Recent Developments in the Application of Hidden Markov Models to Speaker-Independent Isolated Word Recognition

B. H. Juang
L. R. Rabiner
S. E. Levinson
M. M. Sondhi

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT. In this paper we extend previous work on isolated word recognition based on hidden Markov models by replacing the discrete symbol representation of the speech signal by a continuous Gaussian *mixture* density. In this manner the inherent quantization error introduced by the discrete representation is essentially eliminated. The resulting recognizer was tested on a vocabulary of the 10 digits across a wide range of talkers and test conditions, and shown to have an error rate at least comparable to that of the best template recognizers and significantly lower than that of the discrete symbol hidden Markov model system. Several issues involved in the training of the continuous density models and in the implementation of the recognizer are discussed.

I. Introduction

Template based approaches with dynamic programming (DP) have been demonstrated to be one of the most effective methods for isolated word recognition. Several alternative approaches, however, have been proposed, due to:

1. the high computational cost of the DP approach;
2. the difficulties in extending the DP recognition paradigm to more difficult problems — e.g., continuous speech;
3. the desire to use a parametric model to represent the speech, rather than the non-parametric template;
4. the desire to use speech units smaller than words — e.g., syllables, demisyllables, phonemes.

These alternative approaches include using vector quantization (VQ) in the DP computation, using word-based vector quantization to eliminate the DP processing, using VQ as a front end preprocessor, and using hidden Markov models (HMM's) to represent the speech signal. Among these, the HMM recognizer is of great interest because of its potential low cost and its capability of modelling various events (phonemes, syllables, etc.) in the speech signal with efficient parametric representations. In our previous work [1], we studied how to apply discrete observation (i.e., vector quantized LPC vectors from a fixed size codebook) HMM's in isolated word, speaker independent speech recognition applications over dialed-up telephone lines. Work performed at IBM [2], CMU [3], and more recently at Phillips [4] has used continuous HMM's where it was assumed that all parameters of interest had Gaussian distributions. The HMM's to be discussed in this paper are based on continuous, *mixture* density models of the distribution of LPC derived parameter vectors. We have devised training procedures for obtaining maximum likelihood estimates of the parameters of the mixture distribution and applied the models to the problem of recognizing isolated digits. Our results show that the average error rate of such HMM recognizers are essentially identical to if not better than that of the best template approaches using DP methods, and considerably lower than that of an HMM recognizer with a discrete symbol VQ front end.

II. The Continuous Mixture Density HMM

Figure 1 shows the type of HMM we are considering here. It is based upon a left-to-right Markov chain which starts in state 1 and ends in state N. The observed signal is assumed to be a stochastic

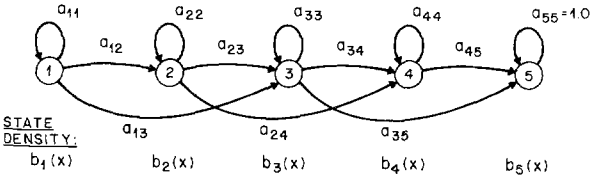

Fig. 1 Representation of left-to-right hidden Markov model.

function of the state sequence of the Markov chain. The state sequence itself is unobservable (hidden). The goal is to choose the parameters of the hidden Markov model to optimally match the observed characteristics of a given signal.

The parameters which characterize the HMM of Figure 1 are the following: 1) N, the number of states in the model; 2) $A = [a_{ij}]$, $1 \leqslant i, j \leqslant N$, the state transition matrix where a_{ij} is the probability of making a transition from state i to state j; and 3) B, the observation probability function. If we assume that the signal to be represented by the HMM consists of a sequence of observation vectors $\mathbf{O} = \{O_1, O_2, \ldots, O_T\}$, where each O_t is a vector which characterizes the signal at time t, then we can consider two types of observation probability functions, namely discrete and continuous. Discrete type HMM's have been extensively discussed in [1]. In the continuous case we have the probability density function $B = \{b_j(\mathbf{x})\}$, $1 \leqslant j \leqslant N$, where $b_j(\mathbf{x})d\mathbf{x}$ is the probability that the vector O_t lies between $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}$. The types of density functions allowed for $b_j(\mathbf{x})$, for which a reestimation algorithm exists, include strictly log concave densities [5], elliptically symmetric densities [6], and more recently mixtures of strictly log concave or elliptically symmetric densities [7]. In this paper we will consider Gaussian mixture densities of the form

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} C_{jk}\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}) \qquad (1)$$

where $\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{U})$ denotes a D-dimensional normal density function of mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{U}$.

To summarize the discussion above, a complete specification of a continuous mixture density HMM requires choosing values (parameter estimates) for the following: 1) N — number of states in the model; 2) M — number of mixtures; 3) D — number of dimensions in each vector; 4) $A = [a_{ij}]$ — state transition matrix; 5) $C = [c_{jk}]$ — mixture gain matrix; 6) $\boldsymbol{\mu} = [\mu_{jkd}]$ — means of the mixture components; and 7) $\mathbf{U} = [U_{jkde}]$ — covariance matrices of the mixture components. For the work to be presented here, we have chosen $N = 5$ states on the basis of previous studies with discrete symbol models [1]. Also our signal observation vectors are cepstral coefficient vectors derived from an 8th order LPC analysis of the speech signal.

2.1 Training the HMM

For each word, v, in a vocabulary of V words ($V = 10$ for the digits), an HMM is designed; i.e., the set of parameters above is estimated from a training set of data representing multiple occurrences of the vocabulary word by multiple talkers. The procedure for obtaining model parameter estimates is shown in Figure 2. We assume a training set of data consisting of Q sequences of observations, where each sequence, $\mathbf{O}^i = \{O_1^i, O_2^i, \ldots, O_{T_i}^i\}$, $1 \leqslant i \leqslant Q$ is the set of

CH2118-8/85/0000-0009 $1.00 © 1985 IEEE

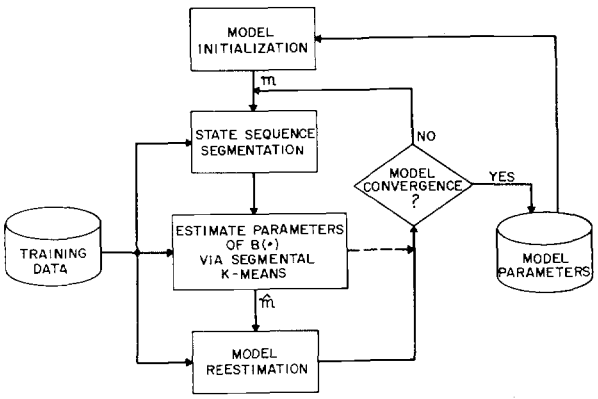

Fig. 2 The training procedure used to estimate parameter values.

vectors (observations) constituting a single occurrence of the word. An initial model estimate is assumed. This initial estimate (unlike the one required for reestimation) can be chosen randomly, or based upon any prior knowledge.

The second step in the training procedure is to segment each word occurrence, $\mathbf{O}^i$, into states based on the current model. This segmentation is achieved by finding the optimum state sequence via the Viterbi algorithm. The result of segmenting each of the Q training sequences is, for each of the N states, a set of the observations that occur within each state according to the current model. This step, in effect, isolates the observation stochastic processes from the underlying Markov chain so that the initial estimate of the observation statistics is not interfered by the Markov chain estimate. To illustrate this and the need for mixture densities, we compare the marginal distribution $b_j(\mathbf{x})|_{\mathbf{x}=\{\cdots x_n \cdots\}}$ of a 5 term Gaussian mixture distribution with diagonal covariances against a histogram of the actual observations in the corresponding state in Figure 3 for a 9 dimensional representation. The need for values of $M > 1$ is seen in the histogram of the 1st, 2nd, 4th, and the 8th parameters.

Following the above segmentation, a segmental K-means procedure is used to cluster the vectors in each state into a set of M clusters (using a Euclidean distortion metric and a VQ design algorithm). From the clustering, an updated set of model parameters is derived as follows: 1) $\hat{c}_{jk}$ = number of vectors classified in cluster k of the jth state/number of vectors in state j; 2) $\hat{\mu}_{jkd}$ = dth component of mean

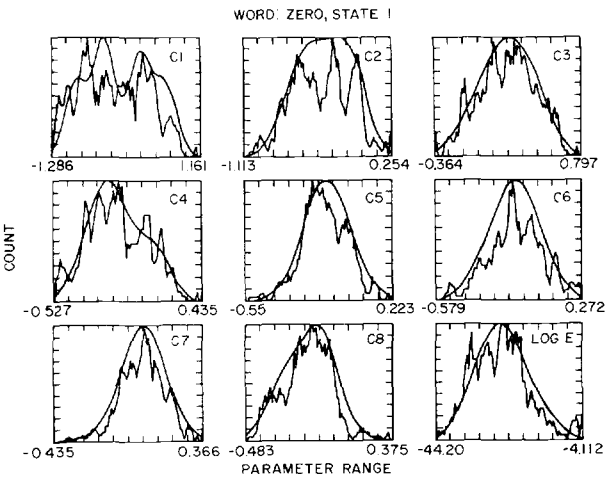

Fig. 3 Comparison of estimated density (jagged contour) and model density (smooth contour) for each of the 9 components of the representation vector.

of vectors classified in cluster k of state j; and 3) $\hat{U}_{jkrs}$ = (r, s)th component of covariance matrix of vectors classified in cluster k of state j. The transition matrix coefficients, a_{ij}, are not changed according to this procedure since no information about state transitions is retained. At this point the formal reestimation procedure is used to reestimate optimal values (in a maximum likelihood sense) of *all* model parameters. The resulting model is then compared to the previous model by computing a distance score which reflects the statistical similarity of the HMM's. If the model distance score exceeds a threshold, the model is updated and the overall training loop is repeated. The iteration stops when the distance falls below a prechosen threshold.

Since the steps of segmenting the training sequences into states, and clustering the vectors via a VQ clustering procedure are relatively inexpensive (in a computational sense), whereas reestimation is an exceedingly costly procedure, a practical implementation of the training procedure of Figure 2 is to bypass the step of model reestimation until local model convergence is obtained, and then apply the reestimation procedure at the final step.

Aside from using the reestimation algorithm to achieve maximum likelihood, we also developed a parameter estimation procedure based upon histogram fitting (for each observation state). The results obtained from this histogram fitting algorithm are comparable to those from the maximum likelihood estimate.

2.2 The HMM Recognizer

Once the HMM's have been trained on each vocabulary word, the recognition strategy is straightforward. Figure 4 shows a block diagram of the recognizer. The speech signal, $s(n)$, for the unknown word is first analyzed using an 8th order LPC analysis. The speech sampling rate is 6.67 kHz, and overlapping sections of 45 msec of speech are analyzed every 15 msec to give a set of 8 LPC coefficients. An LPC transformation algorithm is used to convert the LPC representation to an LPC derived cepstrum. Then, for each vocabulary word model, the optimum state sequence is found via the Viterbi algorithm and the log likelihood score for the optimal path is computed. The decision rule assigns the unknown word to the vocabulary word whose model has the highest log likelihood score.

2.3 Incorporation of Duration into the Recognizer

Inherently, each state in the HMM has an exponential duration probability. A state j, with a probability a_{jj} of returning to itself, has a state duration probability of

$$p_j(\ell) = (1-a_{jj})a_{jj}^{\ell-1}$$

where ℓ is the number of frames occurring in state j. To better exploit the state sequence information in the recognizer, we considered two alternatives, namely modification of the scoring procedure to include an internal durational model, and application of a post-processing durational model on the state sequence as determined by the Viterbi algorithm. In either case, in the training phase, we estimate a state duration probability of the form

$p_j(\ell/T)$ = probability of being in state j for (ℓ/T) of the word, where T is the number of frames in the word and ℓ is the number of frames in state j.

For each word and each state, the quantity $p_j(\ell/T)$ is estimated for 25 values of ℓ/T from 0 to 1.

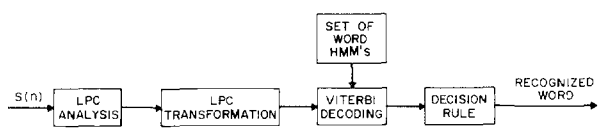

Fig. 4 Block diagram of the HMM recognizer.

For scoring a given observation sequence using the internal durational model, a recursion of the Viterbi procedure is required. The recursion is considerably more costly than the implementation of the standard Viterbi scheme.

The post-processor durational model, on the other hand, uses the original Viterbi alignment procedure. Then for each word, the optimal state sequence is determined, and the duration of each state is obtained via a backtracking procedure. The log likelihood is then augmented by the log duration probabilities (suitably weighted) to give the final score for the recognition decision, as shown in Eq. (2):

$$\log \hat{f} = \log f + \alpha \sum_{j=1}^{N} \log (p_j(\ell_j/T)) \ . \qquad (2)$$

III. Experimental Evaluations

Four sets of spoken digits recorded over standard dial-up telephone lines, were used to evaluate the performance of the HMM recognizer with the mixture density representation. These consisted of the following:

DIG 1 — 100 talkers (50 male, 50 female), 1 replication of each digit by each talker.

DIG 2 — Same 100 talkers and recording conditions as DIG 1; recordings made several weeks later than those of DIG 1.

DIG 3 — 100 new talkers (50 male, 50 female), 1 averaged occurrence of each digit by each talker obtained from averaging a pair of robust tokens of the digit. The transmission conditions (i.e., analog front end, filter cutoff frequencies, etc.) differed slightly from those used in recording the DIG 1 and DIG 2 databases.

DIG 4 — A second group of 100 new talkers (50 male, 50 female), 20 recordings of each digit by each talker. A random sampling of 1 of the recordings of each digit by each talker was used. The transmission conditions again differed somewhat from those used in recording the other sets.

For model training, only one digit set (either DIGI or DIG4) was used; for testing and performance evaluation, each of the 4 sets was used.

3.1 Diagonal Versus Full Covariance Matrices

Two forms for the U matrices of Eq. (1) were considered, namely diagonal matrices (with assumed zero correlation between components of the representation), and full covariance matrices. Recognition tests with diagonal covariance matrices using $M = 1, 3$ and 5, and full covariance matrices using $M = 1$ only were performed. The results showed that performance with the full covariance matrix with $M = 1$ was better than that obtained using only the diagonal covariance matrix with $M = 1$ or 3; however for $M = 5$ the performance with the diagonal matrix was comparable to that of the full covariance matrix with $M = 1$. Both full covariance and diagonal covariance matrices were used in subsequent recognition tests.

3.2 Applicability of Word Clustering to Model Generation

We also considered combining the word clustering procedure with model estimation, to give more than one HMM per word. We tested this idea in the following way. First a single HMM per word was created on training set DIG 1; next the 2 cluster per word template set was used to segment the 100 token training set into two groups. For each group a single HMM was created; hence a total of 2 HMM's per word was used in the performance evaluation. The potential disadvantage of this procedure should be clear, i.e., the training data per model available for estimating HMM parameters is half that used for the single model case and hence the estimates may be less reliable. Results of experiments with each of the 4 test sets and with 1 and 2 models per word showed that in the diagonal covariance case, a performance improvement of 0.65% was realized using 2 models per digit, and in the full covariance case the improvement was 0.25%.

3.3 Effects on Different Number of Mixtures

Using models trained on DIG1, the number of mixtures, M, was varied from 1 to 7, in steps of 2, for the diagonal covariance case, and from 1 to 2 for the full covariance case to see the effects on recognition performance. The results of these tests on the 4 digit data bases are given in Table I. The results show an improvement in performance from an average digit error rate of 2.95% for $M = 1$ down to an average digit error rate of 1.95% for $M = 5$. Results for $M = 7$ show a slight increase in average digit error rate to 2.23%. This result seems to indicate that the improvement in modelling the statistics of the "limited" observations from using more mixture terms is offset by the accompanying effect of broadened fitting range that helps incorrect words during recognition.

For the full covariance case the effect of increasing M from 1 to 2 is an increase in digit error rate by 0.95%. Hence we again see the evidence of sparser training data that makes model estimates unreliable and not robust for open recognition tests over a wide range of signal conditions.

3.4 Effects of Energy and Duration

To study the effects of including energy in the signal representation, and of including the durational model in the testing, a series of recognition runs were made using the diagonal covariance matrix models with $M = 5$, using 2 models per word. The results of these recognition tests are given in Table II. The durational model was implemented as a post-processor computation in all cases. The results show clearly that the addition of either energy or duration uniformly improves the performance of the HMM recognizer. Furthermore the combination of both energy and durational model yields better performance than either factor individually. The biggest improvements in performance were obtained for test sets DIG 3 and DIG 4 where the transmission characteristics of the speech were different from those of DIG 1 and DIG 2. In these cases the addition of energy and duration model make the system more robust because these features are, for the most part, insensitive to differences in transmission conditions.

Covariance Matrix	M	Average Digit Error Rate (%)				
		DIG 1	DIG 2	DIG 3	DIG 4	Overall
Diagonal	1	1.1	1.3	3.2	6.2	2.95
	3	0.2	1.1	4.1	5.2	2.65
	5	0.1	0.7	3.0	4.2	2.0
	7	0.0	0.8	3.1	5.0	2.23
Full	1	0.2	0.9	2.9	4.7	2.18
	2	0.0	1.2	6.0	5.3	3.13

Table I. Comparison of Performance of HMM Recognizer with Different Values of M

Condition	Average Digit Error Rate (%)				
	DIG 1	DIG 2	DIG 3	DIG 4	Overall
No Energy No Duration	0.3	2.5	4.3	8.0	3.78
Energy but No Duration	0.3	0.9	2.5	5.5	2.3
Duration but No Energy	0.1	1.3	3.3	5.4	2.53
Energy and Duration	0.1	0.7	2.8	4.2	1.95

Table II. Comparison of Performance of HMM Recognizer

3.5 Comparison of Internal and Post-Processor Durational Models

The two different implementations of the durational model, namely the internal durational model and the post-processor durational model were also compared. In both cases the same state-duration probability density function was used, with a multiplier of $\alpha = 3.0$. (This factor was locally optimum based on preliminary experimentation.) The results show that the performance of the HMM recognizer with the post-processor duration model was uniformly slightly better than for the recognizer with the internal durational model. Across the 4 data sets the improvement in performance was almost 0.7%.

3.6 Effects of Different Training Sets

As noted earlier, the recognizer used in the above experiments was trained on data set DIG1. To study the effects of training data, we also trained a new series of HMM's on DIG4, which is less homogeneous in recording conditions than the other 3 data sets. These HMM's use single multivariate Gaussian density with a full covariance matrix. Recognition results based on this new set of HMM's are shown in the first row of Table III. Results of the recognizer trained on DIG1 are shown in the second and third rows of Table III. It is seen that this new recognizer achieves a performance improvement of about 0.2%. What is more significant is that the error rate resulting from this has much less variation across the different test data sets than that obtained from the earlier training set.

IV. Comparison with Previous Recognizers

The above results characterize what can be achieved by the continuous mixture density HMM recognizer. Another way of measuring the effectiveness of the approach is to compare the current performance results with those of alternative recognition systems based on discrete densities (i.e., VQ symbols) [1], and based on templates [8]. Such a comparison is given in the last 2 rows of Table III. For the discrete density HMM recognizer, results are given only for the DIG 2 data set where the performance is significantly worse than that of the HMM recognizer with a continuous mixture density. For the template-based DTW recognizer, the results, based on the latest clustering procedure are comparable to those of the continuous density HMM recognizer. Since the template-based DTW recognizer has been studied for about 10 years and has been highly optimized in its performance, the equality between the HMM recognizer and the DTW recognizer, at least for the digits vocabulary, is highly significant.

Training Set	Cov. Type	Type of Recognizer	Average Digit Error Rate (%)				
			DIG 1	DIG 2	DIG 3	DIG 4	Overall
DIG4	Full	HMM-Continuous	2.5	1.7	2.1	0.8	1.78
DIG1	Full	$M = 1$	0.2	0.0	2.2	4.7	1.93
DIG1	Diagonal	HMM-Continuous $M = 5$	0.1	0.7	2.8	4.2	1.95
DIG1	NA	HMM-Discrete	—	2.9	—	—	—
DIG1	NA	DTW-Templates	0.0	0.6	2.7	3.9	1.8

Table III. Comparison of Performance of Several Recognizers

V. Summary

In the previous sections we have developed and tested an HMM isolated word recognition technique which uses a continuous mixture density model for the probability densities of the feature vector. Based on experimentation with the recognizer, in a speaker independent mode, using a vocabulary of 10 digits, the following general results were obtained:

1. The proposed model training procedure, with an iterative K-means loop for estimating initial values for the means and covariances of the components of the mixture model, works extremely well in practice and was able to converge to a local maximum of the likelihood function in a small number of iterations (typically 2-4 in most cases).
2. Mixture models with diagonal covariance matrices need a larger number of mixtures than mixture models with full covariance matrices in order to give the same performance.
3. Combining the techniques of clustering and HMM models can lead to small improvements in the performance of the HMM recognizer.
4. The addition of a word normalized energy contour (as an extra dimension to the feature vector) as well as durational information uniformly improves performance of the HMM recognizer and makes it more robust to differences in talker populations and transmission conditions.
5. The combination of normalized energy and durational information works better than either factor alone in the HMM recognizer.
6. The durational model of the HMM recognizer can be conveniently implemented as a post-processor to the Viterbi decoding procedure.

REFERENCES

[1] 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," *Bell System Tech. J.*, Vol. 62, No. 4, pp. 1075-1105, April 1983.

[2] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proc. IEEE*, Vol. 64, pp. 532-556, April 1976.

[3] J. K. Baker, "The Dragon System – An Overview," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-23, No. 1, pp. 24-9, February 1975.

[4] H. Bourlard, C. J. Wellekins and H. Ney, "Connected Digit Recognition Using Vector Quantization," *Proc. IEEE ICASSP '84*, pp. 26.10.1-26.10.4, San Diego, CA, March 1984.

[5] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Stat.*, Vol. 41, pp. 164-171, 1970.

[6] L. R. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," *IEEE Trans. on Information Theory*, Vol. IT-28, pp. 729-734, September 1982.

[7] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum Likelihood Estimation for Multivariate Normal Mixture Observations of Markov Chains," submitted for publication.

[8] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-27, No. 4, pp. 336-349, August 1979.