TR－A－0130

GPD Training of Dynamic Programming-Based Speech Recognizers

Takashi Komori and Shigeru Katagiri

1992. 1.17

ATR視聴覚機構研究所

〒619-02 京都府相楽郡精華町光台2-2　☎07749-5-1411

ATR Auditory and Visual Perception Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1411
Facsimile: +81-7749-5-1408
Telex: 5452-516 ATR J

Abstract

Although many pattern classifiers based on artificial neural networks have been vigorously studied, they are still inadequate from a viewpoint of classifying dynamic (variable- and unspecified-durational) speech patterns. To cope with this problem, the generalized probabilistic descent method (GPD) has been recently proposed. GPD not only allows one to train a discriminative system classifying dynamic patterns, but also possesses a remarkable advantage, namely the learning optimality guaranteed in a sense of probabilistic descent search. A practical implementation of this theory, however, remains to be evaluated. In this light, we particularly focus on evaluating GPD in designing a widely-used speech recognizer based on dynamic time warping distance-measurement. We also show that a design algorithm appraised in this paper can be considered as a new version of learning vector quantization, which is incorporated with the dynamic programming. Experimental evaluation results in tasks of classifying syllables and phonemes clearly demonstrate the GPD's superiority.

Contents

1. Introduction

Applying artificial neural networks (ANNs) to classifier design has attracted considerable interest in the speech recognition field [1-5]. In particular, due to the inadequacy that most ANNs such as the multi-layer perceptron are originally suited in terms of structure to handling only a *static* (fixed-dimensional) vector, developing a satisfactory method of treating a dynamic speech pattern has been one of the most important research topics in this field. In this light, various new network structures have actually been studied: e.g., time delay neural networks [1] and a shift-tolerant learning vector quantizer [2]. However, compared with the conventional speech recognition approach using the dynamic time warping (DTW) philosophy, the new ANN-motivated approach is not necessarily adequate for representing such dynamics.

An effort to increase the discriminative capability of conventional DTW speech recognizers should thus be a promising alternative to the above ANN approach. These recognizers are usually designed so that class identity, e.g., design sample distribution, can be represented properly. Such a design, without considering classification directly, can not necessarily produce high classification power as a result. Thus, various kinds of improvement have actually been studied in this design framework (e.g. [6]); however, they unfortunately result in no significant advancement mainly because they all are based on heuristics and not on rigorous theoretical considerations. Motivated by this concern, one of the authors and his colleagues proposed a new discriminative learning theory, namely GPD, as one solution to this current difficulty [7-8]. GPD is a novel learning framework formalized by generalizing the classical probabilistic descent method which was developed a quarter of a century ago [9], and includes a family of new discriminative training ideas for various kinds of classifier structures. However, due to its intrinsic generality and complexity of formulation, careful evaluation of implementing GPD still remains an emerging research question, though very recently several evaluation studies have actually

been started [10-12]. This paper is intended to show the results of our research which is a continuation of this evaluation.

There are two main DTW approaches: one is based on a DP-based distance classifier (*DPC*) and one is based on a hidden Markov model (HMM). Comparing DPC and HMM, distance calculation which is time-consuming, but indispensable in DPC, is often criticized. Such criticism, however, holds true only in the scheme of traditional sequential computation. Remarkable progress in recent parallel computation and hardware technology, such as the advent of a fine-grained parallel machine, would remind one of the usefulness of a pattern classifier consisting of simple distance calculation (e.g., [13]). Thus, in this paper, we selected DPC, which has long been studied in speech recognition, as our implementation framework.

One of main goals in this paper is to evaluate a GPD-based discriminative training algorithm in designing a multi-reference DPC. We first introduce a somewhat complicated but general training rule based on the full knowledge of GPD; this rule is referred to as the *G-rule*. Detailed evaluation of the G-rule will be presented separately [12]. However, to increase awareness on application possibilities, we particularly focus on evaluating a more practical and simpler algorithm which can be defined as an extremely simple case of the G-rule; this simple algorithm is referred to as the *S-rule*. Experimental results using this S-rule actually demonstrate that GPD greatly contributes to increasing classification accuracy.

Interestingly, it is also shown that the S-rule can be viewed as an idea of learning vector quantization (LVQ) generalized so as to be suited to the use of DP. An alternative to such integration of LVQ and DP would occur to anyone recalling several intuitive hybrid ideas developed by combining existing algorithms (e.g., [6, 14-15]). However, without doubt, the theoretical appropriateness of a new algorithm should be proved. It is here worthwhile noting that GPD-based rules are guaranteed to be optimal from the viewpoint of a probabilistic descent search because each of them fundamentally inherits all properties of GPD.

This paper is organized as follows. Section 2 will be used to introduce the G-rule in detail. In Section 3, we will first propose the S-rule by simplifying the G-rule and next show that the S-rule can be treated as a generalized LVQ to categorize dynamic patterns. In Section 4, we will show the experimental results on two tasks: English isolated syllable classification and classification of Japanese phonemes extracted from word utterances. The paper will be concluded in Section 5.

2. GPD training for multi-reference DPC; G-rule

Two ideas, DTW and DP, are often used to represent the same technique in speech recognition. However, for clarity we will define these two ideas before starting our main discussion. In this paper, we define DTW as a general concept of nonlinear time warping which allows one to treat any possible, reasonable, time-alignment path between an input and a reference; on the other hand, we define DP as one specific class of DTW-based algorithms, which is usually pursued by the dynamic programming best-path search.

We here present the G-rule by accurately following the GPD formalization for a multi-reference DPC [7-8, 10-11]. Consider an M-class task C_m $(m=1,2,...,M)$. We assume that a speech utterance $\mathbf{x}$ is a variable but finite length sequence of acoustic feature vectors. Each acoustic feature vector has a fixed dimension (S), usually consisting of coefficients based on the linear predictive coding or the Fourier transform. $\mathbf{x}_t$ denotes the t-th acoustic feature vector (frame) of $\mathbf{x}$. We also assume that a classifier consists of a set of reference vectors

$$\Lambda=\left\{\lambda_m=\left\{\mathbf{r}_m^b\right\},\ b=1,2,...,B_m,\ \text{and}\ m=1,2,...,M\right\}, \tag{1}$$

where $\mathbf{r}_m^b$ denotes the C_m's reference pattern b-th closest to $\mathbf{x}$, and B_m is the number of reference patterns for C_m. $\mathbf{r}_m^b$ is also a variable but finite duration (T) sequence of acoustic feature vectors.

We consider a sequential, or adaptive, training scheme where a classifier is adjusted by a small amount every time a single training token is given. According to the GPD idea, we

introduce the following measurable distances. First, to measure the degree to which $\mathbf{x}$ belongs to C_m, a discriminant function, or *a class distance*, is defined by

$$g_m(\mathbf{x};\Lambda)=\left[\sum_{b=1}^{B_m}\left\{D\left(\mathbf{x},\mathbf{r}_m^b\right)\right\}^{-\zeta}\right]^{-1/\zeta}, \tag{2}$$

where $D\left(\mathbf{x},\mathbf{r}_m^b\right)$ is called *a reference distance* between $\mathbf{x}$ and $\mathbf{r}_m^b$, and ζ is a positive number. The reference distance is also defined in the same fashion as (2);

$$D\left(\mathbf{x},\mathbf{r}_m^b\right)=\left[\sum_{\theta=1}^{\Theta}\left\{D_\theta\left(\mathbf{x},\mathbf{r}_m^b\right)\right\}^{-\xi}\right]^{-1/\xi}, \tag{3}$$

where $D_\theta\left(\mathbf{x},\mathbf{r}_m^b\right)$ is *a path distance* accumulated along the θ-th *best* (smallest distance) path selected by the DP-matching between $\mathbf{x}$ and $\mathbf{r}_m^b$ among all the possible Θ paths. Here, the path distance is decomposed as

$$D_\theta\left(\mathbf{x},\mathbf{r}_m^b\right)=\sum_{t=1}^{T} w_{m,t}^b \delta_{m,\theta_t}^b, \tag{4}$$

where $w_{m,t}^b$ is a weighting factor corresponding to the t-th frame of $\mathbf{r}_m^b$, δ_{m,θ_t}^b is *a local distance* between the t-th frame of $\mathbf{r}_m^b$ and the corresponding frame θ_t of $\mathbf{x}$ along the θ-th best path. For simplicity, in this paper we treat the weighting factor as a non-trainable constant, though it was trained in [10, 12]. There are many possible ways of measuring the local distance, but by way of example we use here the Euclidean distance

$$\delta_{m,\theta_t}^b=\sum_{s=1}^{S}\left(\mathbf{r}_{m,t,s}^b-\mathbf{x}_{\theta_t^{m,b},s}\right)^2, \tag{5}$$

where $\mathbf{r}_{m,t,s}^b$ and $\mathbf{x}_{\theta_t^{m,b},s}$ are the s-th elements of $\mathbf{r}_{m,t}^b$ and $\mathbf{x}_{\theta_t^{m,b}}$ respectively, and $\mathbf{x}_{\theta_t^{m,b}}$ is the frame of $\mathbf{x}$ corresponding to $\mathbf{r}_{m,t}^b$ along the θ-th best path.

There are again many possibile of decision (classification) rules, each using the class distances, and we here simply choose the following rule

$$C(\mathbf{x})=C_i, \qquad \text{if } i=\underset{j}{\operatorname{argmin}}\left\{g_j(\mathbf{x};\Lambda)\right\}. \tag{6}$$

Our target is to train Λ such that misclassifications can be minimized given this decision rule. To perform this training, classification results should be embedded in a functional form which some reasonable optimization method, such as the gradient search, can treat

properly. In fact, GPD demonstrated that there were many possible ways for this embedding to take place. In particular, we here use the misclassification measure

$$d_k(\mathbf{x}) = g_k(\mathbf{x};\Lambda) - \left[\frac{1}{M-1}\sum_{j,j\neq k}\left\{g_j(\mathbf{x};\Lambda)\right\}^{-\mu}\right]^{-1/\mu} \tag{7}$$

where μ is a positive number, for a given training token $\mathbf{x} \in C_k$. One notes that a larger $d_k(\mathbf{x})$ implies a more definite misclassification of $\mathbf{x}$ and a negative $d_k(\mathbf{x})$ implies a correct classification.

A general form of cost function is next given as

$$\ell_k(\mathbf{x};\Lambda) = \ell_k\big(d_k(\mathbf{x};\Lambda)\big), \tag{8}$$

where ℓ_k is a monotonically increasing, differentiable function. It is worthwhile noting that the selection of ℓ_k leads to various implementations of training criteria, but the following function

$$\ell_k\big(d_k(\mathbf{x};\Lambda)\big) = \frac{1}{1+e^{-\alpha(d_k(\mathbf{x};\Lambda)+\beta)}}, \qquad \alpha > 0 \tag{9}$$

properly approximates the most important classification criterion, namely, the *minimum classification* error criterion (See [16]). In this light, we will use Eq. (9) in the discussion below.

GPD guarantees that adjusting Λ by $-\varepsilon U \nabla \ell_k(\mathbf{x};\Lambda)$, where ε is a small positive number and $\mathbf{U}$ is a positive definite matrix, leads to at least a locally optimal minimum error classifier in a probabilistic sense; this convergence property was the main result of the original probabilistic descent method [9]. An actual adjustment rule in our situation is given by

$$\left.\begin{aligned} \mathbf{r}^b_{m,t,s}(n+1) &= \mathbf{r}^b_{m,t,s}(n) - 2\varepsilon_n \nu_k w^b_{m,t} \phi_m \psi_m \tau_m, && \text{for } m = k, \\ \mathbf{r}^b_{m,t,s}(n+1) &= \mathbf{r}^b_{m,t,s}(n) + 2\frac{\varepsilon_n \nu_k w^b_{m,t}}{M-1} \kappa_m \phi_m \psi_m \tau_m, && \text{for } m \neq k, \end{aligned}\right\} \tag{10}$$

where

$$\nu_k = \ell_k'\big(d_k(\mathbf{x};\Lambda)\big) = \alpha \ell_k\big(d_k(\mathbf{x};\Lambda)\big)\big\{1 - \ell_k\big(d_k(\mathbf{x};\Lambda)\big)\big\},$$

$$\phi_m = \left[\sum_{b'=1}^{B_m} \left\{ \frac{D\left(\mathbf{x}, \mathbf{r}_m^b\right)}{D\left(\mathbf{x}, \mathbf{r}_m^{b'}\right)} \right\}^{\zeta} \right]^{-\frac{1+\zeta}{\zeta}},$$

$$\psi_m = \left[\sum_{\theta=1}^{\Theta} \left\{ D_\theta\left(\mathbf{x}, \mathbf{r}_m^b\right) \right\}^{-\xi} \right]^{-\frac{1+\xi}{\xi}},$$

$$\tau_m = \left[\sum_{\theta=1}^{\Theta} \frac{\left(\mathbf{r}_{m,t,s}^b - \mathbf{x}_{\theta_t^{m,b},s} \right)}{\left\{ D_\theta\left(\mathbf{x}, \mathbf{r}_m^b\right) \right\}^{\xi+1}} \right], \text{ and}$$

$$\kappa_m = \left[\frac{1}{M-1} \sum_{j, j \neq k} \left\{ \frac{g_m(\mathbf{x};\Lambda)}{g_j(\mathbf{x};\Lambda)} \right\}^{\mu} \right]^{-\frac{1+\mu}{\mu}}.$$

Eq. (10) gives the complete G-rule procedure. Although we only consider handling dynamic patterns in this paper, our result is also applicable to static vector classification because a static vector is merely a limited case of a dynamic pattern.

3. Simplified training; S-rule

Eq. (10) illustrates that all the reference vectors will be adjusted along all the time warping paths every time a single training token is given. However, the computation of all these possible cases would be very time-consuming, and a reasonable way of simplifying Eq. (10) should be attempted.

There are obviously various possibilities in implementing Eq. (10), due to the selection of the parameters such as μ. In this section, we particularly consider an extremely simplified case of G-rule, by setting $\mu, \zeta, \xi \to \infty$. This kind of extreme simplification was discussed in [7] for the case of classifying static vectors and, in fact, a crucial link between LVQ and GPD was revealed. The extreme setting results in the following manifolds.

$$d_k(\mathbf{x}) \approx g_k(\mathbf{x};\Lambda) - g_i(\mathbf{x};\Lambda), \tag{11}$$

where C_i $(i \neq k)$ is the most probable among the incorrect classes.

$$g_m(\mathbf{x};\Lambda) \approx D\left(\mathbf{x},\mathbf{r}_m^1\right) \tag{12}$$

$$D\left(\mathbf{x},\mathbf{r}_m^1\right) \approx D_1\left(\mathbf{x},\mathbf{r}_m^1\right) \tag{13}$$

Here, one should note the following three points: 1) due to Eq. (11), only two classes, the correct class and the most probable among the incorrect classes, concern the classification decision; 2) each class distance is represented by only the reference closest to the input among the same class references; 3) the reference distance is measured along only the corresponding best path usually selected using the DP-based minimum search operation. Eq. (10) is then reduced into

$$\left.\begin{aligned}
\mathbf{r}_{m,t,s}^1(n+1) &= \mathbf{r}_{m,t,s}^1(n) - 2\varepsilon_n \nu_k w_{m,t}^1\left(\mathbf{r}_{m,t,s}^1(n) - \mathbf{x}_{1_t^{m,1},s}\right), && \text{for } m = k,\\
\mathbf{r}_{m,t,s}^1(n+1) &= \mathbf{r}_{m,t,s}^1(n) + 2\varepsilon_n \nu_k w_{m,t}^1\left(\mathbf{r}_{m,t,s}^1(n) - \mathbf{x}_{1_t^{m,1},s}\right), && \text{for } m = i,\\
\mathbf{r}_{m,t,s}^b(n+1) &= \mathbf{r}_{m,t,s}^b(n), && \text{otherwise,}
\end{aligned}\right\} \tag{14}$$

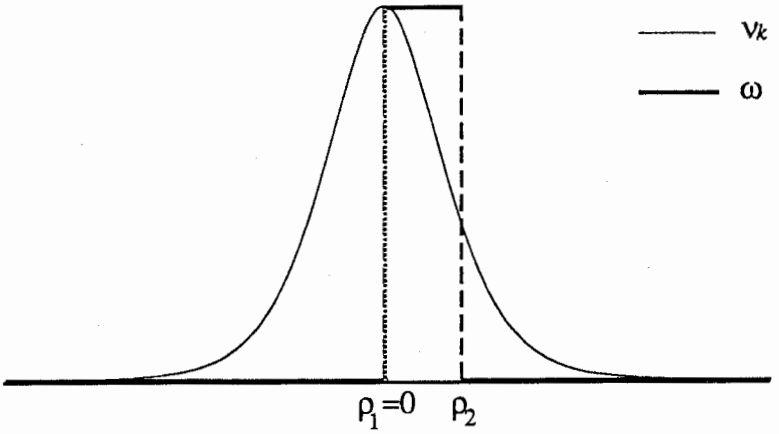
where n is a discrete time index in training, ε_n is a small positive number, and $1_t^{k,1} = \theta_t^{k,1}\big|_{\theta=1}$.

The optimality of probabilistic descent search requires ε_n to meet the conditions in [9], originally based on the stochastic approximation philosophy. However, these conditions (assuming infinite training repetition) are actually never realistic. Thus, we reasonably approximate ε_n as

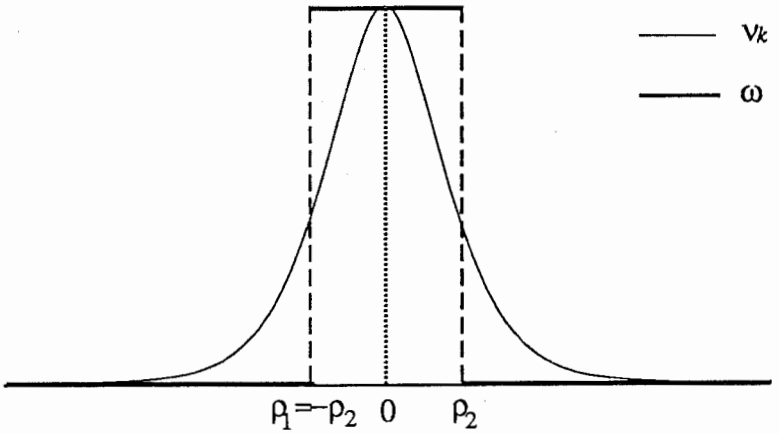
$$\varepsilon_n = \varepsilon_0\left(1 - \frac{n}{N}\right), \tag{15}$$

where ε_0 is a positive small number and N is a large prescribed positive constant.

We next consider the simplification of the derivative form of the loss ν_k. The form ν_k is originally a symmetric function of $d_k(\mathbf{x})$, i.e. unimodal around an actual class boundary. See Figure 1. Taking account of the fact that the *learning-when-incorrect* is essential for minimizing misclassifications on training data, we replace ν_k by the following step function



(a)



(b)

Figure 1. Derivative form of loss (ν_k) and its approximation (ω). The form ω in (a) is designed to minimize misclassifications over training data, and the form ω in (b) is expected to create a robust class boundary for unknown data. The ordinate is here normalized.

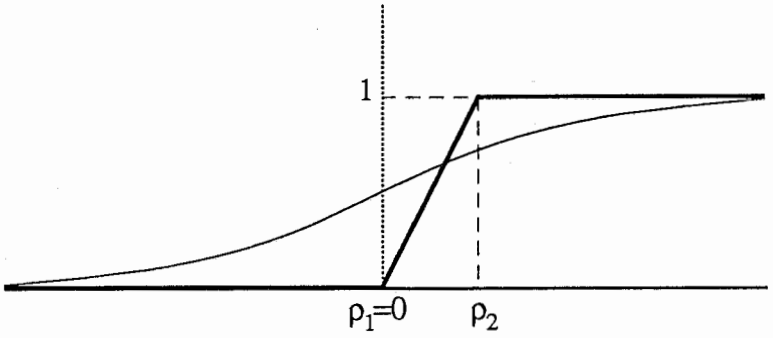


Figure 2. The form of loss (thin line) and a sample of its approximation (thick line).

$$\omega = \begin{cases} 1, & \rho_1 < d_k(\mathbf{x}) < \rho_2, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where ρ_1 is zero and ρ_2 is a positive constant. One may note that ω formally corresponds to the derivative of a piece-wise linear step function approximating the smooth sigmoid loss (See Figure 2). Eq. (14) is now revised to

$$\left.\begin{aligned} \mathbf{r}^1_{m,t,s}(n+1) &= \mathbf{r}^1_{m,t,s}(n) - 2\varepsilon_n \omega w^1_{m,t}\left(\mathbf{r}^1_{m,t,s}(n) - \mathbf{x}_{1^{m,1}_t,s}\right), && \text{for } m = k, \\ \mathbf{r}^1_{m,t,s}(n+1) &= \mathbf{r}^1_{m,t,s}(n) + 2\varepsilon_n \omega w^1_{m,t}\left(\mathbf{r}^1_{m,t,s}(n) - \mathbf{x}_{1^{m,1}_t,s}\right), && \text{for } m = i, \\ \mathbf{r}^b_{m,t,s}(n+1) &= \mathbf{r}^b_{m,t,s}(n), && \text{otherwise.} \end{aligned}\right\} \tag{17}$$

We define here the rule in Eq. (17) as the S-rule.

The above discussion would already suggest that the LVQ adjustment philosophy underlies the S-rule. We discuss the S-rule in the view of a generalized LVQ in the remainder of this section. Consider the case in Figure 1 (a). Eq. (17) then implies that the adjustment occurs only for the misclassification that incurred around the actual class boundary. It is now obvious that the function ω substantially works as the vector space window of LVQ [17]. We then call the function ω *a space window* hereafter. One should note here that the adjustment strategy of Eq. (17) at each frame position is equivalent to the modified version of LVQ2 [18]. Our simplification eventually leads to the following; *an extremely simple GPD training rule for DPC, namely the S-rule, is equivalent to the generalized idea of performing the LVQ training along the DP best path between a dynamic input and the closest reference pattern.*

4. Experiments

Experiments were conducted on two sets of isolated speech tokens: 1) English E-rhyme letters (E-set), and 2) Japanese phonemes (P-set).

Table 1. Acoustic feature extraction conditions for E-set.

sampling frequency	6.67kHz
time window	45msec Hamming, 15msec shift
acoustic feature vectors	24-dimension (12-dim LPC cepstrum & 12-dim delta cepstrum)

4.1 E-set

The E-set is the task of classifying nine English E-rhyme letters, i.e. {b, c, d, e, g, p, t, v, z}. Speech tokens were recorded over telephone lines by one hundred untrained speakers: 50 female and 50 male speakers, which were then converted to acoustic feature vectors using the conditions shown in Table 1. Each speaker voiced each E-rhyme letter twice, once for training and once for testing. Due to their high confusability because of the common following sound / i: /, this task has long been used as a good framework to evaluate many different classifiers. Actually it was reported that one of the baseline systems, namely, a system consisting of continuous HMMs (5-component mixtured Gaussian distributions, 5-state left-to-right structure, and no skip) produced only 61.7% on unknown testing data (80.2% on training data) [14].

There are many possible ways selecting the system size, or the number of references. By way of example, we used the following two classifiers: one consisting of only one reference pattern per class (*R1*) and one consisting of three reference patterns per class (*R3*). Each classifier was first initialized using the modified *k*-means clustering method; two versions of this clustering idea, the minimax method and the pseudo-average method, were adopted for the sake of reliable comparison [19]. We then ran twenty epochs (one epoch = one full presentation of the training data) of the S-rule training for each classifier.

Figure 3 shows a typical training curve, i.e., a recognition rate over training data vs. an epoch. This curve demonstrates that the S-rule can achieve a very high discriminative power, quickly and steadily.

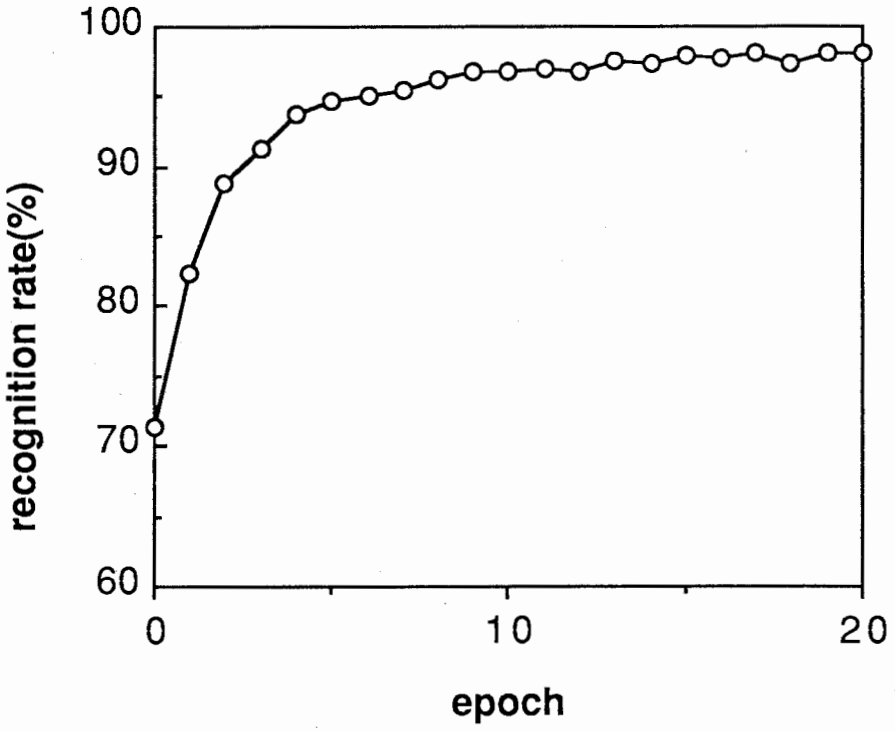


Figure 3. A typical training curve on the E-set task. The recognition rate on training data rises quickly and smoothly.

Table 2. Recognition rates in the E-set task.

Initial Reference Clustering Method	Number of Refs.	Before/After S-rule Training	Results Training	Results Testing
minimax	R1	before	56.7%	55.0%
minimax	R1	after	99.0%	74.2%
pseudo-avg.	R1	before	61.3%	59.8%
pseudo-avg.	R1	after	98.9%	74.9%
minimax	R3	before	72.4%	64.1%
minimax	R3	after	100.0%	72.4%
pseudo-avg.	R3	before	71.3%	64.9%
pseudo-avg.	R3	after	99.8%	74.0%

Table 2 lists the results for several different conditions. Regardless of the selection of the *k*-means clustering procedures, the difference in classification power between the initial situation and the after-training situation is obvious. Without doubt, the results illustrate the high superiority of our GPD-based training. In particular, a training achieved the significant reduction in recognition rates over the training data, ranging from 30% to 45%, as well as an almost perfect accuracy over the training data.

As we pointed out, the adjustment only for misclassification cases is enough to reduce misclassifications over the design samples. However, due to the finite number of design samples, the training, even correct classification cases would help increase the accuracy for future unknown data [20-21]. We then treat another space window; assuming $\rho_1 = -\rho_2$ (See Figure 1 (b)). This implies that the adjustment could occur even if a training token is correctly but somehow insecurely classified. Although further analysis is certainly required, it would be expected that this symmetric window creates a robust class boundary for unknown data (e.g., [16]). Table 3 shows results for this new space window. The results again show the superiority of the S-rule and also suggest the plausibility of the above discussion as concerns robustness.

In parallel with our study, several high recognition rates on this E-set have also been reported in succession [8,10,12,14]. It should be pointed out here that GPD underlies each

Table 3. Recognition rates on the E-set task using a symmetric space window.

Initial Reference Clustering Method	Number of Refs.	Before/After S-rule Training	Results Training	Results Testing
minimax	R1	after	94.1%	75.4%
pseudo-avg.	R1	after	96.1%	74.9%
minimax	R3	after	96.8%	77.2%
pseudo-avg.	R3	after	98.1%	76.7%

of these promising approaches.

4.2 P-set

Although the above results have not reset the record on the E-set (e.g., see [8]), the improvement in recognition rates clearly demonstrated the effectiveness of the proposed algorithm. However, the E-set consisting of only the E-rhyme letters is very special from the viewpoint of speech recognition tasks. We thus tested the algorithm on the second data set consisting of the possible Japanese phonemes, namely the P-set.

The P-set is a set of phoneme segments extracted from the ATR 5240 Japanese common word data, using manually-selected acoustic-phonetic labels [22]. This set was split into two independent sets of roughly equal sizes: one for training and one for testing. Each word was spoken in a soundproof booth by one male professional announcer and was transformed to a sequence of 16-dimensional Mel-scale spectrum vectors using the parameters specified in Table 4. We then used a 112-dimensional acoustic feature vector, concatenating 7 adjacent Mel-scale spectrum vectors. This high-dimensional vector was also normalized so that the average of all the vector components was 0.0.

The classifier was composed of 5 reference patterns per class, for 53 classes. Each of the background segments and contracted sounds such as / kj / was treated as one class. A word-initial vowel was categorized separately from a class of inside vowels and word-final vowels; e.g., a phoneme token / a / in the word beginning was treated as a different class token compared to a token / a / from the other segment positions. Moreover, a long vowel

Table 4. Acoustic feature extraction conditions for P-sets.

sampling frequency	12kHz
time window	20-msec Hamming, 5-msec shift
acoustic feature vectors	16-dimension Mel-scale coefficients (down-sampled from 256-point FFT coefficients)

such as / o: / in the word / o:ki: / was categorized as different from a typical vowel class. It should, however, be noticed that, in the recognition phases, the classifier did not distinguish in the vowel position, either word-initial or other; this implies that classification was evaluated for only 41 classes. Table 5 lists the 41 phoneme categories used in recognition. Taking account of the experimental result in the previous section, here we used the minimax method and the robust version of space window ($\rho_1 = -\rho_2$).

We performed 10 epochs of training, and achieved 98.4% over training data and 96.2% over testing data. This result shows that our discriminative training algorithm works well not only on small, special tasks such as E-sets, but also on ordinary tasks with a large number of classes. Although recognition rates by other algorithms have not been reported for the same class arrangement, namely 41 classes, our results can be considered rather high. Compare them with 95.3% by the shift-tolerant LVQ and 97.2% by LVQ-HMM, both of which were obtained in a 25-class condition using exactly the same database [6]. Consequently, the results here again demonstrated the very high discriminative power of the

Table 5. Forty one phoneme categories of the P-set.

consonants					vowels			background
/p/	/t/	/k/	/pj/	/kj/	/a/	/a:/		/*/
/b/	/d/	/g/	/bj/	/gj/	/i/	/i:/		
/s/	/ʃ/	/h/	/ç/	/ɸ/	/u/	/u:/		
/tʃ/	/ts/	/dʒ/	/dz/		/e/	/e:/	/eĭ/	
/r/	/w/	/j/	/ɲj/		/o/	/o:/	/oŭ/	
/m/	/n/	/N/	/mj/	/nj/				

Note 1: Each of / h /, / ç / and / ɸ / denotes the initial phoneme of the Japanese syllable {ha, hi, he, ho}, {hya, hyu, hyo} and {hu} respectively.

Note 2: / N / includes / m /, / n / and / ŋ /, each corresponding to the Japanese kana ん .

Note 3: / * / denotes a background segment.

S-rule.

5. Conclusion

We presented the new GPD-based discriminative training algorithms, the G-rule and the S-rule, for a multi-reference distance classifier handling dynamic patterns. We also showed that the simpler S-rule could be viewed as a generalized LVQ for dynamic pattern classification. In other words, the S-rule is actually equivalent to a hybrid of the LVQ rule and the DP-based best path search. However, it should be stressed here that the DP search is not essential to the optimality pursued in our approach; the DP-search is merely used for simplification and its optimality in *minimum* operation is not crucial in our optimization based on the gradient search. This point is quite different from several algorithms where the DP-based optimality is used as a part of training optimality (e.g., see [3]).

The simple and practical S-rule was evaluated in the two isolated-mode speech recognition tasks. The experimental results clearly demonstrated that the proposed algorithm contributes towards increasing the discriminative power of the traditional DP-based speech recognizer.

The experiments reported here were designed to show the fundamental effectiveness of our novel algorithms. The superiority of the S-rule was clearly demonstrated. However, the characteristics of our algorithms, particularly of the G-rule, must be studied further. Specifically, using finite values of classification parameters such as ξ in the G-rule would contribute toward increasing classifier robustness, which is an emerging ANN research topic, because the G-rule intrinsically possesses the promising property of smooth decision making based on multi-reference, multi-path, and multi-class measurement [12]. The proposed algorithms should also be studied in a more realistic task such as connected speech recognition.

It is lastly worthwhile nothing that, without any serious modification, the main formulations in this paper are applicable to the so-called batch-type gradient search, e.g.,

the steepest descent method, where all the training tokens are used at the same time for every adjustment. This point would be also tested.

Acknowledgements

It would have been impossible to pursue this study without the collaboration of, and enlightening discussions, with Dr. Biing-Hwang Juang and Dr. Chin-Hui Lee. The study was supported by many people at AT&T Bell Laboratories and ATR Auditory and Visual Perception Research Laboratories, particularly Dr. Lawrence Rabiner and Dr. Yoh'ichi Tohkura. Mr. Hitoshi Iwamida guided us in beginning the experiments. The authors would like to deeply thank all of them for their great help.

References

[1] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition: Neural Networks vs. Hidden Markov Models," IEEE, Proc. of ICASSP, Vol. 1, S3.3, pp. 107-110 (1988. 4).

[2] E. McDermott, and S. Katagiri, "Shift-invariant, Multi-category Phoneme Recognition Using Kohonen's LVQ2," IEEE, Proc. of ICASSP, Vol. 1, S3.1, pp. 81-84 (1989. 5).

[3] K. Iso, and T. Watanabe, "Speaker-independent Word Recognition Using a Neural Prediction Model," IEEE, Proc. of ICASSP, Vol. 1, S8.8, pp.441-444 (1990. 4).

[4] E. Levin, "Word Recognition Using Hidden Control Neural Architecture," IEEE, Proc. of ICASSP, Vol. 1, S8.6, pp.433-436 (1990. 4).

[5] T. Kohonen, "Learning Vector Quantization," Helsinki University of Technology, Report TKK-F-A-601 (1986. 11).

[6] H. Iwamida, S. Katagiri, and E. McDermott, "A Hybrid Speech Recognition System Using HMMs with an LVQ-trained Codebook," ASJ, JASJ (E), Vol. 11, No. 5, pp. 277-286 (1990. 9).

[7] S. Katagiri, C.-H. Lee, and B.-H. Juang, "A Generalized Probabilistic Descent Method," ASJ, Proc. of Fall Conf., 2-P-6, pp.141-142 (1990. 9).

[8] S. Katagiri, C.-H. Lee, and B.-H. Juang, "New Discriminative Training Algorithms Based on the Generalized Probabilistic Descent Method," IEEE, Neural Networks for Signal Processing, pp.299-308 (1991. 9).

[9] S. Amari, "A Theory of Adaptive Pattern Classifiers," IEEE, Trans. of EC, Vol. 16, No. 3, pp.299-307 (1967. 6).

[10] P.-C. Chang, S.-H. Chen, and B.-H. Juang, "Discriminative Analysis of Distortion Sequences in Speech Recognition," IEEE, Proc. of ICASSP91, Vol. 1, S8.6, pp.549-552 (1991. 5).

[11] T. Komori, and S. Katagiri, "A New Discriminative Training Algorithm for Dynamic Time Warping-based Speech Recognition," IEICE, Tech. Report SP91-10, pp.33-40 (1991. 6).

[12] P.-C. Chang, and B.-H. Juang, "Discriminative Training of Dynamic Programming Based Speech Recognizers," submitted for publication (1992. 3).

[13] A. Duchon, S. Katagiri, and E. McDermott, "Implementation of a Prototype-based Speech Classifier on a Fine-grained Parallel Computer," ASJ, Proc. of Fall Meeting, pp.161-162 (1991. 10).

[14] S. Katagiri, and C.-H. Lee, "A New HMM/LVQ Hybrid Algorithm for Speech Recognition," IEEE, Proc. of GLOBECOM, 608.2, pp. 1032-1036 (1990. 12).

[15] S. Mizuta, and K. Nakajima, "Optimum Discriminative Training for HMM with Continuous Mixture Densities," ASJ, Proc. of Spring Conf., 1-3-12, pp.23-24 (1990. 3).

[16] B.-H. Juang, and S. Katagiri, "Discriminative Learning for Minimum Error Classification," under submission.

[17] T. Kohonen, G. Barna, and R. Chrisley, "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," IEEE, Proc. of ICNN, Vol. 1, pp.61-68 (1988. 7).

[18] E. McDermott, "LVQ3 for Phoneme Recognition," ASJ, Proc. of Spring Conf., pp. 151-152 (1990. 3).

[19] J. Wilpon, and L. Rabiner, "A Modified *K*-means Clustering Algorithm for Use in Speaker-independent Isolated Word Recognition," AT&T Tech. Memo., 11227-840103-1 (1984. 1).

[20] S. Amari, *Information Theory II - Geometrical Theory of Information* -, Kyoritsu, pp.108-109 (1968. 1).

[21] A. Ando, and K. Ozeki, "A Clustering Algorithm to Minimize Recognition Error Function (in Japanese)," IEICE, Trans. (A), Vol. J74-A, No. 3, pp.360-367 (1991. 3).

[22] K. Takeda, Y. Sagisaka, S. Katagiri, and H. Kuwabara, "A Japanese Speech Database for Various Kinds of Research Purposes," ASJ, Proc., Vol. 44, No. 10, pp.747-754 (1988. 10).