TR－A－0088

A New HMM /LVQ Hybrid Algorithm for Speech Recognition

Shigeru KATAGIRI *Chin-Hui LEE*

1990. 8. 6

ATR視聴覚機構研究所

〒619-02 京都府相楽郡精華町乾谷 ☎07749-5-1411

ATR Auditory and Visual Perception Research Laboratories

Inuidani,Sanpeidani, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

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

to be published in Proc. of GLOBECOM'90, IEEE, December, 1990.

A NEW HMM/LVQ HYBRID ALGORITHM FOR SPEECH RECOGNITION

Shigeru Katagiri and Chin-Hui Lee*

Speech Research Department
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Abstract

The Learning Vector Quantization (LVQ) training algorithms are capable of producing highly discriminative reference vectors for classifying *static* patterns. The Hidden Markov model (HMM) formulation has also been successfully applied to recognizing *dynamic* speech patterns. In this paper, we present a new HMM/LVQ hybrid algorithm for speech recognition. We show that by combining both the discriminative power of LVQ and the capability of modeling temporal variations of an HMM into a hybrid algorithm, the performance of an HMM-based recognition algorithm is significantly improved. We tested the hybrid algorithm in a multi-speaker, isolated word mode, using a highly confusable vocabulary consisting of the nine English E-set words. The average word accuracy for the original HMM-based system was 62%. When the LVQ classifier is incorporated, the word accuracy increased to 81%.

1. Introduction

In recent years, artificial neural networks (ANN) have been successfully applied to classification of *static* patterns. However, the generalization to incorporate dynamics into neural networks is less satisfactory. On the other hand, hidden Markov model formulations have been proven useful for recognizing *dynamic* speech patterns. In this paper, we propose a new speech recognition algorithm which combines the advantages of both an ANN and an HMM. The proposed hybrid algorithm uses a novel classifier in place of the conventional HMM likelihood comparison. Since the parameters of the classifier can be estimated through adaptive learning rules, the discriminative power of the recognizer is greatly enhanced. Any learnable classifier, such as an ANN, can be used for the discriminative classifier. By way of example, we use an LVQ-based multicategory classifier in this study. Conceptually, the hybrid algorithm consists of three parts, namely: (a) the use of HMM to segment a speech utterance into a fixed number of acoustic segments, each representing one state of the HMM, (b) normalization of speech frames in an acoustic segment so that every speech utterance is represented by the concatenation of a sequence of fixed-dimension vectors each representing a word (or sub-word), and (c) recognition, which is performed by finding the most likely sequence of reference vectors (corresponding to a sequence of words) in an LVQ codebook so that the average distortion is minimized. The LVQ codebook is obtained through a *probabilistic descent method* [1] using segmented and normalized speech tokens as training vectors. To evaluate the performance of the hybrid algorithm, we used a highly confusable vocabulary consisting of the nine English E-set words. The test was conducted in a multi-speaker, isolated-word mode. The database consisted of 100 talkers, 50 males and 50 females, each speaking each of the E-set words twice, once for training and once for testing. The test results showed that the average error rate was reduced from 38% to 19% when the LVQ classifier is incorporated.

2. The Hybrid Algorithm

The key idea in the proposed hybrid algorithm is to use HMM-based segmentation to convert a speech utterance into a sequence of static patterns and to use any trainable classifier for discrimination. A block diagram of the hybrid algorithm is shown in Figure 1. HMMs are trained with a conventional method, e.g. the Baum-Welch algorithm or the segmental *k*-means algorithm. The reader is referred to [2] for a tutorial discussion on the HMM. In this paper, we assume that each word (class) is characterized by a single *n*-state left-to-right HMM with no skips.

A speech token is first translated into a sequence of acoustic feature vectors. The Viterbi segmentation is then performed on the observation vectors to segment the sequence into *n* time-aligned segments, each corresponding to one of the HMM states. According to our modeling assumptions, one expects the feature vectors grouped in the same state to possess common stochastic characteristics. We can therefore represent

* On leave from ATR Auditory and Visual Perception Research Laboratories, Sanpeidani Inuidani, Seika-cho Soraku-gun, Kyoto 619-02 Japan.

all the feature vectors within a state by a single, fixed-dimension vector. Throughout this study, the centroid (mean vector) is used to represent each state. For the purpose of word recognition, we concatenate all state centroids into a word-based, fixed-dimension vector. We call this vector a *time-normalized* (TN) vector, and we use it as an input to the classifier. Since a speech utterance is intrinsically of variable duration, this time normalization procedure alleviates one of major limitations of using an ANN for speech pattern classification. Regardless of the duration of an input token, the size of this new TN vector depends only on the size of the original acoustic feature vector and the number of states in each model.

Once speech utterances are converted into fixed-dimension TN vectors, we can design a classifier in the space of the TN vectors. Any trainable classifier can be used to classify the TN vectors, and this classification procedure is equivalent to doing word recognition. The proposed hybrid algorithm can handle both isolated word and continuous speech utterances. In this study, we focus our discussion on isolated word recognition. Several comparison studies have shown that LVQ can realize discrimination power at least as high as that by ANN. By way of example, we design our classifier based on the LVQ principle.

3. Learning Vector Quantization

Two versions of LVQ have been proposed by Kohonen [3-4]. Previous studies [4,6] have shown that the second version, known as LVQ2, produces better classification results than that of LVQ1. LVQ2 was originally formulated using the Euclidean distance. Since the Euclidean distance measure is not scale invariant, it has difficulty handling vector components with different dynamic ranges. To cope with this problem, we generalize, in this paper, the LVQ2 idea so that *likelihood-based distance measures* can be adopted. We call this new version *LVQ2-L*, and refer to the original version as *LVQ2-E*. We also refer to LVQ2 as a general idea which covers both LVQ2-E and LVQ2-L.

Consider the following classification problem: given an M-dimension vector $\mathbf{x}$, we want to classify $\mathbf{x}$ into one of the K classes, C_1, ..., C_K. Each class C_k is characterized by $r(k)$ multivariate Gaussian distributions with parameters

$$\lambda_k = \left\{ \mathbf{m}_k^j, \mathbf{R}_k^j \right\}_{j=1}^{r(k)}, \qquad \mathbf{m}_k^j \in \Re^M, \quad \text{..... (1)}$$

where $\mathbf{m}_k^j$ and $\mathbf{R}_k^j$ are the mean vector and the covariance matrix of the j-th Gaussian distribution of C_k, respectively. Define the parameter space Λ as the collection of all class parameters,

$$\Lambda = \{\lambda_1, \ldots, \lambda_K\}. \quad \text{..... (2)}$$

Then the likelihood-based distance measure for each Gaussian distribution is a function of the likelihood $L(\mathbf{x})$ of observing a vector $\mathbf{x}$. To simplify our discussion, we assume the features are uncorrelated. We therefore define the distance measure, for the j-th distribution in class C_k, as

$$D_k^j(\mathbf{x}) = \sum_{i=1}^{M} \left\{ \frac{\left(\mathbf{x}[i] - \mathbf{m}_k^j[i]\right)^2}{\mathbf{R}_k^j[i]} + \ln\left(\mathbf{R}_k^j[i]\right) \right\}$$

$$= -2\ln\left\{L_k^j(\mathbf{x})\right\} + \text{constant}, \qquad \text{..... (3)}$$

where $\mathbf{x}[i]$ and $\mathbf{m}_k^j[i]$ are the i-th element of $\mathbf{x}$ and $\mathbf{m}_k^j$, respectively, and $\mathbf{R}_k^j[i]$ is the (i, i)-th diagonal element of $\mathbf{R}_k^j$. To measure the likelihood that the classifier input $\mathbf{x}$ belongs to the class C_k, we define the following *class discrimination function*

$$g_k(\mathbf{x}, \Lambda) = \min_j \left\{ D_k^j(\mathbf{x}) \right\}. \quad \text{..... (4)}$$

Remember that a classification learning procedure using the likelihood-based distance is equivalent to estimating the Bayesian boundary with a set of Gaussian distributions. Adapting the mean vector moves the center location of the corresponding distribution, and adapting the covariance matrix changes the shape of the distribution. One may notice that the mean vector can be adapted more effectively in the same sense as the reference vector adaptation in LVQ2-E. Moving the mean vector closer to the input vector reduces the distance, while opposite movement increases it. Taking into account the correspondence between LVQ2-E and LVQ2-L, we here adapt

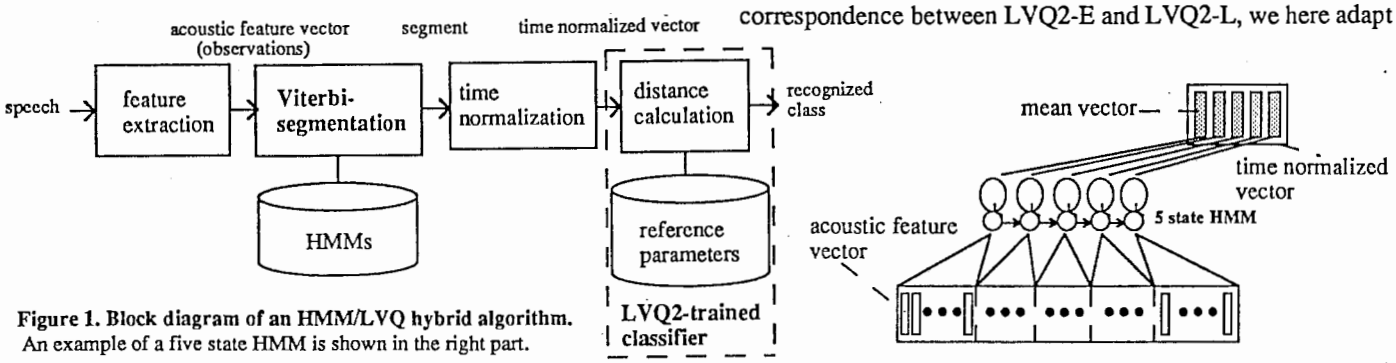

Figure 1. Block diagram of an HMM/LVQ hybrid algorithm. An example of a five state HMM is shown in the right part.

only the mean vectors. All $\mathbf{R}_k^j$'s are estimated once and assumed constant throughout the whole training procedure. $\mathbf{R}_k^j$ is computed for every $\mathbf{m}_k^j$'s cluster which is initialized by a VQ algorithm like the k-means clustering.

To adapt the classifier so as to reduce the number of miscategorizations, we define the following two conditions:

$$\left.\begin{array}{l}(1)\ g_\beta(\mathbf{x}(t),\ \Lambda(t))\ (\beta \neq \alpha) \text{ is the smallest,} \\ (2)\ g_\alpha(\mathbf{x}(t),\ \Lambda(t)) \text{ is the next-smallest.}\end{array}\right\} \quad \text{..... (5)}$$

LVQ2-L requires the above two conditions to be met every time a training vector $\mathbf{x}(t) \in C_\alpha$ is presented. Here t is a discrete time index indicating the order the training tokens are presented in the learning stage. If and only if these two conditions are satisfied, the mean vectors are adapted as follows.

$$\left.\begin{array}{l}\mathbf{m}_\alpha^{d_\alpha}(t+1)[i] = \mathbf{m}_\alpha^{d_\alpha}(t)[i] + w(t)\dfrac{\mathbf{x}(t)[i] - \mathbf{m}_\alpha^{d_\alpha}(t)[i]}{\mathbf{R}_\alpha^{d_\alpha}(t)[i]} \\ \mathbf{m}_\beta^{d_\beta}(t+1)[i] = \mathbf{m}_\beta^{d_\beta}(t)[i] - w(t)\dfrac{\mathbf{x}(t)[i] - \mathbf{m}_\beta^{d_\beta}(t)[i]}{\mathbf{R}_\beta^{d_\beta}(t)[i]}\end{array}\right\}, \quad \text{..... (6)}^1$$

where $d_k = \underset{j}{\operatorname{argmin}}\{D_k^j(\mathbf{x}(t))\}$ and $w(t)$ is a monotonically decreasing (positive value) function of time. Every time a single training vector is presented, the above adaptation of LVQ2-L is repeated in the same way as LVQ2-E learning until a preset convergence criterion is satisfied. Note that the third requirement of LVQ2-E (window condition [4-5]) was removed in LVQ2-L because it is not appropriate in LVQ2-L adaptation.

4. Experiment

We evaluated the proposed hybrid algorithm in a multi-speaker, isolated word recognition mode. The task was recognition of the set of nine English E-rhyme letters, i.e. {b, c, d, e, g, p, t, v, z}. Speech tokens were recorded over dialed-up telephone lines by one hundred untrained talkers: 50 male and 50 female speakers. Each talker spoke each of the E-set letters twice, once for training and once for testing. Table 1 shows a list of all the system parameters used in our study.

4.1 LVQ2 Training and Classification

There are nine hundred tokens available for training the HMMs. In order to account for possible segmentation errors caused by the HMMs and to improve system robustness, we generate, for each training token, all possible TN vectors $\mathbf{x}_1$, ... , $\mathbf{x}_9$, in which $\mathbf{x}_\ell$ is obtained using the HMM for class C_ℓ to segment the input token. Each $\mathbf{x}_\ell$ is assigned the same label as the training token. Therefore, we have a total of 8,100 TN vectors available for training the LVQ2 classifier.

Figure 2 illustrates the recognition procedure in detail. An unknown input token is first converted into a sequence of feature vectors. We then use all the nine HMMs to segment the observation sequence and generated nine TN vectors for classification. Two distance measures are needed: the first we call a *segmentation-oriented* (SO) distance $D(\ell, k)$ and the second we call a *class distance* $D(k)$ for C_k. They are defined as follows.

$$D(\ell, k) = g_k(\mathbf{x}_\ell, \Lambda) \quad \text{.... (7)}$$

$$D(k) = \sum_{\ell=1}^{K} v_\ell D(\ell, k), \quad \text{..... (8)}$$

where v_ℓ is a weighting coefficient. $D(\ell, k)$ is the same as the class discrimination function as defined in (4) with the TN vector $\mathbf{x}_\ell$ as variable. The class distance $D(k)$ is a weighted sum of all SO distances. The input token is categorized as the class with the smallest class distance. Several choices for v_ℓ are possible. In this study, we determine v_ℓ based on the

Table 1. System parameters.

Acoustic Feature Extraction

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

HMM

type: continuous density HMM
structure: 5 state left-to-right
observation probability: 5 component mixtured Gaussian distribution
training: segmental k-means clustering

Classifier

number of distributions $r(k)$ [1]: 1, 2, 3, 4, 5, 7, [2]10, 15
number of classes K: 9
number of training tokens for all the classes N_{tr}: 900
training epochs E [3]: 30
vector space: 120 (5 × 24) -dim time normalization vector space
weight in class distance v_ℓ: 1 or 0
time function: $w(t) = 0.1 \times \{1 - t / (E \times N_{tr} \times K)\}$ [4]

[1] $r(k)$ was set identical for all k's.
[2] This condition was hired only in experiments shown in Figure 5.
[3] The epoch represents the procedure where all the training tokens are used once to train the classifier. This number was selected according to our preliminary experiments.
[4] This time function was selected according to our previous study (See [6]).

[1] Parameters β, d_α, and d_β are also functions of time. However, for simplicity, we omit the index t in these notations.

HMM segmentation likelihoods. In particular, we sort all segmentation likelihoods and set $v_\ell = 1$ for the top P candidates and set $v_\ell = 0$ for the rest of the classes.

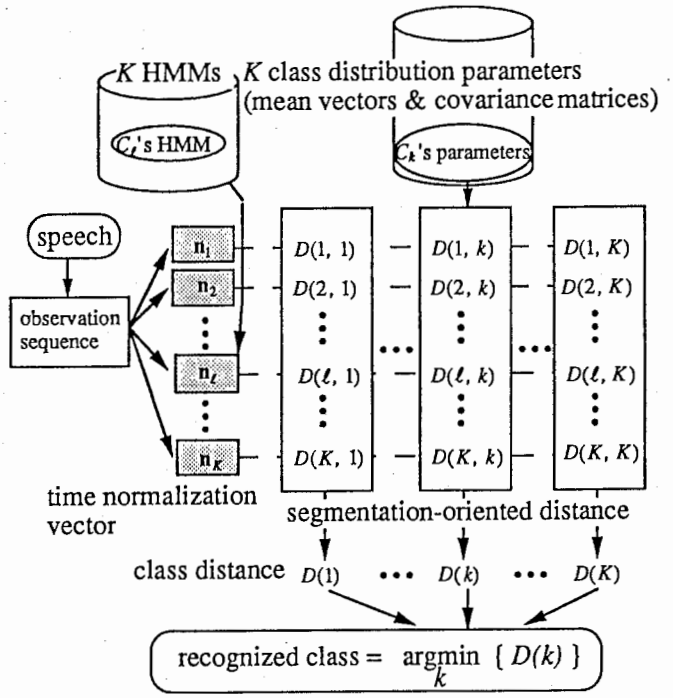

Figure 2. Recognition stage.

4.2 Results and Discussion

We first investigate the effect of several possible SO distances in computing the class distance. Figure 3 shows the relations between P, the number of SO distances, and the recognition rates obtained by the LVQ2-E hybrid systems. These results show that the usage of several probable SO distances helps improve recognition rate, especially in the case of using the small size classifier for the testing tokens. Two key points are worth mentioning: (a) As P increases, the recognition rate on testing data improves almost monotonically. A similar trend was also observed for the results on training data with the small size classifier ($r(k) = 1$). Moreover, the performance curves saturate beyond $P = 5$. (b) The large size classifiers obtain very high recognition rates on training data, regardless of P.

Next, we compare the results of the hybrid algorithms with that of the HMM system. The recognition rates of each system are plotted in Figure 4. The results show that the hybrid algorithm realized a significant increase in recognition rate on testing data for a broad range of classifier sizes. The HMM result (5-state, 5-mixture models) was 61.7% on testing data. The LVQ2-E hybrid system gave an increase of 13.7% in performance, and the LVQ2-L version achieved an additional improvement of 3.6%. The results of the LVQ2-L hybrid system suggest that the selection of a distance measure should be considered more carefully even in the LVQ2 framework. The recognition rate difference between the HMM system and the hybrid systems exceeded 17%. For the training data, the results of the hybrid systems were nearly perfect in most cases. However there were big drops in the recognition accuracy between training data and testing data, even in the hybrid systems. This problem, which is related to robustness in pattern classification and learning, requires more investigation.

We can also incorporate additional knowledge about the task into the recognition systems. For the E-set task, it is well known that tying of the vowel states improves HMM performance. We therefore designed a hybrid system using only the information from the beginning part of an utterance. In particular, we used the first three segments in creating the TN vector (a 72-dimensional vector). To evaluate this idea, we used the LVQ2-L hybrid system with 3, 5 or 7 distributions per class. Figure 5 shows the recognition results on testing data. The recognition rates were again increased. The highest rate achieved was 81.3%, which is the best performance reported on this database.

5. Conclusion

We have presented a new HMM/LVQ hybrid algorithm for speech recognition. We have also generalized LVQ2 algorithm to incorporate likelihood-based distance measures. The evaluation on the E-set task showed that the new algorithm greatly contributed towards increasing recognition rate. The difference between our best recognition rate and that of the conventional HMM was 20%.

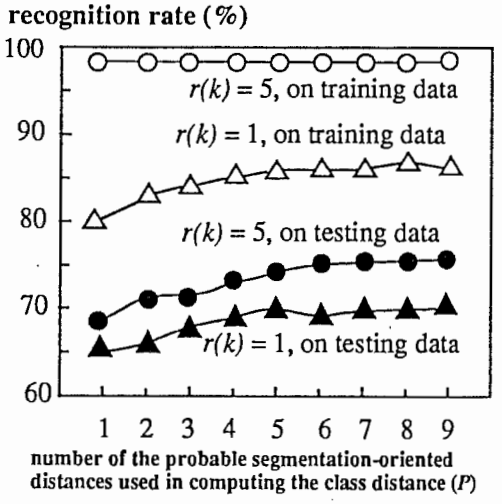

Figure 3. Recognition rates for different numbers of the probable segmentation-oriented distances.
The results were provided by the LVQ2-E systems. The numbers of distributions were identical for all the classes, i.e., $r(k)$ was constant for all k's. Two cases, $r(k) = 1$ and 5, are shown in this figure.

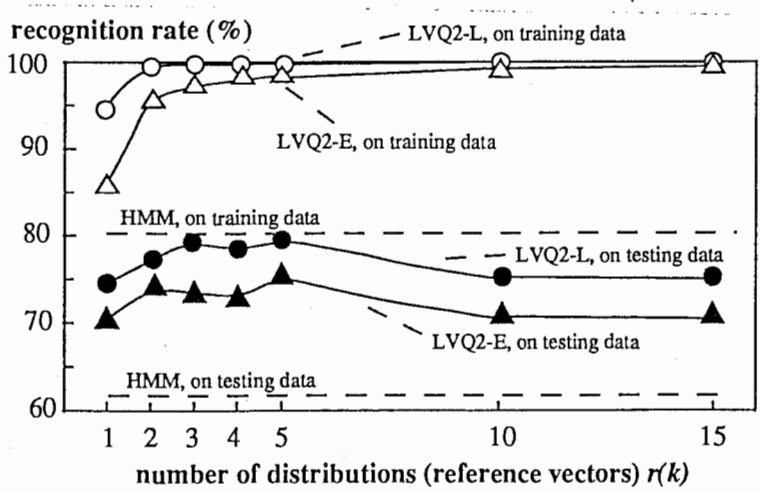

Figure 4. Recognition rates for different numbers of distributions.

The numbers of distributions were identical for all the classes, i.e., $r(k)$ was constant for all k's. The result by the LVQ2 hybrid system shows the highest value among the nine different cases of P (=1, 2, ..., 9).

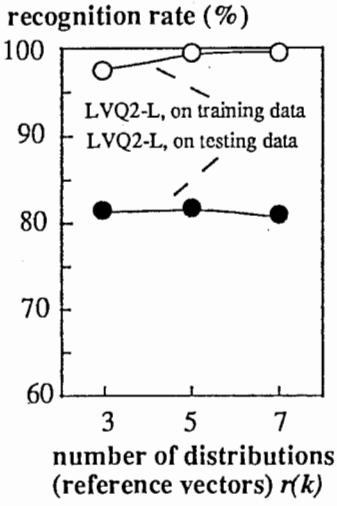

Figure 5. Recognition rates for the time normalized vector based on the first three segments.

The numbers of distributions were identical for all the classes, i.e., $r(k)$ was identical for all k's. The time normalized vector was 72-dimensional. The LVQ2-L hybrid systems were here adopted. Each result shows the highest value among nine different cases of P (=1, 2, ..., 9).

Acknowledgements

The authors would like to thank Dr. L. Rabiner and Dr. Y. Tohkura for arranging this collaboration research. The authors would also like to thank the members of Speech Research Department, AT&T Bell Laboratories, for many helpful comments and suggestions.

References

[1] S. Amari; "A theory of Adaptive Pattern Classifiers," IEEE Trans. on Electronic Computer, Vol. 16, No. 3, pp 299-307, June 1967.

[2] L. R. Rabiner and B.-H. Juang, "An Introduction to Hidden Markov Models," IEEE ASSP Magazine, Vol. 3, No. 1, January 1988.

[3] T. Kohonen; "Learning Vector Quantization for Pattern Recognition," Helsinki University of Technology, Report TKK-F-A601, November, 1986.

[4] T. Kohonen, G. Barna, and R. Chrisly; "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," IEEE, Proc. of ICNN, Vol.1, pp.61-68, July 1988.

[5] E. McDermott and S. Katagiri; "Shift-Invariant, Multi-Category Phoneme Recognition Using Kohonen's LVQ2," Proc. ICASSP89, Glasgow, UK, pp.81-84, May 1989.

[6] E. McDermott and S. Katagiri; "LVQ-Based Shift-Tolerant Phoneme Recognition", ATR, ATR Technical Report, TR-A-0059, August 1989.