3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

ON THE APPLICATION OF MULTIPLE TRANSITION BRANCH HIDDEN MARKOV MODELS TO CHINESE DIGIT RECOGNITION

Xixian Chen, Yinong Li, Xiaoming Ma, and Lie Zhang

Beijing University of Posts and Telecommunications
Beijing, 100088, P.R.China

ABSTRACT

In this paper we propose a multiple branch hidden Markov model(MBHMM) which is different from the conventional ones. In the basic HMMs, there is only one transition branch from one state to another one. Our new model has multiple transition branches between two states. As a result, it can hold much more spectral information in the speech signal than the basic HMMs. The evaluation, decoding, and training algorithms associated with MBHMM are also derived. The resulting recognizer is tested on a vocabulary of ten Chinese digits over 20 speakers. The recognition results show that MBHMM significantly outperforms the conventional discrete HMM(DHMM).

I. INTRODUCTION

The theory of hidden Markov models(HMM) is well established and has been successfully applied to many state-of-art speech recognition systems[1],[2]. Fig.1 illustrates a basic left-to-right HMM with four states. Each directed line is a transition from one state to another state, whose probability is indicated by the number alongside the line. A probability distribution is associated with each state, which governs the observation output symbols. In this paper we propose a multiple branch hidden Markov model(MBHMM) which is different from the conventional ones. In the basic HMMs[1]-[3], there is only one transition branch from one state to another one. Our new model has multiple transition branches between two states. Such a model (left-to-right MBHMM) is shown in Fig.2. Here, the number of states is four, and there are two branches between the states. Since it has multiple transition branches from one state to another one, MBHMM can hold much more spectral information in the speech signal than the old ones. This property is very useful for multi-speaker or speaker-independent word recognition.

This work was supported by the National Natural Science Foundation of China

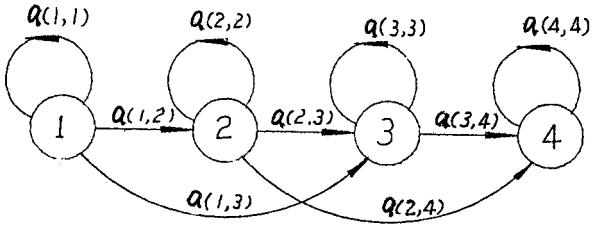

Fig.1 A basic left-to-right hidden Markov model with four states.

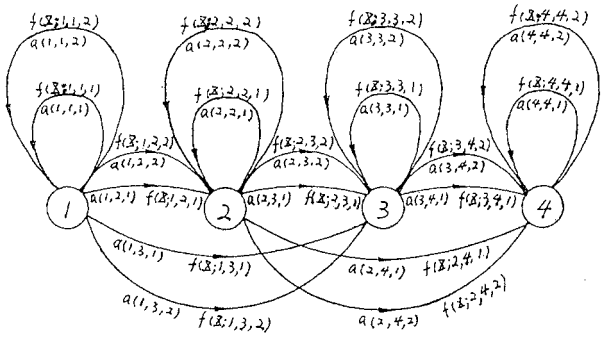

Fig.2 A four-state multiple branch hidden Markov model showing transition and output probabilities. There are two transition branches between the states.

The MBHMM is a double stochastic process of a Markov chain in which the unobservable state transitions are governed by a state transition array, and where each transition branch between the states is associated with an output probability function. To describe MBHMM formally, the following model notation for a MBHMM can be used.

T = length of the observation sequence, O(1),O(2),...,O(T).
N = number of states in the model.
S={s(0),s(1),...,s(T)}, the state sequence, where s(t)∈{1,2,...,N}. For simplicity, state i at time t may be denoted by s(t)=i when ambiguity does not exist.
P = {p(1),p(2),...,p(T)}, the transition

branch sequence, where $p(t)\in\{1,2,\ldots,K(i,j)$, $1\le i,j\le N\}$. $K(i,j)$ denotes the number of transition branches from state i to state j.
$A=\{a(i,j,k),\ 1\le i,j\le N, 1\le k\le K(i,j)\}$, the transition probability array, where $a(i,j,k)=Pr\{s(t)=j,p(t)=k/s(t-1)=i\}$ denotes the transition probability from state i to state j through branch k at time t.
$U=\{u(i),\ 1\le i\le N\}$, the initial probability vector, where $u(i)=Pr\{s(0)=i\}$ denotes the probability that the initial state s(0) is at state i at time t=0. Note that for the parameters to be true probabilities

$$\sum_{j=1}^{N}\sum_{k=1}^{K(i,j)} a(i,j,k) = 1 \quad \text{for any } i, \qquad \sum_{i=1}^{N} u(i) = 1$$

$F=\{f(X;i,j,K),1\le i,j\le N,1\le k\le K(i,j)\}$, the output probability density functions, where $f(X;i,j,K)=f[X/s(t-1)=i,s(t)=j,p(t)=k]$ denotes the probability density function that for a state transition from i to j through branch k at time t, the observation X is produced.

A MBHMM can be represented by using the compact notation M=(A,F,U). Given the definition of MBHMM, we can deduce several probabilistic functions as follows:

1) The joint probability of state sequence S and corresponding branch sequence P being generated by the MBHMM is

$$\begin{aligned} Pr(S,P/M) &= Pr\{s(0),s(1),\ldots,s(T);p(1),p(2),\ldots,p(T)/M\} \\ &= u[s(0)]\prod_{t=1}^{T} a[s(t-1),s(t),p(t)] \end{aligned} \quad (1)$$

2) Given a specific state sequence S and its corresponding branch sequence P generated by the Markov chain, the conditional likelihood of observing sequence $O=O(1),O(2),\ldots,O(T)$ is

$$f(O/S,P,M)=$$

$$f[O(1),\ldots,O(T)/s(0),\ldots,s(T);p(1),\ldots,p(T),M]$$

$$=\prod_{t=1}^{T} f[Ot/s(t-1),s(t),p(t),M] \quad (2)$$

3) The joint likelihood of O, S and P is simply the product of the above two terms:

$$f(O,S,P/M) = f(O/S,P,M)Pr(S,P/M) \quad (3)$$

4) THe likelihood of observing O, f(O/M), is the summation of (3) over all possible state and branch sequences:

$$f(O/M) = \sum_{S,P} f(O,S,P/M) \quad (4)$$

In the following two sections, we will discuss the evaluation, decoding, and training algorithms for MBHMM using the above probability functions.

II. COMPUTING THE LIKELIHOODS

The enumerative evaluation of f(O/M), by using (4), is a computational complex task, which is a sum of all possible state and branch sequences of length T. In a manner similar to the basic HMMs[1]-[3], we can derive a more efficient algorithm called the forward-backward algorithm to solve this problem. Define the forward likelihoods $\alpha(t,i)$ for i=1,2,...,N and t=1,2,...,T

$$\alpha(t,i)= f[O(1),O(2),\ldots,O(T);s(t)=i/M] \quad (5)$$

then the forward likelihoods can be calculated inductively for t=0,1,...,T-1

$$\alpha(t+1,j)=\sum_{i=1}^{N}\sum_{k=1}^{K(i,j)} f[O(t+1)/i,j,k]a(i,j,k)\alpha(t,i) \quad (6)$$

with initial values $\alpha(0,i)$ defined as

$$\alpha(0,i)= u(i) \quad \text{for } i=1,2,\ldots,N \quad (7)$$

Similarly, define the backward likelihoods $\beta(t,i)$ for i=1,2,...,N and t=0,1,...,T-1

$$\beta(t,i)=f[O(t+1),\ldots,O(T)/s(t)=i,M] \quad (8)$$

Then we have the following recursion for t=T-1, T-2,...,0

$$\beta(t,i)=\sum_{j=1}^{N}\sum_{k=1}^{K(i,j)} f[O(t+1)/i,j,k]a(i,j,k)\beta(t+1,j) \quad (9)$$

with initial values $\beta(T,i)$ defined as

$$\beta(T,i)= 1 \quad \text{for } i=1,2,\ldots,N \quad (10)$$

Either of the forward and backward algorithms can be used to evaluate f(O/M)

$$f(O/M)=\sum_{i=1}^{N}\alpha(T,i)=\sum_{i=1}^{N}\beta(0,i)u(i) \quad (11)$$

They can also be used together to formulate a solution to the problem of model parameters' estimation.

Given the observation sequence O, what are the most likely state sequence S=s(0),s(1),...,s(T) and corresponding branch sequence P=p(1),p(2),...,p(T)? One possible optimal criterion is to chose S and P, which are in the best path with the highest probability Pr(S,P/O,M), or equivalently, with maximum f(O,S,P/M). A formal technique for finding the best state and branch sequences as well as the maximum of f(O,S,P/M) is called the Viterbi algorithm, which is described as follows:

1) Let $d(0,i)=u(i)$ for $i=1,2,\ldots,N$

2) Perform the following recursion for $t=1,2,\ldots,T$ and all states j

$$d(t,j)=\max_{i,k}\{d(t-1,i)f[O(t)/i,j,k]a(i,j,k)\}$$
$$=d(t-1,i^*)f[O(t)/i^*,j,k^*]a(i^*,j,k^*)$$

and

$$u(t,j)=i^*, \quad v(t,j)=k^*$$

where i^* and k^* are the optimum choice of the indexes i and k, respectively.

3) Obtain the optimum result of $f(O,S,P/M)$

$$\max_{S,P}\{f(O,S,P/M)\}=\max_{i}\{d(T,i)\}=d(T,i^*)$$

and $s(T)^*=i^*$

4)Backtrack the optimum state and branch sequences indicated by *

$$s(t)^* = u[t+1,s(t+1)^*] \quad \text{for } t=T-1,T-2,\ldots,0$$
$$p(t)^* = v[t,s(t)^*] \quad \text{for } t=T,T-1,\ldots,1$$

Both the forward-backward procedure and Viterbi algorithm can be used for the word recognition task. The former may work more robustly than the later. This is achieved for an increase in computation complexity, for $f(O/M)$ is the summation of $f(O,S,P/M)$ over all possible state and branch sequences. While the Viterbi algorithm only efficiently finds the maximum of $f(O,S,P/M)$ over all S and P. It is more efficient since it can be implemented in the logarithm domain using only additions. Also it is possible to obtain the optimum state and branch sequences at the same time.

III. TRAINING THE MBHMM

The purpose of training the MBHMM is to find an optimum model $M=\{A,F,U\}$ that maximizes the likelihood $f(O/M)$. Unfortunately, such a solution has not been found so far. We can, however, use the Baum-Welth re-estimation algorithm to find a local maximum of $f(O/M)$. Given a set of observation sequences O and an arbitrary model M, the Baum-Welth re-estimation algorithm iteratively finds another model $M'=\{A',F',U'\}$ that leads to $f(O/M')\geq f(O/M)$. The algorithm continually improves the estimates and converges to a local maximum. As an alternative, we will discuss the training procedure based on viterbi algorithm, which is more efficient than the former. Let $O=\{O^1,O^2,\ldots,O^L\}$ denote a training set of L independent observation sequences which may be derived from L repetition of utterances for a certain word in a recognition vocabulary, where $O^l=O^l(1),O^l(2),\ldots,O^l(T^l)$ is the lth training sequence with length of T^l. Now, the objective in training the model becomes the maximization of the joint likelihood

$$\prod_{l=1}^{L}\max_{S_l,P_l}\{f[O^l,S_l,P_l/M]\} \quad (12)$$

The first step in the training procedure is to choose the initial model parameters. These initial estimates can be chosen randomly, or on the basis of any good initial guess. The second step in the training procedure is to segment each training sequence, O^l, into branches based on the current model, M. This segmentation is achieved by finding the optimum state and branch sequences.

Let $S_l^*=s_l(0)^*,s_l(1)^*,\ldots,s_l(T^l)^*$ and $P_l^*=p_l(1)^*,p_l(2)^*,\ldots,p_l(T^l)^*$ denote the optimum state and branch sequences for the lth utterance. Define $r(l,t,i)=\delta[s_l(t)^*-i]$ and $r(l,t,i,j,k)=\delta[s_l(t)^*-i]\delta[s_l(t+1)^*-j]\delta[p_l(t+1)^*-k]$, where $\delta(t)$ is the delta function:

$$\delta(t)=\begin{cases}1\\0\end{cases} \quad (13)$$

The new estimates of the initial state probability u(i) and the transition probability a(i,j,k) may be taken as

$$u'(i)=\sum_{l=1}^{L}r(l,0,i)/L \quad (14)$$

and

$$a'(i,j,k)=N(i,j,k)/N(i) \quad (15)$$

where

$$N(i)=\sum_{l=1}^{L}\sum_{t=0}^{T^l-1}r(l,t,i) \quad (16)$$

is the total number of transitions out of state i, and

$$N(i,j,k)=\sum_{l=1}^{L}\sum_{t=0}^{T^l-1}r(l,t,i,j,k) \quad (17)$$

is the total number of transitions from state i to state j through branch k.

To re-estimate the probability density function $f(X;i,j,k)$, we suppose that $f(X;i,j,k)$ is the multivariate Gaussian density with mean vector $V(i,j,k)$ and covariance matrix $R(i,j,k)$. The improved estimates of $V(i,j,k)$ and $R(i,j,k)$ can be obtained from (18) and (19), respectively.

$$V'(i,j,k)=\sum_{l=1}^{L}\sum_{t=0}^{T^l-1}r(l,t,i,j,k)O^l(t+1)/N(i,j,k) \quad (18)$$

$R'(i,j,k)$

$$=\sum_{l=1}^{L}\sum_{t=0}^{T^l-1} r(l,t,i,j,k)\,|E[O^l(t+1),i,j,k]|^2/N(i,j,k) \quad (19)$$

$$|E[O^l(t+1),i,j,k]|^2$$

$$=[O^l(t+1)-V(i,j,k)][O^l(t+1)-V(i,j,k)]^T$$

where the superscript T denotes the transpose of a vector or matrix. It can be proved that the new, improved estimates obtained from (14), (15), (18), and (19) satisfy

$$\prod_{l=1}^{L} \max_{S_l,P_l}\{f[O^l,S_l,P_l/M']\geq \prod_{l=1}^{L} \max_{S_l,P_l}\{f[O^l,S_l,P_l/M]$$

IV. EXPERIMENTAL RESULTS

The new recognizer based on MBHMM was tested on a vocabulary of ten Chinese digits over 20 speakers(10 males and 10 females). Each speaker voiced 10 utterances/word, and there are 200 utterances for each digit. The first 100 utterances were used as the training data to create the averaged templates and the rest of them were used as the test data to perform the recognition.

The speech is sampled at a rate of 8kHz and divided into frames of 20ms(160 samples) with each consecutive frame spaced 10ms(80 samples). The basic feature vector used for modeling the speech frame is composed of the 12th-order LPC based cepstral coefficients. A left-to-right MBHMM with one skipping, shown in Fig.2, was constructed for each digit. The output probability density, f(X;i,j,k), associated with each branch was selected to be the multivariate Gaussian density with diagonal covariance matrix. We further assumed that f(X;i,j,k) was only related to the current state j and the transition branches towards it. It had no relation with the previous state i, that is, f(X;i,j,k)= f(X;j,k). The number of transition branches from state i to state j was set to be equal, K(i,j)=K.

Both the closed and open tests were performed for six sizes of MBHMMs: MBHMM(5,5) with N=5 and K=5 (5 states and 5 branches), MBHMM(5,10) with N=5 and K=10, MBHMM(8,5) with N=8 and K=5, MBHMM(8,10) with N=8 and K=10, MBHMM(10,5) with N=10 and K=5, and MBHMM(10,10) with N=10 and K=10. The recognition error rates for all the ten digits are shown in Table 1. The results show that in the closed test, the zero error rate is achieved for MBHMM(10,10), while in the open test, the highest recognition accuracy (99.3%) is achieved for MBHMM(8,10) and MBHMM(10,5). For performance comparison, the same vocabulary was also tested using the conventional discrete HMM(DHMM)[3] with 8 states and codebook size of 128. The recognition error rate is also given in table 1 and denoted as DHMM(8,128). It showed that MBHMM significantly outperformed the DHHM.

Table 1. Recognition error rates of MBHMMs and DHMM for ten Chinese digits.

Model	Closed Test	Open Test
MBHMM(5,5)	0.5%	1.1%
MBHMM(5,10)	0.1%	1.2%
MBHMM(8,5)	0.2%	1.6%
MBHMM(8,10)	0.2%	0.7%
MBHMM(10,5)	0.1%	0.7%
MBHMM(10,10)	0%	1.1%
DHMM(8,128)	2%	7.6%

REFERENCES

[1] K.F.Lee, et al, "SPHINX system for large vocabulary, speaker-independent, continuous speech recognition," IEEE Trans. Acoust., Speech & Signal Processing, vol.ASSP-38, no.1, pp. 35-45, January 1990.

[2] L. R. Rabiner, J. G. Wilpon, and F.K. Soong, "High performance connected digit recognition using hidden Markov models," IEEE Trans. Acoust., Speech & Signal Processing, vol.ASSP-37, no.8, pp. 1214-1225, August 1989.

[3] L.R. Rabiner, S.E. Levinson, and M.M.Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," The Bell System Technical Journal, vol.62, no.4, pp.1075-1105, April 1983.