levels. The 20 hidden node networks perform poorly up to 4000 iterations, but perform better than the smaller networks after 20,000 iterations. All the network sizes continue improving with additional training. However, even with the exponential rise in the amount of training, the additional gains in performance seem to diminish above 4000 and 10,000 iterations.

3.2.2 In Search of Overtraining

This experiment was aimed at finding out if indeed there is any overtraining of the binary-pair neural networks used for speaker identification. The results of Experiment 3.2.1 do not show any evidence of overtraining when binary-pair neural networks with up to 20 hidden nodes were trained up to 20,000 iterations with five sentences per category. There are various possible reasons for this. One possibility may be that individual BP networks do indeed overtrain, but that they reach their peaks at different points along the training curve. The average of 190 of such curves then may not show a peak. In this experiment, the test performance of individual networks as a function of training is examined. The six binary-pair classifiers needed to perform 4-way speaker identification were trained. To increase the likelihood of overtraining, only one sentence (SA1) was used for each speaker. The performance of each of the six classifiers was tested after 100, 200, 400, 1000, 2000, 4000, 10,000, 20,000, 40,000, and 100,000 iterations with the same one second test speech segments as used in Experiment 3.2.1. The six resulting curves, along with the average curve (labeled "AV") are shown in Fig. 3-2. Five of the six curves show overtraining, the most dramatic being the classifier
Figure 3-2. The performance of six individual binary-pair classifiers as a function of the amount of training showing the effect of overtraining.
trained to distinguish speakers two and three (labeled "2-3"). It reaches 85.7% recognition rate with 2000 iterations, but then eventually levels off at 75.4% at 40,000 and 100,000.

3.2.3 Variable (Feedback) Training

This experiment was designed to determine if performance improvement and/or training time savings may be realized by varying the amount of training for individual binary-pair neural networks based on training data performance. In this experiment the variable was the training data recognition rate at which training was terminated. The training data recognition rate could be computed two different ways. The static recognition rate is determined by turning off the training and making one complete pass over the entire training data set. The dynamic recognition rate on the other hand is computed while training of the network is taking place uninterrupted. Thus the dynamic recognition rate is only an estimate of the true performance of the network with the training data. The static recognition rate is an exact performance measure, but it requires a substantial computational overhead. In this experiment, as well as all other experiments involving variable training, the training data performance reported is the static recognition rate. It should be noted however, that in a real-world implementation of this system, the dynamically computed estimate of recognition rate would be more practical.

The recognition rate targets used in this experiment were 65%, 75%, 80%, 85%, and 90% for individual frames of the training data. An upper limit of 20,000 iterations was placed on the training of each network of each system. Training data performance
was checked after every 200 training iterations. The experiment was independently performed with six hidden-node and 20 hidden-node systems of networks. The performance of each threshold with one-second test speech segments is plotted in Fig. 3–3 for the six and 20 hidden-node cases. The figure shows that performance in both cases continues to improve as the training performance is raised. There are two possible explanations for this. One is that the binary-pair networks do not overtrain with five training sentences. (In Experiment 3.2.2, only one sentence per speaker was used.) The other is that none of the training data performance thresholds tried consistently predicts the peak testing performance of the 190 binary-pair classifiers.

In addition to performance, the other consideration by which the variable training (or training performance thresholding) method was evaluated is the amount of training time required. Figure 3–3 shows the percent recognition rate with one second test speech segments at the binary-pair level and the average training time required by each of the two network sizes for each of the five thresholds. For comparison, the performance as a function of fixed training, obtained in Experiment 3.2.1, of the same two network sizes is also shown. The variable training method consistently requires less training for the same performance as the fixed training method. For example, comparing the twenty hidden node curves (NH=20), it is apparent that 99 percent performance with one second test speech segments requires 20,000 fixed iterations but only an average of about 6,300 iterations if training is terminated when individual frame training data recognition rate reaches 90 percent.
Figure 3-3. Performance comparison of variable training versus fixed training with one second test speech segments.
3.2.4 The Amount of Training Data

This experiment was designed to determine the effect of varying the number of sentences used for training. Eight systems were trained each consisting of six hidden node classifiers trained 10,000 iterations. Each system was trained with a different number of sentences ranging from one to eight. As mentioned in Section 1.5, the average sentence length is about three seconds, thus with 10 ms frame spacing the average sentence is represented by 300 tokens (frames). The performance of each of the eight systems is shown in Fig. 3–4 with various test speech lengths ranging from individual (32 ms) frames to eight seconds (averaged over 800 frames). The results of this experiment are potentially biased by the fact that some of the sentences (SA) have the same text from speaker to speaker, whereas the SI and SX sentences vary from speaker to speaker. The general trend tends to be a significant improvement from one to two sentences, but then diminishing improvements as more sentences are added to the training set.

3.2.5 Fixed-text vs. Random-text Training Data

This is a comparison of fixed-text versus random-text speech sentences used for training speaker identification system neural networks. The motivation behind this comparison is the fact that the speech signal carries stronger phonetic information than speaker identity information, especially in case of speakers that sound a lot alike. If that is the case, then there is the potential that the neural network (or other classifier) may learn to separate the training data more by phonetic content than speaker identity. The network may learn phonetic content of training data to some extent and may be mislead...
Figure 3-4. Binary classifier performance as a function of the number of training sentences and the test speech length.
during testing. There is no danger of this if the spoken text of the speakers is the same. Also, the chances of highly "lopsided" data are more and more reduced with increased amounts of training data, unless the data is specifically picked to be phonetically unbalanced. Consider, for example, a speech data base in which one of the speakers is represented by the following text: "The rain in Spain falls mainly in the plains." A speaker identification system trained with this data would almost certainly learn to associate the "a" sound with that particular speaker. During testing, the system would tend to pick that speaker whenever it were presented the "ai" sound, regardless of the actual speaker.

Four systems of binary-pair networks were trained to examine the performance degradation that may be expected when random-text speech is used for training, rather than speech with the same text from speaker to speaker. One system was trained with the two "SA" sentences which are the same for all speakers and three were trained with various two sentence combinations of "SI" random text sentences. All other factors affecting performance, including testing data (all five SX sentences) were the same for each system. The recognition rate as a function of test speech is plotted for the binary-pair classifiers of the four systems in Fig. 3–5. The performances of the three random-text trained systems are nearly identical to one-another (within one-half percent with any test speech length) and are not more than two percent worse than the system trained with fixed-text (SA) speech.
Figure 3-5. Performance comparison of binary classifiers trained with random-text (SI) speech versus fixed-text (SA) speech.
3.2.6 Learning Rate and Amount of Training

This experiment is similar to Experiment 3.2.1, but was designed to determine the effect of varying the learning rate and the amount of training. Four identically initialized system of six hidden node networks were each trained with learning rates of 0.1, 0.2, 0.3, and 0.4. The performance of each was checked after 100, 200, 400, 1000, 2000, 4000, 10,000, and 20,000 iterations, with the same one second test speech segments as used in Experiment 3.2.1. The recognition rate results are plotted in Fig. 3-6. Each learning rate performed about the same. The only noticeable differences are that the smallest learning rate (0.1) combined with the fewest training iterations (100) performed poorly, and that the highest learning rate (0.4) combined with the most iterations (20,000) performed relatively poorly. One obvious conclusion is that since the networks learn fastest in the beginning with a larger learning rate and then need a smaller learning rate in the final stages of training to reach optimal performance, the rate should be reduced during training. But since the effect is so small even over a large range of learning rates, finding the optimal learning rate schedule among limitless possibilities appears to be relatively unimportant.

3.2.7 Random Initialization

Experiments 3.2.2 and 3.2.3 show that individual binary-pair networks have highly varying performance levels. Furthermore, some train very quickly, while others take a lot longer to reach peak performance. This, without doubt, is due mainly to varying degrees of similarity between speaker pairs. There is only one other potential source of
Figure 3-6. Binary classifier performance with one second test speech as a function of learning rate and amount of training.
variability, and that is the random initial network weights. In this experiment, an attempt was made to determine the relative affects of data versus initial weights on the performance of binary-pair classifiers trained to distinguish pairs of speakers. Ten six-hidden-node binary classifiers were randomly initialized with weights uniformly distributed from $-0.05$ to $+0.05$. Each were then independently trained 1,000 iterations to distinguish between ten pairs of speakers. The resulting 100 binary classifiers were then each tested (with the training data) to determine the effect of initial weights compared to the effect of differences in training data. The resulting 100 error rates were arranged in a ten by ten matrix, as shown in Table 3-1, where each row represents a set of initial weights and each column a pair of speakers. The average and the standard deviation of each row and each column was then computed. The average of the row and the column standard deviations was then found to be 0.054405 and 0.005861, respectively. In other words, on the average, differences in data (distinguishability of speakers) caused 9.3 times as much deviation in the average error as did the initial neural network weights.

The experiment was repeated with various numbers of training iterations per data/network combinations, ranging from 100 to 10,000. After each set of iterations the overall average error, the average error deviation caused by data, the average error deviation caused by initialization, and the ratio of the two average deviations was computed and are shown in Table 3-2. As expected, the effect of data is consistently greater than that of network initialization. However, the effect of the network initialization never does disappear with more and more training. In fact, after declining in the first 1000 iterations, it makes a gradual comeback in the 2,000 to 10,000 range.
<table>
<thead>
<tr>
<th>NN</th>
<th>1-2</th>
<th>1-3</th>
<th>1-4</th>
<th>1-5</th>
<th>2-3</th>
<th>2-4</th>
<th>2-5</th>
<th>3-4</th>
<th>3-5</th>
<th>4-5</th>
<th>AV</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.2918</td>
<td>.2346</td>
<td>.2433</td>
<td>.1872</td>
<td>.2212</td>
<td>.1850</td>
<td>.1007</td>
<td>.1682</td>
<td>.1413</td>
<td>.2689</td>
<td>.2042</td>
<td>.0559</td>
</tr>
<tr>
<td>3</td>
<td>.2939</td>
<td>.2151</td>
<td>.2333</td>
<td>.1857</td>
<td>.2160</td>
<td>.1853</td>
<td>.1009</td>
<td>.1662</td>
<td>.1385</td>
<td>.2806</td>
<td>.2015</td>
<td>.0566</td>
</tr>
<tr>
<td>5</td>
<td>.2957</td>
<td>.2422</td>
<td>.2294</td>
<td>.1876</td>
<td>.2143</td>
<td>.1816</td>
<td>.1011</td>
<td>.1727</td>
<td>.1386</td>
<td>.2672</td>
<td>.2030</td>
<td>.0559</td>
</tr>
<tr>
<td>AV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.2942</td>
<td>.0559</td>
</tr>
<tr>
<td>SD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.0090</td>
<td>.0038</td>
</tr>
</tbody>
</table>

**Table 3-1.** Average training data error rates after 1000 training iterations for combinations of training data and randomly initialized neural networks.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### Table 3-2

The relative effect of training data versus random initialization on average training data error rates as a function of the amount of training.

<table>
<thead>
<tr>
<th>Training Iterations</th>
<th>Ratio of average standard deviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.818</td>
</tr>
<tr>
<td>100</td>
<td>4.763</td>
</tr>
<tr>
<td>200</td>
<td>5.688</td>
</tr>
<tr>
<td>400</td>
<td>9.435</td>
</tr>
<tr>
<td>1,000</td>
<td>9.283</td>
</tr>
<tr>
<td>2,000</td>
<td>4.917</td>
</tr>
<tr>
<td>4,000</td>
<td>3.407</td>
</tr>
<tr>
<td>10,000</td>
<td>2.365</td>
</tr>
</tbody>
</table>
This clearly indicates that the various initializations do not ultimately end up with the same solution after training, but in fact result in different local error minima. However, the differences in quality of the final solutions tends to be rather insignificant.

3.3 Summary

Several considerations effecting the performance of binary-pair neural network classifiers used for speaker identification were investigated. Some of these, such as network size, learning rate, overtraining, and network initialization, were shown to be relatively unimportant. The most important factor found influencing overall performance is the number of training iterations. Also effecting performance, but to a lesser degree, are the amount of training data (number of training sentences) and text consistency from speaker to speaker. Variable training, i.e. terminating training of individual binary-pair networks based on training data performance, did not significantly improve system level performance with test data, however it did result in substantial training time savings over fixed training.
CHAPTER FOUR

SPEAKER IDENTIFICATION POTENTIAL OF SPEECH DATA
BASED ON PHONETIC GROUP AND SIGNAL ENERGY

4.1 Introduction

In this chapter the effect of phonetic content and the short term energy of the speech signal on automatic speaker recognition will be examined. Potential performance improvements by the selective use of training data will also be examined. Finally, various methods will be compared which combine many short term neural network decisions with phonetic and energy information into decisions on longer speech segments.

4.1.1 Which speech data carry speaker identity information?

All speech data does not carry the same amount of speaker identity information. Complete silence, which may be embedded in continuous speech between words, obviously does not carry useful information. But exactly what threshold of signal energy should be used to separate useful and useless segments of the raw signal is less obvious since even silence has some noise energy. Even if experiments with a specific high quality data base (such as TIMIT) show that there is useful information in very low
energy portions of the signal, in general they should be used with caution. In silence, or near silence, the effects of the recording conditions are more prevalent, such as background noise or reverberation (echo), which can actually be misleading in training since they cannot be counted on to be the same during use (testing) of the speaker recognition system.

The results of past speech recognition research (Jagharghi, 1990, and Nossair, 1989) show that automatic recognition of vowels is much more sensitive to speaker variations than recognition of stop consonants. It seems reasonable to presume that since vowels vary more with speakers, they also carry more speaker identity information. Nasals are also good candidates to carry speaker identification information since their production depends on resonance of the nasal cavity, which varies considerably among humans.

4.1.2 Can performance be improved by selective use of training data?

Training any statistical classifier with data that does not contain relevant information may or may not hurt, but it does not help. The data may even contain systematically misleading information, such as, for example, recording conditions. It may be worth while, therefore to discard certain portions of the training data.

In all experiments prior to this chapter, all speech data was treated the same, both for training and testing of speaker identification systems, without regard to phonetic content or signal energy. Pilot experiments have indicated, however, that very low energy speech data (silence, or near silence) has very small speaker identity information. For this reason, the experiments reported in Chapter 2 were performed with "low energy" speech
frames discarded both for training and testing. Low energy was defined as the first
discrete cosine transform coefficient (DCTC1) being less than one standard deviation
below average. This energy threshold eliminated approximately 15 percent of both
training and testing data. Since this threshold is somewhat arbitrary and to allow
comparison of performance results to those of other methods reported in literature, in
Chapter 3 all, including low energy data (except beginning and ending silence) was used
for testing. For the training data, however, the energy threshold was retained, since it was
believed to cause a performance improvement. One of the aims of this chapter is to
examine more closely the effect of training data energy thresholding on test performance.

An alternative to completely eliminating less useful training data could be to
reduce its relative weight in training. In context of the current neural network method
this could mean reducing either or both the learning rate and the strength of the desired
output (target). In the current study this approach was not examined.

4.1.3 Given a trained set of BPP networks, can their use be improved?

The overall scheme for performing speaker identification with binary-pair
partitioned neural networks involves three basic steps. First, the networks make
independent soft binary decisions on individual short (32 ms) frames of the test speech.
Second, these frame level decisions are combined into segment level binary decisions.
Third, the binary segment level decisions are combined into a top level (N-way) decision
using one of the several methods already discussed in Section 2.2.2. Possible ways to
improve on the first two steps are discussed in the following two subsections.
4.1.3.1 Improving Frame Level Probability Estimates by Incorporating Energy and Phonetic Information

In all previous experiments, the neural network outputs are used as probability estimates. The probability estimated is that the category corresponding to the output node is the correct category. In general, an improved probability estimate (p) may be computed, which includes not only the neural net output (x), but also incorporates the frame energy (y) and the phonetic group (g):

\[ p = f(x, y, g). \]

X and y are continuous valued variables. However, g is not only discrete but non-ordered. For this reason, it is necessary to use a separate function for each value of g:

\[ p = f_g(x, y). \]

There are many possible function types that might be worth investigating. The simplest non-parametric approach would be to place a rectangular grid over the x–y plane and compute a probability for each bin of the grid as the percentage of correct decisions made with data that fell into it. The size of the grid would determine the tradeoff between how accurately the probabilities are computed versus the coarseness of the grid. In densely populated regions of the x–y plane the grid size should be smaller to reduce the coarseness; in sparsely populated regions the grid should be larger to improve the probability estimate. A more general method would be to use a vector quantizer to form evenly populated non-rectangular tessellations of the plane. Also, to reduce the effect of coarseness, some type of interpolation might be used to smooth the probability contour between the anchor points, where each anchor point is the centroid of a tessellation.
A simple parametric function that may be used to compute probability estimates is an N-th order polynomial whose coefficients are derived using the least mean squares (LMS) criterion. For example, the coefficients of the second order polynomial,

\[ p = A_{20}x^2 + A_{02}y^2 + A_{11}xy + A_{10}x + A_{01}y + A_{00} \]

are obtained by solving the following set of simultaneous linear equations:

\[
\begin{bmatrix}
\Sigma x^4 & \Sigma x^3 y & \Sigma x^2 y^2 & \Sigma x^3 & \Sigma x^2 y & \Sigma x^2 \\
\Sigma x^3 y & \Sigma x^2 y^2 & \Sigma x^3 & \Sigma x^2 y & \Sigma x^2 & \Sigma x \\
\Sigma x^2 y^2 & \Sigma x^3 & \Sigma y^4 & \Sigma x^2 & \Sigma x & \Sigma \\
\Sigma x^3 & \Sigma x^2 y & \Sigma x^2 & \Sigma x & \Sigma x & \Sigma \\
\Sigma x^2 y & \Sigma x^2 & \Sigma x & \Sigma x & \Sigma x & \Sigma \\
\Sigma x^2 & \Sigma x & \Sigma y & \Sigma x & \Sigma y & \Sigma \\
\end{bmatrix}
\begin{bmatrix}
A_{20} \\
A_{11} \\
A_{02} \\
A_{10} \\
A_{01} \\
A_{00}
\end{bmatrix}
= 
\begin{bmatrix}
\Sigma p_k x^2 \\
\Sigma p_k x y \\
\Sigma p_k y^2 \\
\Sigma p_k x \\
\Sigma p_k y \\
\Sigma p_k
\end{bmatrix}
\]

where \( k \) is the frame index, \( K \) is the total number of training frames, and \( p_k \) is either 1.0 or 0.0 depending on whether frame \( k \) belongs to the category corresponding to the particular output neuron.

4.1.3.2 Combining Frame Level Probability Estimates into Segment Level Decisions

Another topic considered in this chapter is the method by which individual frame level decisions are combined into segment level decisions. Reliable text-free speaker identification requires speech segments of one to ten seconds. Therefore, with a 10 ms frame spacing 100 to 1,000 individual decisions must be integrated. Generally, given M
independent estimates of the probability of an event \( \{x=A\} \), denoted by \( p_i \) for the \( i^{th} \) estimate, then

\[
P\{x=A\} = \frac{\prod_{i=1}^{M} p_i}{\prod_{i=1}^{M} p_i + \prod_{i=1}^{M} (1-p_i)}.
\]

In this case \( x \) is the unknown speaker which is either A or B. The probability \( P\{x=B\} = 1 - P\{x=A\} \) is estimated by \( 1 - p_i \). To decide between A and B, therefore, it is sufficient to compare the numerators of Eq. (4-1) applied to A and B, since their denominators are the same. In implementing Eq. (4-1), it is useful to sum the logarithms of the probability estimates rather than multiplying them directly, to prevent numeric underflow. Even so, this method of combining frame level decisions is obviously very sensitive to small errors in probabilities that are close to zero. To reduce this sensitivity, a small constant (0.01 to 0.1) may be added to the probability estimate before computing its logarithm. Another way of combining frame level decisions to arrive at the segment level decision is to simply time integrate (add) them. This is the method that has been used in previous experiments. Potential advantages over the implementation of Eq. (4-1) are that it is less sensitive to small errors in probability estimates that are close to zero, and numeric underflow is not possible.
4.2 Experiments

The objectives of the experiments presented in this chapter are the following:

(1) to determine how phonetic category and short term speech energy affect both the training and use (testing) of speaker identification classifiers with respect to performance;

(2) to investigate the possibility of improved performance by eliminating some of the less useful data from the training set based on energy or phonetic content;

(3) to incorporate speech energy and phonetic group information to compute improved probability measures at the individual speech frame level;

(4) to compare methods of combining frame-level probability estimates into segment-level decisions; and

(5) to evaluate the usefulness of augmenting the feature set with phonetic group information.

The following conditions were the same in all the experimental runs presented in this chapter, unless specifically stated otherwise. In each case a binary-pair partitioned system of neural network classifiers were used. The architecture of each network was the same: two-layer, memoryless, feed-forward, with 15 inputs, six hidden nodes, and two output nodes. The networks were trained with error backpropagation with a learning rate of 0.2, and momentum term of 0.7. Each system was trained to identify the same 20 speakers as in Chapter 3, therefore each consisted of 190 binary-pair subclassifiers. The binary-pair classifiers were combined to make 20-way decisions using the "tree search based on hard decisions" method. In most of these experiments the training and testing
sentences were swapped as compared to experiments in other chapters. Here, the SX sentences were used for training because they are phonetically balanced. Generally, all frames of the speech data were used for both training and testing, except beginning and ending silence. This contrasts with Chapters 2 and 3, in which low energy data frames were discarded as stated.

4.2.1 Intrinsic Potential of Data Based on Phonetic Category

Two experiments were performed to determine the relative merit of speech data for speaker identification based on the phonetic group and relative signal energy.

In the first experiment, a system of networks was trained using five complete sentences per speaker, and tested with individual frames at the binary-pair level. The recognition rates of each phoneme were then computed individually. The results are shown in Table 4–1, along with the percentage of occurrence (based on number of frames) and average normalized energy of each of the phonemes. Note that the closure and burst regions of stop consonants are considered separately. Table 4–2 shows the same information for five broad phonetic groups that Table 4–1 does for individual phonemes. The five groups are: (1) stop consonants, (2) nasals, (3) fricatives, (4) liquids, glides, and h, and (5) vowels. Liquids, glides, and h are lumped together because they are most similar and are not represented by enough speech frames to warrant individual groups. Combined, they represent 11.9% of speech frames, as compared to 47.5% for vowels, as shown in Table 4–2. The overall conclusion of this experiment is that the phonetic groups perform very differently. Vowels are best with 88.3% and stops are
<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Occurrence (%)</th>
<th>Recognition Rate (%)</th>
<th>Av. Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>.217</td>
<td>77.8</td>
<td>.047</td>
</tr>
<tr>
<td>d</td>
<td>.836</td>
<td>64.0</td>
<td>.021</td>
</tr>
<tr>
<td>g</td>
<td>.803</td>
<td>60.0</td>
<td>-.336</td>
</tr>
<tr>
<td>p</td>
<td>.655</td>
<td>66.2</td>
<td>-.159</td>
</tr>
<tr>
<td>t</td>
<td>1.675</td>
<td>68.2</td>
<td>-.012</td>
</tr>
<tr>
<td>k</td>
<td>1.900</td>
<td>66.8</td>
<td>-.078</td>
</tr>
<tr>
<td>dx</td>
<td>.601</td>
<td>75.0</td>
<td>-.038</td>
</tr>
<tr>
<td>q</td>
<td>1.914</td>
<td>74.9</td>
<td>-.478</td>
</tr>
<tr>
<td>bcl</td>
<td>.749</td>
<td>64.6</td>
<td>-1.437</td>
</tr>
<tr>
<td>dcl</td>
<td>2.001</td>
<td>69.6</td>
<td>-1.376</td>
</tr>
<tr>
<td>gcl</td>
<td>1.042</td>
<td>71.6</td>
<td>-1.120</td>
</tr>
<tr>
<td>pcl</td>
<td>1.002</td>
<td>60.5</td>
<td>-1.624</td>
</tr>
<tr>
<td>tcl</td>
<td>3.260</td>
<td>63.8</td>
<td>-1.495</td>
</tr>
<tr>
<td>kcl</td>
<td>3.115</td>
<td>60.7</td>
<td>-1.510</td>
</tr>
<tr>
<td>m</td>
<td>1.512</td>
<td>85.9</td>
<td>-.460</td>
</tr>
<tr>
<td>n</td>
<td>2.873</td>
<td>90.1</td>
<td>-.492</td>
</tr>
<tr>
<td>ng</td>
<td>.666</td>
<td>85.6</td>
<td>-.562</td>
</tr>
<tr>
<td>em</td>
<td>.040</td>
<td>82.8</td>
<td>-.606</td>
</tr>
<tr>
<td>en</td>
<td>.554</td>
<td>85.8</td>
<td>-.594</td>
</tr>
<tr>
<td>eng</td>
<td>.000</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>nx</td>
<td>.206</td>
<td>84.1</td>
<td>.294</td>
</tr>
<tr>
<td>f</td>
<td>1.201</td>
<td>62.3</td>
<td>-.322</td>
</tr>
<tr>
<td>s</td>
<td>7.211</td>
<td>75.2</td>
<td>-.109</td>
</tr>
<tr>
<td>z</td>
<td>1.675</td>
<td>76.1</td>
<td>-.303</td>
</tr>
<tr>
<td>sh</td>
<td>2.417</td>
<td>60.5</td>
<td>.025</td>
</tr>
<tr>
<td>zh</td>
<td>.018</td>
<td>69.5</td>
<td>-.287</td>
</tr>
<tr>
<td>v</td>
<td>.807</td>
<td>78.7</td>
<td>-.416</td>
</tr>
<tr>
<td>th</td>
<td>.242</td>
<td>67.2</td>
<td>-.643</td>
</tr>
<tr>
<td>ch</td>
<td>.130</td>
<td>48.4</td>
<td>-.047</td>
</tr>
<tr>
<td>jh</td>
<td>.268</td>
<td>58.5</td>
<td>-.138</td>
</tr>
<tr>
<td>dh</td>
<td>1.035</td>
<td>72.6</td>
<td>-.518</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Occurrence (%)</th>
<th>Recognition Rate (%)</th>
<th>Av. Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>3.314</td>
<td>85.3</td>
<td>.141</td>
</tr>
<tr>
<td>r</td>
<td>.856</td>
<td>83.8</td>
<td>.233</td>
</tr>
<tr>
<td>y</td>
<td>1.364</td>
<td>82.3</td>
<td>-.086</td>
</tr>
<tr>
<td>w</td>
<td>2.406</td>
<td>80.2</td>
<td>-.313</td>
</tr>
<tr>
<td>hh</td>
<td>.821</td>
<td>74.3</td>
<td>-.054</td>
</tr>
<tr>
<td>hv</td>
<td>.405</td>
<td>74.6</td>
<td>.224</td>
</tr>
<tr>
<td>iy</td>
<td>6.267</td>
<td>88.0</td>
<td>.260</td>
</tr>
<tr>
<td>ih</td>
<td>3.441</td>
<td>90.3</td>
<td>.441</td>
</tr>
<tr>
<td>eh</td>
<td>3.075</td>
<td>92.1</td>
<td>.814</td>
</tr>
<tr>
<td>ae</td>
<td>5.869</td>
<td>90.4</td>
<td>.763</td>
</tr>
<tr>
<td>ux</td>
<td>1.172</td>
<td>86.7</td>
<td>.312</td>
</tr>
<tr>
<td>ix</td>
<td>2.916</td>
<td>88.6</td>
<td>.253</td>
</tr>
<tr>
<td>ax</td>
<td>.930</td>
<td>88.2</td>
<td>.247</td>
</tr>
<tr>
<td>ah</td>
<td>1.730</td>
<td>89.3</td>
<td>.654</td>
</tr>
<tr>
<td>uw</td>
<td>.883</td>
<td>80.5</td>
<td>.029</td>
</tr>
<tr>
<td>uh</td>
<td>.213</td>
<td>82.0</td>
<td>.385</td>
</tr>
<tr>
<td>ao</td>
<td>3.455</td>
<td>86.0</td>
<td>.436</td>
</tr>
<tr>
<td>aa</td>
<td>3.419</td>
<td>85.6</td>
<td>.836</td>
</tr>
<tr>
<td>ey</td>
<td>2.261</td>
<td>93.2</td>
<td>.602</td>
</tr>
<tr>
<td>ay</td>
<td>2.240</td>
<td>89.9</td>
<td>.707</td>
</tr>
<tr>
<td>oy</td>
<td>1.009</td>
<td>91.2</td>
<td>.415</td>
</tr>
<tr>
<td>aw</td>
<td>.850</td>
<td>89.1</td>
<td>.804</td>
</tr>
<tr>
<td>ow</td>
<td>2.569</td>
<td>89.9</td>
<td>.396</td>
</tr>
<tr>
<td>er</td>
<td>2.290</td>
<td>86.3</td>
<td>.291</td>
</tr>
<tr>
<td>axr</td>
<td>2.178</td>
<td>81.4</td>
<td>.054</td>
</tr>
<tr>
<td>el</td>
<td>.731</td>
<td>79.5</td>
<td>.054</td>
</tr>
</tbody>
</table>

**Table 4-1.** Percentage of occurrence, binary level talker recognition rate, and average normalized energy of individual 32 ms speech frames of each of the 57 phonemes.
Table 4-2. Percentage of occurrence, binary level talker recognition rate, and average normalized energy of individual 32 ms speech frames of each of the five phoneme groups.
worst with 66.3%. (For binary decisions, 50% recognition rate represents zero information.) Generally, there is a high correlation between signal energy and recognition rate, with the notable exception of the nasals. The nasals perform almost as well as vowels (87.8%) but have less average energy. Figure 4–1 is a scatter plot showing the correlation between energy and recognition rate for the same data as in Table 4–1.

The results of experiment one show the relative performance of the five basic phonetic groups with a system trained with natural speech consisting of complete sentences. Natural speech is phonetically unbalanced because the frequency and duration of various phonemes is uneven. This unevenness of the training data potentially biased the results of experiment one in favor of more frequently occurring and longer duration phonemes. Experiment two was performed to give a relative measure of usefulness of each phoneme group both for training and testing without the bias present in experiment one.

In the second experiment, five independent 20 speaker (190 pair) systems of networks were trained, one on each phonetic group. The amount of training data was the same for each of the twenty speakers and each of the five phonetic groups. This amount, 45 frames, was limited by the fact that the training data (five SX sentences) of one of the speakers contains only this many frames of one of the phonetic groups (nasals). After training, the individual binary classifiers of each of the systems was tested on each phonetic group individually. The results are a five by five matrix of training/testing combinations shown in Table 4–3. The basic conclusion is that nasals and vowels are best for both training and testing of speaker identification systems. One interesting result
Figure 4-1. Speaker identification performance and energy of individual 32 ms speech frames averaged for each phoneme and grouped into five broad phonetic categories (from Table 4-1).
<table>
<thead>
<tr>
<th>Training data phoneme group</th>
<th>Testing Data Phoneme Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stops</td>
</tr>
<tr>
<td>Stops</td>
<td>59.2</td>
</tr>
<tr>
<td>Nasals</td>
<td>58.9</td>
</tr>
<tr>
<td>Fricatives</td>
<td>55.8</td>
</tr>
<tr>
<td>Liq,Glides,H</td>
<td>55.8</td>
</tr>
<tr>
<td>Vowels</td>
<td>56.5</td>
</tr>
</tbody>
</table>

**Table 4-3.** The relative potential of phoneme groups for speaker identification with the bias due to uneven distribution of training data removed.
is that nasals and vowels perform better than liquids, glides, and h even on the system trained with liquids, glides, and h, (73.1, 75.3, and 72.6%, respectively). This is unusual since statistical classifiers generally perform better with test data that is more similar to the training data. One possible explanation of the results is that the speaker identification information present in the various phonemes is similar, but some phonemes contain more of it than others. Another reason may simply be coarticulation, i.e., the phonemes being somewhat modified by their immediate neighbors, as well as the presence of in-between regions that are somewhat arbitrarily assigned to one of the two surrounding phonemes.

4.2.2 With testing fixed, can training be improved?

In experiment three, two possible training improvements were investigated. The first is the elimination of low energy training data. Five different energy thresholds were compared to the control case of using all training data. The second approach was to eliminate training data based on phonetic group. Only one of many possibilities was investigated, that of training with vowels and nasals only, discarding the all others.

Two sets of experimental runs were made. Each set consisted of seven systems of networks trained and tested. Each of the seven were the same with respect to network architecture, training method, evaluation (testing) method and testing data. The only variable between the systems was the training data. The first was trained on all data. Five of the systems were trained with data that were required to meet minimum energy thresholds. These thresholds were specified in terms of the number of standard deviations away from average energy. These were -2, -1.5, -1, -0.5, and 0. The seventh system
was trained with nasals and vowels only. In the first set of experimental runs the five SX sentences were used for training and the two SA and three SI sentences for testing. Four hidden-node binary-pair networks, each trained 400 iterations, were used. The 20-way classification results of each of the seven systems of experimental run one are shown as a function of test speech length in Fig. 4–2. The results indicate that eliminating low energy training data tends to improve performance even if all, including low energy data is used for testing. Performance does not seem to be sensitive to the threshold; thresholds of one to two standard deviations below average energy work equally well.

The second experimental run differs from the first mainly in that the extra variable of number of training iterations is added. Performance of each of the seven systems is checked after 100, 200, 400, 1000, and 2000 iterations. In this case six hidden node networks were used. The SA and SI sentences were used for training and SX sentences for testing, which is consistent with experiments performed for other chapters. In the first experimental run the data partitioning was consistent with experiments described in this chapter. The 20-way classification results with one-second test speech lengths for this set of runs are shown in Fig. 4–3. The results indicate that the ideal energy threshold is a function of the amount of training. The general trend is that with very limited training it is advantageous to use only better quality (higher energy) data. With more and more training, however, it is better to use more of (or all of) the available training data. The systems trained with high energy thresholds, as well as the one trained with nasals and vowels only, consistently performed poorly. One reason for this could be the reduction of the amount of training data. Figure 4–4 shows the average and minimum training data
Figure 4-2. The relative performance of the seven training data selection criteria as a function of test speech length. (Numbers indicate minimum normalized energy, N&V indicates nasals and vowels.)
Figure 4-3. One second test speech recognition rates as a function of training data selection criterion and amount of training (varied from 100 to 2,000 iterations).
Figure 4-4. The amount of training speech available for 20 speakers depending on the selection criterion.
that remain for the 20 speakers for each of the six data elimination schemes as compared
to retaining all the data. The performance results strongly suggest that the reduction in
the quantity of training data based on energy or phonetic considerations does not justify
the improvement in quality if the neural network is to be thoroughly trained.

4.2.3 With training fixed, can testing be improved?

Two considerations were investigated. The first is the estimation of frame level
probabilities and the second is the method by which they are combined to make a
segment level binary decision. Theoretically, given additional apriori knowledge of how
well testing data should do based on phonetic group and/or energy, this information may
be used to help combine small (frame level) decisions to make improved segment level
binary decisions.

Five methods to estimate frame level probabilities were compared:

(1) using the neural net output (x) directly without energy (y) or phonetic group
information, i.e., \( p_i = x \);

(2) computing probabilities using the same second order polynomial regardless of
phonetic group,

\[
p_i = A_{20}x^2 + A_{02}y^2 + A_{11}xy + A_{10}x + A_{01}y + A_{00};
\]

(3) computing probabilities using the same third order polynomial regardless of phonetic
group,

\[
p_i = A_{30}x^3 + A_{03}y^3 + A_{21}x^2 y + A_{12}xy^2 +
+ A_{20}x^2 + A_{02}y^2 + A_{11}xy + A_{10}x + A_{01}y + A_{00};
\]
(4) same as (2) except a separate 2nd order polynomial is used for each of the five phonetic groups;

(5) same as (3) except a separate 3rd order polynomial is used for each of the five phonetic groups. The polynomial coefficients were computed using the least mean squares criterion from performance with the training data.

Note that a separate set of polynomial coefficients could be computed and used for each of the BPP classifiers, or for each of the speakers, or a single set could be used globally. In this case, a separate set of coefficients was computed for each of the speakers. Since the number of coefficients is much smaller than the amount of training data per speaker, computing a separate set of coefficients for each BPP classifier would have probably performed best.

Four methods for combining frame level probabilities to estimate the relative segment level probability were used:

(a) summing the frame level probabilities \( p_i \);

(b) summing \( \log(p_i) \), (which is equivalent to multiplying them)\(^1\);

(c) summing \( \log(p_i+0.01) \); and

\(^1\) Note that when determining segment level binary hard decisions, it is sufficient to compute a "relative" probability for each of the two candidates. Consider \( \alpha \) and \( \beta \) as the two "true" probabilities. If \( \alpha > \beta \), then \( f(\alpha) > f(\beta) \) if \( f(\cdot) \) is any monotonically increasing function. In this case \( f(\alpha) \) and \( f(\beta) \) may be termed the relative probabilities and can be used in place of the true probabilities because the only consideration is which of the two is greater.
(d) summing \( \log(p_i + 0.1) \),

where \( i \) is the frame index spanning the test speech segment.

Each of the five methods for estimating frame level probabilities was combined with each of the four methods for combining frame probabilities (20 combinations). These were tested with one second test speech lengths and the resulting error rates at the binary level are shown in Table 4-4. Each of the 20 error rates was obtained from 10,127 recognition attempts, implying a high level of statistical reliability in the results (see Appendix B). The methods incorporating energy alone did reduce error rates (misclassifications) by about one third, as compared to using neural net outputs alone. The further addition of phonetic group information however, did not significantly reduce error rates. The third order polynomial performed only slightly better than the second order. Of the four methods for combining frame level decisions, three (a, c, and d) perform about the same and method (b), summing log probabilities is inferior, undoubtably because of errors made with values of \( p \) close to zero.

An examination of the average polynomial coefficients computed for probability estimations reveals why the addition of energy and phonetic group information resulted in relatively small drops in error rates. For example, the LMS polynomial coefficients for method (4) are given in Table 4-5. The coefficients are very similar for all five phonetic groups, indicating only a small effect due to the phonetic group. Apparently the neural network outputs already account for most of the phonetic information. The energy also contributes relatively small additional information since the coefficients pertaining to the energy terms (\( A_{02}, A_{11}, \) and \( A_{01} \)) are small compared to \( A_{10} \), the coefficient of the
Method of estimating frame level probabilities from neural network output (x), frame energy (y), and phonetic group (g)

<table>
<thead>
<tr>
<th>Method of estimating frame level probabilities from neural network output (x), frame energy (y), and phonetic group (g)</th>
<th>(1) (p_i = x) (NN output alone)</th>
<th>(2) same 2nd order polynomial of x and y for each g</th>
<th>(3) same 3rd order polynomial of x and y for each g</th>
<th>(4) separate 2nd order polynomial of x and y for each g</th>
<th>(5) separate 3rd order polynomial of x and y for each g</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) (\sum p_i)</td>
<td>1.19</td>
<td>.80</td>
<td>.76</td>
<td>.81</td>
<td>.72</td>
</tr>
<tr>
<td>(b) (\sum \log(p_i))</td>
<td>1.28</td>
<td>5.87</td>
<td>3.50</td>
<td>5.71</td>
<td>4.42</td>
</tr>
<tr>
<td>(b) (\sum \log(p_i + .01))</td>
<td>1.17</td>
<td>.80</td>
<td>.74</td>
<td>.68</td>
<td>.72</td>
</tr>
<tr>
<td>(b) (\sum \log(p_i + .1))</td>
<td>1.23</td>
<td>.74</td>
<td>.78</td>
<td>.73</td>
<td>.70</td>
</tr>
</tbody>
</table>

**Table 4-4.** Binary level error rates with one second test speech lengths with various combinations of evaluation methods.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4-5. Average polynomial coefficients derived by the Least Mean Squares method and used for probability estimation with Method 4.
neural network output. The fact that $A_{10}$ for all five phonetic groups is very close to 1.0 means the probability is well estimated by the neural network output alone, i.e., $p_i = x$.

4.2.4 Augmenting the Feature Set with Phonetic Group Information

In this experiment the input feature set (15 DCTC's) were augmented by five more parameters, each representing one of the five phonetic categories. The parameter corresponding to the correct phonetic group was assigned a one, while the other four parameters were assigned zeroes. This was done in both training and testing. The performance of the 20 speaker system is plotted as a function of test speech length in Fig. 4–5. The performance of a similar system whose feature set is not augmented with phonetic group information is also plotted for comparison. The results show that the addition of phonetic group information to the speaker identification system does not significantly improve its performance.

4.3 Conclusions

Experiments described in this chapter have shown that the short term energy and phonetic content of the speech signal have considerable effect on its usefulness toward identifying the speaker. A number of methods incorporating energy and phonetic group information into the neural network speaker identification system were investigated. The performance improvements were fairly small. This suggests that the neural networks already extract most of the available useful information from the feature set to determine the speaker identity.
Figure 4-5. The effect on speaker identification of augmenting the acoustic feature set with phonetic group information.
CHAPTER FIVE

ACOUSTIC FEATURE SELECTION
FOR SPEAKER IDENTIFICATION

5.1 Introduction

Speaker identification is a statistical pattern recognition problem that involves two basic issues. One is the statistical classifier to be used, the other is the basic feature set used to characterize the categories. In Chapters 2, 3, and 4, the focus was on issues relating to the classification method. In this chapter, the problem of extracting features from the speech signal for speaker identification is examined.

There are two main reasons for not using raw, digitized speech data for speech analysis. One is that the speech signal contains a great deal of redundancy. There exist many coding schemes which can encode speech without perceptible loss of quality, while realizing large savings of storage (or transmission) requirements (for example, Zahorian and Rothenberg, 1981). The other, and probably more important reason is that the original time domain representation of speech contains a lot of random variability which can be reduced by a frequency domain representation. Such effects as random phase alignment, variable pitch, and varying talker rate do not allow time domain template
matching. A memoryless feed-forward neural network architecture with sigmoidal non-linearities does have the ability to implement DFTs (or other FIR filter banks) but cannot compensate for the random phase component. (This is normally dealt with by computing the magnitude only and discarding the phase. This cannot be done with the "conventional" NN.) Therefore, the approach of using neural networks directly with raw data would require precise segmentation (time alignment) or adding to the network the ability to rectify and low pass filter (smooth) the detected frequency components. Neural networks could be modified to accomplish this, but at the expense of vastly increased size and training time.

Feature selection for speech recognition and for speaker recognition have diametrically opposing aims. In the first case, the features should retain the phonetic content and suppress or eliminate as much as possible variations due to other factors, such as the talkers identity. In the second case, the ideal feature set would maximize discriminability of the speaker while ignoring everything else. Thus presumably, the ideal features for the two tasks are not identical. However, since in both cases the human auditory system is the "natural" "front end" signal processing system, presumably features for both cases would be well represented in the auditory system. In this search for a feature set for speaker identification, the starting point was a feature set which works well for automatic speech recognition. An attempt was made to fine-tune this feature set for use with speaker identification. For example, variables such as optimum window length for short-time spectral analysis may be different for the two tasks.
5.1.1 Spectral Magnitude Features

Short-time spectral features are commonly used both for speech and speaker recognition. The spectral features are usually extracted from the acoustic speech signal by the use of FFTs, filter banks, or LPC (linear predictive coding) analysis. The short-time (10 to 100 ms) magnitude spectrum may be characterized by its peaks (formants) or its general shape (low order DCTCs). The earliest studies used pitch and energy contours for speaker recognition (Atal, 1972). Later studies showed that spectral shape features are more suitable (Atal, 1974; Furui, 1981). One of the latest studies (Matsui and Furui, 1991) reported error rates dropping from 1.5% to 1.0% when both types of features are used versus spectral shape only.

The basic feature set used throughout this study are a modified version of low order cepstral coefficients or Discrete Cosine Transform (DCT) Coefficients. The cepstrum is defined as the Fourier transform of the logarithm of the magnitude of the spectrum. The main modification to the cepstrum is the use of frequency warping. Frequency warping stretches the low frequency range and at the expense of the high frequency end. Such warping is based on known psychophysical properties of hearing. In particular, the Bark\(^1\) frequency scale is often used because it is directly modeled on the frequency resolution of the ear (Syrdal and Gopal, 1986). Another related warping function is bilinear frequency warping, which is more flexible with regard to the degree of warping. The bilinear warping is of the form

\[ B = 13\cdot\tan^{-1}(0.76f) + 3.5\cdot\tan^{-1}(f/7.5)^2, \text{ where } f \text{ is in kHz.} \]
\[ f' = f + \frac{1}{\pi} \tan^{-1} \left( \frac{\alpha \cdot \sin(2\pi f)}{1 - \alpha \cdot \cos(2\pi f)} \right), \]  

(5-1)

where \( f \) is the original normalized frequency, \( f' \) is the warped normalized frequency, and \( \alpha \) controls the degree of warping. Figure 5-1 shows the bilinear frequency warping for three values of \( \alpha \) (0, 0.25, and 0.5) along with Bark warping. Note that with \( \alpha=0.55 \), the bilinear warping is a very close match to Bark warping.

The frequency warping can be done two ways. One method (Nossair, 1989) is to first reconstruct the continuous spectrum from the discrete spectrum via interpolation. The continuous spectrum is then resampled at the discrete warped frequencies. The DCTCs are then computed with the use of cosine basis functions or an FFT. Since only the low order DCTCs are needed, the use of the FFT does not provide much computational savings.

An alternative method is presented here which combines the frequency warping into the cosine basis functions. The basic idea is that the cosine basis vectors are nonlinearly "stretched" and changed in amplitude, such that they vary more rapidly and are larger in amplitude at low frequencies relative to high frequencies. This modification of the basis vectors is based on whichever warping function is selected. A mathematical derivation of the basis vector modification is given in Appendix A. A sketch of modified basis vectors is given in Fig. 5-2. Note that in comparison to the method used by Nossair, this method for frequency warping does not require any interpolation and does not skip any frequency samples. Therefore this method should be a more precise implementation of warping than the previous method.
Figure 5.1. Bark and bilinear frequency warping with three values of alpha (α).
Figure 5-2. Examples of cosine basis functions modified to include bilinear frequency warping with $\alpha=0.5$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Note that some additional details of the DCTC computations were given in Section 1.4. Summarizing briefly, the sampled speech signal (16 kHz, 16 bit) is high-frequency pre-emphasized \((1-0.9z^{-1})\), Hamming windowed, (with a 32 ms window, except where noted differently), processed by a 512 point FFT, and converted to log magnitude. The DCTC's are then computed using the basis functions which include frequency warping as mentioned above. In computing the log magnitude, the dynamic range was limited using a floor of \(-50\) dB relative to the peak level of each frame. This limiting eliminates large negative "spikes" in the log magnitude spectrum due to zeroes. Since the zeroes of the spectrum are greatly effected by acoustic noise as well as computational round-off errors, and since the \(-50\) dB limitation of dynamic range was found to improve performance in speech recognition, (Zahorian and Jagharghi, 1991), the \(-50\) dB limit was also used for the speaker identification experiments.

In using DCTCs as features for speaker recognition, several basic issues can be examined. First, since there is a basic tradeoff between frequency resolution \((\Delta f)\) and time resolution \((\Delta t)\) dependent on the analysis frame length (or window length), performance is likely to be a function of the frame length. This tradeoff is quantified by the Heisenberg inequality:

\[
\Delta t \cdot \Delta f \geq \frac{1}{4\pi}
\]

(Riou and Vetterli, 1991). For speaker recognition, since most evidence from previous studies implies that static information is more important than transitional information,
presumably the "best" window length will be relatively long, corresponding to poor time resolution and good frequency resolution. Another issue is the number of DCTCs to be used, and the relative importance of the various DCTCs. Note that the number of DCTCs used is also related to the frequency resolution issue, since the degree of implied spectral smoothing is dependent on the number of DCTCs. Finally, the optimal degree of frequency warping, if warping is required at all, may be different for speaker identification than for speech recognition.

5.1.2 Transitional Spectral Features

The usefulness of transitional spectral information, i.e., features which encode the trajectories of static spectral features, to speaker recognition has been studied extensively. Earlier research, such as Furui (1981), concentrated mainly on the relative importance of transitional or dynamic information under various conditions (i.e. with various types and amounts of data). These investigations showed that while the bulk of information is carried by the static (instantaneous) power spectrum, the dynamics (changes, or time derivatives of the spectrum) also carry significant useful speaker identity information. More recent studies address the problem of how to best incorporate dynamic information.

There are two fundamentally different approaches for using dynamic information. The first is to simply augment the feature set with parameters which encode transitional information. Several studies employ this approach (Furui, 1981; Rosenberg and Soong, 1987; Mason and Zhang, 1991). Any classifier can be used with this approach, such as a memoryless neural network or a vector quantizer (VQ) codebook. The other
fundamental approach is to use static features only, but with a classifier capable of detecting temporal variations. Examples of this approach are time-delay neural networks (TDNNs) (Bennani and Gallinary, 1991), predictive neural networks (Hattori, 1992) and hidden Markov models (HMMs) (Poritz, 1982; Tishby, 1991).

5.1.3 Spectral Phase Features

The cepstral coefficients commonly used for speaker recognition are computed from the magnitude of the spectra only. The phase component is generally discarded. Acoustic features which retain the phase component, such as the complex spectrum are not commonly used for speaker (or speech) recognition for two reasons. One reason is that the spectral phase is somewhat difficult to compute, at least in unwrapped form. The other is that the phase is not generally believed to be useful. Many researchers (such as Hattory, 1992) have developed and used elaborate auditory models for speech processing, including speaker recognition. The human ear is considered to be insensitive to the relative phase of the frequency components in sounds; therefore the phase is discarded in speech processing based on auditory models.

Others researchers (such as Fussel, 1991) argue that for speaker recognition, machines can outperform humans, and thus de-emphasize the importance of auditory models. Another relevant point is that speech synthesizers, which do not preserve phase information, generate "speech" that is understandable but artificial sounding. This may be due to a lack of talker characteristics, suggesting the possibility that spectral phase may carry speaker identity information, even if it does not carry phonetic information. It

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
seems worthwhile, therefore to investigate the use of spectral phase information for
speaker recognition.

Tribolet (1977) presents an algorithm for computing the complex cepstrum, using
the unwrapped phase of the discrete Fourier transform, which retains both magnitude and
phase information. Alternatively, the Fourier transform may be replaced by the Hartley
transform, which inherently combines magnitude and phase spectral information of real
sequences, to compute a feature set (Bracewell, 1986). However, an experiment intended
to isolate the relative contribution of the phase information would be difficult to design
using features containing both magnitude and phase information. The performance
difference between magnitude and phase versus magnitude alone may include secondary
effects which may be difficult to eliminate. Therefore a procedure was developed to
compute phase features independently of the magnitude to determine their potential
contribution to a complete acoustic feature set.

The computation of the spectral phase is not as straightforward as that of the
spectral magnitude. The reason is that the "extended" arctangent function provides only
the principal component of the phase. The extended arctangent differs from the regular
arctangent in that its range is between plus and minus $\pi$ (180 degrees)—thus it preserves
the quadrant information. The actual phase, referred to as the unwrapped phase, is the
principle phase plus an integer multiple of $2\pi$ (360 degrees), such that the phase of the
continuous spectrum is continuous. Note that the definition of the unwrapped phase of
a discrete sequence is in terms of its continuous valued spectrum. The continuous
spectrum must be evaluated with sufficient density at each frequency range to reliably establish phase continuity.

The DFT of a sequence is its z-transform evaluated at discrete points along the unit circle. A zero of the z-transform which is very close to and inside the unit circle contributes a sharp positive 180 degree swing to the spectral phase in the frequency range closest to the zero. Similarly, a zero close to but slightly outside the unit circle contributes a sharp negative 180 degree swing. The sharpness of the phase swing (i.e. its derivative) at the frequency closest to the zero is given by:

\[ \frac{d\Theta}{d\omega} = \frac{\Delta\Theta}{\Delta\omega} = \pm \frac{1}{a}, \text{ at } \omega = \omega_0. \tag{5-3} \]

where \( a \) is the shortest distance to the unit circle at \( \omega_0 \), and the phase \( \Theta \) and the frequency \( \omega \) are both in radians. A zero which is exactly on the unit circle represents a frequency at which the phase is not defined. For a sequence with a zero on the unit circle the unwrapped phase is not defined since the continuity requirement cannot be met. Due to round-off errors, the positions of zeroes may move slightly. The only case in which this may make a significant difference is when a zero is very close to the unit circle. For sequences with zeros on, or very close to the unit circle the spectral phase cannot be unwrapped reliably. There are two measures of closeness to the unit circle of zeros which may be used to eliminate "problem" segments. One is based on the magnitude of the spectra, the other on the phase derivative. In the first case, speech segments are discarded if their spectral magnitude is lower than a certain threshold anywhere in the continuous
frequency range. In the second case, a speech segment is discarded if its phase derivative goes outside an acceptable range at any frequency.

A simple phase unwrapping procedure is as follows. First, the FFT of the sequence is computed. Since the time domain signal is real, the transform is symmetric, thus the upper half of the FFT is discarded. The principle phase component of the spectrum is then computed at each of the remaining points from the rectangular coordinates (real and imaginary) using the extended arctangent function. The next step is to determine the "continuous" or unwrapped phase at each point. The unwrapped phase is the sum of the principle phase and an integer multiple of 360 degrees, such that the difference between adjacent points is less than some arbitrarily small threshold. The phase of the first point (zero frequency) is zero. Thus the determination of the unwrapped phase begins with the second point and progresses upward on the frequency scale one point at a time. Thus the initial estimate of the unwrapped phase is the principle component plus the integer multiple of 360 degrees needed to determine the unwrapped phase of the previous point. If the transition is not greater than the threshold, then this estimate is accepted. Otherwise, decrementing and then incrementing by 360 degrees is attempted. If one of these results in an acceptably low transition, then it is accepted. Otherwise, the frequency range between the two points is successively halved (from here on referred to as frequency splitting) and the phase component of the DFT of the original sequence at each necessary frequency is computed until the phase transition between adjacent frequency values is not greater than the threshold.
5.1.4 Telephone vs. Laboratory Quality Speech

One of the ways in which speaker recognition studies vary is by the quality of the speech data used. Some use "telephone" quality (noisy, band-limited, nonlinear distortions, echoes) and others use "laboratory" quality speech. The ideal acoustic features for these cases are not likely to be the same. For example, features developed for laboratory quality speech may waste power to capture frequency components that are not present in telephone speech. As one simple test to verify the "robustness" of the features used in this study, one experiment, as described later, was conducted to examine performance degradation on speech bandlimited to approximate telephone bandwidth.

5.1.5 Outline of experiments

In the first set of experiments some basic issues in computing DCTCs are investigated. These are the degree of frequency warping to be used, the number of DCTCs to be used, the relative importance of some of the DCTCs, and the speech frame window length to be used.

The second set of experiments involve the use of transitional information. All other experiments were performed with static cepstral speech information. That is, decisions were made on individual frames independently of adjacent frames with a memoryless NN classifier. Two approaches to using dynamic information were investigated. One was to augment the feature set with parameters representing the dynamics of the DCTCs. The second approach was to add short term memory to the neural net classifiers to allow them to respond to temporal changes in the incoming
sequence of static parameters. Both time delay neural networks (TDNN) and recurrent neural networks (RNN) were investigated.

The third set of experiments used features which encode phase information. First, a method for reliable determination of the unwrapped spectral phase was developed. Then a set of experimental runs were made to attempt to train neural networks to distinguish speakers based solely on phase parameters, without any spectral magnitude information.

The last experiment was designed to gauge the degradation in performance of the existing speaker identification system when used with severely bandlimited speech data. The acoustic feature set was optimized for best performance with high quality, full bandwidth data. The same features, with no modifications, were then used with the bandlimited data. The experiment thus gives a measure of robustness of the features.

5.2 Optimizing the DCTCs

Three major issues in the use of DCTCs for speaker identification were addressed. These are the degree of frequency warping to be used, how many and which DCTCs to use, and the window length of speech frames. All experimental results were based on the first 20 male speakers from the Northern dialect region of the TIMIT data. All classifiers were trained on the two SA and three SI and tested with the five SX sentences. All results given in this section are for the test sentences only. In some cases results are given at the binary level, because of increased statistical reliability. Where top-level (20-way) results are given, the evaluation method used was the global soft search as
defined in Section 2.2.2. In these tests, networks with ten hidden nodes and two output nodes were used. Networks were trained for 10,000 iterations using a learning rate of 0.2 and a momentum term of 0.7.

5.2.1 Degree of Frequency Warping and Number of DCTCs

In the first experiment combinations of degrees of frequency warping and groups of DCTCs were tested. The results of these tests, in terms of binary-level recognition rates, are shown in Table 5–1. The frequency warping made surprisingly small difference. Alpha (α) values of 0.25 and 0.5 perform about equally well, and no warping (α=0) is only slightly inferior. The results shown in Table 5–1 also demonstrate that most of the information is in the first 15 DCTCs. Using the first 30 DCTCs does improve performance slightly. However, increasing the number of DCTC’s to 45 generally results in slight drop in performance. These tests also show that the first 15 DCTC’s convey substantially more information than higher order DCTC’s, such as DCTC’s 16–30. Note that even small changes in the binary error rates can result in relatively large changes in error rate at the top-level. Thus, for example, an increase in performance at the binary level from 99.59% to 99.92%, which is a factor of five decrease in error, results in an increase in recognition rate at the top level from 96.80% to 99.65% with global soft search or 97.71% to 99.60% with tree search based on hard decisions.
### Table 5-1

The relative performance of groups of DCTCs and the degree of frequency warping for speaker identification.

<table>
<thead>
<tr>
<th>Degree of frequency warping</th>
<th>DCTCs used</th>
<th>Binary level test data recognition rates (percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Single frame (32 ms)</td>
</tr>
<tr>
<td>(\alpha = 0)</td>
<td>1 - 15</td>
<td>82.94</td>
</tr>
<tr>
<td></td>
<td>1 - 30</td>
<td>82.96</td>
</tr>
<tr>
<td></td>
<td>1 - 45</td>
<td>82.55</td>
</tr>
<tr>
<td></td>
<td>16 - 30</td>
<td>64.45</td>
</tr>
<tr>
<td></td>
<td>31 - 45</td>
<td>60.52</td>
</tr>
<tr>
<td>(\alpha = 0.25)</td>
<td>1 - 15</td>
<td>83.04</td>
</tr>
<tr>
<td></td>
<td>1 - 30</td>
<td>83.57</td>
</tr>
<tr>
<td></td>
<td>1 - 45</td>
<td>83.11</td>
</tr>
<tr>
<td>(\alpha = 0.5)</td>
<td>1 - 15</td>
<td>82.90</td>
</tr>
<tr>
<td></td>
<td>1 - 30</td>
<td>84.10</td>
</tr>
<tr>
<td></td>
<td>1 - 45</td>
<td>85.20</td>
</tr>
<tr>
<td></td>
<td>16 - 30</td>
<td>70.60</td>
</tr>
<tr>
<td></td>
<td>31 - 45</td>
<td>70.02</td>
</tr>
</tbody>
</table>
5.2.2 The Relative Importance of each of the first 15 DCTCs

The results of the experiment described above show that the first 15 DCTCs as a group carry the bulk of the speaker identity information. Another experiment was designed to determine the relative importance of each of these individually. Fifteen separate, independent 20-way speaker identification systems were trained and tested. Each of the systems used 14 DCTCs, i.e., the "test" DCTC was deleted. The performance of each of the 15 systems was compared to that of the baseline system using all 15 DCTCs. The performance of each system was degraded, showing that all 15 DCTCs make individual contributions to performance as shown in Fig. 5–3.

5.2.3 The Window Length

The performance of four window lengths, 16, 32, 64, and 128 ms were compared and the results are shown in Fig. 5–4. In each case the frame spacing was 10 ms, and the FFT length was 2048 points. Zero padding was therefore required in all except the 128 ms case. Performance of the four window lengths is shown as a function of test speech length. With one second test speech lengths the 32 ms window length clearly performs best, and 128 ms window worst. With two seconds of test speech, the 32 and 64 ms perform best and about equally well. Generally, the trend is that shorter windows work best with short test speech lengths and longer ones with longer test speech. In general, performance changes only slightly as the window length is varied. However, the 32 ms window is overall somewhat better than other window lengths therefore appears to be the best choice.
Figure 5-3. The relative worth of the first 15 DCTCs (with $\alpha=0.5$) to speaker identification with three test speech lengths.
Figure 5-4. A performance comparison of four window lengths for speaker identification.
5.3 The use of Transitional Information

5.3.1 Enhancing the Feature Set with Transitional Information

In this experiment the contribution of transitional features computed from the static cepstral features was examined. The general approach to compute these features was to encode sections consisting of several adjacent frames of static features as the coefficients in a two term basis vector expansion. The transitional features thus represent the smoothed trajectory of each static feature (i.e., each DCTC), over a short time interval centered about each frame. The transitional features were computed for each frame and could be used either alone (dynamic information only) or to augment the original static features. These features, which are estimates of the first and second derivatives of the smoothed trajectory, are therefore also called velocity and acceleration features. The dynamic features were computed from the static features by the use of the weighting functions shown in Fig. 5-5. Three types of weighting functions were used: polynomial, sinusoidal, and Hamming weighted polynomial. The first-order windows span 150 ms (15 frames), and the second order 250 ms (25 frames), as recommended for speaker identification from no-noise and low-noise speech by Mason and Zhang (1991). To minimize artifacts at the ends of each sentence, the beginning and ending of each sentence was padded with up to 12 frames of average normalized DCTCs computed from the beginning and ending silence.

Recall as previously mentioned in Section 1.4, features to be used for the neural network training should be scaled to have the same mean and standard deviation for each
1st and 2nd order polynomial windows

1st and 2nd order sinusoidal windows

1st and 2nd order Hamming weighted polynomial windows

**Figure 5-5.** Weighting functions (windows) used for computing dynamic features from static features.
feature. Since the static features were scaled for zero mean and unity standard deviation, the dynamic features should also be scaled similarly. The average value of each of the six weighting functions was set to be zero to result in zero short term average of the dynamic parameters. Note that the global mean of the dynamic features would be zero in any case since the static features are zero mean. The magnitude of each window was adjusted so that the corresponding dynamic features would have a standard deviation close to one. To make all 15 dynamic DCTCs have exactly unity standard deviation would have required each window having 15 different magnitudes. However, somewhat surprisingly, in each of the six cases, the same window magnitude resulted in each of the 15 dynamic features with standard deviations no less than 0.7 and no greater than 1.5. Therefore, the windows were not individually scaled for each DCTC in the experiments reported below.

Several speaker identification experimental runs were made with various combinations of dynamic and static features. The results of some of these are shown in Table 5-2. Each run was made with 20 speakers, resulting in a system of 190 binary-pair neural net classifiers. Each of the test speech segments was evaluated with the 19 relevant classifiers. Since the number of parameters varied from 15 to 45, completely unbiased comparisons were difficult to make. If the number of hidden nodes were to be held constant, then the number of weights would vary nearly three to one, favoring the case with more parameters. Making the number of weights equal, however, would disadvantage the larger parameter sets, because some of the parameters carry far more information than others, and the number of connections to these would be dramatically
<table>
<thead>
<tr>
<th>Window Type</th>
<th>Features used</th>
<th>Test data binary-level error rates (percent) with speech lengths of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Static</td>
<td>1st order dynamic</td>
</tr>
<tr>
<td>polynomial</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>sinusoidal</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>sinusoidal</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>polynomial</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>sinusoidal</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Hamming windowed polynomial</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>N/A</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5-2. The relative performance of static and dynamic acoustic features for speaker identification.
reduced. Therefore, in this experiment, a compromise between the two extremes was used as follows: 15, 30, and 45 input (parameter) networks had six, five, and four hidden nodes, respectively. Each binary-pair neural network weight and offset was randomly initialized with values uniformly distributed between plus and minus 0.01. Each network was trained 10,000 iterations with a learning rate of 0.2 and momentum term of 0.7. Each time a speech frame is presented to the network and the weights and offsets are adjusted is considered one iteration. After each iteration the corresponding data pointer is advanced by 11, 13, or 17 frames, and the categories are alternated, as previously discussed in Section 1.3 as a method to prevent the network from temporarily learning phonetic discrimination rather than speaker discrimination.

The first two runs were made with first order dynamic parameters alone, computed with different window types, as shown in Table 5-2. As expected, some speaker discrimination is possible using only these features, although they perform significantly worse than the static features. With dynamic features alone, the polynomial window results in somewhat lower error rates than does the sinusoidal window. The combination of first order dynamic features and static features is slightly better than static features alone. Experiments were also conducted for each of the three window types with all 45 features. For these tests, the sinusoidal windows performed slightly better than the polynomials, although slightly worse than the Hamming windowed polynomials. This may simply be due to the windows being somewhat too long, since the windows performing better are larger magnitude close to the center and fall off in magnitude near the endpoints.
It is clear that some performance improvement is possible by the use of dynamic parameters. It is also evident, however, that with the TIMIT data base, even if the system were highly tuned, the improvement would not be dramatic. It is possible, however, that the dynamic features are less sensitive to noise due to the smoothing performed by the weighting functions, and therefore would cause more performance improvement when used with telephone quality data.

5.3.2 Extracting Dynamic Information by adding Short Term Neural Net Memory

In this experiment the performances of time-delay (TDNN) and recurrent (RNN) neural networks were compared to that of the memoryless feed-forward (MLFF) architecture used in all other speaker identification experiments. The potential advantage of the short term memory present in these architectures is that these networks may be able to learn and take advantage of the dynamics of the incoming sequence of instantaneous parameters. The architectures of the neural networks used in this experiment are shown in Fig. 5-6.

One of the difficulties that arise in training TDNNs and RNNs is the need to treat training data in units of consecutive frames from complete sentences, rather than independent short term frames, which was the case when training memoryless networks. One aspect of the algorithm used to train memoryless networks is that the samples presented to the network for training are alternated between the two categories. Another is that after each iteration the data pointer is incremented by a value well over one (typically 11 or more) to prevent a speaker being represented by the same phoneme for
Figure 5-6. Time-Delay (TDNN) and Recurrent (RNN) neural network architectures.
too many consecutive iterations. In order to accommodate both the restrictions imposed by the addition of short term memory and the desire to rapidly alternate phonetic categories in a training sequence, a modified version of data management was implemented as described in the following paragraph.

The neural net to be trained was allocated twelve separate "scratch-pad" sets of short term memories. These memories stored delayed values for twelve different positions in the training data. Thus, there was a high likelihood that each position corresponded to a different phonetic segment. Each temporary memory was considered to belong to one of twelve training "sessions." Training was cycled in a circular fashion among the twelve sessions. The training sessions were evenly divided between two speakers and six data pointers corresponding to each speaker were evenly spaced along the training data for that speaker. Note that the same single set of weights and offsets were updated by each training session. Each time a session computed a set of changes to the weights, the weights were immediately updated before cycling to the next session. After each weight update, the data pointers were incremented by one frame, thus insuring that data was presented as consecutive frames within each session. Preliminary experiments showed that weight updates after completing each session resulted in faster convergence than accumulating weight changes and making an update only after each of the sessions completed an iteration.

Another issue that must be addressed with TDNNs and RNNs is the initialization of the short term memory. The ideal initial state for a network used to classify acoustic data would be the final steady state it came to rest at after "listening" to silence (or
background noise) for a while. Since this silence steady state is a function of the network weights, it would have to be recomputed every time one of the sessions was about to begin training with a new sentence. To eliminate the need for these extra computations, a more straightforward initialization was used. Since the input parameters were normalized to zero average value, the delayed inputs (relevant to TDNN only) are initialized to zero. The delayed outputs of the hidden nodes are initialized to one half, since this is the neutral, or non-committed output of the sigmoidal type neuron.

Several experimental runs were made with each network type. The best result achieved by each of the three network types is given in Table 5–3. (In the table "NH", "NDI", and "NDH" mean number of hidden nodes, number of delays at the input node level, and number of delays at the hidden node level, respectively.) Since the results with the TDNN and RNN networks suggest the possibility of overtraining, Table 5–3 also shows the TDNN networks performance at three earlier points along its training.

This experiment was performed with 20 speakers, each represented by five training and five testing sentences. Each of the 100 testing sentences were used to evaluate 19 binary-pair classifiers, resulting in 1,900 trials of each system. Therefore, 114 failures of the RNN network represents a failure rate of 6.00%. The dynamic information improves performance in an average sense, since most sentences are recognized with less average error. However, for a few sentences the dynamic information proved to be misleading. Thus, for the conditions tested, the "bottom-line" performance, i.e., sentence error rate for test data, was always degraded relative to performance obtained with the memoryless network.
<table>
<thead>
<tr>
<th>Neural Network Description</th>
<th>Training Iterations (thousands)</th>
<th>Training data performance</th>
<th>Testing data performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Av. sentence error</td>
<td># of failures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average</td>
<td>St.Dev.</td>
</tr>
<tr>
<td>Memoryless NH=20</td>
<td>96</td>
<td>.1086</td>
<td>.0425</td>
</tr>
<tr>
<td>TDNN NH=5 NDI=3 NDH=5</td>
<td>96</td>
<td>.0552</td>
<td>.0442</td>
</tr>
<tr>
<td></td>
<td>9.6</td>
<td>.1566</td>
<td>.0848</td>
</tr>
<tr>
<td></td>
<td>19.2</td>
<td>.1257</td>
<td>.0733</td>
</tr>
<tr>
<td></td>
<td>28.8</td>
<td>.1121</td>
<td>.0676</td>
</tr>
<tr>
<td>RNN NH=6</td>
<td>96</td>
<td>.0047</td>
<td>.0076</td>
</tr>
</tbody>
</table>

Table 5-3. The relative performance of memoryless, time delay, and recurrent neural network architectures for speaker identification.
5.4 Spectral Phase for Speaker Identification

Two experiments were performed involving spectral phase. The first was designed to investigate and verify the method for computing the unwrapped phase for 32 ms (512 point) segments of speech. The second was a set of experimental runs attempting speaker identification using features derived from the unwrapped phase.

5.4.1 Unwrapping the Phase

This experiment was designed to verify the method for unwrapping the phase component of the FFT of short speech segments. The speech segments were obtained as follows. The digitized raw speech signal was high frequency pre-emphasized with a filter with the transfer function of \(1-0.9z^{-1}\). Speech segments of 32 ms (512 points) were extracted, Hamming windowed, zero padded various amounts, and converted to the frequency domain via an FFT. Zero padding and computing a larger FFT was used in order to oversample the spectrum. The first step in developing a phase unwrapping algorithm was to determine the degree to which oversampling is needed to establish phase continuity. Table 5–4, which shows the percentage of undetermined transitions for various conditions, implies that a long FFT length is needed for reliable phase unwrapping.

Figure 5–7 also illustrates the need for oversampling for phase unwrapping. In particular, oversampling by a factor of 16, ie. an FFT length of 8192 rather than 512, results in a much smoother phase component of the spectrum. Thus an 8192 point FFT length was used for the primary phase unwrapping experiments. Since the time domain
<table>
<thead>
<tr>
<th>FFT size</th>
<th>Number (and percentage) of undetermined transitions with thresholds of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>45 degrees (π/4)</td>
</tr>
<tr>
<td>512</td>
<td>479 (93.7)</td>
</tr>
<tr>
<td>1024</td>
<td>954 (93.3)</td>
</tr>
<tr>
<td>2048</td>
<td>1096 (53.5)</td>
</tr>
<tr>
<td>4096</td>
<td>101 (2.45)</td>
</tr>
<tr>
<td>8192</td>
<td>19 (0.23)</td>
</tr>
</tbody>
</table>

Table 5-4. The number of undetermined transitions encountered in unwrapping the spectral phase of 32ms (512 point) speech segments as a function of frequency oversampling (FFT size) for two uncertainty thresholds.
Figure 5-7. The principle phase component of the DFT of a typical 32 ms speech segment computed from 512 and 8192 point FFTs.
signal is real, the transform is symmetric, thus the upper 4095 points of the FFT are discarded. The principle phase component of the spectrum is then computed at each of the 4097 remaining points from the rectangular coordinates (real and imaginary) using the extended arctangent function which preserves the quadrant information. The next step is then to determine the "continuous" or unwrapped phase at each point. The acceptable transition threshold used was 45 degrees, since a transition of 45 degrees or less is far more likely than a transition of 315 degrees or greater. If the transition is not greater than 45 degrees, then this estimate is accepted. Otherwise, decrementing and then incrementing by 360 degrees is attempted. If one of these results in an acceptably low transition, then it is accepted. Otherwise, the frequency range between the two points is successively halved (from here on referred to as frequency splitting) and the phase component of the DFT of the original 512 points at each necessary frequency is computed until the phase transition between adjacent frequency values is not greater than 45 degrees.

Typically, the unwrapped phase can be determined without frequency splitting at all except ten to 20 of the original 4096 points. Occasionally, the frequency range must be split as many as ten times (requiring double precision DFTs) before acceptable phase continuity can be established. About one out of a thousand 32 ms speech segments, have a frequency point at which even this much frequency splitting is unable to establish phase continuity. This is due to the spectrum passing through what is effectively a zero. In this case the particular speech segment was discarded.
The criterion used for discarding segments with zeros on the unit circle was as follows. Initially the 8192 point FFT provides a frequency resolution of 1.95 Hz. This is split (halved) up to ten times, which corresponds to a maximum frequency resolution of 0.00191 Hz. If the phase changes more than 45 degrees (π/4) over this range, the segment is discarded. Using Eq.(5–3) above, this can be determined to correspond to a zero within a distance of 0.954·10^{-6} of the unit circle.²

5.4.2 Speaker Identification using DSTCs

This experiment was designed to determine the speaker identification potential of acoustic features computed from the phase component of the spectra. The features were computed as follows. The speech segments were extracted and their unwrapped spectral phase was computed as described in the previous section. The next step in computing parameters from the spectral phase was the removal of its tilt. Any shift in the time domain of the sample window causes a linear phase shift (tilt) in the frequency domain which is arbitrary and should be discarded. The tilt was removed by adding a value to each point which was directly proportional to its frequency and which resulted in the 4097th point (corresponding to 8,000 Hz) becoming zero. Since the phase of the transform of a real sequence has odd symmetry (with or without tilt removal) the phase cepstrum is characterized entirely by its discrete sine transform (DST).

² Δω = 0.00191 Hz · (2π/16,000 Hz) = 0.750·10^{-6} radians
ΔΘ = 45° · (2π/360°) = π/4 = 0.785 radians
therefore a=0.954·10^{-6}
Several experimental runs were conducted to determine the usefulness of DST coefficients computed from spectral phase to speaker identification. Speech from three speakers were used. All 256 DSTCs were computed for each 32 ms speech frames, and various groups of these were tested, as shown in Table 5-5. The best results are 57.32% with individual (32 ms) frames and 87.40% with one second test speech segments obtained with 50,000 training iterations. Note that the best results with 30 DCTCs reported earlier were 83.57% and 99.92%, respectively, obtained with only 10,000 iterations. Therefore, parameters derived from spectral phase (DSTCs) contain far less speaker identification information than spectral magnitude features (DCTCs). Because of this very low performance with phase features, no attempt was made to combine phase feature with magnitude features. In all subsequent experiments, phase information was not used.

5.5 The Performance of Band-limited Speech

The final experiment for this chapter was designed to determine the extent to which performance degrades when the acoustic features optimized for laboratory quality speech are used with severely band-limited speech. The frequency range of 300 to 3300 Hz, which corresponds approximately to telephone bandwidth, was used. Band limiting was accomplished by discarding (changing to zero) the values of the FFT of points one through ten and points 107 through 256. Note that this procedure results in a sharper cutoff than the relatively gentle rolloff of a good quality telephone connection. Note also that the remaining points of the FFT (points 11 through 106) correspond to a frequency
<table>
<thead>
<tr>
<th>DSTCs used</th>
<th>Network size and training</th>
<th>Binary level recognition rates (percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of hidden nodes (&amp; learning rate)</td>
<td>training iterations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-30</td>
<td>10 (332) 0.2</td>
<td>10,000</td>
</tr>
<tr>
<td>31-60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61-90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91-120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>121-150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>151-180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>181-210</td>
<td></td>
<td></td>
</tr>
<tr>
<td>211-240</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-256</td>
<td>4 (1038) 0.2</td>
<td>50,000</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 (2592)</td>
<td></td>
</tr>
<tr>
<td>1-30</td>
<td>10 (332) 0.05</td>
<td></td>
</tr>
</tbody>
</table>

Table 5-5. Summary of automatic speaker identification attempts using DSTCs computed from unwrapped and tilt normalized spectral phase of the speech signal.
range of 312.5 Hz to 3281 Hz. The first 30 DCTCs were computed for each speech frame of ten male speakers. The DCTCs were computed with bilinear frequency warping with a coefficient of 0.5. Two independent ten-way speaker identification systems were constructed, each consisting of ten hidden node BPP neural networks, each trained 10,000 iterations. One used all 30 DCTCs as input features, the other only the first 15 DCTCs. The systems were tested using the global soft search method, and with speech data band-limited the same as the training data. As the control experiment, the two ten-way speaker identification systems were reproduced with all variables unchanged, except with the band-limiting omitted (for both training and testing). The performance as a function of test speech length of each of the four systems is shown in Fig. 5-8. As expected, the bandlimited speech does perform somewhat poorer than the original, high quality speech, however it is still useful for speaker identification. Somewhat surprising, however, is that using 30 DCTCs rather than 15 improved performance more with the bandlimited speech than with the full frequency range speech. In general, one would expect that when less information is available, it can be encoded in fewer parameters.

5.6 Overall Conclusions

In this chapter, acoustic feature computation for speaker identification was examined. Three types of feature sets were investigated. The first type, frequency warped cepstral coefficients (DCTCs), were used all along in all previous experiments, and are the most common features used by other researchers. The specific issues regarding DCTCs that were experimentally examined are: the window length, the degree
Figure 5-8. A performance comparison of bandlimited (300-3,300 Hz) and non-bandlimited (0-8,000 Hz) speech with 15 and 30 DCTCs computed with $\alpha=0.5$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
of frequency warping, the number of DCTCs to use and the relative contribution of individual DCTCs. The best performing feature set is the first 30 DCTCs computed from 32 ms windows with a bilinear frequency warping coefficient of 0.25. The speaker identification performance is not very sensitive to changes in the way the DCTCs are computed. The 64 ms window performs almost as well, although significantly better than the 16 and 128 ms windows. The number of DCTCs used is also not very important; 15, 45, or even as few as ten perform almost as well as 30. It is very important, however, that the low order DCTCs be included in the feature set, since their contribution is far greater than the high ordered ones. Performance was also not sensitive to the degree of frequency warping.

The usefulness of dynamic information to speaker identification was also investigated using two very different methods. The first method was to enhance the static feature set (DCTCs) with additional features representing the frame to frame dynamics of the static features. The other method was to add short-term memory to the neural networks to allow them to internally compute dynamics of the static input features. Both TDNN and RNN network architectures were examined to a limited extent. Both methods of using dynamic information failed to improve speaker identification performance. This is somewhat surprising, since most researchers have reported at least a slight performance improvement with the addition of dynamic information to the static features. However, many studies are conducted with noisy, telephone quality data, the use of which is helped by the averaging performed by the dynamic feature computations.
The usefulness of spectral phase information for speaker identification was also examined. Acoustic features (DSTCs) computed from the unwrapped phase of the Fourier transform of 32 ms speech segments were used for speaker identification, without any spectral magnitude information. The basic conclusion of this experiment is that the spectral phase carries some speaker identification information, but much less than the spectral magnitude.
CHAPTER SIX

OVERVIEW AND CONCLUSIONS

6.1 Overview of Method

In this dissertation a novel approach to statistical pattern classification was presented and applied to the problem of automatic speaker identification. This method contrasts with most speaker identification approaches mainly in that with most methods a model is computed for each speaker independently of the other speakers. A potential weakness of this is that each model learns characteristics which may be common to all or most categories, rather than tuning in to the individual characteristics that actually distinguish the categories. In contrast, with the binary-pair partitioned (BPP) approach, one classifier is trained to capture only the differences that separate the individual pairs of categories. This method is inherently based on discriminative training. One drawback of this method, of course, is that the total number of classifier parameters is on the order of $N^2$, as opposed to order $N$. This, however, tends to represent the true complexity of the problem. For this reason, the BPP method scales up to large numbers of categories, while most others do not.
With the method studied, the N-way speaker identification task is partitioned into \( N \cdot (N-1)/2 \) binary-pair classifications. The binary-pair classifications are performed with "small" neural nets, each trained to make independent binary decisions on small fragments of speech data. Each of the small networks is trained to discriminate between two speakers, independently of all others. The approach is called binary-pair partitioning (BPP) to distinguish it from binary group partitioning. While the binary-pair approach suffers from the requirement for many subclassifiers, its advantage over group partitioning (binary or otherwise) is that the categories (speakers in this application) need not fall into distinct, easily identifiable groups. Furthermore, each BPP classifier trains rapidly because it is trained only with data from its corresponding two categories.

A final experiment was performed to verify that the basic methodology does indeed scale up to many speakers. The experiment is presented in this final chapter because its description is a review of the basic method that has been investigated in all previous chapters. In designing the system for this experiment, a few tradeoffs between performance and overall simplicity were made. Some of the techniques investigated resulted in varying degrees of performance improvement but at the cost of increased complexity and thus were not included in this experiment. The experiment was first performed with the 120 male speakers belonging to the Northern, New England, and North Midwest dialect regions of the TIMIT database. It was then repeated with all 130 female speakers of the database. The experiment was also intended to show performance as a function of the speaker population size. The conditions under which this experiment was performed was as follows. The acoustic feature set computed was 30 DCTCs, as
described in Section 1.4. Each BPP network consisted of 30 input, ten hidden, and two output nodes, and therefore a total of 332 weights and offsets. The frame level probability estimation was by neural network output alone without energy or phonetic group information. The neural network outputs were time integrated using simple addition. The global soft search evaluation method was used to combine BPP decisions into top-level (N-way) decisions. The performance of each system as a function of test speech length is shown in Fig. 6-1. For each curve at least 100 speakers were used. For example, the five-speaker curve is the average of 20 independent five-way systems, each with different speakers. Each system reaches 100% performance with six to eight seconds of test speech.

It may be worth noting that real-time implementation of a 100 speaker identification system on a personal computer is possible without difficulty. The network parameters would require about one to two mega-bytes of memory and the processing requirement is well within the capability of an Intel 386/387 or 486 based computer with a circuit board containing a TMS320C30 DSP chip and an analog-to-digital converter. The training required for the addition of an individual speaker would require only a few minutes of processing time.

6.2 Summary of Results and Conclusions

Many issues regarding the neural network based binary-pair partitioned speaker identification approach were investigated. Some of these are: the effect of the type and amount of training data; acoustic feature selection; neural network size, initialization, and
Figure 6-1. Speaker identification performance as a function of the number of speakers and test speech length.
training; methods of incorporating speech energy and phonetic content information to compute probability estimates at the individual speech frame level; methods of combining binary frame-level decisions into a binary segment-level decisions; methods of combining the binary segment-level decisions into a single N-way segment level decision.

In Chapter 2, the BPP approach was shown to have about the same speaker identification performance as the best obtainable with the use of a single large neural network. The main advantage of the BPP approach, however, is a vast savings of training time. The time required to train a single network to perform N-way classification is nearly proportional to the exponential of N. In contrast, the BPP approach requires training times on the order of $N^2$. With as few as 20 speakers, the savings is nearly two orders of magnitude. Another advantage of partitioning is that the required training time becomes much more predictable. If the BPP networks are trained serially (one at a time) then at any time during training we can determine the percentage of training completed as the percentage of the total number of required networks already trained. With the one large network approach, the remaining training time needed is difficult to predict from the current level of performance. Another advantage of the BPP system is that it is inherently more modular than one large network. Categories are easy to add or remove without the need to retrain already existing portions of the system. Several methods for combining BPP decisions into top-level (N-way) decisions were investigated. Each have advantages over the others, therefore the ideal choice depends on the constraints and specifications of a particular implementation.
In Chapter 3, several issues were investigated concerning the performance of neural networks used for elementary two-way decisions. Performance was fairly insensitive to several parameters, such as network size, network initialization, and even learning rate. With the large amount of training data available (approximately 1300 frames from each of two speakers), overtraining did not prove to be a significant problem.

In Chapter 4, it was found that phonetic content and short term energy of the speech signal both have a strong correlation to its usefulness for speaker identification. By incorporating broad phonetic group and energy information into the decision process, error rates were reduced by about one third. It was also found, however, that the neural network outputs are good indicators of the probability that a particular frame was from a particular speaker.

In Chapter 5 the acoustic feature set was examined. The best set of features found for speaker identification are 30 cepstral coefficients, computed over a frequency range of 0 to 8 kHz with bilinear frequency warping with a coefficient of 0.25, and a time window of 32 ms. The experiments also determined that the low order coefficients carry most of the speaker specific information (the first ten to fifteen coefficients), but that small improvements are obtained as additional cepstral coefficients are added up to about 30. The addition of dynamic information to the feature set did not improve performance. Features computed from spectral phase (DSTCs) carry some useful speaker identity information, but are not as useful as features derived from the spectral magnitude (DCTCs).
6.3 Contribution to Information Technology

The primary contribution of the research reported in this dissertation was the development and investigation of a novel technique for scaling up a classification problem to a very large size. The basic method extends beyond the speaker identification problem and beyond the use of neural networks. Prior to this work, the idea of partitioning a classification task in terms of elementary differential two-way classifiers has only been considered and investigated in a limited context, i.e., as a method of training a single large neural network. The idea of a flexible method for combining group partitioning and binary-pair partitioning to make full use of the advantages of each is unique. The advantages and limitations of the binary-pair partitioned approach were investigated both analytically and experimentally. The technique was thoroughly tested in the context of automatic speaker identification and the state of knowledge in this area was also advanced. The binary-pair partitioned classification technique is particularly well suited to the speaker identification problem, since the problems of interest involve many speakers.

The field of automatic speaker identification has matured to the point where machine recognition surpasses that possible by most humans. The algorithm presented potentially extends this capability even further. This technology can become a working reality in its own right (for such applications as security) or as a companion to the very difficult problems in automatic speech recognition.

The partitioning approach to pattern classification also has potential to aid other statistical pattern classification. The power of the method stems from exploiting the
fundamental notion that each pair of categories must be separable. By explicitly concentrating on the fundamental problem, in addition to solving the scaleability problem, another major benefit is derived. This method can use substantially more input features, which is very helpful with applications for which the optimal feature set has not been determined, or which inherently require a large feature set.

6.4 Future Research

Although the basic methodology has already been developed and extensive experimental results were obtained with the BPP approach for speaker identification, much additional work remains for investigation. Future research topics fall into three categories: (1) further improvements on current system while continuing to use the TIMIT database; (2) adapting system to real-world use, including the use of telephone quality speech; and (3) exploring applications beyond speaker identification.

Studies aimed at improving the current system might address the following: finding automatic data-driven group partitioning to reduce the number of BPP subclassifiers needed by the combined use of group and binary-pair partitioning (as discussed in Section 2.2.4); adding pitch information to the feature set; automatic feature selection using modified neural networks trained on raw data; developing features computed from the Hartley transform (Bracewell, 1986) to retain spectral phase; using Wavelets (Rioult and Vetterli, 1991) to get optimal time-frequency resolution at each frequency range; using multiple window lengths depending on how rapidly the spectrum changes, again for improved time-frequency resolution; and investigating neural network
training techniques which minimize the probability of error, rather than minimizing mean square error.

All results so far were obtained with laboratory quality speech, obtained in a single recording session for each speaker. A more demanding, and more potentially useful task, is to use telephone speech obtained over multiple sessions with varying channel characteristics. With this type of data, modifications in the input features may be required to reduce the effects of channel characteristics. Although the cepstral coefficients (DCTCs) work very well for speaker identification from "clean" speech, they may not be as well suited for telephone quality speech. The use of dynamic information has already been tested with two vastly different approaches with the TIMIT data base and no significant improvement in performance were found with either method. However, there may be a benefit in using dynamic information for the case of noisy and otherwise degraded telephone quality speech. Methods to compensate for systematically misleading training data should be investigated, such as uneven noise, echoes, and unequal frequency range.

A potential application of speaker identification worth investigating is the improvement of speaker-independent speech recognition by using a speaker normalization trained for a particular speaker or class of speakers. A direct extension of speaker identification is the use of the basic binary-pair partitioned technique for speaker verification. Other closely related applications of the BPP and/or data partitioning methods might be automatic gender, age group, mood, dialect, and language identification.
LIST OF REFERENCES


APPENDIX A

Frequency warping by the use of modified cosine transform basis functions

Cepstral coefficients are commonly used as acoustic features for speaker recognition as well as speech recognition. The cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the magnitude of the Fourier transform. For computational efficiency, and because the speech signal is often discretized to begin with, the Fourier transform is computed using one of the FFT algorithms. Since the logarithm of the magnitude of the transform is real, its inverse Fourier transform is fully captured by its cosine transform. Since generally only a few of the low order cepstral coefficients are desired, it is efficient to compute them as the dot product of the spectral magnitude and the corresponding cosine basis function. Since the original speech signal is real, its Fourier transform is even, therefore the dot product need only be computed over half the frequency range.

The procedure described above is an efficient way to compute spectral coefficients. It is often desirable, however, to frequency warp the spectrum (Fourier transform) before computing the discrete cosine transform coefficients (DCTCs) (Syrdal and Gopal, 1986). The traditional method (Nossair, 1989) has been to first reconstruct the continuous Fourier transform from the discrete transform (FFT) by using interpolation, and then resample at the warped frequencies. There are two disadvantages to this approach. First of all, the interpolation procedure introduces small errors, second, there are additional computations that have to be performed. The interpolation errors could be reduced by oversampling.
the frequency range by computing a larger, zero-padded FFT, at the expense of even more computations. Alternatively, the need for interpolation could be eliminated by computing the DFT at each of the desired frequencies directly, thereby giving up the computational efficiency of the FFT entirely.

An improved method of computing frequency-warped cepstral coefficients (DCTCs) is introduced. Its advantage is to eliminate the need for interpolation, while retaining the computational efficiency of the FFT. Given the magnitude of the Fourier transform of a signal as \( X(f) \). Let the frequency warped version of \( X(f) \) be \( Y(f') \) such that

\[
Y(f') = X(f), \quad (A-1)
\]

where \( f' = W(f) \) is the warped frequency and \( W(f) \) is the frequency warping function. We wish to compute

\[
DCTC(i) = \int Y(f') \cdot \cos[2\pi(i-1) \cdot f'] \cdot df' \quad (A-2)
\]

The problem with (A-2) is that \( Y(f') \) or a uniform sampling of it is not readily available to perform the numerical integration as a dot product. The traditional solution has been to interpolate between the available points and then resample uniformly. As an alternative, consider substituting (A-1) into (A-2) and using \( W(f) \) in place of \( f' \). The result is

\[
DCTC(i) = \int X(f) \cdot \{ \cos[2\pi(i-1) \cdot W(f)] \cdot [dW(f)/df] \} \cdot df \quad (A-3)
\]

The advantage of (A-3) is that the uniformly sampled \( X(f) \) is readily available from the FFT. The basis functions (within \( \{ \cdot \} \)) are computed only once and have the frequency warping already incorporated. For examples of cosine basis functions modified to include bilinear frequency warping see Fig. 5–2.
Statistical significance of experimental results

Student’s t-test (Nie et al., 1975; Cooper and McGillem, 1986) is a commonly used method to determine the statistical significance of experimental results used to compare two methods. Consider that two methods, X and Y, are each tested by N samples, resulting in sequences x(n) and y(n), respectively, where n=1,...,N. The computation of t is as follows:

\[
\begin{align*}
\text{AV}(X) &= \frac{\Sigma x(n)}{N}, \quad \text{AV}(Y) = \frac{\Sigma y(n)}{N}, \\
\text{AV}(X^2) &= \frac{\Sigma x^2(n)}{N}, \quad \text{AV}(Y^2) = \frac{\Sigma y^2(n)}{N}, \quad \text{AV}(X \cdot Y) = \frac{\Sigma x(n) \cdot y(n)}{N}, \\
S_X &= \text{AV}(X^2) - [\text{AV}(X)]^2, \quad S_Y = \text{AV}(Y^2) - [\text{AV}(Y)]^2, \\
\text{COV}(XY) &= \text{AV}(X \cdot Y) - \text{AV}(X) \cdot \text{AV}(Y), \\
\text{SD} &= \left\{ [S_X + S_Y - 2 \cdot \text{COV}(XY)]/N \right\}^{1/2}, \quad \text{and} \\
t &= \frac{[\text{AV}(X) - \text{AV}(Y)]}{\text{SD}}.
\end{align*}
\]

The value of t determines the degree of certainty that AV(X) would continue to be higher than AV(Y) if the experiment were to be repeated infinite times. Values of t of 1.28, 1.645, 2.33, and 3.29 correspond to confidence levels of 90%, 95%, 99%, and 99.95% respectively, if N is at least about 300.

Consider for example two of the results shown in Table 4-4. With method (1-a) the error rate was 1.19% and with method (2-b) it was 0.74%. Each was averaged from 10,127 recognition attempts. This information alone is not sufficient to compute t because it is not known how many samples that failed with one method passed with the other. However, an upper and lower bound on t can still be computed, by using maximal
and minimal correlation, respectively. Thus, consider \( x(n) = 0.0 \) for method X having correctly classified sample \( n \), and \( x(n) = 1.0 \) for \( n \) having been misclassified. Similarly assign 0.0 and 1.0 to \( y(n) \) for correct and incorrect classifications. Then \( N = 10,127 \), \( \text{AV}(X) = \text{AV}(X^2) = 0.0119 \), and \( \text{AV}(Y) = \text{AV}(Y^2) = 0.0074 \). \( \text{AV}(X-Y)_{\text{MIN}} = 0.0 \) and \( \text{AV}(X-Y)_{\text{MAX}} = 0.0074 \). The range of \( t \) then can readily be computed as \( t_{\text{MIN}} = 3.29 \) and \( t_{\text{MAX}} = 6.77 \). Thus, even the lower bound suggests a confidence level of 99.95%. Note that the arbitrary values of 0.0 and 1.0 assigned to the outcomes could have been any two real numbers without effecting the computed range of \( t \).