A Discriminative Hidden Markov Model Recognizer with Neural Network Postprocessor

Sung-Bae Cho
ATR Human Information Processing Research Laboratories

***Abstract*— This paper is concerned with the problem of improving recognition accuracy of hidden Markov models (HMM) for sequential pattern recognition. It is argued that maximum-likelihood estimation of the HMM parameters via the forward-backward algorithm may not lead to values which maximize recognition accuracy. We introduce a hybrid method with neural network postprocessor which is aimed at minimizing the number of recognition errors. This method exploits the discrimination capability of neural network classifier while using HMM formalism to capture the dynamics of input patterns. Although it has not been proved that the presented method is a kind of maximum mutual information estimation, experimental results with on-line handwriting characters suggest that it leads to fewer recognition errors than can be obtained with the conventional recognition method.**

I. Introduction

Neural network classifiers have been recognized as a powerful tool for pattern classification problems [1]. Their properties are the discrimination power and the capability to learn and represent implicit knowledge, but they are generally used for classification of static patterns without sequential processing. Several researchers have proposed original architectures having feedback loops for providing dynamic and implicit memory [2, 3, 4]. However, current neural network topologies are inefficient in modeling temporal structures.

An alternative approach is to use hidden Markov models (HMM). HMM provides a good probabilistic representation of temporal sequences having large variations, and has been widely used for automatic speech recognition [5]. The main drawback of HMMs trained independently, however, is the weak discrimination power. The maximum likelihood (ML) estimation procedures typically used for training HMMs can be suitable to model the time sequential order and variability of input observation sequences, but the recognition task requires more powerful discrimination.

A solution is to train the HMMs with the maximum mutual information (MMI) criterion, which provides more discrimination but the mathematics are more complex; The standard ML optimization yields

$$\max_{\omega_i} P(X^i|\omega_i),$$

where X^i is a separate training sequence of observations and ω_i is a model, while MMI criterion should optimize

$$\max_{\omega} \left\{ \sum_{i=1}^{c} \left[\log P(X^i|\omega_i) - \log \sum_{j=1}^{c} P(X^i|\omega_j) \right] \right\}.$$

MMI training involves a number of practical difficulties. The Baum-Welch algorithm [5] is a robust and efficient algorithm for ML estimation, but it cannot be applied directly to MMI. As a result, the work on MMI training was forced to use slow and somewhat unreliable gradient descent methods [6]. To alleviate this problem, various attempts have been made to combine the classification power of neural networks with the temporal sequence modeling capability of HMM.

This paper is inspired by previous attempts for combining HMM with neural networks, and presents a method in which HMMs provide the neural network with input vectors of which the temporal variations are filtered through HMMs. This method makes use of the HMM formalism to capture the dynamics of input patterns, and then the neural network postprocessor to additionally classify the internal outputs of HMMs. The experimental results with the recognition problem of on-line handwriting characters confirm the usefulness of the presented method.

II. Hidden Markov models

An HMM can be thought of as a directed graph consisting of N nodes (states) and arcs (transitions)

The author is with ATR Human Information Processing Research Laboratories, 2-2 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-02, JAPAN. E-mail: sbcho@hip.atr.co.jp, Fax: +81 7749 5 1008.

0-7803-1901-X/94 $4.00 ©1994 IEEE

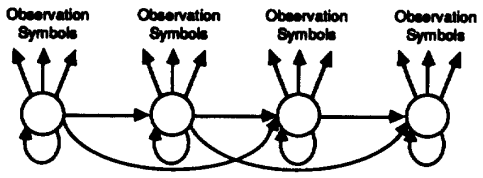

Figure 1: A left-right HMM allowing skip states.

representing the relationships between them. Each node stores the initial state probability, π_i, and the observation symbol probability distribution, $b_j(X_t) = P(X_t|q_t = j)$, and each arc contains the state transition probability distribution, $a_{ij} = P(q_{t+1} = j|q_t = i)$. Using these parameters, the observation sequence can be modeled by an underlying Markov chain of which state transitions are not directly observable. An example of such an HMM is shown in Fig. 1. Here, N is 4 and the model is constrained such that it is a so-called left-right model allowing skips of only one state.

The conventional HMM approach makes a Markov model for each class. Recognition with the HMMs of c classes involves accumulating scores for an unknown input across the nodes in each class model, and selecting that class model which provides the maximum accumulated score. Given a model, $\omega_i = (A, B, \pi)$, and an unknown input sequence, $X = X_1, X_2, \ldots, X_T$, the matching score, the probability of a sequence of observation X having been generated by a model ω_i, is given by

$$P(X|\omega_i) = \sum_{Q} \pi_{q_1} b_{q_1}(X_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(X_t),$$

where $Q = \{q_1, q_2, \ldots, q_T\}$. Then, we select the maximum as

$$i^* = \arg\max_i P(X|\omega_i), \quad 1 \le i \le c,$$

and classify the input data as class i^*.

For a given ω, an efficient method for computing the matching score, known as the forward-backward algorithm [5], is as follows:

- Initialization:

$$\alpha_1(i) = \pi_i b_i(X_1), \quad 1 \le i \le N.$$

- Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(X_{t+1}), \quad 1 \le t \le T-1.$$

Then, the matching score can be calculated by

$$P(X|\omega) = \sum_{i=1}^{N} \alpha_T(i). \tag{1}$$

To train an HMM means to adjust the parameters of the given model, ω, such that for an observation sequence X, $P(X|\omega)$ is maximized.

III. The Hybrid Method

The task in pattern classification problems is to assign an input vector, $X = (X_1, X_2, \ldots, X_T)$, to one of c classes $\{\omega_1, \omega_2, \ldots, \omega_c\}$. Bayesian classifiers perform this task by calculating the Bayesian probability, $P(\omega_i|X)$, for each class, and assigning the input to the class with the highest probability. The Bayesian probability $P(\omega_i|X)$ is expressed as follows:

$$P(\omega_i|X) = \frac{P(X|\omega_i)P(\omega_i)}{P(X)},$$

where $P(X|\omega_i)$ is the conditional probability of producing the input if the class is ω_i, $P(\omega_i)$ is the *a priori* probability of occurrence of the class, and $P(X)$ is the probability of the input. Since $P(X)$ is common to all classes, it is usually omitted and $P(X|\omega_i)P(\omega_i)$ is used for classification.

For a 1 of M problem, when network parameters are chosen to minimize a squared-error cost function, each output neuron of the neural network classifier estimates the Bayesian probability as a nonlinear equation with parameters $\{w_{kj}^{mi}, w_{ik}^{om}\}$:

$$P(\omega_i|X) \approx f\left\{\sum_{k=1}^{H} w_{ik}^{om} f\left(\sum_{j=1}^{T} w_{kj}^{mi} X_j\right)\right\},$$

where the w_{kj}^{mi} is a weight from the jth input neuron to the kth hidden neuron, w_{ik}^{om} is a weight from the kth hidden neuron to the ith class output, and f is a sigmoid function such as $f(x) = 1/(1+e^{-x})$.

Apart from the ignorance of *a priori* probabilities $P(\omega_i)$, it should be noted that the hidden nodes of a neural network classifier and the states of an HMM may not have specific physical meaning. They may just reflect some clustering properties of the observation data in the feature space. Thus, when the training data have some unfavorable distributions in the feature space, using each method directly may cause some problems. That is, in some case, the matching score of observation sequences generated by the correct model may be less than those generated by alternative models.

To overcome these shortcoming, many researchers have attempted to combine the advantages of the time-alignment function of HMM and the powerful discrimination capability of neural networks. Some researchers have shown that HMMs can be considered as a subset of recurrent neural networks, resulting in the use of several alternatives to the traditional HMM training algorithms. Bourlard *et al.*

have also proposed this kind of architecture, called discriminative HMMs, for utilizing the advantages of neural network classifier in the HMM framework [7]. Other researchers have used neural networks for pre-processing, one-unit-at-a-time, and/or for post-processing, to refine or integrate information at a static pattern level, leaving temporal processing to HMM [8, 9].

In this paper, we present an alternative hybrid method for classifying the nodal matching scores,—each $\alpha_T(k)$ in equation (1),—of all models by the feedforward neural network, instead of simply selecting the model generating the maximum accumulated score. The hybrid method takes the likelihood patterns inside hte HMMs and presents them to a neural network classifier to estimate posterior probabilities as follow:

$$P(\omega_i|X) \approx f\left\{\sum_{k=1}^{H} w_{ik}^{om} f\left(\sum_{j=1}^{T}\sum_{l=1}^{N} w_{kjl}^{mi}\,\alpha_T(j,l)\right)\right\}.$$

In this method, the HMM yields a kind of static patterns whose temporal variations have been processed, and the neural network classifier discriminates them as belonging to one particular class. It is hoped that the pattern of $\alpha_T(k)$'s for the classes can play an important role in classifying each sample to the correct class.

IV. Experiments

A. On-line handwriting character recognition

The successes of HMM in speech recognition area have stimulated many researchers to considerable research efforts of applying to the problems of handwriting script recognition [10, 11, 12]. The reason of this trend is that the rules about the interpretation of temporal patterns can be clearly specified by the HMM trained with examples in the data whether they be speech features or image features.

We have used a data set of handwriting characters as a source of training and test samples to give an idea of the practical application of the presented method in pattern recognition. An input character consists of a set of strokes, each of which begins with a pen-down movement and ends with pen-up movements. Several preprocessing algorithms were applied to successive data points in a stroke to reduce quantization noises and fluctuations of the writer's pen motion. The processes used are as follows: the wild point reduction, the dot reduction, the hook analysis, the three point smoothing, the peak preserving filtering, and the N point normalization. A sequence of preprocessed data points is approximated by a sequence of 8-directional straight-line segments which is the same as the chain code used by Freeman.

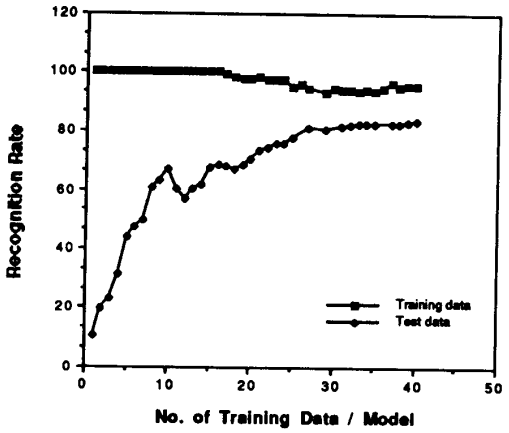

Figure 2: The recognition rates of the HMM with respect to the increase of the number of training samples.

For the HMM was used the left-right model in which no transitions are allowed to states whose indices are lower than that of the current state. It was composed of the ten nodes and the eight observation symbols in each node. The ten nodal matching scores of all models provided as inputs to the neural network classifier of the hybrid method.

B. Simulation results

In the simulation, handwriting characters were inputted to the computer (SUN workstation) by an LCD tablet of Photron FIOS-6440 which samples 80 dots per second. The tasks were to classify the Arabic numerals, the uppercase letters, and the lowercase letters which were collected from 13 writers. For training HMMs and neural network classifier, 40 examples for each class were used, while for recognition a further 500 examples were used as test inputs.

First of all, we tried to investigate the recognition rate of the HMM with different numbers of training samples, from one to 40 for each class. Fig. 2 shows that the recognition rates of classifying the total 500 test samples with respect to the various numbers of training samples. It is seen that the correct recognition rate tends to increase with the increase of the training samples. However, after the number of training samples exceeded 30, the increasing rate of the recognition performance was greatly reduced. This is a strong indication that the accuracy of the HMM is increased as using more training data, but the recognition rate arrives at the limit around the 40 training samples per model.

In order to apply the presented hybrid method for

Table 1: Number of recognition errors.

	HMM	Hybrid	Reduction
Numeral	82	73	11%
Uppercase	118	85	30%
Lowercase	140	116	17%
Total	340	274	19%

the numerals recognition, we implemented a two-layered multilayer perceptron which has 100 input nodes, 20 hidden nodes and 10 output nodes. The input was provided by the ten HMM models consisting of ten nodes. This network, however, did not converge because the nodal matching scores of the HMM were too small floating point numbers. Therefore, we encoded each output value of the HMM states as one of the ten values between zero and one: we assigned 1.0 if $0.1 < \alpha_T(i)$, 0.9 if $0.01 < \alpha_T(i) \leq 0.1$, and so on.

A comparative result for the methods are summarized in Table 1. The overall recognition rate for the ten classes with hybrid method is 85.40% on a total of 500 characters. This is a significant improvement over the performance obtained with HMMs trained with ML optimization (83.60% recognition rate). This improvement may be practically significant, but it is not impressive for a method which should give some net benefit by construction. However, the fact that similar (or bigger) improvements were obtained for upper and lower case letters provides evidence that this is a real effect. The hybrid method has a small, but statistically significant ($p > 0.995$), advantage in the recognition rates than the HMMs where the training sample size is 40. The statistical comparisons are based on a paired sample t test with 6 degrees of freedom.

V. Concluding remarks

In this paper, we present a hybrid method of HMMs and a neural network classifier to improve the recognition accuracy. From the results of the limited experiment of recognizing on-line handwriting characters, we have seen that the presented hybrid method has performed well despite some limitations on the coding scheme. We believe that with additional work on the encoding method not only of the neural network but also of the HMMs, there is potential for this hybrid method to be used for recognizing continuous script in much the same way it has been for continuous speech recognition.

Several works are remained for further research. For example, a hybrid system based on *alternative* HMM and neural network models may be more suitable for handwriting character recognition—the left-right HMM and the multilayer perceptron may not be the most appropriate models for this task.

Acknowledgments

The author would like to thank Dr. Katsunori Shimohara, Head of Department 6 at ATR HIP laboratories, for supporting this work.

References

[1] R.P. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Communications Magazine*, 47–64, November, 1989.

[2] M.L. Jordan, "Serial order: a parallel distributed processing approach," Univ. California, Davis, *Tech. Rep. 8604*, 1986.

[3] J.L. Elman, "Finding structure in time," Univ. California, San Diego, *CRL Tech. Rep. 8801*, 1988.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay networks," in *Proc. ICASSP-88*, New York, 1988.

[5] L.R. Rabiner, "A Tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, **77** (2), 257–286, 1989.

[6] L.R. Bahl, P.F. Brown, P.V. de Souza and R.L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP'86*, 49–52, April, 1986.

[7] H. Bourlard and N. Morgan, "Merging multilayer perceptrons and hidden Markov models: some experiments in continuous speech recognition," *Neural Network Advances and Applications*, 215–240, 1991.

[8] R.P. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, **1**, 1–38, 1989.

[9] H. Bengio, R.D. Mori, G. Flammia and R. Kompe, "Global optimization of a neural network - hidden Markov model hybrid," in *Proc. IJCNN-91*, **II**, 789–794, Seatle, 1991.

[10] R. Nag, K.H. Wong and F. Fallside, "Script recognition using hidden Markov models," in *Proc. ICASSP-86*, 2071–2074, Tokyo, April, 1986.

[11] A. Kundu and P. Bahl, "Recognition of handwritten script: a hidden Markov model based approach," in *Proc. ICASSP-88*, 928–931, 1988.

[12] C.C. Tappert, "Online handwriting recognition with hidden Markov models," in *Proc. Fifth Handwrit. Conf. Int. Graphonomics Soc.*, 204–206, October, 1991.