A PARTITIONED NEURAL NETWORK APPROACH FOR VOWEL CLASSIFICATION USING SMOOTHED TIME/FREQUENCY FEATURES

Stephen A. Zahorian and Zaki B. Nossair
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, Virginia 23529

ABSTRACT

A novel pattern classification technique and a new feature extraction method are described and tested for vowel classification. The pattern classification technique partitions an N-way classification task into N*(N-1)/2 two-way classification tasks. Each two-way classification task is performed using a neural network classifier that is trained to discriminate the two members of one pair of categories. Multiple two-way classification decisions are then combined to form an N-way decision. Some of the advantages of the new classification approach include the partitioning of the task allowing independent feature and classifier optimization for each pair of categories, lowered sensitivity of classification performance on network parameters, a reduction in the amount of training data required, and potential for superior performance relative to a single large network. The features described in this paper, closely related to the cepstral coefficients and delta cepstra commonly used in speech analysis, are developed using a unified mathematical framework which allows arbitrary nonlinear frequency, amplitude, and time scales to compactly represent the spectral/temporal characteristics of speech. This classification approach, combined with a feature-ranking algorithm which selected the 35 most discriminative spectral/temporal features for each vowel pair, resulted in 71.5 % accuracy for classification of 16 vowels extracted from the TIMIT database. These results, significantly higher than other published results for the same task, illustrate the potential for the methods presented in this paper.

EDICS: SA1.6.3, SA1.6.1
Correspondence author: Stephen A. Zahorian
telephone: (757)-683-3741 fax: (757)-683-3220 email: zahorian@ece.odu.edu

1. INTRODUCTION

Accurate automatic speech recognition (ASR) requires both features that are highly discriminative with respect to the categories of interest and a classifier which can form arbitrary boundaries in the feature space. Some classifiers are less sensitive to scaling, interdependence, and other "shortcomings" of the features. However, for a given feature set, there is an upper limit to classification accuracy, dependent on separability of the categories with respect to the features. Another important consideration for ASR is that statistical models using many parameters which must be estimated require an extremely large amount of training data. Since the number of model parameters generally increases with the number of features, this consideration often manifests itself in the so-called "curse of dimensionality" [2]. That is, for a fixed set of training data, classifier performance improves on the training data as additional features or dimensions are added, but degrades on test data. Therefore a compact set of highly discriminative features is preferred to a large set of features which contain the same information. Some classifiers are more powerful than others, not only in their ability to form more complex decision regions in a feature space, but also in terms of their ability to utilize higher-dimensionality feature spaces.

In this paper, we discuss a classification method which is based on the idea that a successful classifier must be able to discriminate every pair of categories. In particular, this classification approach uses a system of pair-wise classifiers, with one elementary classifier for each pair of categories which are to be distinguished. With this technique, both the classifier and features can be individually optimized for each pair-wise discrimination task, which in turn can lead to higher classification performance. We describe the structure of this classification method and experimentally demonstrate the benefits of this approach using vowel classification experiments.

To illustrate the interaction between features and classifiers, we also evaluate three feature sets for speech recognition, all based on a cosine transform of the short-time magnitude spectrum. The first two feature sets encode only static spectral information whereas the third set encodes trajectories of spectral features over several frames. These feature sets, although similar to cepstral features commonly used in speech analysis, are developed with a mathematical framework which allows arbitrary nonlinear resolutions in frequency and time. We show that the incorporation of

temporal information substantially improves classification performance. We also show experimentally that some classifiers are less affected by details of the feature calculations than are other classifiers.

In section 2, we briefly survey literature relevant to this paper. In section 3, we describe the classification methods used in this study and, in section 4, the feature sets used in this study. In section 5, we present the experimental results. In the last section, we give conclusions and discuss the potential applications and implications of this work.

2. BACKGROUND

Note that the techniques described and the experimental data provided within this paper directly address only the *classification* paradigm (i.e., labeled endpoints provided for each phonetic segment) rather than the more open-ended *recognition* problem for which the segmentation problems are solved automatically. However, since the ability to make fine phonetic distinctions is a very important component of a speech recognition system (for example, see [17]), and since a phonetic classification subsystem can be incorporated into a speech recognition system, knowledge gained from improving phonetic classification can be used to improve recognition systems.

Over the last two decades, hidden Markov model (HMM) speech recognition systems have generally provided the best results for continuous speech recognition. Reasons for the superiority of this technique include the ability to use large amounts of training data, the well-developed mathematical framework, the ability to automatically segment, the good ability to account for timing variations in speech, and convenient mechanisms for incorporating diverse sources of knowledge such as phonetic context and language models. HMMs are not, however, normally used for classification studies, such as those reported in this paper.

There are many papers in the literature which report phonetic classification and/or recognition results using the TIMIT database. Several of these also specifically address feature/classification issues for vowels. Published work on vowel classification (using 16 vowels from TIMIT) includes [9] (54% using 99 features extracted from an auditory model and a multi-layer perceptron (MLP) neural network with 32 hidden units; 67% correct using the same method with context also explicitly (i.e., non automatically encoded); [10] (about 63% using 99 features extracted from an auditory model and an MLP classifier with 256 hidden nodes); [6] and [7] (69% using 90 features computed with mel frequency cepstral analysis and gender specific models and a Gaussain full covariance matrix maximum likelihood estimator); [12] (66% using 120 features computed from an auditory model and a neural network classifier with 32

hidden units); [15] (71% using cepstral trajectory features and a neural network classifier). Table I gives a brief summary of these results obtained with automatic methods and results from two studies with human listeners.

Recently several investigators have used hybrid HMM/neural systems (for example [13], [19], and [24]) to take advantage of some of the good temporal modeling properties of HMMs and the discriminative properties of the neural subsystems. In a typical implementation of such a hybrid system (e.g., page 34 of [13]), the neural network is effectively a classifier, using labeled training segments, automatically determined from the previous step of the Viterbi alignment of the HMM. The process is repeated iteratively, thus requiring several cycles of neural network training. Robinson [19] uses a recurrent neural network as a frame level classifier (which can also be viewed as a posterior probability estimator). Some of the highest reported results for phonetic recognition use these hybrid systems [19].

See [18] for a tutorial on "state of the art" speech signal modeling techniques which are typically used for ASR; FFT derived cepstral coefficients and the first derivative of cepstra are the parameters most directly related to those proposed in this paper. The binary paired classifier approach used in this study, as well as some of the issues involved for combining the outputs of individual classifiers, have been reported in the statistical literature (for example, [8]). There is little reported use of this method for speech processing applications, with the exception of our past work on speaker identification ([20] and [21]), and one conference paper on vowel classification [16].

3. CLASSIFICATION METHODS

The primary classification method used in this study, which we call binary-pair partitioning (BPP), partitions an *N*-way classification task into $N * (N-1)/2$ two-way classification tasks. Each two-way classification task is performed using a neural network classifier which is trained to discriminate the two members of one pair of categories. With this approach, there are two distinct steps in the classification process: (1) "Elemental" classifiers are used to discriminate between every pair of categories (i.e., vowel pairs for the present study). (2) The pair-wise decisions from step one are combined to form the final "top-level" *N*-way decision. Overall, however, this method is input/output equivalent to a single large neural network, but with a very different internal configuration.

Although any type of elemental classifier can be used, in the experiments reported in this paper, all BPP classifiers consisted of memoryless feedforward fully interconnected networks, with one hidden layer, one output node, and unipolar sigmoid nonlinearities. These networks were trained with back-propagation to discriminate between each

pair of vowels, using training data selected only from those vowels.

Methods for combining the binary decisions from individual pair-wise classifiers to obtain final decisions can be derived using a probabilistic representation for the outputs of the pair-wise classifiers. In particular, the outputs of the pair-wise networks which are relevant to a certain category (G_i) can be considered as probability estimates for that category ([3], [4]), but conditioned on the two categories of data for which that network was trained. Thus, an estimate of the overall posterior probability of that category can be developed as follows. Using the notation and assumptions

$$P_i = P(G_i), \tag{1a}$$

$$P_{ij} = P(G_i / G_i \text{ or } G_j) \approx \text{output of network } ij, \tag{1b}$$

In Appendix A, we show that

$$P_i = \prod_{\substack{j=1 \\ j\neq i}}^{N} P_{ij} \Big/ \left(\sum_{\substack{k=1 \\ k\neq i}}^{N} \left(\prod_{\substack{j=1 \\ j\neq i \\ j\neq k}}^{N} P_{ij} \right) - (N-2) \prod_{\substack{j=1 \\ j\neq i}}^{N} P_{ij} \right) \tag{2}$$

From another point of view, assuming that each of the P_{ij}, for a fixed i represents conditional probability estimates for category G_i, and that each of the conditions are independent and equally likely, the overall posterior probability is proportional to the sum of the relevant network outputs, that is,

$$P_i \propto (1/N) \sum_{\substack{j=1 \\ j\neq i}}^{N} P_{ij} . \tag{3}$$

In either case, the classification rule is to choose whichever category has the highest P_i. Although, the independence assumption used for Equation 3 is clearly not true, in practice we found that classification results based on the two methods give nearly identical performance. Equation 3 was used for the experimental results reported in this paper.

To assess the relative merits of the BPP classifier, two control classifiers were also used. One of these was a "large" feedforward fully interconnected neural network (LNN) with one hidden layer and one output node for each category. This type of network, trained with back propagation, is typically used for pattern recognition with neural

networks. The other control classifier was a Gaussian, full-covariance matrix Bayesian maximum likelihood classifier (BML), which is theoretically optimum if the features are multi-variate Gaussian [2]. For this classifier, each category is "modeled" by its mean vector and covariance matrix.

In all experiments described in this paper, the network weights and offsets were initialized with random values uniformly distributed from -.5 to .5. For both neural classifiers, the networks were trained with backpropagation using a momentum term of .6. (See [11] for a tutorial on some neural network basics.) The learning rate for each BPP network was initially .45, and was reduced during training by a multiplicative factor of .96 for every 5000 network updates. For the large network, the rate began as .15 and was reduced during training by a multiplicative factor of .96 for every 16000 network updates. For all three classifiers, features were also scaled to have a mean of zero and standard deviation of .2 (i.e., a range of approximately +-1). These network learning conditions, determined from pilot experiments, are similar to values used in other studies reported in the literature.

Although we do not know of a fundamental theoretical advantage to a partitioned classifier, for some applications, the BPP method is superior to a single large classifier with respect to both performance and classifier architecture. In particular, for vowel identification based on neural networks, we show that the binary-pair classifier has the potential for higher identification rates than does a single large network. The explicit partitioning of the classifier is also advantageous for issues such as parallel computer implementations and ease of expanding (or reducing) the number of classification categories. A primary advantage for the BPP approach is that it allows for the individual optimization of features and classifier for each pair of categories, thus enabling improvement of overall performance. A potential disadvantage of the BPP approach is the requirement for a large number of elemental classifiers. For example, 16 categories require 120 binary-pair classifiers.

4. FEATURES

A. Introduction

To illustrate some of the interactions between classification performance and the features used, and also to ultimately determine a very good feature set in terms of classification accuracy, experiments were performed with three different feature sets and with a varying number of features from each set. In all cases, the speech signal (16 kHz sampling rate, 16 bit resolution) was first preemphasized using the second order equation

$$y[n] = x[n] - 0.95\, x[n-1] + 0.49 y[n-1] - 0.64\, y[n-2]. \quad (4)$$

Pilot experiments demonstrated that this preemphasis, which has a peak at approximately 3 kHz, and which is a reasonably good match to the inverse of an equal-loudness contour, results in slightly better performance than does a first order pre-emphasis, (*y[n] = x[n] - .95 x[n-1]*). The next step was to compute the magnitude spectrum of each frame of data with an FFT. Each frame was Kaiser "windowed," using a coefficient of 5.33 (i.e., approximately a Hamming window). The frame length was 20 ms for the first two feature sets described below. For the third feature set, frame lengths ranging from 2.5 ms to 60 ms were investigated. Each FFT magnitude spectrum was also "peak" smoothed using the morphological dilation operator over a range of +-75 Hz. This smoothing, which emphasizes peaks in the spectrum and eliminates very small values between pitch harmonics, was also found to provide a small improvement [15]. The next step in processing was to compute a cosine transform of the scaled magnitude spectrum, but with variations and additions as described below for the three feature sets.

B. Static Features (Set 1 and Set 2).

For these cases, a single frame of data, centered at the midpoint of each vowel token, was used. The features consisted of twenty-five coefficients in a basis vector representation of the magnitude spectrum over a selected frequency range. In effect, the features represented the smoothed spectral envelope. These coefficients were computed using the following approach, which allows flexibility in nonlinear frequency and amplitude scaling.

First, let *X(f)* be the magnitude spectrum represented with linear amplitude and frequency scales and let *X′(f′)* be the magnitude spectrum as represented with perceptual amplitude and frequency scales. Let the relations between linear frequency and perceptual frequency, and linear amplitude and perceptual amplitude, be given by:

$$f' = g(f), \qquad X' = a(X). \quad (5)$$

For convenience of notation in later equations, f and f' are also normalized, using an offset and scaling, to the range {0,1}.

The acoustic features for encoding the perceptual spectrum are computed using a cosine transform,

$$DCTC(i) = \int_o^1 X'(f') \cos(\pi i f') df', \quad (6)$$

where *DCTC(i)* is the *ith* feature as computed from a single spectral frame.

Making the substitutions

$$f' = g(f),\ X'(f') = a(X(f)),$$

and

$$df' = \frac{dg}{df}\, df, \tag{7}$$

the equation can be rewritten as

$$DCTC(i) = \int_o^1 a(X(f)) \cos[\pi\, ig(f)]\, \frac{dg}{df}\, df. \tag{8}$$

We therefore define modified basis vectors as

$$\phi_i(f) = \cos[\pi\, ig(f)]\, \frac{dg}{df} \tag{9}$$

and rewrite the equation as

$$DCTC(i) = \int_o^1 a(X(f))\, \phi_i(f)\, df. \tag{10}$$

Thus, using the modified basis vectors, all integrations are with respect to linear frequency. In practice, therefore, Eq. 10 can be implemented as a sum, directly using spectral magnitude values obtained from an FFT. Any differentiable warping function can be precisely implemented, with no need for the triangular filter bank typically used to implement warping.

Except for the frequency warping method and other spectral "preprocessing" refinements as mentioned above, the terms computed with Eq. 10 *(DCTC(i))* are equivalent to cepstral coefficients. However, to emphasize the underlying cosine basis vectors and the calculation differences relative to most cepstral coefficient computations, we call them the Discrete Cosine Transform Coefficients *(DCTCs)*, consistent with terminology in our previous related work ([14], [22]).

Feature set 1 was computed using Eq. 10 with a linear magnitude and frequency scale (i.e., a and *g* are identity functions)[1]. Feature set 2 was computed with Eq. 10 using a logarithmic amplitude scale (i.e., a is the log function) and bilinear warping with a coefficient $\alpha = .45$:

$$f' = f + \frac{1}{\pi} \tan^{-1}\left\{ \frac{\alpha \sin(2\pi f)}{1 - \alpha \cos(2\pi f)} \right\} \tag{11}$$

The first three basis vectors, incorporating the bilinear warping, are shown in Fig. 1. Pilot experiments were conducted to select the "optimum" value of the warping coefficient. For both features sets (and also for set 3 described below), a frequency range of 75 Hz to 6000 Hz was used. To help illustrate the processing described, Figure 2 shows an

original FFT spectrum, the spectrum after morphological filtering, and the spectrum recomputed from feature sets 1 and 2, using 15 *DCTCs* for each case. The curves in figure 2 were drawn with a log amplitude scale and linear frequency scale.

C. Trajectory Features (Set 3).

These features were computed so as to encode the trajectory of the smoothed short-time spectra, but typically with better temporal resolution in the central region than for the end regions. Using the processing as described for the second feature set, *P DCTCs* (typically 10 to 15) were computed for equally-spaced frames of data spanning a segment of each token. Each *DCTC* trajectory was then represented by the coefficients in a modified cosine expansion over the segment interval. The equations for this expansion, which are of the same form as for Equations 5-10 above, allow non-uniform time resolution as follows:

Let the relation between linear time and "perceptual time" (i.e., with resolution over a segment interval proportional to estimated perceptual importance) be given by:

$$t' = h(t) \qquad (12)$$

For convenience, t and t′ are again normalized to the range {0,1}. The spectral feature trajectories are encoded as a cosine transform over time using

$$DCSC(i,j) = \int_o^1 DCTC'(i,t')\cos(\pi\, jt')\, dt'. \qquad (13)$$

The *DCSC(i,j)* terms in this equation are thus the new features which represent both spectral and temporal information ("dynamic") over a speech segment. Making the substitutions

$$t' = h(t), DCTC'(i,t') = DCTC(i,t), \qquad (14)$$

and

$$dt' = \frac{dh}{dt}\, dt,$$

the equation can be rewritten as

$$DCSC(i,j) = \int_o^1 DCTC(i,t)\cos[\pi\, jh(t)]\, \frac{dh}{dt}\, dt. \qquad (15)$$

We again define modified basis vectors as

$$\theta_j(t) = \cos[\pi\, jh(t)]\, \frac{dh}{dt} \qquad (16)$$

and rewrite the equation as

$$DCSC(i, j) = \int_{o}^{I} DCTC(i,t)\, \theta_j(t)\, dt. \qquad (17)$$

Using these modified basis vectors, feature trajectories can be represented using the static feature values for each frame, but with varying resolution over a segment consisting of several frames. The terms computed in Eq. 17 are referred to as Discrete Cosine Series Coefficients *(DCSCs)* to emphasize the underlying cosine basis vectors and to differentiate between expansions over time *(DCSC)* versus *DCTC* expansions over frequency. In general, each *DCTC* (index i in Eq. 17) was represented by a multi-term *DCSC* (index *j* in Eq. 17) expansion.

In our work, the function *h(t)* was chosen such that its derivative, *dh/dt*, which determines the resolution for *t'*, was a Kaiser window. By varying the Kaiser beta parameter, the resolution could be changed from uniform over the entire interval *(beta = 0)*, to much higher resolution at the center of the interval than the endpoints (beta values of 5 to 15). Figure 3 depicts the first three *DCSC* basis vectors, using a coefficient of 5 for the Kaiser warping function. The motivation for these features was to compactly represent both spectral and temporal information useful for vowel classification, with considerable data reduction relative to the original features. For example, 12 *DCTCs* computed for each of 50 frames (600 total features) can be reduced to 48 features if 4 *DCSC* basis vectors are used for each expansion.

5. EXPERIMENTS

A. Introduction

Several experiments were conducted to investigate the BPP classifier for vowel classification and to examine tradeoffs between classifiers and features. In the first series of experiments, conducted with the first two feature sets mentioned above, we examined performance as a function of the number of input features, number of hidden nodes, amount of neural network training, etc. We also compared performance with the two control classifiers.

In the second series of experiments, conducted with feature trajectories, we used the BPP classifier to investigate and optimize a number of issues relative to spectral/temporal features described in section 4C. In particular, we describe experiments designed to determine the length of the speech segment, the degree of time warping for the selected segment, and time/frequency resolution issues including frame length and the number of *DCTCs* and *DCSCs* used. Finally, we

present experimental results based on a BPP classifier using features separately optimized for discrimination of each vowel pair.

A complete joint optimization of all parameters investigated, particularly for the spectral/temporal features (set 3), would have required a prohibitive number of experiments. Numerous pilot experiments were conducted to determine approximate values for these parameters. Each of the experiments was repeated several times with "adjustments" of fixed parameters based on previous tests, and with a modified range for those variables under investigation. We report the results of selected experiments, designed to show the main effects of parameter variations. In each case, some of the parameter values were held constant, using values determined from the previous experiments.

B. Database

The experiments were performed with vowels extracted from the TIMIT database. A total of 16 vowels were used, encompassing 13 monophthongal vowels /iy,ih,eh,ey,ae,aa,ah,ao,ow,uh, ux,er,uw/, and 3 diphthongs /ay,oy,aw/. We used four different data sets, SX1, SX2, SXI (SX + SI sentences), and SXIA (SX + SI + SA sentences), as noted in Table II. The SX1 data set was used primarily for parameter optimization. The SX2 data set was then used in order to check the results with different training and test speakers. The last two sets of data were used to make tests with different constraints on phonetic context. The speakers used in these tests were the same as those used in previously reported tests with the TIMIT vowels ([6], [7], [9], and [10]) so that our results could be directly compared with previously published data.

C. Experiments with Static Features

Experiment C1

Tests were conducted with one frame of spectral features (the first 15 DCTCs from feature set 2) to compare the BPP network approach with a single large network for a variety of network sizes, and for various amounts of training. The number of hidden nodes per network was varied from 5 to 250 using 11 steps. For the large neural network, a total of 4,000,000 network updates were made for each case, with performance checked on training and test data every 80,000 iterations. For the BPP network, the network was trained for 200,000 iterations with performance checked every 10,000 iterations. These experiments were performed with data set 1 from Table II.

Table III summarizes the percentage of vowels correctly classified for the two classifiers for both training

and test data as a function of the network size, for a fixed number of training cycles, for the two types of networks. The data in this table were obtained using 2,000,000 training cycles for the large network, and 160,000 training cycles for the BPP network. As illustrated by the curves in Fig. 4, test performance changed vary little after these numbers of training iterations (i.e., 80% of the total training iterations for the BPP, and 50% of the total training iterations for the LNN). The experimental data show that the BPP test performance changes little as the number of hidden nodes is varied from 5 to 75, with some decrease in performance for 100 or more hidden nodes (perhaps due to insufficient training for these cases). The large neural network has fairly consistent test performance as the number of nodes is varied from 15 to 250. For both networks, performance on the test data is typically within 3% of performance on the training data[2].

Figure 4 depicts the performance bounds (minimum and maximum test classification rates) for different network sizes for the BPP and large networks as a function of the number of training iterations. The difference in performance between the best network and the worst network is typically no more than 3% for the BPP network and no more than 4% for the large network, provided the networks are "well trained " (about 160,000 iterations for the BPP and 2,000,000 iterations for the large network). Note that, as verified in a later section, differences of greater than 1% are significant at the 90% confidence level, and differences of greater than 2% are significant at the 99% confidence level.

In general, the maximum performance of the LNN is higher than that of the BPP. The minimum performance is higher for the BPP. Thus, the performance bounds for the BPP case are more consistent than are the bounds for the large network, as also indicated by the smoothness of the BPP curves relative to the LNN curves. In general, the experimental data in this section illustrate that both types of networks result in similar performance for vowel classification based on 15 features from one frame of data. Both networks also perform well over a large range of network sizes and for varying amounts of training. For additional experiments reported in this section, a BPP with 5 hidden nodes, trained with 160,000 iterations, was used because this network configuration performed well. Similarly, for the large network, a network with 35 hidden nodes trained with 2,000,000 iterations was used. These values were chosen based on the desire for good performance and rapid training.

Note that the total number of "parameters" in the large network was 1085, whereas the total number of parameters for the system of BPPs was 9600. The training time for a single BPP network was approximately 1/170 of the total training time for the large network. The total training time for the BPP networks was approximately 70% of the

training time for the large network, since 120 BPP networks were required.

Experiment C2

Another experiment was conducted to compare the performance of the BPP to the performance of the large network for reduced amounts of training data. The first 15 *DCTCs* of feature set 2 were used. Networks were trained for 1%, 2%, 4%, 10%, 20%, 50%, and all of the training data of data set SX1. The entire test set of SX1 was used for evaluation for each case. Figure 5 shows the performance of both classifiers as a function of the amount of training data. The figure shows that the BPP gives higher performance than the large network for smaller amounts of training data. Although the differences between the test results for the two classifiers are not dramatic (typically 2 to 5 %), with all other factors equal, any statistical classifier which is less sensitive to reductions in training data is preferred.

Experiment C3

In order to more fully examine the relationship between classifiers and features, several tests were made with the three classifiers (BPP, LNN, and BML) for the first two feature sets described using data set SX1. For each feature set and each classifier, the number of features was varied from 1 to 25. Figure 6a depicts test results for *DCTCs* computed from linear/linear scaled spectra and Fig. 6b depicts test results obtained from log/bilinear scaled spectra. For the *DCTCs* computed from the log/bilinear spectra, all three classifiers achieve very similar levels of performance, with the two neural network classifiers slightly superior to the Gaussian classifier. Vowel classification of approximately 55% is achieved with both neural network classifiers provided at least 14 *DCTCs* are used. For the linear/linear features, overall performance is lower, as expected. However, there are dramatic differences between the performances of the Gaussian classifier and the two neural network classifiers. In particular, the performance of the Gaussian classifier is approximately one-half (about 27%) of that achieved with the log/bilinear feature set whereas the performance of the neural network classifiers degrades to approximately 50% (from 55%) .

Summary of Experiments with Static Features

The first two experiments of this section show that the BPP classifier is less sensitive to network size, amount of network training, and amount of training data, as compared to a single large network. Best performance of the two neural network classifiers is approximately the same. Experiment C3 illustrates that one benefit of both neural network classifiers over the Gaussian classifier is the relative insensitivity to details of the nonlinear scaling used in the

feature extraction process. The results for feature set 1 (slightly higher results for the BPP classifier than for the LNN) imply that the BPP classifier is somewhat less sensitive than the LNN to feature deficiencies. However, the best performance obtained with a single frame of static features, approximately 55%, is far less than that obtained for vowel classification in other studies, thus illustrating the need for an improved feature set which incorporates temporal information.

D. Experiments with Feature Trajectories

As noted above, the BPP classifier was found to perform at least as well as the LNN, but with certain advantages as noted. Therefore, only this classifier was used for the majority of the experiments with feature trajectories (set 3). Based on an experiment similar to that described in section 5C1, a BPP with 10 hidden nodes per vowel pair was trained with 100,000 iterations and used for all experiments with feature trajectories. For tests with an LNN, a network with 100 hidden nodes trained with 1,000,000 iterations was used. Our objective in these experiments was to investigate several details related to spectral/temporal trajectories with a single classifier. Results from several previous studies ([9], [12], and [22]) indicate that even for vowels, spectral trajectory information is required for good automatic classification performance. The experiments in this section were designed to examine details in the methods for extracting this trajectory information.

Experiment D1

This experiment was designed to optimize feature computations with respect to time/frequency resolution. That is, we wanted to determine the number of *DCTCs* (a measure of frequency resolution) and the number of *DCSCs* (a measure of time resolution) to use in representing each vowel. The combinations tested were 8, 10, 12, and 15 DCTCs and 3, 4, 5, and 6 *DCSCs* (16 combinations ranging from 24 to 90 features). In addition, the frame length, which also affects time and frequency resolution, was tested with lengths of 2.5, 5, 10, 20, 40, and 60 ms, for each of the above 16 combinations (96 total cases). The frame spacing was 1/4 of the frame length for each case (1.25 ms to 15 ms). All computations were based on a 300 ms segment centered at the midpoint of each vowel, using a time warping factor of 10 (i.e., beta value for the Kaiser window for the basis vectors over time (Eq. 16)). The results of this experiment were remarkably consistent over all conditions tested with the best result of 70.8% (obtained with 12 *DCTCs,* 4 *DCSCs*, and a 10 ms frame length), only 9% higher than the lowest result of 61.8% (obtained with 8 *DCTCs*, 3 *DCSCs*, and a 2.5 ms

frame). For the 10 ms window case, the results ranged from 67.2% to 70.8%, despite the wide variation in the number of features tested. These data indicate that a fairly smooth time-frequency representation is adequate for representing vowel spectral/temporal phonetic information. Since generally the results based on a 10 ms frame were the highest, for additional experiments a 10 ms frame length was used.

Experiment D2

The objective of this experiment was to examine the role of segment duration and the degree of time warping over the segment in computing feature trajectories. That is, we performed a more detailed examination of temporal information for vowel classification. Since these two effects would seem to be closely related, i. e., more time warping would be required for longer segments than for short segments, we varied the time duration from 50 ms up to 300 ms and also varied the time warping factor from 0 to 12 for each duration. For each case, we used 48 features consisting of 4 *DCS* coefficients for each of 12 *DCTCs*. Figure 7 shows the performance as a function of the degree of time warping for different segment lengths. As we can see from the figure, when the segment length increases, better performance is obtained with more time warping. The best test performance from these tests was achieved using a 300 ms segment and a warping factor of 10. This result is reasonable, since as the segment length increases, more contextual information is included.

Note that this method for representing contextual information is considerably different and more simplified than the typical method currently used in automatic speech recognition--context sensitive hidden Markov models (HMMs). That is, most high-performance HMM systems use several models to represent each phone, depending on left and right phonetic context. This greatly increases the total number of phonetic models required (typically in excess of 1000, rather than around 40, to represent all the phones in English), thus adding complexity to the overall speech recognition process. In contrast, our approach represents context using features which encode the adjacent segments with lowered resolution relative to the vowel midpoints.

Experiment D3

A fundamental difficulty of statistical pattern recognition is that there is virtually never enough data to train a classifier when a large number of features are used. Therefore, techniques such as discriminant analysis or feature selection (i.e., choosing only the "best" features, and eliminating the others) can be used for feature reduction. In

this experiment, we used feature selection with the BPP classifier to determine a relatively small set of "optimal" features for each vowel pair. We first computed a large number of features (as described below) and then used a feature selection algorithm, based on a maximum likelihood criterion, to select a subset of features which are highly discriminative for the classes under investigation [22]. For the BPP classifier structure, this feature selection can be performed independently for each pair of classes, thus potentially improving performance over that obtained using a single optimization process for all classes.

For this experiment, the following 101 features were first computed for each token. The first 45 features encoded 15 *DCTC* trajectories over 100 ms centered at each vowel midpoint (20 ms frame length, 5 ms frame spacing, 15 *DCTCs* times 3 *DCSCs*, with a warping factor of 2). The next 56 features encoded 8 *DCTC* trajectories over 300 ms centered at each vowel midpoint (10 ms frame length, 2.5 ms frame spacing, 8 *DCTCs* times 7 *DCSCs*, with a time warping of 4). Thus, these features were a collection based on the experiments described above. They include varying degrees of time-frequency resolution, based on the conjecture that different vowel pairs might be best discriminated with features varying with respect to these resolutions.

This collection of 101 features was then used for the feature selection program in each of two modes. In the first mode, the program selected all subsets of features, up to 40 total features, such that the 16 vowel classes were best classified (as judged by training data) with those features, using the same Gaussian, full-covariance matrix Bayesian maximum likelihood classifier (BML) mentioned previously. This classifier was used, since it requires much less "training " time than a neural network classifier, thus making it feasible to evaluate the large number of feature combinations required. Additionally, this method insured that the features were not tuned more to one of the neural network classifiers than the other. Using a modified "add-on" procedure ([14], [22]), the program determined the best single feature, the best set of two features, the best set of three features, and so on up to 40 features. In the second mode, the program also ranked feature sets for all sizes up to 40, but repeated this process separately for all pairs of vowels. That is, for the 16 vowels, the program selected 120 sets of features for each number of features up to 40. These feature subsets were then used as inputs to the BPP networks (and also LNN for some cases) to measure the classification performance as a function of the number of features.

Figure 8 depicts test vowel classification results as a function of the number of features for the two feature

selection methods for the SX1 data set. For a small number of selected features (up to about 15), the results obtained using pair-wise optimized features are significantly higher than those obtained using features optimized across all vowels. For more than 15 features, the pair-wise optimized features still give better performance than for the features optimized across all vowels, although by a smaller margin. Performance for both cases levels off at about 30 features. Additional comparisons were made with the other data sets using 35 features since the absolute best performance with the SX1 data was obtained with 35 features. Table IV shows training and test results for all 4 data sets used in this study for the case of 35 selected features, and also for all 101 features. Similar results are included for a large neural network.

The results for this last experiment demonstrate the primary advantage of the BPP approach. That is, if both the features and the classifier are optimized for every pair of categories, higher performance can be obtained than with a common optimization over all categories. Using 35 features optimized for every pair of vowels, classification test results of 73.6% were obtained for the SX1 data set, higher than for any other set of features we tested. The results are 1.6% higher than for results based on all 101 features (significant at the 90% level, as mentioned in section 5E). The results for the SX2 data set are also best for the case of the individually optimized features and the BPP classifier. For the other two feature sets, SXI and SXIA, the results are higher for both the BPP and LNN classifiers, using all 101 features rather than 35 selected features. However, since features were optimized on the SX1 data, they would not be expected to be optimum for the other data sets, since the phonetic contexts were systematically different. The results of this experiment thus support the idea that the feature selection approach can be used to greatly reduce the number of features needed for classification. With careful "tuning," feature selection can be used for a modest increase in classification accuracy (i.e., as demonstrated by the results in Table IV for the SX1 and SX2 data sets). The results for the SXI data set are within 1.5% of those obtained for the SX2 data (except for the LNN network and 101 features). In contrast the results for the SXIA data are typically about 3% higher than the SX2 results.

E. Statistical Significance Issues

Two experiments were performed to address the issue of the statistical significance for the results reported in this paper. In the first test, a correlated one-tailed t test was performed using the first 15 features of feature set 2, comparing the BPP and LNN classifiers, for the 49 test speakers of the SX1 data set. For this test, the t value of .197 indicated that the results for the two classifiers were not significantly different (mean of 53.71% classification for the BPP versus

53.56% classification for the LNN). The test showed that a difference of .96% in classification rates would have been required for significance at the 90% confidence level, and a difference of 1.75% would have been required for significance at the 99% confidence level. Additionally the variability in classification results across the 49 test speakers (standard deviations of 9.7 and 10.3 for the BPP and LNN respectively) implies a standard deviation of approximately 1.4% with respect to the means obtained from speaker groups of size 49.

As another examination of statistical significance, the best 35 pair wise features were compared to the best 35 common features using the BPP classifier (Experiment D3, Table IV-b and IV-c, first row), again using a correlated one-tailed t test. For this test, the t value of 1.72 was significant at the 95% confidence level. This test also showed that a difference in means of .96% would be required for significance at the 90% confidence level, and a difference of 1.73% for significance at the 99% confidence level. Finally, the results also implied a standard deviation of approximately 1.1% for classification results obtained from speaker groups of size 49.

Summarizing, these tests imply that differences of approximately 1% are significant at the 90% confidence level, and differences of about 2% are significant at the 99% confidence level. If classification were to be performed using different speaker populations, using speech materials with phonetic content similar to those used in the tests reported in this paper, the results obtained would be expected to be within about 3% of those reported here (assuming results would be within two standard deviations of the mean.)

6. Summary

A. Conclusions

A partitioned neural-network classifier structure, which we call binary-pair partitioning, and a new segment based feature set, have been described and evaluated for vowel classification. The classifier is developed using a separate classifier for every pair of discriminations which must be made. This classifier and these features have been used to obtain vowel classification results of 71.5% for 16 vowels excised from the TIMIT data base (data set SX2, experiment D3). As can be seen by inspection of Table I, which lists TIMIT vowel classification results obtained both with automatic methods and from listeners, the results obtained in this study are better than the best previously reported results for the same task.[3]

The results obtained in this study lead to the following specific conclusions with respect to features for vowel classification and a binary-paired partitioned neural network classifier:

1. Accurate vowel classification can be performed with features which represent the smoothed trajectory of vowel spectra over a short segment centered at the vowel midpoint. These features can be derived from FFT-computed spectra.

2. Temporal information is clearly important for vowel classification as evidenced by the large difference between the best results obtained from a single frame of data versus the best results obtained with features extracted from a segment (55% for 15 features extracted from a single frame versus 73.6% for 35 features extracted from a 300 ms interval for the SX1 data).

3. Primary coarticulation effects can be accounted for using a feature set which combines information from an interval approximately 300 ms in length, using the methods described in this study. The most comparable method and results to those obtained in this study appear to be those of [6] and [7] who also modeled spectral dynamics and obtained 68.9% vowel classification with gender specific phone models, as compared to 71.5% obtained in the present study using a common model for males and females.

4. The optimum features for discriminating vowels depend to some degree on the particular vowel pairs. Using 35 features optimized for each vowel pair, results of 71.5% were obtained versus results of 69.7% for 35 features optimized across all vowels.

5. The BPP classifier is relatively insensitive to linear and nonlinear transformations of the input feature space..

6. There is a small but consistent performance advantage with reduced data using the BPP versus the LNN. The results depicted in Figure 5 indicate that the BPP performance on test data is typically 3 to 5% higher than performance for the LNN if the training data is less than 1/3 of the full amount available from TIMIT.

7. The BPP classifier is more consistent in performance with respect to number of nodes and number of training iterations than is the LNN classifier. For the case of vowel classification, the accuracy is also generally higher by a small amount.

B. Potential Applications, Observations and Implications

Although there is no theoretical basis for expecting improved classification performance from a

partitioned classifier, the structure provides a convenient framework for selecting and optimizing features separately for each pair of categories. The ability to individually refine features for each category pair, coupled with the ability to effectively use a large number of features, and also the overall high-performance of the classifier, help to insure that classification rates based on a BPP classifier can be extremely high. These properties make the BPP classifier very attractive for other applications, including a recognition system for all phones.

The automatic classification results obtained in this study are significantly higher than the best reported human listener results for vowels extracted from TIMIT ((58.8%) in [16] and (65.9%) in [1]). We thus contend that at a phone recognition level, machine recognition can surpass that possible by humans. Other refinements, such as the use of a speaker normalization model or simply more training data, could potentially improve classification accuracy beyond the results presented here.

Both the BPP network structure and the trajectory features can be extended to other classes of phones and could be combined with HMMs to improve the phonetic recognition component of continuous recognition systems. The BPP structure, after all network outputs are combined, can be viewed as a single large network. Thus, it can be used where ever a single large network is used. The trajectory features presented in this paper could be used in a recognition system (which does not rely on labeled midpoints or endpoints) in either of two ways. First, in conjunction with the BPP neural network, they could be used to rescore sentence hypotheses generated by an HMM recognizer, using the segmentation obtained by the recognizer. This approach would require that the features first be computed for HMM training data, also automatically segmented by the HMM. Alternatively, the trajectory features could be computed for sliding "blocks" of frames, using a fixed block length and block spacing before any segmentation occurs. This is similar to the technique of augmenting cepstral features with delta cepstra. These "new" features could then used for any recognition system, including HMM and hybrid HMM/neural network systems. We have already used this approach for isolated word recognition ([23]) but have not yet tested the method for continuous speech recognition.

ACKNOWLEDGMENT

Portions of this work were supported by NSF grants IRI-9217436, and BES-9411607. Also thanks to Terry Nearey, University of Alberta, Edmonton, for helpful comments on an earlier draft of the manuscript.

FOOTNOTES

1. Note that the features in set 1 were not autocorrelation coefficients, since the cosine transform was computed using the magnitude spectrum rather than magnitude squared spectrum. However, as for autocorrelation features, there was no expectation that these features would work well for speech classification.

2. Note that in this section "test" performance is somewhat misleading, since performance on test data was checked several times. Thus, test results from data set SX1 were not used as final test results. Rather results were used from experiments such as this one to determine parameter settings for other experiments—using different data sets.

3. The 71% result in [15] was based on Data Set SX1, which really should be viewed as a "developmental" data base, as noted in footnote 2. The best test results obtained with Data Set SX1 in the present study were 73.6%.

REFERENCES

[1] R. A. Cole and Y. K. Muthusamy, "Perceptual studies on vowels excised from continuous speech," *Proc. ICSLP-92*, pp. 1091-1094, 1992.

[2] R. O. Duda and P. E. Hart, Pattern Analysis and Scene Classification, (*Wiley, New York, 1973*).

[3] A. El-Jaroudi and J. Makhoul , "A new error criterion for posterior probability estimation with neural nets," *Proc. IJCNN-90*, pp. III: 185-192, 1990.

[4] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," *Proc. ICASSP-90*, pp. 1361-1364, 1990.

[5] H. Gish and K. Ng , "A segmental speech model with applications to word spotting," *Proc. ICASSP-93,* pp. II-447-II-450, 1993.

[6] W. D. Goldenthal, "Statistical trajectory models for phonetic recognition," Ph. D. Thesis, Massachusetts Institute of Technology, Sept. 1994.

[7] W. D. Goldenthal and J. R. Glass, "Modeling spectral dynamics for vowel classification," *Proc. EUROSPEECH-93*, pp. 289-292, 1993.

[8] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," The Annals of Statistics, Vol 26, no. 2, pp. 1-18, 1998.

[9] H. Leung and V. Zue, "Some phonetic recognition experiments using artificial neural nets," *Proc. ICASSP-88*, pp. I: 422-425, 1988.

[10] H. Leung and V. Zue, "Phonetic classification using multi-layer perceptrons," *Proc. ICASSP-90*, pp. I: 525-528, 1990.

[11] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine,* pp. 4-22, April 1987.

[12] H. M. Meng and V. Zue, "Signal representations for phonetic classification," *Proc. ICASSP-91*, pp. 285-288, 1991.

[13] N. Morgan and H. Bourlard, "Continuous speech recognition," *IEEE signal process. magazine*, pp. 25-42, May 1995.

[14] Z. B. Nossair and S. A. Zahorian, "Dynamic spectral shape features as acoustic correlates for initial stop consonants," *J. Acoust. Soc. Am., vol.* 89, no. 6, pp. 2978- 2991, 1991.

[15] Z. B. Nossair, P.L. Silsbee, and S. A. Zahorian, "Signal modeling enhancements for automatic speech recognition," *Proc. ICASSP-95*, pp. 824-827, 1995.

[16] M. S. Phillips, "Speaker independent classification of vowels and diphthongs in continuous speech," *Proc. of the Eleventh International Congress of Phonetic Sciences*, vol. 5, pp. 240-243, 1987.

[17] J. Picone, "Continuous speech recognition using Hidden Markov Models," IEEE ASSP Magazine, vol 7, no. 3, pp. 26-41, July 1990.

[18] J. Picone, "Signal modeling techniques in speech recognition," IEEE Proceedings, vol 81, no. 9, pp. 1215-1247, 1993.

[19] A. J. Robinson , "An application of recurrent nets to phone probability estimation," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 298-305, 1994.

[20] L. Rudasi and S. A. Zahorian , "Text-independent talker identification with neural networks," *Proc. ICASSP-91*, pp. 389-392, 1991.

[21] L. Rudasi and S. A. Zahorian , "Text-independent speaker identification using binary-pair partitioned neural networks," *Proc. IJCNN-92*, pp. IV: 679-684, 1992.

[22] S. A. Zahorian and A. J. Jagharghi, "Spectral-shape features versus formants as acoustic correlates for vowels," *J. Acoust. Soc. Am, vol* 94, no. 4, pp. 1966-1982, 1993.

[23] S. A. Zahorian , D. Qian, and A. J. Jagharghi, "Acoustic-phonetic transformations for improved speaker-independent isolated word recognition," *Proc. ICASSP-91*, pp. 561-564, 1991.

[24] G. Zavaliagkos, Y. Zhao, R. Schwartz, and J. Makhoul, "A hybrid segmental neural net/hidden Markov model system for continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 1, pp. 151-160, 1994.

Researcher	Feature set	Method	Accuracy
Leung and Zue [10]	Auditory Model Outputs	Neural Networks	63%
Goldenthal and Glass [7]	MFCCs	Gaussian Multi-state/ Spectral Trajectories	68.9%
Meng and Zue [12]	Auditory Model Outputs	Neural Networks	66.1%
Gish and Ng [5]	MFCCs+Differences+ Durations	Segmental Speech Modeling	65.5%
M. S. Phillips [16]	Labeled Segments	Human Listeners	58.8%
Cole and Muthusamy [1]	Labeled Segments	Human Listeners	65.9%
Nossair et al [15]	Cepstral Plus Delta Ceptra	Neural Networks	61.3%
Nossair et al [15]	Cepstral Trajectories	Neural Networks	71.0%

Table I. Vowel classification rates from previous work.

Data set	Training Speakers	Test Speakers	Training Tokens	Test Tokens
SX1	450	49	18441	2076
SX2	499	50	20517	1877
SXI	499	50	34556	3240
SXIA	499	50	45041	4250

Table II. Data sets used for the experiments.

Number of Nodes	BPP Training	BPP Test	LNN Training	LNN Test
5	55.7	54.8	49.1	50.2
10	56.7	54.8	51.1	52.2
15	57.0	54.2	52.8	54.5
25	56.8	54.9	54.2	54.1
35	56.8	54.7	54.9	55.3
50	56.6	54.4	55.7	54.5
75	56.5	54.4	55.8	53.7
100	56.3	53.8	56.4	53.3
150	55.5	53.8	56.4	52.9
250	51.4	52.6	56.4	54.4

Table III. Percent of vowels correctly classified for the BPP and LNN classifiers as a function of the network size for a fixed amount of training.

Table IV-a. 101 features

	BPP		LNN	
Data Set	Training	Test	Training	Test
SX1	91.2	71.9	87.3	66.2
SX2	91.3	71.2	87.6	67.7
SXI	85.0	71.7	82.1	72.4
SXIA	84.8	75.5	82.8	75.3

Table IV-b. Best 35 common features

	BPP		LNN	
Data Set	Training	Test	Training	Test
SX1	79.7	72.3	78.2	68.6
SX2	79.4	69.7	78.0	69.7
SXI	75.3	70.9	75.1	70.9
SXIA	76.6	73.5	76.9	73.9

Table IV-c. Best 35 pairwise features

	BPP	
Data Set	Training	Test
SX1	82.0	73.6
SX2	81.1	71.5
SXI	77.0	70.6
SXIA	78.1	74.1

Table IV. Percent of vowels correctly classified for the BPP and LNN classifiers for various conditions.

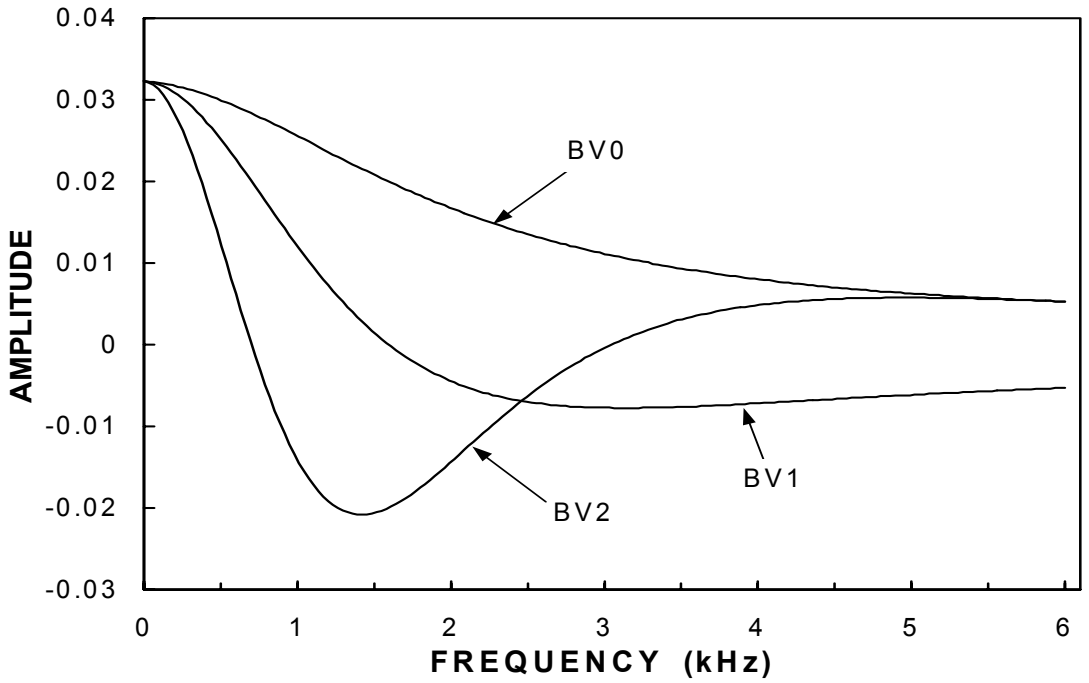

Figure 1. The first three DCTC basis vectors, with a warping factor of 0.45.

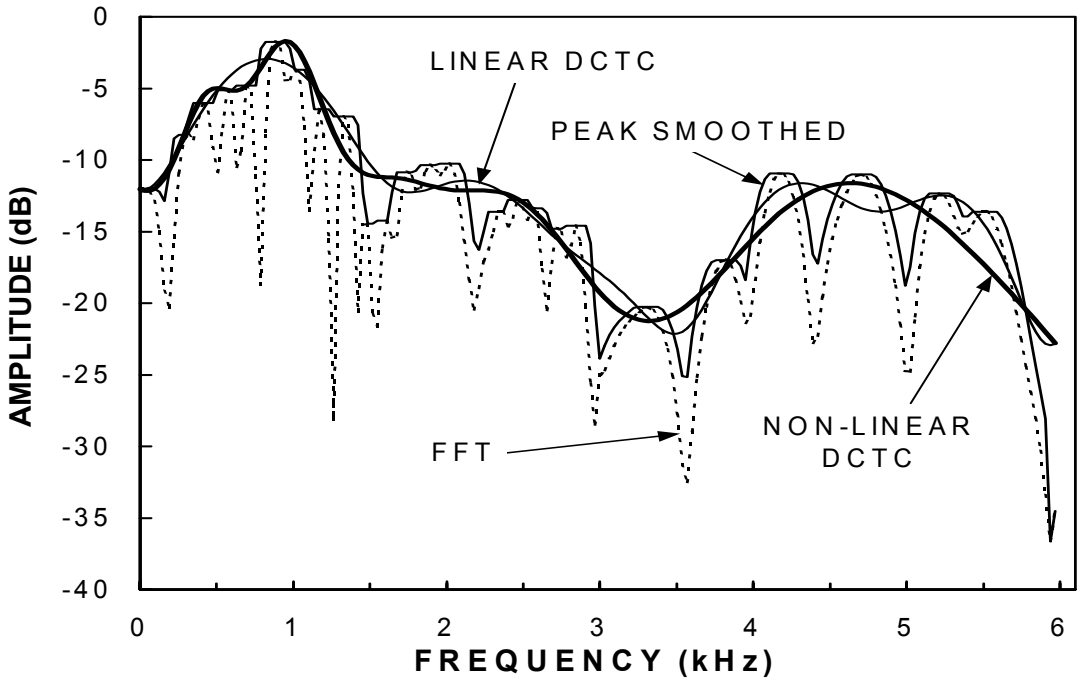

Figure 2. The original FFT spectrum, the spectrum after peak smoothing, the spectrum recomputed from feature set 1 (linear DCTCs), and the spectrum recomputed from feature set 2 (non-linear DCTCs).

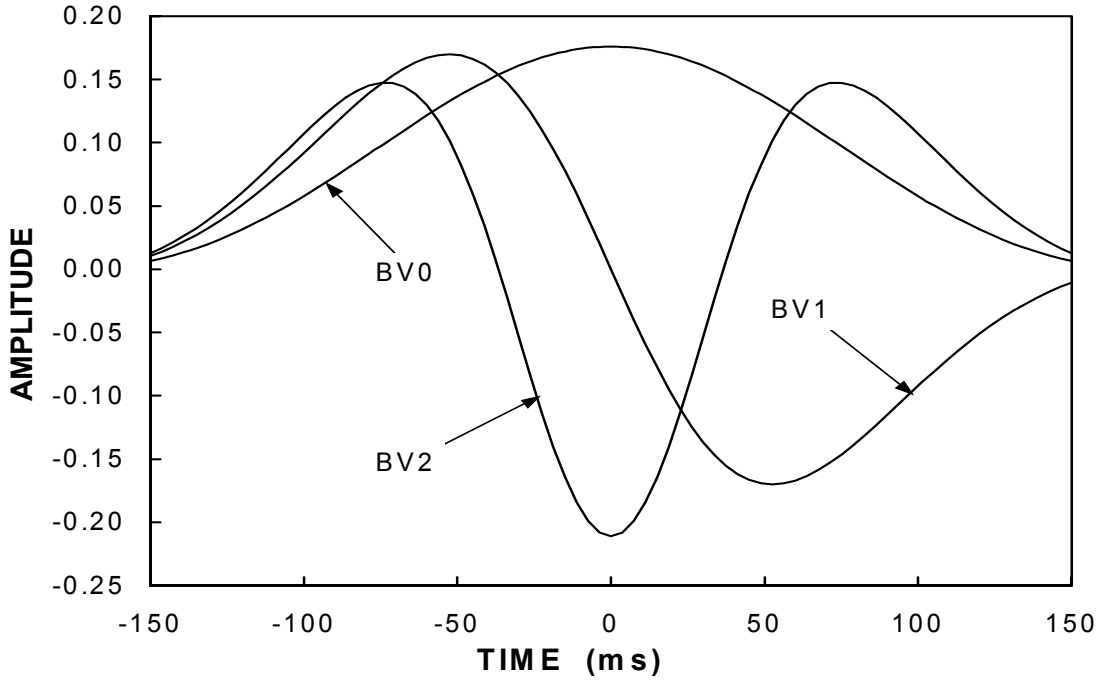

Figure 3. The first three DCSC basis vectors, with a coefficient of 5 for the Kaiser warping.

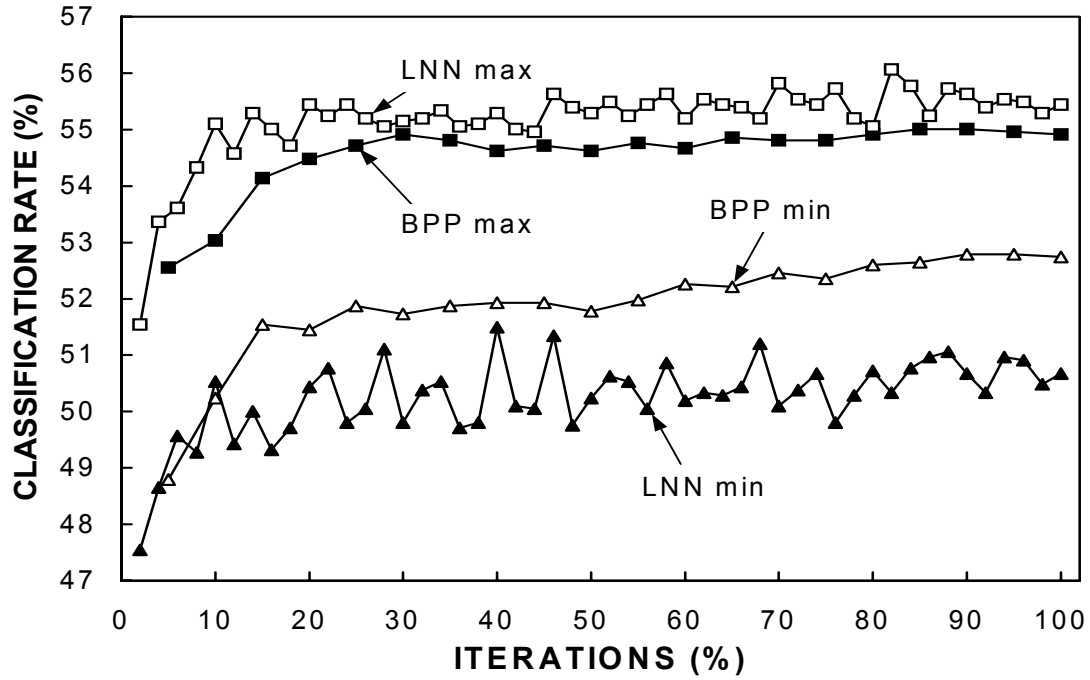

Figure 4. The performance bounds (minimum and maximum test classification rates) for different network sizes for the BPP and large networks as a function of the number of training iterations

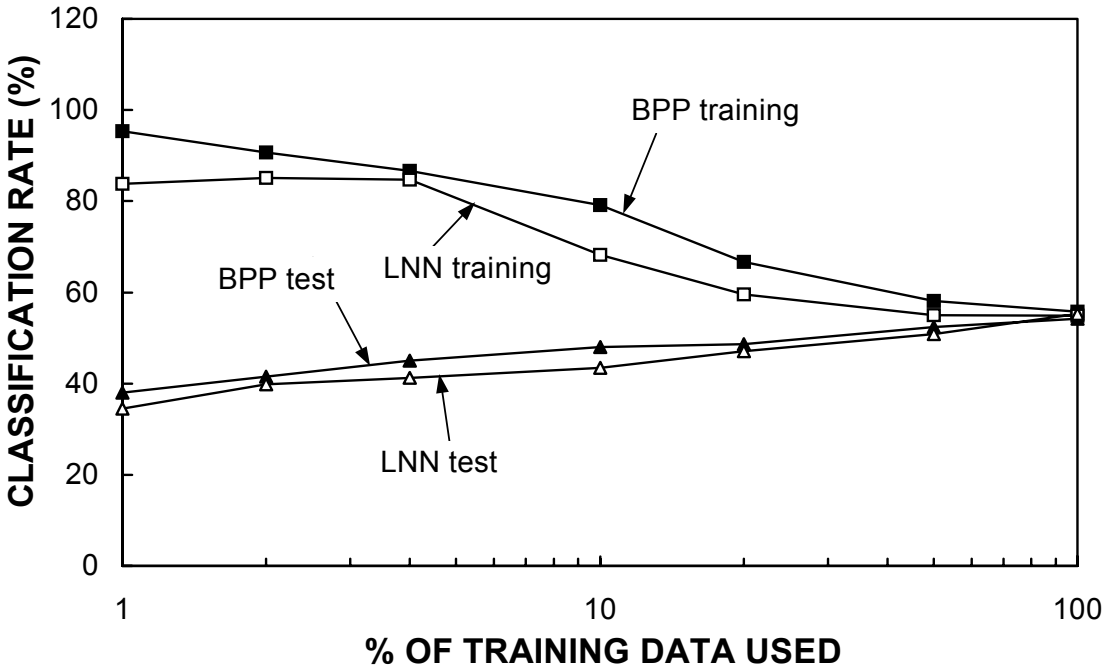

Figure 5. Percent of vowels correctly classified for the BPP and LNN as a function of the amount of training data.

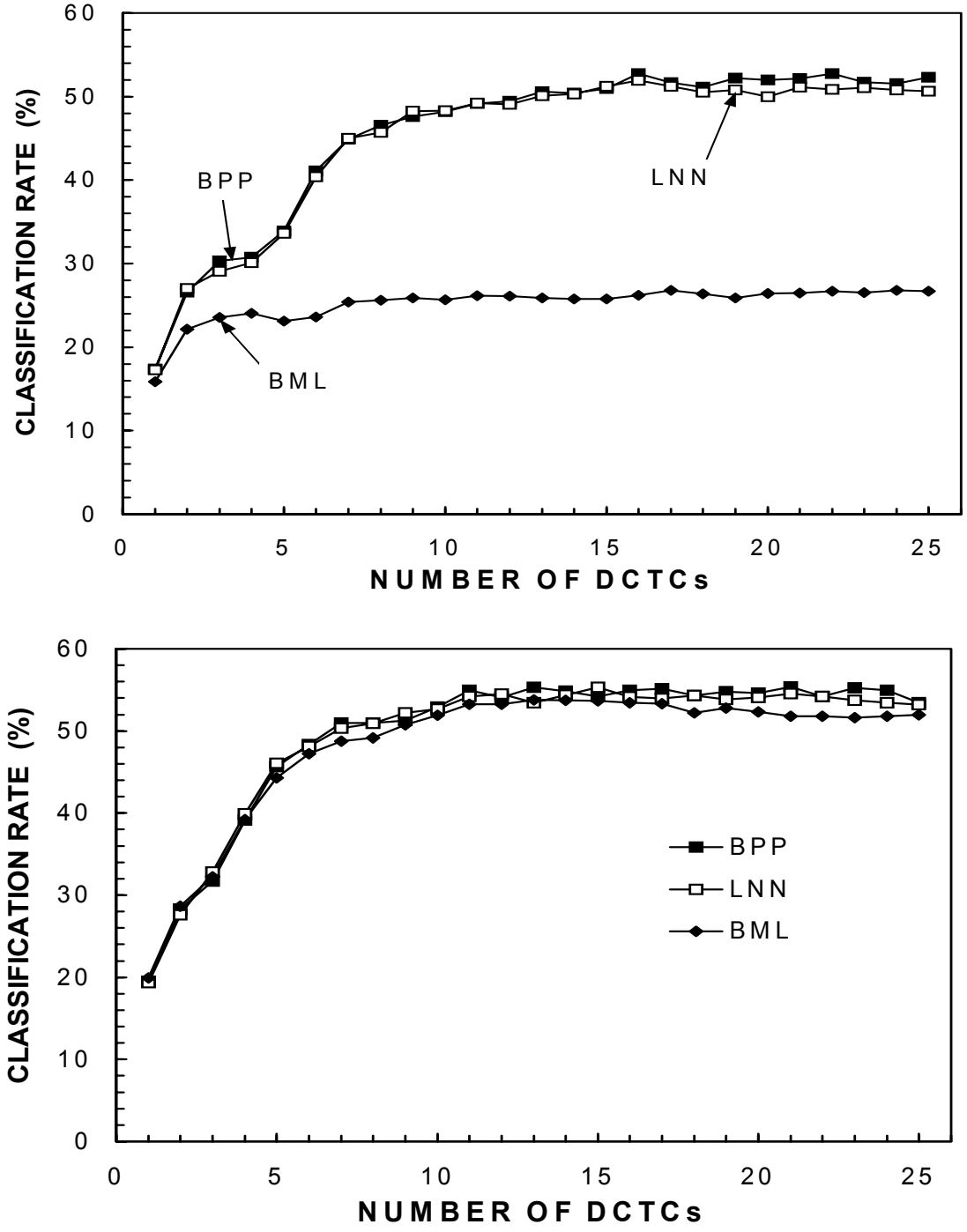

Figure 6. Vowel classification rates for three classifiers using varying number of DCTCs computed from linear amplitude/linear frequency spectra (panel a) and from log amplitude/bilinear frequency spectra (panel b)..

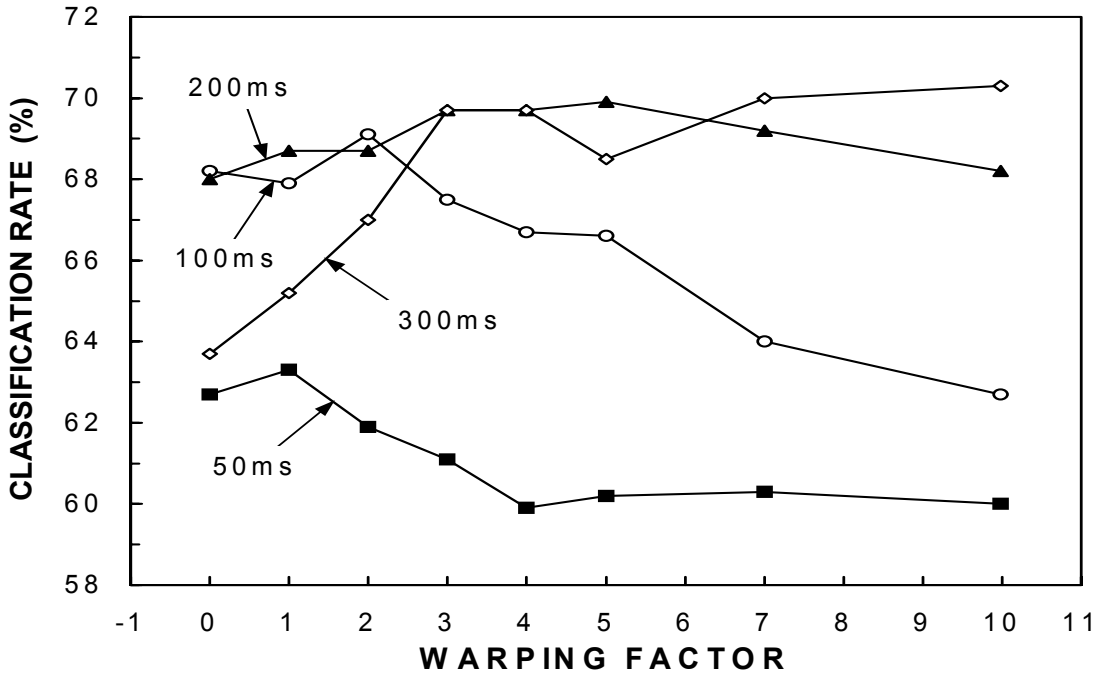

Figure 7. Vowel classification rates obtained with a BPP classifier and 48 trajectory features as a function of segment duration and degree of time warping.

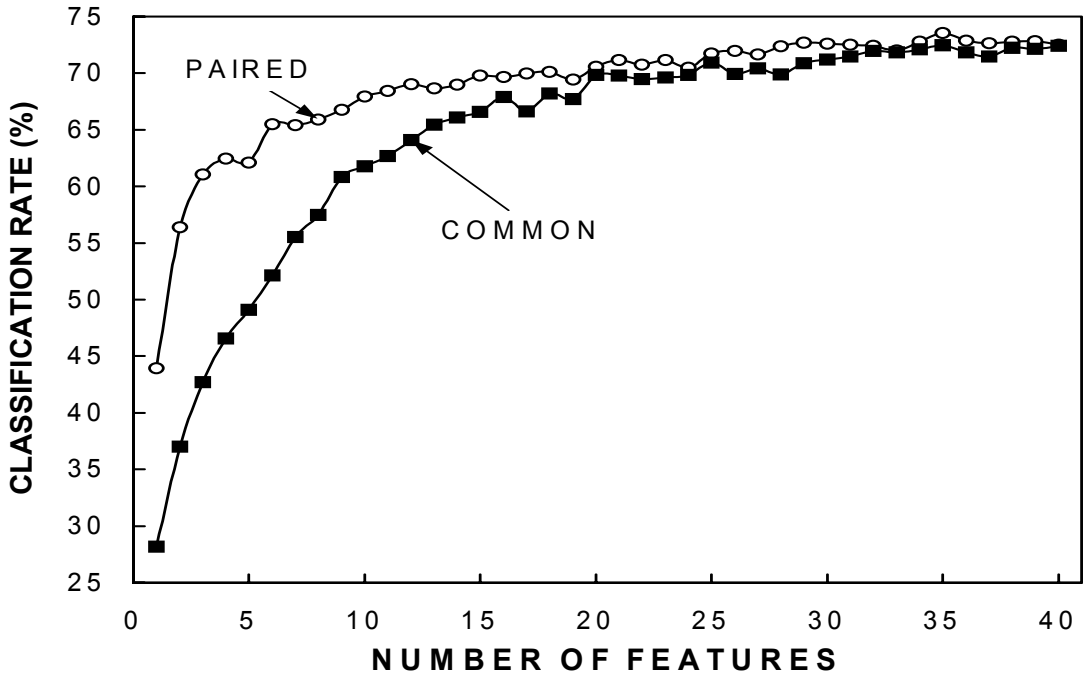

Figure 8. Vowel classification rates with the BPP classifier as a function of the number of features for two feature selection methods for the SX1 data set.

Appendix A

Derivation of posterior probabilities of individual classes from pairwise conditional probabilities.

With

$$p_i \equiv p(G_i), \quad 1 \le i \le N,$$

and

$$p_{ij} \equiv p[G_i / (G_i \text{ or } G_j)], \qquad 1 \le i \le N, \quad 1 \le j \le N, \quad i \ne j$$

and since $G_i \bigcap G_j = \phi$ for all $i \ne j$,
we also have

$$p_{ij} = \frac{p_i}{p_i + p_j} \text{ for } \quad 1 \le i \le N, \quad 1 \le j \le N, \quad i \ne j \qquad \textbf{(1)}$$

Additionally, it must be that

$$\sum_{i=1}^{N} p_i = 1. \qquad \textbf{(2)}$$

Using Eq. 1, for a particular i, i = m, and j = N, and also making use of Eq. 2, then

$$p_{mN} = \frac{p_m}{1 - \sum_{\substack{i=1 \\ i \ne m}}^{N-1} p_i}, \qquad \textbf{(3a)}$$

or

$$p_m = p_{mN} \left\{ 1 - \sum_{\substack{i=1 \\ i \ne m}}^{N-1} p_i \right\} \qquad \textbf{(3b)}$$

Each of the equations in 1 can be converted to the form

$$p_j = \frac{p_m(1-p_{mj})}{p_{mj}}$$

$$= \frac{p_m}{p_{mj}} - p_m, \quad \text{for} \qquad 1 \le j \le N\ ,\ j \ne m \tag{4}$$

Combining Eq. 3b and Eq. 4,

$$p_m = p_{mN}\left\{1 - \sum_{\substack{j=1 \\ j\neq m}}^{N-1} \frac{p_m}{p_{mj}} + (N-2)p_m\right\} \tag{5}$$

Expanding the sum in Eq. 5, and using a common denominator,

$$p_m = \frac{p_{mN}\left\{1 - p_m\left\{\sum_{\substack{j=1 \\ j\neq m}}^{N-1}\left(\prod_{\substack{k=1 \\ k\neq j \\ k\neq m}}^{N-1} p_m\right)\right\} + (N-2)p_m\right\}}{\prod_{\substack{j=1 \\ j\neq m}}^{N-1} p_m} \tag{6}$$

Simplifying, and combining terms, we have

$$p_m\left(\prod_{\substack{j=1 \\ j\neq m}}^{N-1} p_{mj}\right) = \prod_{\substack{j=1 \\ j\neq m}}^{N} p_{mj} - p_m\left\{\sum_{j=1}^{N-1}\prod_{\substack{k=1 \\ k\neq m \\ k\neq j}}^{N} p_{mk}\right\} + (N-2)p_m\prod_{\substack{j=1 \\ j\neq m}}^{N} p_{mj} \tag{7}$$

Solving for p_m

$$p_m = \frac{\prod_{\substack{j=1 \\ j\neq m}}^{N} p_{mj}}{\sum_{\substack{j=1 \\ j\neq m}}^{N}\left\{\prod_{\substack{k=1 \\ k\neq m \\ k\neq j}}^{N} p_{mk}\right\} - (N-2)N\prod_{\substack{j=1 \\ j\neq m}}^{N} p_{mj}} \tag{8}$$

This holds for $1 \le m \le N$.