AT&T Technical Journal
Vol. 64, No. 6, July–August 1985
Printed in U.S.A.

Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities

By L. R. RABINER, B.-H. JUANG, S. E. LEVINSON, and M. M. SONDHI*

(Manuscript received November 15, 1984)

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 with 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 ten digits across a wide range of talkers and test conditions and shown to have an error rate comparable to that of the best template recognizers and significantly lower than that of the discrete symbol hidden Markov model system. We discuss several issues involved in the training of the continuous density models and in the implementation of the recognizer.

I. INTRODUCTION

In the literature a wide variety of approaches have been proposed to recognize isolated words, based on standard statistical-pattern-recognition techniques.[1–6] The most successful of these has been the template-based recognizer approach, which uses Dynamic Programming (DP) as the method for comparing patterns. Although the template-based approach using DP has been very successful, alternative recognition strategies have been studied because of

* Authors are employees of AT&T Bell Laboratories.

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

For one or more of the above reasons, several different approaches have been proposed, such as using Vector Quantization (VQ) in the DP computation,[4] using word-based vector quantization to eliminate the DP processing,[5] using VQ as a front-end preprocessor,[7] and using Hidden Markov Models (HMMs) to represent the speech signal.[6,8–11] Although the VQ-based recognizers have performed very well in isolated-word recognition tasks, and have significantly reduced the computational costs, they have done very little to alleviate the difficulties in extending template-based approaches to large vocabulary connected and continuous speech recognition applications. As such, the HMM recognizer has been and will continue to be of great interest both because of its potential low cost, and because it is a parametric model of the speech signal that can model various events (phonemes, syllables, etc.) in the speech signal.

Although HMMs have been used in a wide variety of speech systems,[6,8–11] our experience with their application to speech recognition systems has been considerably less than with that of template-based approaches. Hence each new experiment using HMMs gives us a better understanding of the strengths and weaknesses of such models as applied to different speech recognition tasks. In particular, in our own work, we have been studying how to apply HMMs in isolated-word, speaker-independent speech recognition applications over dialed-up telephone lines. In a previous investigation,[6] we studied HMMs based on observations consisting of discrete symbols from a finite alphabet (i.e., vector-quantized LPC vectors from a fixed-size code book). Work performed at IBM,[12] CMU,[9] and more recently at Phillips[13] has used continuous HMMs where it was assumed that all parameters of interest had Gaussian distributions.

The HMMs to be discussed in this paper are based on continuous, mixture density models of the distribution of Linear Predictive Coefficient (LPC)-derived parameter vectors (e.g., cepstral vectors, log-area ratio vectors, etc.). We have devised training procedures for obtaining maximum-likelihood estimates of the parameters of the mixture distribution. We have applied the models to the problem of recognizing isolated digits. Our results show that the average error rates of such HMM recognizers are essentially identical to those of the best template approaches using DP methods, and considerably

lower than those of an HMM recognizer with a discrete symbol VQ front end.

This paper is organized as follows. In Section II we present the continuous mixture density model. We show how we obtain the maximum-likelihood estimates of the model parameters from a training set of data, and how the overall recognition system is implemented. In Section III we describe a series of experimental evolutions of the recognizer and present results on HMM systems with several different sets of parameters. In Section IV we discuss the results and relate them to earlier work with template-based approaches. We also discuss computational aspects of the system in this section. Finally, in Section V we summarize our results.

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 that starts in state 1 and ends in state N. The observed signal is assumed to be a stochastic 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 HMM to optimally match the observed characteristics of a given signal.

The parameters that characterize the HMM of Fig. 1 are

1. N, the number of states in the model.
2. $A = [a_{ij}]$, $1 \leq i, j \leq N$, the state transition matrix, where a_{ij} is the probability of making a transition from state i to state j. As shown in Fig. 1b, for left-to-right models we use the constraint $a_{ij} = 0, j < i, j > i + 2$.
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 $\mathbf{O} = \{O_1, O_2, \cdots, O_T\}$, where each O_i is a vector that characterizes the signal at time $t = i$, then we can consider two types of observation probability functions, namely, discrete and continuous. For the discrete type we replace O_i by one of M

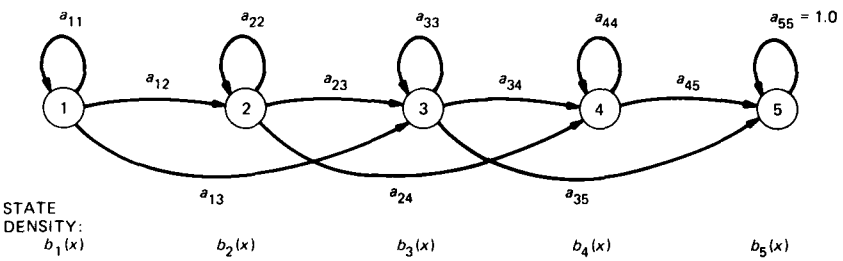

Fig. 1—Representation of a left-to-right hidden Markov model with five states.

possible symbols (via some type of VQ) such that the distortion in quantizing O_i is minimum. Let j be the state at time t. Then $B = [b_{jk}]$, $1 \le j \le N$, $1 \le k \le M$ is the probability of observing symbol k, in state j.

In the continuous case we have the probability density function $B = \{b_j(\mathbf{x})\}$, $1 \le j \le N$, 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,[14] elliptically symmetric densities,[15] and, more recently, mixtures of strictly log-concave or elliptically symmetric densities.[16] In this paper we will consider Gaussian mixture densities of the form

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{x}, \mu_{jk}, \mathbf{U}_{jk}), \tag{1}$$

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

To summarize the discussion above, a complete specification of a continuous mixture density HMM requires choosing values (and/or parameter estimates) for the following:

N—number of states in the model
M—number of mixtures
D—number of dimensions in each vector
$A = [a_{ij}]$—state transition matrix
$C = [c_{jk}]$—mixture gain matrix
$\mu = [\mu_{jkd}]$—means of the mixture components
$\mathbf{U} = [U_{jkde}]$—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.[6] Also, our signal observation vectors (e.g., cepstral vectors, log-area ratios, etc.) are derived from the LPC vector of an eighth-order model 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 a wide range of talkers. Since a convergent reestimation procedure exists for the continuous mixture model,[16] it is, in theory, possible to randomly choose initial values for each of the model parameters (subject to the stochastic constraints) and let the reestimation procedure determine the optimum (maximum-likelihood) values. However, experience with the reestimation procedure[17] has

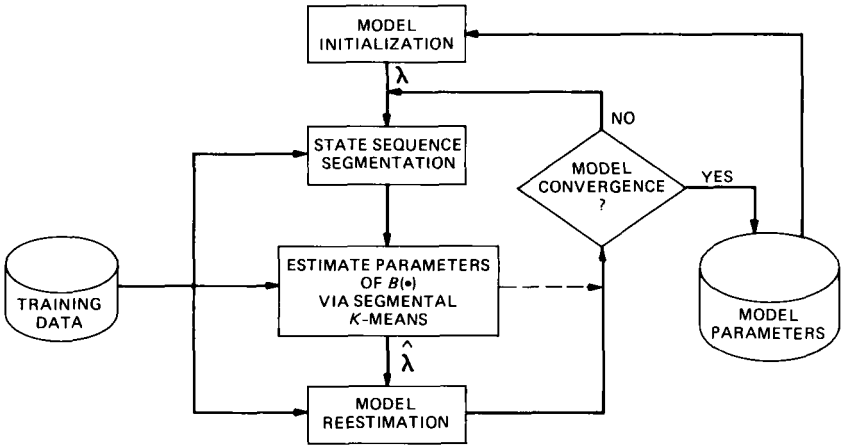

Fig. 2—The training procedure used to estimate parameter values for the optimal continuous mixture density fit to a finite number of observation sequences.

shown that the maximum-likelihood estimates of the means, μ, are quite sensitive to the initial estimate. Hence a procedure for providing good initial estimates of μ for each mixture and each state was required.

Based on previous experience with a K-means iterative procedure for clustering data,[18] a procedure for obtaining model parameter estimates was devised and is shown in Fig. 2. (The analysis used to give the LPC-derived vectors is reviewed in Section 2.2.) We assume a training set of data consisting of Q sequences of observations, where each sequence, $\mathbf{O}^i = \{O_1^i, O_2^i, \cdots, O_{T_i}^i\}$, $1 \le i \le Q$, is the set of vectors (observations) constituting a single occurrence of the word. The total observation vector is $O = \{\mathbf{O}^1\mathbf{O}^2 \cdots \mathbf{O}^Q\}$. The first step in the training procedure is to choose an initial model estimate. This initial estimate (unlike the one required for reestimation) can be chosen randomly, or on the basis of any good initial guess. (The procedure to be described here works well for a wide range of initial guesses.)

We denote the N states in the HMM as q_i, $1 \le i \le N$. The second step in the training procedure is to segment each word occurrence, $\mathbf{O}^i$, into states based on the current model, λ. This segmentation is achieved by finding the optimum state sequence, via the Viterbi algorithm, and then backtracking along the optimal path. This procedure is illustrated in Fig. 3, which shows a log-energy plot, an accumulated log-likelihood plot, and a state segmentation for one occurrence of the digit six. Figure 3 shows that the states correspond roughly to the sounds in the word six.

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 q_i according to the current model. Since the assumed distribution

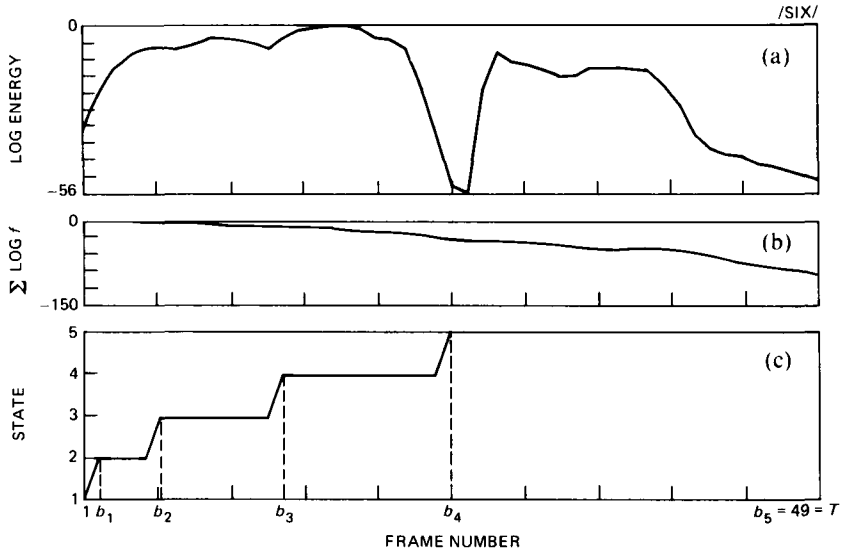

Fig. 3—Plots of (a) log energy, (b) accumulated log likelihood, and (c) state assignment for one occurrence of the word six.

of the observations, within the jth state, is $b_j(\mathbf{x})$, a comparison can be made of the marginal distributions $b_j(\mathbf{x})\,|_{\mathbf{x}=\{\cdots x_n \cdots\}}$ against a histogram of the actual observations (i.e., vectors assigned to that state). Such a comparison is given in Fig. 4 for a $D = 9$ dimensional representation with $M = 5$ mixtures. (The covariance matrices are assumed to be diagonal in this example.) The nine dimensions consist of the eight dimensions of a cepstral representation, and the normalized log energy as the ninth parameter. The results in Fig. 4 are for the first state of the digit 0. The need for values of $M > 1$ is seen in the histogram of the first parameter (the first cepstral component), which is inherently multimodal; similarly, the second, fourth, and eighth cepstral parameters show the need for more than a single Gaussian to provide good fits. Many of the other parameters appear to be well fitted by a single Gaussian curve; however, in some cases even $M = 5$ mixtures do not provide a very good fit.

Following the segmentation into states of all Q training sequences, a segmental K-means procedure is used to cluster the vectors in each state, q_i, into a set of M clusters (to do this we use a Euclidean distortion metric and a VQ design algorithm). From the clustering, an updated set of model parameters is derived as follows:

$\hat{c}_{jk}$ = Number of vectors classified in cluster k of the jth state divided by the number of vectors in state j

$\hat{\mu}_{jkd}$ = dth component of the sample mean of the vectors classified in cluster k of state j

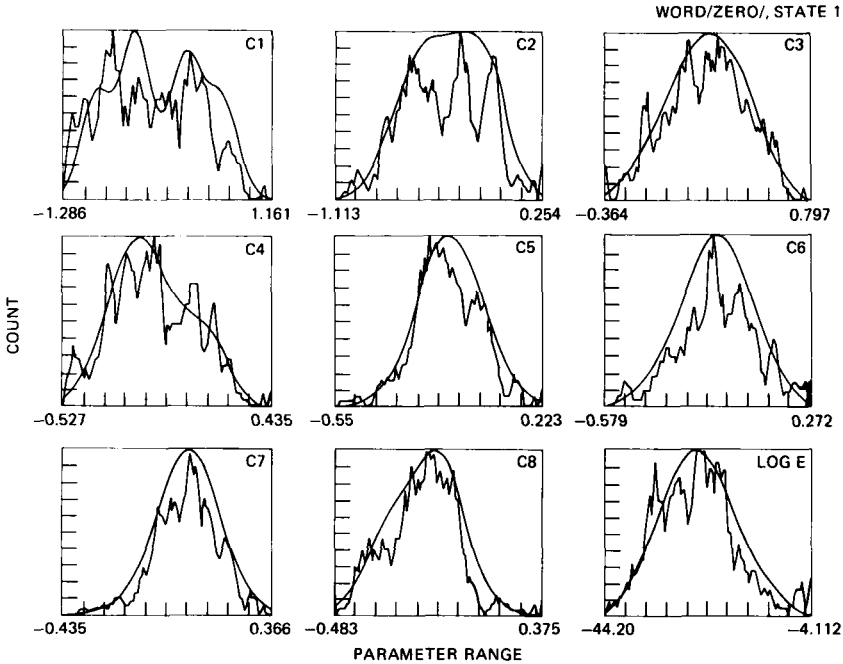

Fig. 4—Comparison of estimated density (jagged contour) and model density (smooth contour) for each of the nine components of the representation vector (eight cepstral components, one log-energy component) for state 1 of the digit zero.

$\hat{U}_{jkrs}$ = (r, s)th component of the sample covariance matrix of the vectors classified in cluster k of state j.

The state transition matrix coefficients, a_{ij}, are not changed according to this procedure. The new model, $\hat{\lambda} = (A, \hat{B}, \hat{\mu}, \hat{U})$, is obtained from the updated estimates $\hat{B}$, $\hat{\mu}$, and $\hat{U}$, and the original A matrix. 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 that reflects the statistical similarity of the HMMs[19]). If the model distance score exceeds a threshold, then the old model, λ, is replaced by the new model, $\hat{\lambda}$ (the result of reestimation), and the overall training loop is repeated. If the model distance score falls below the threshold, then model convergence is assumed and the final model parameters are saved.

As an alternative to using the sample means, $\hat{\mu}_{jkd}$, and sample covariance matrix, $\hat{U}_{jkrs}$ (which are the maximum-likelihood estimates for a Gaussian distribution), we also investigated a method of fitting a single Gaussian distribution to an observed histogram within each cluster of each state. For the case when $\hat{U}$ is diagonal, a histogram

with NB bins is made for each component of the vector, and the model parameters (i.e., μ and σ) are chosen so as to minimize the cost function

$$\rho = \sum_{i=1}^{NB} \frac{(\hat{h}_i - h_i)^2}{h_i},$$

where h_i is the observed frequency of occurrence in the ith bin, and $\hat{h}_i$ is the corresponding model estimate for that bin. The minimization for the two-parameter case (i.e., μ and σ) can be trivially carried out by several different procedures.

For the case when $\hat{\mathbf{U}}$ is a full covariance matrix, the histogram-fitting procedure could, in principle, be extended to D-dimensional histograms with correlated components. However, the amount of training data available was insufficient for the number of parameters being fitted. Instead, the histogram-fitting procedure that we used was as follows. The sample covariance matrix, $\hat{\mathbf{U}}$, was estimated from the training data (as above), and decomposed as

$$\hat{\mathbf{U}} = T' \Lambda T,$$

where Λ is a diagonal matrix. The original vectors, c, were transformed by the relation $w = Tc$. In this manner the components of w were uncorrelated with diagonal correlation matrix Λ; hence the histogram-fitting procedure, described above, could be used along each transformed dimension separately. In practice we have found that the transformation to uncorrelated components and the Gaussian fitting gave somewhat better model parameter estimates than the sample estimates for the full covariance case.

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), and reestimation is an exceedingly costly procedure, a practical implementation of the training procedure of Fig. 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. This procedure works well in practice, particularly when used for left-to-right models where the sequential characteristics of the process are of vital importance.

2.2 The HMM recognizer

Once the HMMs have been trained on each vocabulary word, the recognition strategy is straightforward. Figure 5 shows a block diagram of the recognizer. The speech signal, $s(n)$, for the unknown word is first analyzed using an eighth-order LPC analysis. The speech sampling rate is 6.67 kHz, and overlapping sections of 45 ms of speech are analyzed every 15 ms to give a set of eight LPC coefficients. An LPC transformation algorithm is used to convert the LPC representation

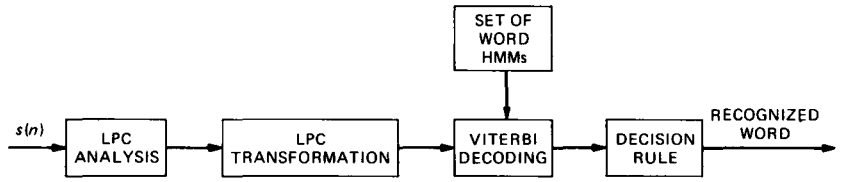

Fig. 5—Block diagram of the HMM recognizer based on continuous mixture densities.

to the desired one for the recognizer. In particular we have considered the following possibilities:

1. LPC-derived cepstrum of eighth-order (the zeroth-order term is not used)
2. LPC-derived log-area ratios
3. Autocorrelation coefficients normalized by energy
4. Residual normalized autocorrelation coefficients
5. Autocorrelation of LPC coefficients.

The vector representation used in training is the one used in the recognizer.

The next step in recognition is to find the optimum state sequence corresponding to the HMM for each vocabulary word, λ^v, $1 \leq v \leq V$, and compute the log-likelihood score for the optimal path. The decision rule assigns the unknown word to the vocabulary word whose model has the highest log-likelihood score.

The optimum path is obtained by the well-known Viterbi algorithm,[20] which can be compactly stated as:

1. Initialization—$\delta_1(1) = \log[b_1(O_1)]$
   $$\delta_1(i) = -\infty \qquad i \neq 1$$
2. Recursion $$-\delta_t(j) = \max_{j-2 \leq i \leq j} \{\delta_{t-1}(i) + \log a_{ij}\} + \log[b_j(O_t)],$$
   $$2 \leq t \leq T, \qquad 1 \leq j \leq N$$
3. Termination—$\log f = \delta_T(N)$.

2.3 Incorporation of duration into the recognizer

Inherently, each state in the HMM has a geometric duration probability. Thus, 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. Experience has shown that exponentials are not good models for state duration probabilities. Thus, we have considered two alternative ways of incorporating state duration information in the recognizer, namely, modification of the scoring procedure to include an internal duration model, and application of a post-processing duration model on the maximum-

likelihood 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 exactly (ℓ/T) of the word, where T is the number of frames in the word and ℓ is the number of frames spent in state j.

The quantity ℓ/T, which ranges from 0 to 1, is the normalized duration within a given state. For each word, and for each state, the quantity $p_j(\ell/T)$ is estimated (via a simple counting procedure on the training sequences) for 25 values of ℓ/T from 0 to 1.

The state duration probability, $[p_j(\ell)$ or $p_j(\ell/T)]$, is not estimated as part of the training procedure, but instead is computed directly from the training sequences based on the models obtained from the training procedure. Hence the estimates of $p_j(\ell/T)$ are strictly heuristic ones, not maximum-likelihood estimates. Unfortunately, direct reestimation of the maximum-likelihood estimate of $p_j(\ell/T)$ is, at present, totally impractical both because of the excessive computation required, and because of the sparsity of training data for estimating the increased number of model parameters.

A typical set of histograms of $p_j(\ell/T)$ for a five-state model for the word six is shown in Fig. 6. Although the states are hidden, examination of the results of segmentation of typical utterances (of the word six) into states shows that the first two states are essentially the initial /s/, the third state is a transition to the vowel /i/, the fourth state is the vowel, and the fifth state is the stop and the final fricative /s/. As seen from Fig. 6, the average duration of the first state is generally very brief; the second and third states have somewhat longer average durations; the fourth state has a well-defined peak in the density with an average duration of about 20 percent of the word; the final state (the stop plus the fricative) has an average duration of about 50 percent of the word.

For scoring a given observation sequence using the internal duration model, the recursion step of the Viterbi procedure is modified to the form

$$\delta_t(j) = \max_{j-2\le i\le j-1} \max_{0\le \ell/T\le 1.0} \left\{ \delta_{t-\ell}(i) + \log a_{ij} + \alpha \log p_j(\ell/T) + \sum_{\tau=1}^{\ell} \log[b_j(O_{t-\tau})] \right\}. \quad (2)$$

Note that in eq. (2) the duration term appears only when the state changes. Furthermore, a multiplier factor α on the log-duration probability is used to adjust the importance of the duration part of the scoring.

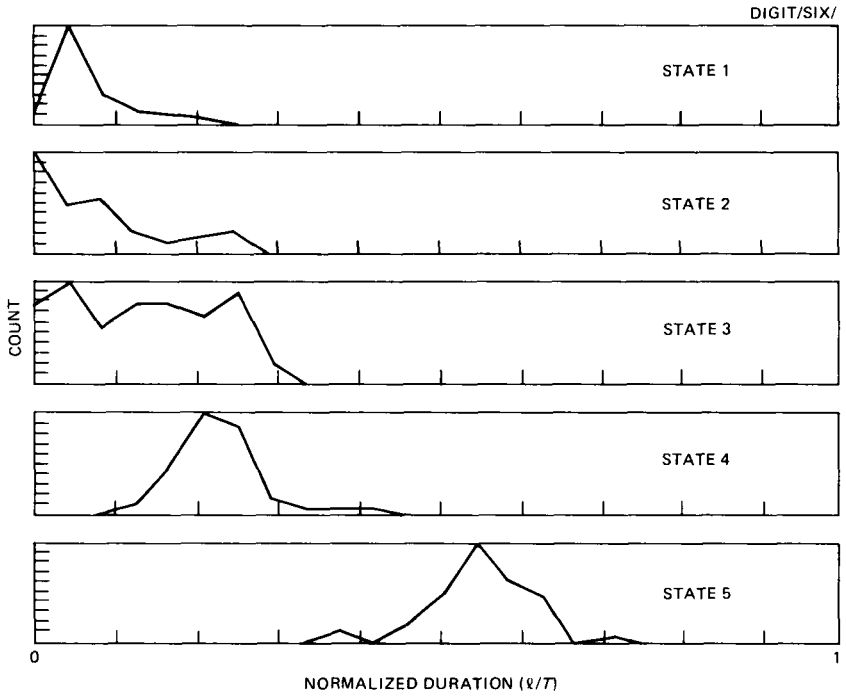

Fig. 6—Histograms of the normalized duration density for the five states of the digit six.

The implementation of the recursion of eq. (2) is considerably more costly than the implementation of the standard Viterbi recursion, since the values $\delta_{t-\ell}(i)$ must be retained for a large range of ℓ values, and since the $\sum \log[b_j(O_{t-\tau})]$ computation must be repeatedly done during each iteration. In practice we have measured an increase of 15 to 20 times in computation for the internal duration model over the standard Viterbi algorithm. For these reasons we have also considered a much simpler post-processor duration model in which the original Viterbi alignment is performed, the maximum-likelihood state sequence is determined, and the duration of each state is obtained via a backtracking procedure. The post-processor then increments the log-likelihood score by the log-duration probabilities (suitably weighted again) to give:

$$\log \hat{f} = \log f + \alpha \sum_{j=1}^{N} \log[p_j(\ell_j/T)], \tag{3}$$

where ℓ_j/T is the normalized time spent in state j along the optimal alignment path.

The incremental cost of the computation for the post-processor

duration model is essentially negligible, and we will see in Section III that it works as well or better than the internal duration model discussed above.

III. EXPERIMENTAL EVALUATION

To evaluate the performance of the HMM recognizer with the mixture density representation, a series of experiments were run in which several parameters of the models were varied. All evaluations were performed on a database of isolated digits recorded over standard dialed-up telephone lines. Four sets of spoken digits were used. These consisted of the following:

DIG 1—100 talkers (50 male, 50 female), one replication of each digit by each talker.[21] These recordings have been used as a training set in a wide variety of evaluations of isolated-word recognizers at AT&T Bell Laboratories. The nominal bandwidth of these recordings was 100 to 3200 Hz.

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), one averaged occurrence of each digit by each talker obtained from averaging a pair of robust tokens of the digit.[22,23] 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.[24] A random sampling of one of the recordings of each digit by each talker was used. The transmission conditions differed substantially from those used in recording the other databases. The nominal bandwidth of these recordings was 200 to 3200 Hz.

Thus, each of the four sets of digits contained 1000 digits. For training the models, only the digits in set DIG 1 or set DIG 4 were used; for testing and evaluating the performance of the recognizer, each of the four sets of digits was used.

3.1 Pilot recognition experiments to determine representation

A series of pilot experiments was run to determine a good set of parameters for use in the HMM recognizer. The five parameter sets (transformations of the LPC parameters) mentioned in Section 2.2 were studied. Results indicated that the best performance was obtained from the cepstral parameters; however, almost the same performance was obtained from the log-area ratio parameters. The remaining three parameter sets—i.e., energy-normalized autocorrelations, residual-normalized autocorrelations, and autocorrelation of LPC coeffi-

cients—all gave significantly poorer performance. This was due to the use of a Euclidean distance metric in the clustering part of the training procedure. Each of the poor representations had the property that one or more of the coefficients in the vector contributed significantly more variance in the distance calculation than the remaining coefficients; hence a large sensitivity to the details of the training set resulted, and very poor estimates of the means and covariances of the parameters were obtained. Such problems could have been alleviated by replacing the Euclidean distance metric with a covariance weighted metric; however, we did not do this because of the greatly increased computational burden.

As a result of the pilot experiment, the parameter set chosen was an eighth-order cepstral vector with the option of appending a peak-normalized log energy as a ninth component of the vector.

3.2 Diagonal versus full covariance matrices

Two forms for the $\mathbf{U}$ matrices of eq. (1) were considered, namely, diagonal matrices (with assumed zero correlation between components of the representation), and full covariance matrices. The advantage of the diagonal covariance matrix is that the computation of $b_j(\mathbf{x})$ reduces to a simple sum of products of Gaussians, whereas for a full covariance matrix the computation of $b_j(\mathbf{x})$ requires a matrix multiply. The disadvantage of the diagonal covariance matrix representation is that, in general, for correlated vector components, a larger value of M (the number of mixtures) is needed to give an adequate model than for a full covariance matrix representation. Neither representation has any particular advantage in terms of ease of making initial estimates or ease of reestimation.

A series of recognition tests was run with diagonal covariance matrices using $M = 1$, 3, and 5, and full covariance matrices using $M = 1$ only. 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 $M = 1$ and 3, and comparable to the performance with $M = 5$. Hence, in all subsequent recognition tests we will consider both diagonal and full covariance models.

3.3 Applicability of word clustering to model generation

In the model training procedure all 100 tokens of each word were used to derive a single HMM for the word. In earlier work, with template-based approaches,[21] it was shown how word clustering techniques could be used to design a set of templates to represent a broad population of talkers. Thus, one question of interest was whether the word clustering procedure could be combined with the model generation technique to give more than one HMM per word with better

performance than the single HMM system. This idea was tested as follows. First, a single HMM per word was created on training set DIG 1; next the two-cluster-per-word template set was used to partition the 100-token training set into two groups. For each group a single HMM was created; hence a total of two HMMs per word was used in the performance evaluation. The potential disadvantage of this procedure should be clear, namely, that the training data per model available for estimating HMM parameters is half that used for the single model case. Hence there is a good possibility of obtaining less reliable estimates of the model parameters.

This procedure could be continued as above for three or more template solutions; however, experience indicated that a two-model solution was about the limit for 100 training tokens. Beyond this point the unreliability of the estimates was the dominant factor.

Results of a formal series of experiments with each of the four test sets and with one and two models per word are given in Table I, which shows average digit error rates for both diagonal covariance models (part a), and full covariance models (part b) using normalized log-energy and cepstral coefficients, and with the post-processor duration model. For the diagonal covariance models, the results show that for the reference set (DIG 1), the average error rate was essentially 0 for both one- and two-model-per-word systems. For the test set DIG 2 the average error rate for the two-model-per-word system was slightly smaller (by 0.4 percent) than for the one-model-per-word system. For the test sets DIG 3 and DIG 4, the average error rate for the two-model-per-word system was 1 percent smaller than for the one-model-per-word system. Overall, averaged across the three independent test sets, the two-model-per-word system had a 0.7 percent smaller error rate than the one-template-per-word system. This difference, although small, is significant at the 95-percent level for a test with 4000 digits.

Table I—Comparison of performance of HMM recognizer with one and two models per word*

Number of Models per Word	Average Digit Error Rate (%)				
	DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
(a) Diagonal covariance models, $M = 5$ mixtures					
1	0.2	1.1	3.9	5.2	3.4
2	0.1	0.7	2.8	4.2	2.67
(b) Full covariance models, $M = 1$ mixture					
1	0.2	0.9	2.9	4.7	2.83
2	0.2	0.6	2.2	4.7	2.5

* Both energy and duration were used in the evaluation. The training set was DIG 1.

For the full covariance models, the improvement in performance in going from one to two models per digit was smaller than that obtained in the diagonal covariance case. On average, the improvement in error rate was only 0.33 percent, and for two of the four sets (the training set DIG 1, and the testing set DIG 4) there was no improvement in performance with two models per digit. Thus, for the full covariance models a single model per word was adequate for the data.

3.4 Effects of different number of mixtures

Using the two-model-per-word system for the diagonal covariance case, the number of mixtures, M, was varied from 1 to 7, in steps of two, to see the effects on recognition performance. The results of these tests on the four-digit databases are given in Table IIa. The results show an improvement in performance from an average test set digit error rate of 3.57 percent for $M = 1$ down to an average test set digit error rate of 2.57 percent for $M = 5$; results for $M = 7$ show a slight increase in average test set digit error rate to 2.97 percent. This increase in error rate for the $M = 7$ case is primarily due to a 0.8-percent increase in error rate for test set DIG 4. This result seems to indicate that no real improvement in modeling the statistics of the observations is obtained with $M = 7$; instead, a somewhat broader range for fitting incorrect words is achieved, thereby raising the error rate on DIG 4. Based on the results of Table II, a value of $M = 5$ was deemed most appropriate for the recognizer.

Another test was run for the diagonal covariance case, in which the value of M was made variable with each state of the model. The chosen value was based on the average distortion in the initial modeling section of the training loop. Thus, large values of M (on the order of

Table II—Comparison of performance of HMM recognizer with different values of M*

	Average Digit Error Rate (%)				
M	DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
(a) Results on diagonal covariance models for two models per digit					
1	1.1	1.3	3.2	6.2	3.57
3	0.2	1.1	4.1	5.2	3.47
5	0.1	0.7	2.8	4.2	2.57
7	0.0	0.8	3.1	5.0	2.97
(b) Results on full covariance models for one model per digit					
1	0.2	0.9	2.9	4.7	2.83
2	0.0	1.2	6.0	5.3	4.17

* Both energy and duration were used in the evaluation. Training set was DIG 1.

10 to 15) were required for some states, whereas very small values of M (1 to 2) were required for others. Recognition tests using the variable M models gave very poor results (i.e., error rates considerably higher than those for fixed M models). An analysis of the errors showed a greater increase in the likelihood for incorrect models than for the correct model. Hence we concluded that variable M models were not a viable alternative for HMM recognizers.

Using the one-model-per-word system for the full covariance case, a similar test was performed in which values of 1 and 2 were used for M. The results, listed in Table IIb, show a degradation in performance from an average test set digit error rate of 2.83 percent for $M = 1$ to an average test set digit error rate of 4.17 percent for $M = 2$. Most of the increase in error rate occurs for test set DIG 3, where the error rate increases by 3.1 percent. This result again indicates that a single full covariance matrix provides an adequate fit to the training data, and that increases in M primarily decrease the amount of training data per model and therefore lead to poorer parameter estimates and worse recognition performance.

3.5 Effects of energy and duration

To study the effects of including energy in the signal representation, and of including the duration model in the testing, a series of recognition runs were made with the $M = 5$, diagonal covariance, two-model-per-word system, and the $M = 1$, full covariance, two-model-per-word system. The results of these recognition tests are given in Table III. The duration model was implemented as a post-processor computation in all cases.

Table III—Comparison of performance of HMM recognizer with and without energy and with and without duration model (training set was DIG 1)

Condition	Average Digit Error Rate (%)				
	DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
(a) Results on diagonal covariance models, $M = 5$, with two models per digit					
No energy, no duration model	0.3	2.5	4.3	8.0	4.93
Energy, no duration model	0.3	0.9	2.5	5.5	2.97
No energy, duration model	0.1	1.3	3.3	5.4	3.33
Energy, duration model	0.1	0.7	2.8	4.2	2.57
(b) Results on full covariance models, $M = 1$, with two models per digit					
No energy, no duration model	0.2	2.0	2.8	7.0	3.93
Energy, no duration model	0.2	1.2	2.1	4.4	2.57
No energy, duration model	0.3	1.3	3.0	5.9	3.4
Energy, duration model	0.2	0.6	2.2	4.7	2.5

The results show clearly that the addition of either energy or duration uniformly improves the performance of the HMM recognizer, although energy is more important than duration for the full covariance models. Furthermore, the combination of both energy and duration 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 makes the system more robust because these features are, for the most part, insensitive to differences in transmission conditions.

3.6 Comparison of internal duration model and post-processor duration model

The next set of experiments compared the two different implementations of the duration model, namely, the internal duration model and the post-processor duration model. In both cases the same (suboptimal) state-duration probability density function was used, with a multiplier of $\alpha = 3.0$. (This factor was optimized based on preliminary experimentation.) The results of the two runs are given in Table IV.

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 duration model. Across the three test sets the improvement in performance was about 0.9 percent, and for the two data sets with different transmission conditions (DIG 3 and DIG 4), the improvement was 0.9 percent and 1.7 percent, respectively. In addition, the computational load was between one and two orders of magnitude lower for the post-processor duration model than for the internal duration model. Hence the results given in Table IV strongly justify the use of the "suboptimal" post-processor duration model as an alternative to using the inherent exponential distributions for each state. The major problem with the use of any duration model is the difficulty of making reliable estimates of the density function

Table IV—Comparison of performance of HMM recognizer with two types of duration models*

	Average Digit Error Rate (%)				
Duration Model	DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
Internal in Viterbi search	0.2	0.9	3.7	5.9	3.5
Post-processor	0.1	0.7	2.8	4.2	2.57

* Results given on diagonal covariance models, $M = 5$, with two models per digit, and with energy used as a feature. Training set was DIG 1.

from a finite training set of observations (as is invariably the case for most speech processing applications). However, the improvements in performance obtained from using the duration model more than justify its use in the HMM recognizer.

3.7 Effects of different training sets

The last set of experiments investigated the effects of different training sets on the overall recognizer performance. The results of these experiments are given in Table V, which shows average test set digit error rate as a function of training set, model type, and number of models per digit. (All models used both energy and the post-processor duration model.) The results of Table V show that when the set DIG 4 was used as the training set, the performance among all four test sets was more uniform than when the set DIG 1 was used as the training set. The results also show that with a single model per digit, the performance with the DIG 4 training models was comparable or better than the performance of the DIG 1 training models with two models per digit. Thus the use of the slightly narrower bandwidth training data led to models that were more robust to small differences in recording conditions than those obtained from the broader bandwidth training data.

IV. DISCUSSION

4.1 General results

In the previous sections we have proposed and tested an HMM isolated-word recognizer that 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 ten 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

Table V—Comparison of performance of the recognizer as a function of the training set, model type, and number of models per digit*

Training Set	Model Type	Number of Models per Digit	Average Digit Error Rate (%)				
			DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
DIG 1	Diagonal covariance	2	0.1	0.7	2.8	4.2	2.57
DIG 1	Full covariance	2	0.2	0.6	2.2	4.7	2.5
DIG 4	Diagonal covariance	1	2.5	2.4	2.8	0.5	2.57
DIG 4	Full covariance	1	2.5	1.7	2.1	0.8	2.1
DIG 4	Full covariance	2	2.3	2.2	2.1	0.5	2.2

* All models used both energy and the post-processor duration model.

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 to 4 in most cases).

2. Several speech parameters (most notably the set of cepstral parameters and the set of log-area ratios) are well represented by the continuous mixture density, and give good recognition performance in the HMM recognizer. Other speech parameters (e.g., energy-normalized autocorrelation parameters, residual-normalized autocorrelation parameters, etc.) are not well represented by the continuous mixture density, and give relatively poor performance in the HMM recognizer when a Euclidean distance measure was used.

3. Mixture models with diagonal covariance matrices need a somewhat larger number of mixtures (typically, $M = 3$ to 5) than mixture models with full covariance matrices ($M = 1$) in order to give the same performance.

4. Combining the techniques of clustering and HMM model building can lead to small improvements in the performance of the HMM recognizer.

5. The addition of a word-normalized energy contour (as an extra dimension to the feature vector) uniformly improves performance of the HMM recognizer and makes it more robust to differences in talker populations and transmission conditions.

6. The addition of duration information, on a state-by-state basis, into the HMM recognizer uniformly improves performance and increases robustness of the recognizer to different talkers and transmission conditions.

7. The combination of normalized energy and duration information works better than either factor alone in the HMM recognizer.

8. Word models with variable number of mixture densities per state (based on clustering distortion statistics) yield significantly worse performance than models with a fixed number of mixture densities per state.

9. The duration model of the HMM recognizer can be conveniently (and suboptimally) implemented as a post-processor to the Viterbi decoding procedure. In practice the performance of this system is actually better (somewhat) than the recognizer with the duration model built into the Viterbi decoding procedure.

The above results are one measure of the success achieved by the continuous mixture density HMM recognizer. Another way of measuring the success is to compare the current performance results with those of alternative recognition systems based on templates[21] and based on discrete densities (i.e., VQ symbols).[6] Such a comparison is given in Table VI for the case when the data of set DIG 1 was used as

Table VI—Comparison of performance of three types of recognizers on the digits database*

Type of Recognizer	Average Digit Error Rate (%)				
	DIG 1	DIG 2	DIG 3	DIG 4	Test Set Average
HMM—Continuous density	0.1	0.7	2.8	4.2	2.57
HMM—Discrete density	—	2.9	—	—	—
DTW—Templates	0.0	0.6	2.7	3.9	2.4

* Training set was DIG 1.

the training set. 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 Dynamic Time Warping (DTW) recognizer, the results, based on the latest clustering procedure,[25] are comparable to those of the continuous density HMM recognizer. Since the template-based DTW recognizer has been studied for about ten 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.

4.2 Computational considerations

To calculate the computation required in the HMM recognizer, and to contrast it with that required by the DTW recognizer, we must define the following:

N = Number of states in HMM model
M = Number of mixture densities per state
D = Dimensionality of vectors in signal representation
T = Average number of frames (observations) per word
R = Number of HMMs per word
V = Number of words in vocabulary
Q = Number of templates per word in DTW system
P = Order of LPC analysis.

The computation for the HMM recognizer, in the Viterbi decoding algorithm, is

$$C_V = R \cdot V \cdot N \cdot T \cdot C_b,$$

where C_b is the computation required to evaluate the mixture density $b_j(\mathbf{x})$. Assuming that multiplications, divisions, exponentiation, and logarithms all take a single multiply-add time (somewhat optimistic calculations), then

$$C_b \approx 3DM \ *, + \qquad \text{(diagonal covariance)}$$

or

$$C_b \approx D^2M \ *, + \qquad \text{(full covariance)}$$

and

$$C_V = 3D \cdot M \cdot R \cdot V \cdot N \cdot T \ *, + \qquad \text{(diagonal covariance)}$$

or

$$C_v = \mathrm{D^2M \cdot R \cdot V \cdot N \cdot T} \ *, + \qquad \text{(full covariance).}$$

The standard DTW recognizer requires

$$C_{\mathrm{DTW}} = Q \cdot V \cdot \frac{T^2}{3} \cdot (p + 1) \ *, + \ .$$

Hence the ratio of C_V to C_{DTW} is

$$\mathrm{RATIO} = \frac{C_V}{C_{\mathrm{DTW}}} = \frac{3DMRN}{Q\frac{T}{3}(p + 1)} \qquad \text{(diagonal covariance)}$$

or

$$\mathrm{Ratio} = \frac{D^2MRN}{Q\frac{T}{3}(p + 1)} \qquad \text{(full covariance).}$$

If we choose typical values of $N = 5$, $M = 5$, $D = 9$, $T = 40$, $R = 2$, $V = 10$, $Q = 12$, $p = 8$, for the diagonal covariance case, we get

$$\mathrm{RATIO} = 15/16,$$

and using $R = 1$, $M = 1$ for the full covariance case we get

$$\mathrm{RATIO} = 9/32,$$

i.e., the computation of the HMM recognizer is essentially that of the DTW recognizer for the diagonal covariance case, and about one-quarter that of the DTW recognizer for the full covariance case. This situation is very different from the discrete symbol HMM recognizer, where the computation was an order of magnitude smaller than that of the DTW recognizer. The problem with the continuous mixture density recognizer is the $b_j(\mathbf{x})$ computation, which is extremely expensive, especially for values of $M > 1$.

V. SUMMARY

In this paper we have extended our experimental investigations of HMM recognizers to include the case where the density function for the observations in each state is represented by a continuous mixture of Gaussian variables. We have shown how the parameters of such a

signal representation can be optimally estimated (in a maximum-likelihood sense) from a finite training set of data, and have given a simple way of implementing the training procedure based on a K-means iteration. We have tested the HMM recognizer, in a speaker-independent mode, on a vocabulary of the ten digits, and shown that the error rate of the system is smaller than that obtained from the discrete density (VQ-based) HMM recognizer, and comparable to that of the multiple template-based DTW recognizer.

REFERENCES

1. F. I. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoust., Speech, and Signal Proc., *ASSP-23,* No. 1 (February 1975), pp. 67–72.
2. G. M. White and R. B. Neely, "Speech Recognition Experiments With Linear Prediction, Bandpass Filtering, and Dynamic Programming," IEEE Trans. Acoust., Speech, and Signal Proc., *ASSP-24,* No. 2 (April 1976), pp. 183–8.
3. G. R. Doddington and T. B. Schalk, "Speech Recognition: Turning Theory to Practice," IEEE Spectrum, *18* (September 1981), pp. 26–32.
4. K. Shikano, "Spoken Word Recognition Based Upon Vector Quantization of Input Speech," Trans. Comm. Speech Res. (December 1982), pp. 473–80.
5. J. E. Shore and D. K. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," IEEE Trans. Inform. Theory, *IT-29,* No. 4 (July 1983), pp. 473–91.
6. 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," B.S.T.J., *62,* No. 4 (April 1983), pp. 1075–105.
7. K. C. Pan, F. K. Soong, and L. R. Rabiner, "A Vector Quantization Based Preprocessor for Speaker Independent Isolated Word Recognition," Trans. Acoust., Speech, and Signal Proc., *ASSP-33,* No. 3 (June 1985).
8. A. B. Poritz, "Linear Predictive Hidden Markov Models and the Speech Signal," Proc. IEEE, *ICASSP '82,* Paris, France (May 1982), pp. 1291–4.
9. J. K. Baker, "The Dragon System—An Overview," IEEE Trans. Acoust., Speech, and Signal Proc., *ASSP-23,* No. 1 (February 1975), pp. 24–9.
10. L. R. Bahl, F. Jelinek, and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, *PAMI-5,* No. 2 (March 1983), pp. 179–90.
11. R. Billi, "Vector Quantization and Markov Source Models Applied to Speech Recognition," Proc. IEEE, *ICASSP '82,* Paris, France (May 1982), pp. 574–7.
12. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," Proc. IEEE, *64* (April 1976), pp. 532–56.
13. H. Bourlard, C. J. Wellekins, and H. Ney, "Connected Digit Recognition Using Vector Quantization," Proc. IEEE, *ICASSP '84,* San Diego, Calif. (March 1984), pp. 26.10.1–4.
14. 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., *41* (1970), pp. 164–71.
15. L. R. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," IEEE Trans. Inform. Theory, *IT-28* (September 1982), pp. 729–34.
16. B.-H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum Likelihood Estimation for Multivariate Normal Mixture Observations of Markov Chains," IEEE Int. Symp. Inform. Theory, Brighton, England, June 23–28, 1985.
17. L. R. Rabiner, B.-H. Juang, S. E. Levinson, and M. M. Sondhi, "Some Properties of Continuous Hidden Markov Model Representations," AT&T Tech. J., *64,* No. 6 (July–August 1985), pp. 1251–70.
18. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," IEEE Trans. Commun., *COM-28,* No. 1 (January 1980), pp. 84–95.
19. B.-H. Juang and L. R. Rabiner, "A Probabilistic Distance Measure for Hidden Markov Models," AT&T Tech. J., *64,* No. 2, Part 1 (February 1985), pp. 391–408.
20. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically

Optimum Decoding Algorithm," IEEE Trans. Inform. Theory, *IT-13* (April 1967), pp. 260–9.
21. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoust., Speech, and Signal Proc., *ASSP-27,* No. 4 (August 1979), pp. 336–49.
22. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Am., *68,* No. 5 (November 1980), pp. 1271–6.
23. J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker Independent Isolated Word Recognition Using a 129-Word Airline Vocabulary," J. Acoust. Soc. Am., *72,* No. 2 (August 1982), pp. 390–6.
24. A. E. Rosenberg, K. L. Shipley, and D. E. Bock, "A Speech Data Base Facility Using a Computer Controlled Cassette Tape Deck," J. Acoust. Soc. Am., Suppl. 1, *72* (Fall 1982), p. 580.
25. J. G. Wilpon and L. R. Rabiner, "A Modified *K*-Means Clustering Algorithm for Use in Speaker Independent Isolated Word Recognition," Trans. Acoust., Speech, and Signal Proc., *ASSP-33,* No. 3 (June 1985).

AUTHORS

Biing-Hwang Juang, B.Sc. (Electrical Engineering), 1973, National Taiwan University, Republic of China; M.Sc. and Ph.D. (Electrical and Computer Engineering), University of California, Santa Barbara, 1979 and 1981, respectively; Speech Communications Research Laboratory (SCRL), 1978; Signal Technology, Inc., 1979–1982; AT&T Bell Laboratories, 1982; AT&T Information Systems Laboratories, 1983; AT&T Bell Laboratories, 1983—. Before joining AT&T Bell Laboratories, Mr. Juang worked on vocal tract modeling at Speech Communications Research Laboratory, and on speech coding and interference suppression at Signal Technology, Inc. Presently, he is a member of the Acoustics Research Department, where he is researching speech communications techniques and stochastic modeling of speech signals.

Stephen E. Levinson, B.A. (Engineering Sciences), 1966, Harvard; M.S. and Ph.D. (Electrical Engineering), University of Rhode Island, Kingston, Rhode Island, 1972 and 1974, respectively; General Dynamics, 1966–1969; Yale University, 1974–1976; AT&T Bell Laboratories, 1976—. From 1966 to 1969, Mr. Levinson was a design engineer at Electric Boat Division of General Dynamics in Groton, Connecticut. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University. In 1976, he joined the technical staff at AT&T Bell Laboratories, where he is pursuing research in the areas of speech recognition and cybernetics. In 1984 he held a visiting fellowship in the Engineering Department at Cambridge University. Member, Association for Computing Machinery, editorial board of Speech Technology; fellow, Acoustical Society of America, senior member, IEEE; associate editor, IEEE Transactions on Acoustics, Speech and Signal Processing; vice chairman, ASSP technical committee on speech.

Lawrence R. Rabiner, S.B. and S.M., 1964, Ph.D., 1967 (Electrical Engineering), The Massachusetts Institute of Technology; AT&T Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at AT&T Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Currently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975) and *Digital*

Processing of Speech Signals (Prentice-Hall, 1978). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi, The National Academy of Engineering. Fellow, Acoustical Society of America, IEEE.

Man Mohan Sondhi, B.Sc. (Physics), Honours degree, 1950, Delhi University, Delhi, India, D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S., 1955, and Ph.D. (Electrical Engineering), 1957, University of Wisconsin, Madison, Wisconsin; AT&T Bell Laboratories, 1962—. Before joining AT&T Bell Laboratories, Mr. Sondhi worked for over a year at the Avionics division of John Oster Manufacturing Company, Racine, Wisconsin; for a year at the Central Electronics Engineering Research Institute, Pilani, India; and taught for a year at the University of Toronto. At AT&T Bell Laboratories his research has included work on speech signal processing, echo cancellation, adaptive filtering, modeling of auditory and visual processes, acoustical inverse problems, and HMM of speech. From 1971 to 1972, Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.