Mixture Autoregressive Hidden Markov Models for Speaker Independent Isolated Word Recognition

B. H. Juang
L. R. Rabiner

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT. In this paper a signal modeling technique based upon finite mixture autoregressive probabilistic functions of Markov chains is developed and applied to the problem of speech recognition, particularly speaker-independent recognition of isolated digits. Two types of mixture probability densities are investigated: finite mixtures of Gaussian autoregressive densities (GAM) and nearest-neighbor partitioned finite mixtures of Gaussian autoregressive densities (PGAM). In the former (GAM), the observation density in each Markov state is simply a (stochastically constrained) weighted sum of Gaussian autoregressive densities, while in the latter (PGAM) it involves nearest-neighbor decoding which, in effect, defines a set of partitions on the observation space. In this paper we discuss the signal modeling methodology and give experimental results on speaker independent recognition of isolated digits.

I. Introduction

Signal modeling based upon hidden Markov models (HMM's) may be viewed as an effective technique that extends conventional stationary spectral analysis principles to the analysis of time-varying signals [1-2]. The primary concern in the hidden Markov modeling technique is the estimation of model parameters from observed sequences. A reestimation algorithm due to Baum & Eagon [3] is usually used for this purpose. In this paper, we concentrate primarily on mixtures of Gaussian autoregressive densities which are the bases in the maximum likelihood formulation of the ubiquitous linear prediction analysis [1,2,4]. Poritz was the first to show how the ideas of linear prediction analysis could be welded into the hidden Markov model methodology [1]. However, in his work, Poritz only considered a single Gaussian autoregressive density per state. Our work extends this initial work to the case of a mixture of Gaussian autoregressive densities.

We consider two types of mixture densities in this paper. The first type is simply a finite mixture of Gaussian autoregressive densities, denoted as GAM (Gaussian Autoregressive Mixture) for brevity, which allows straightforward maximum likelihood estimation. The second type is a finite mixture of Gaussian autoregressive densities with nearest-neighbor decoding, denoted as PGAM (P for Partitioned). In this type of mixture density, the vector space is implicitly partitioned into regions. Each region is defined by a Gaussian autoregressive density. To evaluate the pdf for a point in the measure space, the appropriate region to which the point belongs is first found by a nearest-neighbor criterion. This resembles a vector quantization operation in source coding. The goal is to further link the vector quantization technique for source coding to a super-structure, namely a Markov chain, so as to exploit the non-memoryless nature of speech signals.

II. Mixture Autoregressive Hidden Markov Models

We consider an N-state homogeneous Markov chain with state transition matrix $A = [a_{ij}]$, $i,j = 1,2,\ldots,N$. Associated with each state j of the unobservable Markov chain is a probability density function $b_j(\mathbf{x})$ of the observed K-dimensional random vector $\mathbf{x}' = [x_0, x_1, \ldots, x_{K-1}]$ (K consecutive samples of the speech signal). We will use the notation $\mathbf{b}_j$ to denote the parameters defining $b_j(\mathbf{x})$. Also let $\mathbf{O}$ be the observed sequence, $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$, where T is the duration of the sequence and each $\mathbf{o}_t$ is an observed vector $\mathbf{x}$. The probability density function for $\mathbf{O}$ in the T-fold cartesian product of the K-dimensional vector space, $\mathcal{R}_K^T = \mathcal{R}_K \times \mathcal{R}_K \times \cdots \times \mathcal{R}_K$, is then

$$f(\mathbf{O}|\lambda) = \sum_S f(\mathbf{O},S|\lambda)$$

$$= \sum_S \pi_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(\mathbf{o}_t) \ . \qquad (1)$$

In Eq. (1), we use λ to denote the hidden Markov model, $\lambda = (\pi, A, B)$ where π is the initial state probability vector, A is the transition matrix and B is the set of parameters defining $\{b_j(\mathbf{x})\}$, $j = 1,2,\ldots,N$. S is a state sequence, $S = (s_0, s_1, \ldots, s_T)$, $s_t \in \{1,2,\ldots,N\}$, and the summation is over all possible state sequences S.

In GAM, the observation density $b_j(\mathbf{x})$, for $j = 1,2,\ldots,N$, has the form

$$\text{GAM}: \qquad b_j(\mathbf{x}) = \sum_{m=1}^{M} c_{jm} b_{jm}(\mathbf{x}) \qquad (2)$$

where M is the number of mixture components, c_{jm} is the weight for the mth mixture component, and the double-subscripted function $b_{jm}(\mathbf{x})$ is the basis probability density function for the mth mixture component, all related to state j. The mixture weight c_{jm} must satisfy the stochastic constraint

$$\sum_{m=1}^{M} c_{jm} = 1 \ , \qquad j = 1,2,\ldots,N \qquad (3)$$

so that

$$\int_{\mathcal{R}_K} b_j(\mathbf{x}) d\mathbf{x} = \sum_{m=1}^{M} c_{jm} \int_{\mathcal{R}_K} b_{jm}(\mathbf{x}) d\mathbf{x} = 1 \ .$$

In PGAM, on the other hand, the observation density $b_j(\mathbf{x})$ assumes the form

$$\text{PGAM}: \qquad b_j(\mathbf{x}) = \max_{m=1,2,\ldots,M} c_{jm} b_{jm}(\mathbf{x}) \ . \qquad (4)$$

It is clear that, in PGAM, only the most likely mixture component is chosen as the observation density for each particular vector $\mathbf{x}$. There is thus a built-in classification rule that implies a partition on R_K. Let Ω_{jm}, $m = 1,2,\ldots,M$ be the partitioned regions. In each region Ω_{jm}, every $\mathbf{x} \in \Omega_{jm}$ has

$$c_{jm} b_{jm}(\mathbf{x}) \geq c_{j\ell} b_{j\ell}(\mathbf{x})$$

for all $\ell \neq m$. The stochastic constraint becomes

$$\int_{R_k} b_j(\mathbf{x}) d\mathbf{x} = \sum_{m=1}^{M} c_{jm} \int_{\Omega_{jm}} b_{jm}(\mathbf{x}) d\mathbf{x} = 1 \ . \qquad (5)$$

The constraint on c_{jm} that results from Eq. (5) is discussed below.

Parameters of the model to be estimated therefore include: 1) the transition matrix $[a_{ij}]$, $i,j = 1,2,\ldots,N$; 2) the mixture weight (for GAM) $[c_{jm}]$, $j = 1,2,\ldots,N$ and $m = 1,2,\ldots,M$; and 3) all

CH2243-4/86/0000-0041 $1.00

necessary parameters defining the set of basis probability densities $\{b_{jm}(\mathbf{x})\}$, $j = 1,2,\ldots,N$ and $m = 1,2,\ldots,M$. As will be shown shortly, the stochastic constraints on the transition probabilities, i.e. $\sum_{j=1}^{N} a_{ij} = 1$, as well as the mixture weights as required in Eq. (3) for the GAM case can be automatically satisfied in the reestimation algorithm. To satisfy the requirement of Eq. (5) for PGAM, nevertheless, is not a trivial task during reestimation where an increase in likelihood after each iteration must be maintained. We therefore use a simplified expression for $b_j(\mathbf{x})$ in the PGAM case:

$$\text{PGAM}: \quad b_j(\mathbf{x}) = \frac{1}{M} \max_{m=1,2,\ldots,M} b_{jm}(\mathbf{x}) \ . \tag{6}$$

Note that the use of Eq. (6) excludes the mixture weights as the model parameters since $c_{jm} = \frac{1}{M}$ for all j and m is implied. In addition, we assume that

$$\int_{\Omega_{jm}} b_{jm}(\mathbf{x})\, d\mathbf{x} \simeq 1 \quad \text{for all } j,m$$

for the constraint of Eq. (5) to be approximately satisfied. Experimentally, we found this to be the case for typical values of N and M chosen in this study.

2.1 Gaussian Autoregressive Densities

We assume that the basis probability density function for the observation vectors is Gaussian autoregressive of order p, i.e.

$$f(\mathbf{x}) \simeq (2\pi)^{-K/2} \exp\{-\tfrac{1}{2}\delta(\mathbf{x};\mathbf{a})\} \tag{7}$$

where

$$\delta(\mathbf{x};\mathbf{a}) = r_a(0)r(0) + 2\sum_{i=1}^{p} r_a(i)r(i) \ , \tag{8}$$

$\mathbf{a}' = [1, a_1, a_2, \ldots, a_p]$, the autoregression coefficient vector,

$$r_a(i) \triangleq \sum_{n=0}^{p-i} a_n a_{n+i} \quad \text{with } a_0 = 1 \tag{9}$$

and

$$r(i) \triangleq \sum_{n=0}^{K-i-1} x_n x_{n+i} \ . \tag{10}$$

We assume in the above observation vector $\mathbf{x}$ has already been properly scaled by the square root of the *average* residual energy resulting from LPC analysis. Note that r_a's are the autocorrelation of the autoregressive coefficients and r's are the autocorrelation of the (normalized) observation samples. Maximum likelihood estimation of the autoregressive coefficients from $\mathbf{x}$ requires minimization of $\delta(\mathbf{x};\mathbf{a})$, a procedure equivalent to the autocorrelation method in linear prediction analysis.

We use Eq. (7) as the basis density function:

$$b_{jm}(\mathbf{x}) = (2\pi)^{-K/2} \exp\{-\tfrac{1}{2}\delta(\mathbf{x};\mathbf{a}_{jm})\} \ , \tag{11}$$

where $\mathbf{a}_{jm}$ is clearly the parameter vector defining the density for the mth mixture component in state j.

2.2 Reestimation Transformation

For a given observation $\mathbf{O}$, the reestimation algorithm starts with an initial guess of the model λ. A transformation that maps the parameter space into itself is then obtained based upon λ. The transformation leads to new model $\bar{\lambda}$ which has $f(\mathbf{O}|\bar{\lambda}) > f(\mathbf{O}|\lambda)$ unless λ is a fixed point of the transformation. The procedure is iterated after replacing the old model with the new model, and stops when a fixed point, which corresponds to a critical point of $f(\mathbf{O}|\lambda)$, is reached. The procedure guarantees an increase in likelihood after each iteration and will converge to a local optimum, when proper densities are used. The details of the reestimation formulas are given in Reference [5].

III. Implementation of the Model Estimation Procedure

A crucial prerequisite of the entire estimation procedure is a reliable and meaningful method for initialization. Obtaining such reliable initial estimates is often non-trivial since they are strongly affected by the prescribed Markov chain constraints. For left-to-right Markov models, which have been shown to be extremely useful in speech modeling, a modified training procedure was developed in which very good initial estimates of model parameters were obtained via a segmental k-means procedure, and then the formal reestimation algorithm was used as a model refinement tool [6]. (A segmental k-means procedure is a k-means clustering procedure operated over properly selected segments of the training sequence). A block diagram of this modified training procedure is given in Figure 1. An initial model λ is assumed. This initial model can be chosen via a variety of procedures including random initial guesses of model parameters (subject to the required stochastic constraints). Based on the initial model, each of a set of L training sequences is segmented into the maximum likelihood state sequence (via a Viterbi decoding procedure). For each state of the model, a k-means procedure, specifically the LPC vector quantizer design algorithm [7], clusters all the observation vectors within that state into a set of M clusters, based on a nearest neighbor classifier, using the distance measure of Eq. (8). Based on the vectors within each cluster, the initial estimates of the parameters of B are given as

$$\hat{c}_{jm} = \frac{\text{Number of vectors in cluster } m, \text{ state } j}{\text{Number of vectors in state } j}$$

$\hat{\mathbf{a}}_{jm}$ = LPC vector corresponding to the centroid (of the normalized autocorrelation) of cluster m, state j

The transition coefficients, a_{ij}, are not modified in the segmental k-means procedure. The new model $\hat{\lambda} = (\pi, A, \hat{B})$ is used as the initial estimate for the model reestimation algorithm which leads to a new $\bar{\lambda}$. A check on model convergence, in which the new model, $\bar{\lambda}$, is compared to the previous model, λ, is used to determine if the model estimation has converged. If no convergence is obtained, a new iteration of the training procedure is carried out with $\lambda = \bar{\lambda}$. Practically we have found that rapid convergence is obtained in all cases.

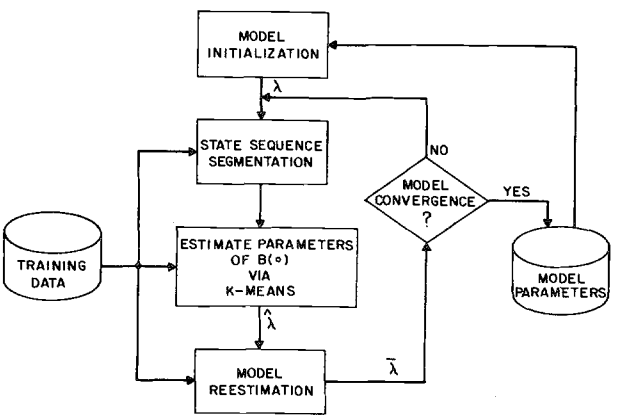

Fig. 1 Block diagram of model training procedure.

IV. Incorporation of Model State Duration and Log Energy Estimates

For speech recognition applications, it is often desirable to incorporate state duration and energy information in the model. Thus, calculation of the likelihood involves evaluation of the following equivalent distance (besides looking up the state transition probability matrix):

$$d(\mathbf{x},\ell/T,\tilde{\epsilon};\mathbf{a}_{jm},p_j,w_j) = \hat{K}\left[\frac{1}{K}\delta(\mathbf{x};\mathbf{a}_{jm}) - 1\right] - \frac{\gamma_D}{T}\log p_j(\ell/T) - \gamma_E \log w_j(\tilde{\epsilon}) \quad (12)$$

where $\hat{K}$ is the effective length of each data vector, ℓ/T is the fraction of the word spent in state j accumulated up to the observation of $\mathbf{x}$, p_j is the duration probability of state j; $\tilde{\epsilon}$ is the quantized log energy of the current frame, w_j is the log energy probability of state j, and γ_D and γ_E are experimentally determined positive scaling constants for the log probabilities of the duration and the log energy respectively.

The effective length, $\hat{K}$, needs further explanation. Speech signal analysis is generally performed on frames of K samples, with consecutive frames being taken after a shift of K_s samples. A speech signal of L_T samples is hence segmented into L_T/K_s frames. These frames are overlapped if the analysis frame size, K, satisfies $K > K_s$. Then, $\mathbf{x}$ is used to represent K samples of the speech signal and $[\delta(\mathbf{x};\mathbf{a}_{jm})/K] - 1$ becomes the well known likelihood ratio distortion measure. If we denote this quantity by $d_{LR}(\mathbf{x};\mathbf{a}_{jm})$, Eq. (11) becomes

$$b_{jm}(\mathbf{x}) = (2\pi)^{-K/2}\exp(-K/2)\exp\{-\frac{K}{2}d_{LR}(\mathbf{x};\mathbf{a}_{jm})\}. \quad (13)$$

It is therefore clear that $d_{LR}(\mathbf{x},\mathbf{a}_{jm})$ can be regarded as the average cross entropy *per sample* between the observed vector $\mathbf{x}$ and an autoregressive source characterized by $\mathbf{a}_{jm}$. Using $\{\mathbf{x}_i\}$ and $\{\mathbf{a}_i\}$, $i = 1,2,\ldots,L_T/K_s$, to denote the L_T/K_s frames of speech data and the sequence of autoregressive sources for comparison respectively, we note that the cross entropy between $\{\mathbf{x}_i\}$ and $\{\mathbf{a}_i\}$, with a memoryless vector source assumption, is

$$K_s\sum_{i=1}^{L_T/K_s} d_{LR}(\mathbf{x}_i;\mathbf{a}_i) = K_s\sum_{i=1}^{L_T/K_s}[\frac{1}{K}\delta(\mathbf{x}_i;\mathbf{a}_i) - 1] \quad (14)$$

if the original time scale of the speech samples is to be maintained, independent of the analysis length K. This becomes more crucial when an underlying Markov chain, instead of a memoryless source, is assumed, since the transition structure contributes to the cross entropy. $\hat{K}$ in Eq. (12) thus allows adjustment on the relative cross entropy contributions between spectral parameters and the Markov chain. The relative rate of information in the model contributed by spectral, durational and energy parameters, respectively, can therefore be adjusted by the three scaling factors, $\hat{K}$, γ_D and γ_E.

V. Isolated Digit Recognition Experiments

One direct application of the above modeling/estimation technique is in isolated digit recognition. To evaluate the performance of the HMM recognizer with the GAM and PGAM mixture densities, a series of experiments was run using a database of isolated digits recorded over standard dialed-up telephone lines. The database consists of four sets of spoken digits, description of which can be found in [6].

5.1 The HMM Recognizer

For training the recognizer, for each digit in the training set, the 100 versions were first clustered into 2 sets (of about 50 versions each), and for each set a left-to-right HMM was designed using the procedure given in Section III. Thus, the output of the training procedure was a set of 2 HMM's per digit. Each HMM had N=5 states, with M=5 mixture densities per state for both the GAM and PGAM models.

For testing the recognizer, each of the 20 models (2 models × 10 digits) was scored using a Viterbi alignment procedure to give the optimal state sequence alignment of the unknown observation sequence to the input model. The distance measure of Eq. (12) was used to score the optimal alignment path.

5.2 Experimental Results

Several experiments were run using different training sets, model types (GAM, PGAM), and choices for γ_D and γ_E, the scaling constants in the distance measure, and the results of these experiments are given in Table I. Based on preliminary experimentation it was found that a value of $\gamma_D = 10.0$ (with $\gamma_E = 0$) gave the best performance when duration (without energy) was incorporated into the distance measure. As such the results in Table I primarily show the effects of different values of γ_E when γ_D was either 0 (no duration used) or 10.0 (the best duration scaling value).

The results given in Table I show that the 1) performance of the two model types (GAM and PGAM) was almost identical across all test conditions. To gain perspective into how the performance of these recognizer compares to that of previous HMM recognizer, the bottom two lines of Table I summarize results from previous work with a cepstral model using diagonal covariance matrices (CEP/DC) [6]. It can be seen that the best GAM performance is about 1% worse than the CEP/DC model performance, and the best PGAM performance is about 1.2% worse than the CEP/DC model performance.

Training Set	Model Type	γ_D	γ_E	Average Digit Error Rate (%) DIG1	DIG2	DIG3	DIG4	Test Set Average
DIG1	GAM	0.0	0.0	1.2	3.3	5.8	9.2	6.1
DIG1	GAM	0.0	3.0	0.9	2.3	4.3	6.0	4.2
DIG1	GAM	10.0	1.0	0.3	2.1	3.7	5.2	3.67
DIG1	GAM	10.0	3.0	0.3	1.8	3.4	4.1	3.1
DIG1	GAM	10.0	10.0	0.4	1.2	3.5	4.6	3.1
DIG4	GAM	0.0	0.0	7.1	7.4	4.1	1.7	6.2
DIG4	GAM	0.0	3.0	4.9	3.9	3.1	0.8	3.98
DIG4	GAM	10.0	1.0	5.7	4.3	2.7	0.5	4.23
DIG4	GAM	10.0	3.0	4.9	3.5	2.9	0.5	3.7
DIG4	GAM	10.0	10.0	3.9	2.9	3.7	0.8	3.5
DIG1	PGAM	0.0	0.0	1.1	4.1	5.3	8.3	5.9
DIG1	PGAM	0.0	3.0	1.0	2.8	3.8	6.4	4.67
DIG1	PGAM	10.0	1.0	0.7	2.1	3.4	6.0	3.83
DIG1	PGAM	10.0	3.0	0.7	2.0	3.1	5.9	3.67
DIG1	PGAM	10.0	10.0	0.5	1.7	3.6	5.0	3.43
DIG4	PGAM	0.0	0.0	7.0	7.3	4.0	1.6	6.1
DIG4	PGAM	0.0	3.0	5.3	4.9	2.9	1.4	4.33
DIG4	PGAM	10.0	1.0	5.3	4.2	3.2	1.1	4.23
DIG4	PGAM	10.0	3.0	4.9	3.4	2.3	0.8	3.53
DIG4	PGAM	10.0	10.0	3.4	2.9	3.7	1.0	3.33
DIG1	CEP/DC	10.0	–	0.1	0.7	2.8	4.2	2.57
DIG4	CEP/DC	10.0	–	2.5	1.7	2.1	0.8	2.1

Table I

Comparison of Performance of the HMM Recognizer as a function of the Training Set, Model Type, γ_D, and γ_E.

5.3 Computational Complexity

The computation required in the recognizer, using a Viterbi decoding algorithm, is

$$C_V = N \cdot T \cdot (p+1) M \cdot V \quad \text{multiplication/addition operations}$$

for a V word vocabulary, average word length T frames, N states per model, M mixtures per state, and pth order LPC representation.

A standard DTW recognizer requires

$$C_{\text{DTW}} = Q \cdot V \cdot \frac{T^2}{3} (p+1) \quad \text{multiplication/addition operations}$$

for a Q template per word system. The ratio of computation is thus

$$\text{RATIO} = \frac{C_V}{C_{\text{DTW}}} = \frac{NM}{QT/3}$$

which for $N = M = 5$, $Q = 12$, $T = 40$, gives

$$\text{RATIO} = \frac{5}{32} = \frac{1}{6.4}$$

i.e. a 6.4 to 1 reduction in computation is achieved for the LPC HMM recognizer over the standard DTW recognizer.

The computational advantage of LPC HMM over CEP/DC is mainly in the distortion computation. LPC HMM requires computation of a dot product as in Eq. (9), while CEP/DC requires computation of a weighted Euclidean distance. Practically speaking, of a reduction in computation of from 2 to 4 can be achieved.

VI. Discussion and Summary

We have shown that the ubiquitous LPC analysis technique can be consistently welded into the general hidden Markov model methodology. The resultant autoregressive HMM's are powerful for modeling time-varying signal sources. For short speech signals such as discrete digit utterances, however, such an extensive stochastic modeling may not be necessary. A simple segment registration procedure may be adequate in coping with the sequentially changing characteristics of speech. We have also demonstrated how the two procedures, namely the segmental k-means and the probabilistic Markov chain, can be successfully combined. Since the segmental k-means procedure gives good initial model estimates, the models resulting from reestimation always worked well in practice.

REFERENCES

[1] A. B. Poritz, "Linear Predictive Hidden Markov Models and the Speech Signal," *Proc. ICASSP 82,* pp. 1291-1294, Paris, France, May 1982.

[2] B. H. Juang, "On Hidden Markov Model and Dynamic Time Warping for Speech Recognition — A Unified View," *AT&T BLTJ,* Vol. 63, No. 7, pp. 1213-1243, September 1984.

[3] L. E. Baum and J. A. Eagon, "An inequality with Applications to Statistical Prediction for Functions of Markov Processes and to a Model for Ecology," *Bull. Amer. Math. Soc.,* 73 (1963), 360-363.

[4] F. Itakura, *Speech Analysis and Synthesis Systems Based on Statistical Method,* Doctor of Engineering Dissertation, Dept. of Engineering, Nagoya University, Japan, 1972 (in Japanese).

[5] B. H. Juang, L. R. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals," *IEEE Trans. Acoustics, Speech, and Signal Processing,* Vol. ASSP-33, No. 6, Dec. 1985.

[6] L. R. Rabiner, B. H. Juang, S. E. Levinson and M. M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities," submitted for publication.

[7] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech Coding Based Upon Vector Quantization," IEEE Trans. ASSP-28, pp. 562-574, Oct. 1980.