Discriminative Analysis of Distortion Sequences in Speech Recognition

Pao-Chung Chang, Sin-Horng Chen, and Biing-Hwang Juang, *Fellow, IEEE*

***Abstract*—In a traditional speech recognition system, the distance score between a test token and a reference pattern is obtained by simply averaging the distortion sequence resulted from matching of the two patterns through a dynamic programming procedure. The final decision is made by choosing the one with the minimal average distance score. If we view the distortion sequence as a form of observed features, a decision rule based on a specific discriminant function designed for the distortion sequence obviously will perform better than that based on the simple average distortion. We, therefore, suggest in this paper a linear discriminant function of the form $\Delta = \sum_{i=1}^{T} w(i) * d(i)$ to compute the distance score Δ instead of a direct average $\Delta = 1/T \sum_{i=1}^{T} d(i)$. Several adaptive algorithms are proposed to learn the discriminant weighting function in this paper. These include one heuristic method, two methods based on the error propagation algorithm [1], [2], and one method based on the generalized probabilistic descent (GPD) algorithm [3]. We study these methods in a speaker-independent speech recognition task involving utterances of the highly confusible English E-set (b, c, d, e, g, p, t, v, z). The results show that the best performance is obtained by using the GPD method which achieved a 78.1% accuracy, compared to 67.6% with the traditional unweighted average method. Besides the experimental comparisons, an analytical discussion of various training algorithms is also provided.**

I. Introduction

In A traditional speech recognizer, an unknown input utterance is compared to the stored reference patterns according to a certain distortion/dissimilarity measure. To cope with the inherent variations in speaking rate and articulation timing, a dynamic programming procedure, usually imbedded in algorithms such as dynamic time warping (DTW) and hidden Markov models (HMM), is necessary to obtain the final dissimilarity score. The dynamic time alignment procedure produces a sequence of distortions of individual frames, $\{d(i)\}$. The index of the distortion sequence usually denotes the frame number associated with either the test token or the reference pattern. The final dissimilarity score is then obtained by averaging all distortions in the sequence. After calculating all dissimilarity scores, the reference pattern with minimum dissimilarity score is determined as the recognized word.

If we view the distortion sequence resulted from dynamic programming as a form of observed features, it becomes obvious that a decision rule based on the simple average distortion seldomly guarantees an optimal classification. Such a problem has been considered previously [4], [5]. In this paper, we explore the possibility of extensive discriminative analysis of the distortion sequence so as to improve the recognition performance of a traditional speech recognizer. Although discriminative training of reference models are currently being pursued, we in this paper concentrate on the analysis of discrimination for distortion sequences only, without changing the structure and parameters of the reference patterns. This may be attractive for existing systems since no major revision of the design would be required.

In our discrimination analysis, a linear discriminant function of the form $\Delta^j = \sum_{i=1}^{T} w^j(i) * d^j(i)$ is used. The superscript j denotes the fact that the comparison is done on the jth reference template. It essentially treats the distortion sequence as a feature vector. The weighting function $w^j(i)$ is trained using a large set of utterances. Unlike the traditional case of learning discriminant function in a fixed dimensional vector space [6], the feature vector, i.e., the distortion sequence, of our linear machine is of varying length and dynamic in nature. This is the problem that hampers the use of many neural networks related algorithms in speech recognition.

The method of using a weighted distance instead of a simple average distance has been considered by Rabiner and Wilpon [4]. The weighting function in their work was determined by a statistical analysis method. In this paper, several *discriminative* training algorithms are proposed. These include one heuristic method, two methods based on the error propagation algorithm, and one method based on the generalized probabilistic descent (GPD) algorithm [3]. Performances of these training algorithms are evaluated using a series of challenging experiments involving recognition of the highly confusible English E-set utterances. In our experiments, a conventional DTW algorithm is applied first for an input utterance to obtain the distortion sequence of best match with each reference template. The discriminant weighting functions are then trained in the training phase using distortion sequences of all training utterances. During recognition test, linear discriminant functions are calculated using the well-trained weighting functions after distortion sequences are obtained via dynamic programming. Finally the recognizer uses the linear discrimination functions rather than the average distortions for classification. The discrimination ability of various recognizers using the above training algorithms will be examined based on the recognition results.

The remaining of the paper are organized as follows: In Section II, the training methods for the discriminant weighting

Manuscript received January 25, 1991; revised June 3, 1992. The associate editor coordinating the review of this paper and approving it for publication was Dr. David Nahamoo.

P.-C. Chang is with the Telecommunication Laboratories, Ministry of Chung-Li Communications, Taiwan, R.O.C.

S.-H. Chen is with the National Chiao Tung University, Hsin-tsu, Taiwan, R.O.C.

B.-H. Juang is with AT&T Bell Laboratories, Murray Hill, NJ 07974.

IEEE Log Number 9208579.

1063–6676/93$03.00 © 1993 IEEE

function used in this paper are introduced. A series of experimental results are reported in Section III. Discussions and analyses of the experimental results are presented in Section IV. Conclusions are given in the last section.

II. Discriminative Training of Weighting Functions

In an N-word speech recognition system, let us assume that each word is represented by a single reference template. So, a total of N reference templates $\{R^j, j = 1, 2, \cdots, N\}$ are involved in this task. When a test token is given, it is first compared to each reference template by a dynamic programming (DP) procedure. After matching, a distortion sequence $D^j = \{d^j(i), i = 1, 2, \cdots, m^j\}$ associated with the best matched path is obtained. Here, m^j is the number of frames of R^j. Traditionally, the distance score for a reference template is calculated by simply averaging the individual frame distortions $d^j(i)$ in the corresponding distortion sequence, i.e.,

$$\Delta^j = \sum_{i=1}^{m^j} \frac{1}{m^j} d^j(i). \tag{1}$$

The final decision is made by choosing the one with minimal distance score. That is, the test token will be classified as the lth word if

$$\Delta^l = \min_{1 \le j \le N} \Delta^j. \tag{2}$$

We now take the distortion sequence as the observed feature and introduce a linear discriminant function

$$\Delta^j = \sum_{i=1}^{m^j} w^j(i) d^j(i) \tag{3}$$

as the new distance score for decision. The weighting function $W^j = \{w^j(i), i = 1, 2, \cdots, m^j\}$ is to be trained using the distortion sequences of all training utterances. In the sequel, we use D^j and W^j to denote sequences as well as column vectors which should pose no ambiguity.

According to the suggested discrimination scheme, each candidate contains two components. One is the reference template R^j and the other is the weighting function W^j. Though discriminative training of the reference template R^j is also important, we here focus our discussion on the training of the weighting function W^j. This means the reference template R^j is trained by a traditional method [7], before and independent of the weighting function training. In the following, several training algorithms for the weighting function will be discussed. These include a heuristic method, an error propagation algorithm, a single-layer perceptron with error propagation, and a generalized probabilistic descent method.

A. A Heuristic Method

Let $D_l^j = \{d_l^j(i), i = 1, 2, \cdots, m^j\}$ be the distortion sequence resulted from matching a training token of the lth word to the reference template of the jth word. We shall attempt to adaptively adjust the weighting functions $\{W^j, j = 1, 2, \cdots, N\}$ using the training distortion sequences. In doing so, the goal is to have

$$\begin{aligned} W_{t+1}^{l*} D_l^l &\le W_t^{l*} D_l^l \\ W_{t+1}^{j*} D_l^j &\ge W_t^{j*} D_l^j, \qquad \text{for all } j, j \ne l \end{aligned} \tag{4}$$

where the index t indicates the tth iteration, and $*$ is the operator of matrix transposition. Here, the adaptation is done sequentially, token by token if

$$\Delta_l^j \le \Delta_l^l + c \tag{5}$$

for any j, where

$$\begin{aligned} \Delta_l^j &\triangleq W_t^{j*} D_l^j \\ \Delta_l^l &\triangleq W_t^{l*} D_l^l \end{aligned}$$

and c is a threshold. One iteration corresponds to one adaptation.

If we constrain the weighting functions so as to satisfy the following conditions

$$\sum_{i=1}^{m^j} w^j(i) = 1, \qquad w^j(i) \ge 0 \tag{6}$$

for all i and j, then one simple adjusting method to accomplish (4) is

$$\begin{aligned} w_{t+1}^{l'}(i) &= w_t^l(i) + \eta d_l^{l'}(i) \\ w_{t+1}^l(i) &= w_{t+1}^{l'}(i) \Big/ \sum_{k=1}^{m^l} w_{t+1}^{l'}(k) \end{aligned} \tag{7}$$

where

$$d_l^{l'}(i) = \max_{i'} d_l^l(i') - d_l^l(i) \tag{8}$$

or

$$\begin{aligned} d_l^{l'}(i) &= 1, \qquad \text{if } d_l^l(i) = \min_{i'} d_l^l(i') \\ &= 0, \qquad \text{otherwise} \end{aligned} \tag{9}$$

and for all j, $j \ne l$,

$$\begin{aligned} w_{t+1}^{j'}(i) &= w_t^j(i) + \eta d_l^{j'}(i) \\ w_{t+1}^j(i) &= w_{t+1}^{j'}(i) \Big/ \sum_{k=1}^{m^j} w_{t+1}^{j'}(k) \end{aligned} \tag{10}$$

where

$$d_l^{j'}(i) = d_l^j(i) \tag{11}$$

or

$$\begin{aligned} d_l^{j'}(i) &= 1, \qquad \text{if } d_l^j(i) = \min_{i'} d_l^j(i') \\ &= 0, \qquad \text{otherwise} \end{aligned} \tag{12}$$

This is a heuristic training method. According to the conditions of (4), the weighting function is adjusted to reduce Δ_l^l and to increase all $\Delta_l^j, j \ne l$, when the training token belongs to the lth word. Obviously, if we decrease the components of W^l corresponding to the larger distortion components of

D_l^l and increase the components of W^l corresponding to the smaller distortion components of D_l^l, then Δ_l^l will become smaller due to the constraints of (6). We therefore, in (7), adjust each $w^l(i)$ by adding $\eta(\max_{i'} d_l^l(i') - d_l^l(i))$ or, alternatively, only add a small positive constant η to the component of W^l associated with the minimum of the distortion sequence. After normalization, the new W^l will satisfy the conditions of (4) and (6). For the weighting functions of other words, the adjusting momentum is contrast to that of the lth word. Here, the momentum means the adjustment direction and quantity of the weighting functions.

According to the above adjusting method, we iteratively adjust the weighting functions until some conditions are met. Note that the condition of (5) indicates that not all distortion sequences D_l^j but those that would cause confusion are used to adjust the weighting functions W^j. If the condition of (5) is not satisfied at all in one cycle, and thus none of weighting functions is adjusted, the iteration procedure stops naturally. Here, it is counted as one cycle when all training tokens are processed one time. Alternatively, the iteration procedure also stops if the iteration cycle exceeds one specified limit.

B. Error Propagation

According to (7) and (10), there are two adjusting momenta in the above heuristic training method. They are used to decrease or increase the corresponding distance scores such that the decision boundary will be correctively moved. As the training process progresses, the training tokens will be gradually separated into their own groups.

The error propagation algorithm can be used to simulate these two adjusting momenta. If we treat each W^j as the parameters of one single-layer perceptron, then a total of N single-layer perceptrons will be involved in our present task. When one training token of the lth word is provided as input, we can use the distortion sequence D_l^j to train the jth perceptron by using the error propagation algorithm.

Basically, the error propagation algorithm is an iterative gradient descent algorithm designed to minimize the mean square error between the desired target values (usually binary) and the actual output activation values of a multilayer perceptron [1], [2]. However, as shown in Fig. 1, our present network for each word is simply a single-layer perceptron with only one output unit. Basically, the learning procedure involves two steps:

1) *Forward step:* Given a set of weights, activation values caused by a training pattern are calculated. Computations are propagated forward through the network.
2) *Backward step:* Error signals between the output activation values and the target values for neurons in the network are calculated and then used in weight update. Computations are carried out by propagating the error signals backward from the output to the input neurons.

According to the network shown in Fig. 1, given the weights W^j and the training pattern D_l^j, the activation value a^j in the forward step is calculated as

$$b^j = \sum_{i=1}^{m^j} w^j(i) d_l^j(i) + \phi^j$$

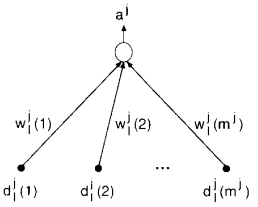

Fig. 1. The network structure of a single-layer perceptron with only one output unit.

$$a^j = f(b^j) = \frac{1}{1 + e^{-b^j}} \tag{13}$$

where ϕ^j is an offset and $f(\cdot)$ is an activation function. The selection of activation function plays an important role in the error propagation algorithm. In general, the sigmoid function as shown in (13) is used. In the backward step, the weights W^j and the offsets ϕ^j are adjusted according to

$$W_{t+1}^j = W_t^j + \eta\mu D_l^j$$
$$\phi_{t+1}^j = \phi_t^j + \eta\mu$$

where

$$\mu = a^j(1 - a^j)(o^j - a^j)$$
$$o^j = 0, \quad \text{for } j = l$$
$$= 1, \quad \text{otherwise.} \tag{14}$$

In (14), the adjustment of the weights W^j and the offsets ϕ^j mainly depend on the error term, $o^j - a^j$, where o^j is the desired target value. If $o^j = 0$, the parameters are adjusted to decrease the actual output value a^j. According to (13), it's equivalent to decreasing the distance score. It therefore will increase the distance score if $o^j = 1$.

In fact, the above method can be analyzed as a scaled single-layer perceptron with N output units in which each output unit connects to its own input units. All weighting functions will therefore be adjusted during one iteration.

C. Fully Connected Single-Layer Perceptron

A fully connected single-layer perceptron as shown in Fig. 2 is further considered. There are a total of N output units and each output unit is connected to the same $M(= \sum_{j=1}^{N} m^j)$ input units. When a training token of the lth word is presented as input, the input vector to the perceptron consists of N distortion sequences $\{D_l^j, j = 1, 2, \cdots, N\}$. It is different from the case in the previous section where the input vector connected to the jth output unit is the sequence D_l^j only. This network is also trained by the error propagation algorithm.

D. The Generalized Probabilistic Descent (GPD) Method

The GPD method was extended from the probabilistic descent (PD) method [8] which was developed as a generalized adaptive training scheme for classifying static patterns. The GPD method provides an enhanced capability in classifying dynamic patterns, of which the dimension of the feature vector

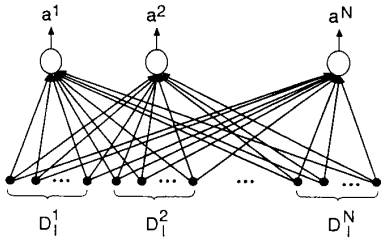

Fig. 2. The network structure of a fully connected single-layer perceptron.

may be varying. Due to the fact that the characteristics of speech is dynamic in nature, such a capability is very important for speech recognition. In fact, the dynamic properties of speech have hampered the use of many neural networks related algorithms in speech recognition.

Again, let us consider the adjustment of all the weighting functions $\overline{W} = \{W^j, j = 1, 2, \cdots, N\}$ when one training token, $X = \{x(i), i = 1, 2, \cdots, m^x\}$, of the lth word is presented. According to the GPD method, the weighting function adjustment is

$$\overline{W}_{t+1} = \overline{W}_t + \delta\overline{W}(X, \overline{R}, \overline{W}, l) \tag{15}$$

where $\overline{R} = \{R^j, j = 1, 2, \cdots, N\}$. That is, the weighting functions are adapted by a small amount $\delta\overline{W}(X, \overline{R}, \overline{W}, l)$ every time a single training input token is presented. The correction parameter $\delta\overline{W}(X, \overline{R}, \overline{W}, l)$ is a function of $X, \overline{R}, \overline{W}$, and l. If we set

$$\delta\overline{W}(X, \overline{R}, \overline{W}, l) = -\varepsilon U \nabla_{\overline{W}} \ell_l(X, \overline{R}, \overline{W}) \tag{16}$$

it can be shown that the adaptive training of (15) converges at least to a local optimum solution [3]. Here ε is a small positive constant, U is a positive-definite matrix, and $\nabla_{\overline{W}}\ell_l(X, \overline{R}, \overline{W})$ is the gradient of a prescribed loss measure $\ell_l(X, \overline{R}, \overline{W})$. Note that the GPD method allows adaptive training of the reference templates $\overline{R}$ and the discriminative weighting functions $\overline{W}$ simultaneously if we view $\overline{R}$ and $\overline{W}$ as part of the parameter set of the overall classifier. In our present study, we limit our attention to the weighting functions $\overline{W}$ only. To discuss the term $\nabla_{\overline{W}}\ell_l(X, \overline{R}, \overline{W})$, we should first define the discriminant measure, the misclassification measure, and the loss measure.

A discriminant measure $g_j(X, \overline{R}, \overline{W})$ which is defined as a function of $X, \overline{R}, \overline{W}$, and j, is used to indicate how likely the input X belongs to the jth word. Here, the discriminant measure should be differentiable with respect to the weighting functions. In our case, it can be defined as

$$g_j(X, \overline{R}, \overline{W}) = \ln\left\{\sum_{i=1}^{T_j}\sum_{\theta}\exp(-D_\theta(X, R_i^j, W^j)\zeta)\right\}^{-1/\zeta} \tag{17}$$

where $T_j(= 1$, in the current example) is the number of reference template of the jth word, θ is an index warping path between X and R_i^j, and $D_\theta(X, R_i^j, W^j)$ is the distance score between X and R_i^j evaluated along the path θ. As $\zeta \to \infty, g_j(X, \overline{R}, \overline{W})$ can be approximated by the minimal distance of the DP matching derived from the optimal path, i.e.,

$$g_j(X, \overline{R}, \overline{W}) \cong \min_\theta D_\theta(X, R_i^j, W^j) \tag{18}$$

where

$$D_\theta(X, R^j, W^j) = \sum_{i=1}^{m^j} w^j(i)(x(i_\theta) - r^j(i))^2 = \sum_{i=1}^{m^j} w^j(i)d^j(i) \tag{19}$$

$R^j = \{r^j(i), i = 1, 2, \cdots, m^j\}$ and i_θ denotes the warped index function.

According to the above definition of the discriminant measure, the weighting function, in a general case, should be included in the DTW algorithm for the search of the optimal path. However, in our current simulation, we simply choose the optimal path based on the conventional DTW algorithm without involving the weighting function. After obtaining the optimal path, the discriminant measure is calculated using the distortion sequence and the pre-stored weighting function. This leads to no major revision for a conventional DP-based speech recognition system.

A misclassification measure $\delta_l(X, \overline{R}, \overline{W})$ is used to indicate the amount that an input token of the lth word deviates from the decision boundary. Here, larger $\delta_l(X, \overline{R}, \overline{W})$ implies that more likely the input is misclassified. Moreover, $\delta_l(X, \overline{R}, \overline{W})$ must be differentiable with respect to the weighting functions (or the complete set of classifier parameters if we also attempt to adjust the template parameters). Based on the above discussion, the misclassification measure used in our present study is defined as

$$\delta_l(X, \overline{R}, \overline{W}) = g_l(X, \overline{R}, \overline{W}) - \ln\left\{\frac{1}{N-1}\sum_{j, j\neq l}\exp(-g_j(X, \overline{R}, \overline{W})\zeta')\right\}^{-1/\zeta'} \tag{20}$$

When $\zeta' \to \infty$, (20) can be approximated by

$$\delta_l(X, \overline{R}, \overline{W}) \cong g_l(X, \overline{R}, \overline{W}) - g_{j'}(X, \overline{R}, \overline{W}) \tag{21}$$

where $g_{j'}(X, \overline{R}, \overline{W})$ is the discriminant measure of the most probable incorrect word j'.

A loss measure,

$$\ell_l(X, \overline{R}, \overline{W}) = \ell(\delta_l(X, \overline{R}, \overline{W})) = \frac{1}{1 + e^{-\delta_l(X, \overline{R}, \overline{W})}} \tag{22}$$

is used to show the cost of misclassifying an input token X of the lth word. It is required that the loss function $\ell(\cdot)$ be a differentiable, monotonically non-decreasing function. Here, a sigmoid function as (22) is used as it approximates well the 0–1 cost function for classification error.

According to (17), (20), and (22), and assuming in (17) $\zeta \to \infty$, we can easily compute $\nabla_{\overline{W}}\ell_l(X, \overline{R}, \overline{W})$. In particular,

$$\nabla_{W^l}\ell(\delta_l(X, \overline{R}, \overline{W})) = \nu D_l^l \tag{23}$$

$$\nabla_{W^j}\ell(\delta_l(X,\overline{R},\overline{W})) = -\nu\frac{e^{-g_j(X,\overline{R},\overline{W})\zeta'}}{\sum_{k,k\neq l} e^{-g_k(X,\overline{R},\overline{W})\zeta'}}D_l^j, \quad \text{for all } j, j\neq l \tag{24}$$

where

$$\nu = \ell(\delta_l(X,\overline{R},\overline{W})) \cdot (1-\ell(\delta_l(X,\overline{R},\overline{W}))). \tag{25}$$

According to (15) and (16), and by setting U to be an identity matrix, the rules for adaptive weighting function adjustment in the GPD training method can be expressed as

$$W_{t+1}^l = W_t^l - \varepsilon\nu D_l^l \tag{26}$$

$$W_{t+1}^j = W_t^j + \varepsilon\nu\frac{e^{-g_j(X,\overline{R},\overline{W})\zeta'}}{\sum_{k,k\neq l} e^{-g_k(X,\overline{R},\overline{W})\zeta'}}D_l^j, \quad \text{for all } j, j\neq l. \tag{27}$$

If we further let in (27) $\zeta' \to \infty$, the adjusting rule can be simplified as

$$W_{t+1}^l = W_t^l - \varepsilon\nu D_l^l \tag{28}$$

$$W_{t+1}^{j'} = W_t^{j'} + \varepsilon\nu D_l^{j'} \tag{29}$$

$$W_{t+1}^j = W_t^j, \quad \text{for all } j, j\neq l, j'. \tag{30}$$

Therefore, in an extreme case where ζ and $\zeta' \to \infty$, only W^l and $W^{j'}$, which is the weighting function of the most probable incorrect word j', will be adjusted. According to (25), the ν shown in the above equations serves as an adaptive step size of weight adjustment. When the absolute value of $\delta_l(X,\overline{R},\overline{W})$ is small, which implies that the training token is likely to be confused with the wrong word, ν will be large, leading to a substantial amount of adjustment. When the absolute value of $\delta_l(X,\overline{R},\overline{W})$ is large as in the case where the input token is either unlikely to cause confusion or obviously an extreme outlier, the amount of adjustment is therefore accordingly reduced. Finally, we note that the rules shown in (26)–(29) certainly meet the requirements of (4) because the term ν is always nonnegative.

We should emphasize here that with the choice of a proper loss function, such as a 0-1 function to which the sigmoid function of (22) is an approximation, the GPD method is aiming at direct minimization of the classification error. The error propagation method, on the other hand, attempts to match the activation value to a (binary valued) desired output using a mean square error measure and thus does not necessarily minimize the error rate [1], [2].

III. Experimental Results

A series of experiments were conducted to examine the effectiveness of the proposed training algorithms. The experiments involved recognition of the highly confusible English E-set. In this E-set database, it contains utterances of 9 English alphabets, namely, b, c, d, e, g, p, t, v, and z. The speech signals were recorded from 100 native Americans, including 50 males and 50 females, through local dialed-up telephone lines. The sampling rate was 6.67 kHz and the bandwidth of the anti-aliasing filter was from 100 to 3200 Hz. Each talker spoke each word twice to produce two sets of databases. One was used as the training set and the other was used as the test set. During analysis, each data frame consisted of 300 samples with the last 200 samples overlapped with the next frame. An eighth-order LPC analysis was performed on each frame of data and the resultant coefficients were transformed into 12 cepstral coefficients (with bandpass cepstral liftering [9]) and 12 delta-cepstral coefficients [10].

A. The Conventional DTW Algorithm

For performance comparison, a conventional DTW algorithm using an average distortion (1) as the distance score was first implemented. The experimental results are listed in Table I. In the 1st row of Table 1, $rf\ n$ denotes the case in which each word in the recognizer is represented by n reference templates. In the second & third rows denoted by D1 and D2, we show the results based on the k-NN rules with $k = 1$ and 2, respectively. The DTW algorithm was implemented by warping the reference template to the test token. The length of the distortion sequence in terms of the number of frames is therefore identical to that of the test token. Usually, DTW algorithm is implemented in this way because the distance score can be obtained directly by summing the distortion sequence without a frame normalization. The best recognition rate is 69.6% for $rf11$ and $k = 2$. Another type of DTW algorithm was also implemented by warping the test token to the reference template. In this case, the operation of frame normalization is necessary to compute the distance score. Experimental results are shown in the fourth row (D3) and the fifth row (D4) for k-NN rules with $k = 1$ and 2, respectively. We provide only two result entries for D3 and D4 to show the effect of warping direction. A recognition rate of 67.6% was achieved for $rf12$ and $k = 2$. Compared to the recognition rate of 61.7% achieved by a continuous HMM recognizer with 5 states and 5 mixtures per state [11], recognition results of these two conventional DTW algorithms are reasonably good.

In the following experiments, the simple average distortion of (1) is replaced by the weighted distortion of (3) in the DTW implementation. We also choose to warp the test sequence to the reference sequence because it avoids the need of multiple weighting functions for each reference sequence due to temporal variations. The k-NN rule with $k = 2$ was used when each word was represented by 12 reference templates.

B. The Heuristic Method

The performance of the heuristic method is tabulated in Table II. In the second row (H1), the results were obtained using (8) and (11). In the third row (H2), the results were obtained using (9) and (12).

Judged from the results shown in Table II, the recognition accuracy indeed has been improved by the heuristic method

TABLE I
RECOGNITION RESULTS BASED ON THE TRADITIONAL DTW ALGORITHM WITH DIFFERENT WARPING TYPES (Type 1: D1, D2; Type 2: D3, D4) AND DIFFERENT k-NN RULES (k = 1: D1, D3; k = 2: D2, D4)

	rf 1	rf 2	rf 3	rf 4	rf 5	rf 6	rf 7	rf 8	rf 9	rf 10	rf 11	rf 12
D1	58.8%	54.7%	57.8%	59.7%	58.7%	60.2%	63.0%	65.8%	66.2%	66.2%	65.9%	66.1%
D2		49.9%	54.6%	58.2%	61.3%	63.2%	63.1%	63.4%	65.6%	68.1%	69.6%	69.3%
D3	59.8%											
D4												67.6%

TABLE II
RECOGNITION RESULTS BASED ON THE HEURISTIC METHOD WITH DIFFERENT ADJUSTMENT STRATEGIES (H1: ALL COMPONENTS OF THE WEIGHTING FUNCTIONS WERE ADJUSTED IN ONE ITERATION; H2: ONLY ONE COMPONENT WAS ADJUSTED)

	rf 1	rf 12
H1	60.2%	71.0%
H2	66.6%	75.8%

TABLE III
RECOGNITION RESULTS BASED ON THE ERROR PROPAGATION ALGORITHM WITH DIFFERENT ADJUSTMENT STRATEGIES (E1: ALL WEIGHTING FUNCTIONS WERE ADJUSTED IN ONE ITERATION; E2: ONLY TWO WEIGHTING FUNCTIONS WERE ADJUSTED)

	rf 1	rf 12
E1	61.9%	65.7%
E2	62.7%	72.1%

as expected. In particular, the results corresponding to H2 indicate a significant reduction in error rate, compared to the benchmark results shown in Table I. The conventional unweighted method resulted in 40.2% and 32.4% error rates for the case of 1 reference template per word and 12 reference templates per word respectively while the H2 weighting method produced 33.4% and 24.2% error rate in the corresponding cases. The performance of H2 where only one component of each weighting function corresponding to the maximal or minimal component in the distortion sequence was adjusted is better than that of H1 where all components of the weighting functions were adjusted at the same time.

C. The Error Propagation Method

The experimental results based on (14) are shown in the second row (E1) of Table III. As mentioned in Section II-B, the modified error propagation method can be analyzed as a scaled single-layer perceptron. In this case, all weighting functions were adjusted in each iteration. Alternatively, we may adjust only two weighting functions associated with the correct and the most probable incorrect words in each iteration as suggested by (28)–(30) of the GPD method. The results are shown in the 3rd row (E2).

As can be seen from Tables II and III, the modified error propagation method didn't perform as well as the heuristic method (H2). The best results with error propagation learning were obtained using the simplified method (E2). The improvements are only 2.9% and 4.5% in error rate reduction for the cases of 1 and 12 reference templates per word, respectively.

TABLE IV
RECOGNITION RESULTS BASED ON THE SINGLE-LAYER PERCEPTRON

	rf 1	rf 2
S1	63.9%	58.3%

D. Fully Connected Single-Layer Perceptron

The fully connected single-layer perceptron described in Section II-C was designed only for $rf1$ and $rf2$ due to complexity considerations. Experimental results are shown in Table IV.

From Table IV the result of $rf2$ is worse than that of $rf1$. It is probably due to the reason that the structural complexity of the case $rf2$ is too excessive to be trained with the current limited database. Compared with previous methods, the recognition rates achieved by the fully connected single-layer perceptron are relatively low.

E. The GPD Method

Table V displays the performance of the GPD method. Two cases were considered first. The results shown in the second row (G1) and the third row (G2) were obtained by using (26) and (27) with $\zeta' = 30$, and (28)–(30) respectively. The difference between these two cases is that in G1 all weighting functions were adjusted during one iteration while in G2 only two specified weighting functions were adjusted. In fact, as mentioned in Section II-D, G2 is an extreme case of G1 when ζ' in (27) approaches ∞. Alternatively, since (28) and (29) also meet the heuristic requirements of (4), adjustment of all weighting functions in one iteration based on these two equations was also implemented. Specifically, the index j' in (29) was extended to all j except l. Since the adaptive step size ν depends on $g_l(X, \overline{R}, \overline{W}) - g_{j'}(X, \overline{R}, \overline{W})$ according to (21) and (25), the weighting functions are, therefore, adjusted in pairs. That is, in the modified method, we treated each reference pattern $R^j, j = 1, 2, \cdots, N, j \neq l$, as a potential contender of R^l, the correct class reference. There were thus $N - 1$ pairs of weighting functions to adjust in each iteration for an N reference pattern set. As a result, each W^j was adjusted one time but W^l was adjusted $N-1$ times in one iteration. The experimental results are shown in the fourth row (G3). Note that in the above three cases, according to (26)–(29) all components of the weighting functions were adjusted during each iteration. Alternatively, like the extreme case of H2, we may change the policy of weight adjustment in these three cases by only adjusting the component of W^l corresponding to the minimal

TABLE V
RECOGNITION RESULTS BASED ON THE GPD METHOD WITH DIFFERENT ADJUSTMENT STRATEGIES (G1, G4: ALL WEIGHTING FUNCTIONS WERE ADJUSTED IN ONE ITERATION; G2, G5: ONLY TWO WEIGHTING FUNCTIONS WERE ADJUSTED; G3, G6: WEIGHTING FUNCTIONS WERE ADJUSTED IN PAIRS)

	rf 1	rf 12
G1	70.0%	76.2%
G2	69.9%	76.8%
G3	67.7%	78.1%
G4	60.1%	71.0%
G5	59.8%	70.7%
G6	62.1%	69.3%

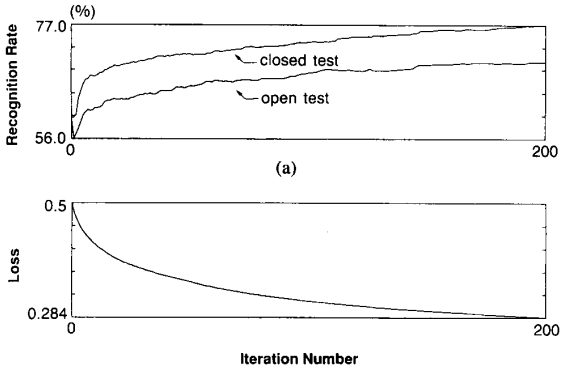

Fig. 3. Recognition results of G1. (a) The recognition results of close-set and open-set tests. (b) The loss measure.

component of D_l^l and adjusting the component of $W^j (j = 1, 2, \cdots, N, j \neq l)$ corresponding to the maximal component of D_l^j. The results of single component adjustment corresponding to the cases of G1, G2, and G3 are shown in the fifth row (G4), the 6th row (G5) and the 7th row (G6), respectively.

From Table V, it is found that the best recognition results of the GPD method are 70.0% (G1) and 78.1% (G3) when each word is represented by 1 and 12 reference templates, respectively. It can also be found that the results of G1, G2, and G3 are better than that of G4, G5, and G6. The result is reverse compared to that of the heuristic method shown in Table II. From the results shown in Tables I and V, the GPD method outperforms the conventional unweighted DTW method by about 10% in recognition rate. This is a significant improvement.

To analyze the convergence characteristics of the GPD method, recognition results of the close-set and the open-set tests, and the loss measure pertaining to the case of G1 are shown in Fig. 3. The recognition rates and the loss measures were evaluated at the end of every cycle after all training tokens were run once. As shown in Fig. 3, the recognition rates for both the close-set and open-set tests were gradually increased, and the loss measure was monotonically decreased. While the recognition results of G1 and G2 corresponding to $\zeta' = 30$ and $\zeta' \to \infty$ in (27), respectively, are almost the same as shown in Table V, the adaptation algorithm of G1 appears

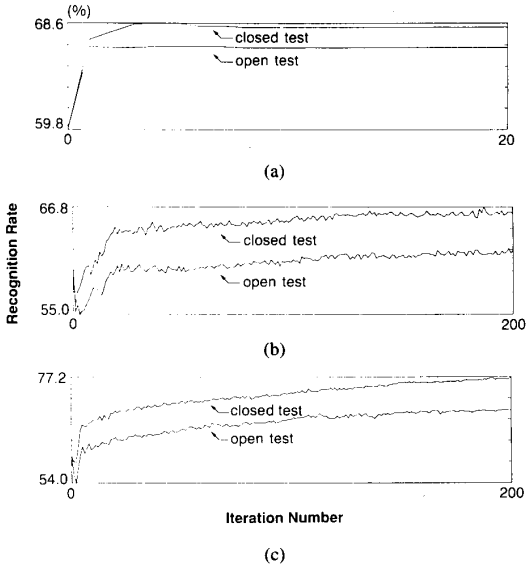

Fig. 4. Recognition results of close-set and open-set test using (a) the heuristic method, (b) the error propagation algorithm, (c) the GPD method.

to produce a smoother learning curve (Fig. 3(a)) than that of G2 (Fig. 4(c)). In the next section, we provide more detailed discussions on various algorithms.

IV. DISCUSSIONS

We find from the experimental results that the best performance was obtained by the GPD method. The heuristic method also led to good results. They are better than the other two methods based on the error propagation algorithm and the fully-connected single-layer perceptron. For performance comparison, we show in Fig. 4 recognition results of the close-set and the open-set tests pertaining to (a) the heuristic method (H2 in Table II), (b) the error propagation algorithm (E2 in Table III), and (c) the GPD method (G2 in Table V). From Fig. 4, it is found that the convergence speed of the heuristic method is the fastest. On the other hand, the recognition results using the GPD method are the best in both the closed test and the open test recognition experiments.

Fig. 5 displays the weighting functions of the word 'C' based on the heuristic method, the GPD method, and the error propagation algorithm. The most significant weights appear at the beginning part of the utterance in all three cases. This matches the characteristics of the vocabulary in the E-set in which the acoustic differences mainly manifest at the beginning of the utterance.

Comparing (14) derived from the error propagation algorithm and (28) derived from the GPD method, we find that they are very similar. The main difference is in the term μ in the error propagation algorithm and the term ν in the GPD method. In fact,

$$\mu \propto a^l(1 - a^l) = \ell(b^l)(1 - \ell(b^l))$$

and

$$\nu \propto \ell(\delta_l(X, \overline{R}, \overline{W}))(1 - \ell(\delta_l(X, \overline{R}, \overline{W})))$$

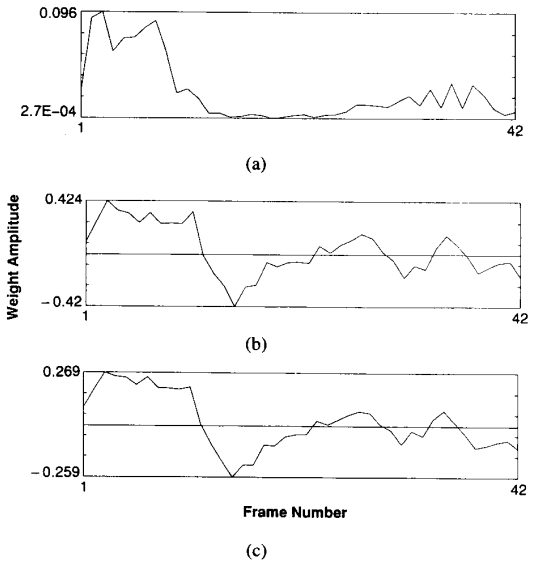

Fig. 5. The weighting function of word 'C' trained by (a) the heuristic method, (b) the error propagation algorithm, (c) the GPD method.

where

$$b^l = \sum_{i=1}^{m^l} w^l(i) d_l^l(i) + \theta^l$$

$$\ell(x) = \frac{1}{1+e^{-x}}$$

and

$$\delta_l(X, \overline{R}, \overline{W}) = \sum_{i=1}^{m^l} w^l(i) d_l^l(i) - \sum_{i=1}^{m^{j'}} w^{j'}(i) d_l^{j'}(i).$$

From the above analysis we can see that the convergence will be reached for the error propagation algorithm when $b^l \ll 0$ and $b^{j'} \gg 0$, and for the GPD method when $\delta_l(X, \overline{R}, \overline{W}) \ll 0$. However, the final convergence goals for the error propagation algorithm and the GPD method are quite different. The error propagation algorithm only attempts to match the desired output value while the GPD method directly aims at minimization of the error rate. That is probably the reason why the GPD method performed better than the error propagation algorithm in our experiments.

V. Conclusions

In this paper, the discriminative characteristics of distortion sequences generated by the DTW algorithm have been analyzed. A linear discriminant function is generated from the distortion sequence to serve as a substitute for the conventional dissimilarity score of average distortion. No major revision of the conventional system is required to incorporate the explicit discriminative analysis. Several training algorithms were suggested to train the weighting function of the discriminant function. The proposed recognition scheme has been confirmed to significantly outperform the conventional DTW-based recognition system. The best recognition rate of 78.1% was obtained by using the GPD training algorithm, tested on a highly confusible speaker independent English E-set database, compared to 67.6% with a conventional system.

To further improve the performance, two extensions may be worth studying. One is to use a nonlinear discriminant function instead of a linear one. The other is to discriminatively adjust reference templates as well as the weighting functions.

References

[1] D. E. Rumelhart, J. L. McClelland, etc., *Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition (Vol. 1)*, MIT Press, 1986.
[2] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4–22, Apr. 1987.
[3] S. Katagiri, C. H. Lee, and B. H. Juang, "A theory on adaptive training for pattern classification," to be published.
[4] L. R. Rabiner and J. G. Wilpon, "A two-pass pattern-recognition approach to isolated word recognition," *Bell Syst. Tech. J.*, pp. 739–766, 1981.
[5] A. Waibel and K. F. Lee, *Readings in Speech Recognition*. chap. 7.
[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[7] J. G. Wilpon and L. R. Rabiner, "A modified K-means clustering algorithm for use in isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 587–594, 1985.
[8] Shunichi Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Electron. Comp.*, pp. 299–307, June 1967.
[9] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, "On the use of bandpass liftering in speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 947–954, 1987.
[10] F. K. Soong and A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 871–879, 1988.
[11] K. Y. Su and C. H. Lee, "Robustness and discrimination oriented speech recognition using weighted HMM and subspace projection approaches," in *Proc. ICASSP-91*, pp. 541–544, 1991.

Pao-Chung Chang, for a photograph and biography please see page 143 of the April 1993 issue of this TRANSACTIONS.

Sin-Horng Chen received the B.S. degree in communication engineering and the M.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, Republic of China, in 1976 and 1978, respectively, and the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, in 1983.

From 1978 to 1980, he was an Assistant Engineer for Telecommunication Laboratories, Taiwan. He became an Associate Professor at the Department of Communication Engineering, National Chaio Tung University in August 1983, and a professor in August 1990. He also became the Chairman from August 1985 to June 1988, and from October 1991 till now. He is currently doing research in the areas of digital communication and speech processing, specially concentrating on the problems of Mandarin speech recognition and text-to-speech.

Biing-Hwang Juang (S'79–M'81–SM'87–F'92), for a photograph and biography please see page 24 of the January 1993 issue of this TRANSACTIONS.