Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers

Vassilios V. Digalakis, Peter Monaco, and Hy Murveit, *Associate Member, IEEE*

Abstract— **An algorithm is proposed that achieves a good tradeoff between modeling resolution and robustness by using a new, general scheme for tying of mixture components in continuous mixture-density hidden Markov model (HMM)-based speech recognizers. The sets of HMM states that share the same mixture components are determined automatically using agglomerative clustering techniques. Experimental results on ARPA's Wall Street Journal corpus show that this scheme reduces errors by 25% over typical tied-mixture systems. New fast algorithms for computing Gaussian likelihoods—the most time-consuming aspect of continuous-density HMM systems—are also presented. These new algorithms significantly reduce the number of Gaussian densities that are evaluated with little or no impact on speech recognition accuracy.**

I. Introduction

HIDDEN Markov model (HMM)-based speech recognizers with *tied-mixture* (TM) observation densities [1]–[3] achieve robust estimation and efficient computation of the density likelihoods. However, the typical mixture size used in TM systems is small and does not accurately represent the acoustic space. Increasing the number of the mixture components (also known as the codebook size) is not a feasible solution, since the mixture-weight distributions become too sparse. In large-vocabulary problems, where a large number of basic HMM's is used and each has only a few observations in the training data, sparse mixture-weight distributions cannot be estimated robustly and are expensive to store.

HMM's with continuous mixture densities and no tying constraints (*fully continuous* HMM's), in contrast, provide a detailed stochastic representation of the acoustic space at the expense of increased computational complexity and lack of robustness. Each HMM state has associated with it a different set of mixture components that are expensive to evaluate and cannot be estimated robustly when the number of observations per state in the training data is small. A detailed representation is critical for large-vocabulary speech recognition. It has recently been shown [4] that, in large-vocabulary recognition tasks, HMM's with continuous mixture densities and no tying consistently outperform HMM's with tied-mixture densities. To overcome the robustness issue, continuous HMM systems use various schemes. Gauvain [5] smooths the mixture-component parameters with maximum *a-posteriori* (MAP) estimation and implicitly clusters models that have small amounts of training via backoff mechanisms. Woodland in [6] uses clustering at the HMM state level and estimates mixture densities only for clustered states with enough observations in the training data.

In this work, and in order to achieve the optimum tradeoff between acoustic resolution and robustness, we choose to generalize the tying of mixture components. From the fully continuous HMM perspective, we improve the robustness by sharing the same mixture components among arbitrarily defined sets of HMM states. From the tied-mixture HMM perspective, we improve the acoustic resolution by simultaneously increasing the number of different sets of mixture components (or codebooks) and reducing each codebook's size. These two changes can be balanced so that the total number of component densities in the system is effectively increased. We propose a new algorithm that automatically determines the sets of HMM states that will share the same mixture components. The algorithm can also be viewed as a method that transforms a system with a high degree of tying among the mixture components to a system with a smaller degree of tying. The appropriate degree of tying for a particular task depends on the difficulty of the task, the amount of available training data, and the available computational resources for recognition, since systems with a smaller degree of tying have higher computational demands during recognition.

In Section II of this paper we present the general form of mixture observation distributions used in HMM's and discuss previous work and variations of this form that have appeared in the literature. In Section III we present the main algorithm. In Section IV we present word recognition results using ARPA's Wall Street Journal speech corpus. To deal with the increased amount of computation that continuous-density HMM's require during decoding, we present algorithms for the fast evaluation of Gaussian likelihoods in Section V. Conclusions are given in Section VI.

II. Mixture Observation Densities in HMM's

A typical mixture observation distribution in an HMM-based speech recognizer has the form

$$\boldsymbol{p}(x_t|s) = \sum_{q \in Q(s)} \boldsymbol{p}(q|s) f(x_t|q) \tag{1}$$

Manuscript received June 9, 1994; revised November 1, 1995. This work was supported by the Advanced Research Projects Agency under Contract ONR N00014-92-C-0154. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xuedong Huang.

V. V. Digalakis was with the Speech Technology and Research Laboratory, SRI International, Menlo Park, CA. He is now with the Electronic and Computer Engineering Department, Technical University of Crete, Kounoupidiana, Chania, 73100 Greece.

P. Monaco and H. Murveit were with SRI International, Menlo Park, CA 94025 USA. They are now with Nuance Communications, 333 Ravenswood Avenue, Menlo Park, CA 94025 USA.

Publisher Item Identifier S 1063-6676(96)05069-9.

where s represents the HMM state, x_t the observed feature at frame t, and $Q(s)$ the set of mixture-component densities used in state s. We shall use the term *codebook* to denote the set $Q(s)$. The stream of continuous vector observations can be modeled directly using Gaussians or other types of densities in the place of $f(x_t|q)$, and HMM's with this form of observation distributions appear in the literature as continuous HMM's [7].

Various forms of tying have appeared in the literature. When tying is not used, the sets of component densities are disjoint for different HMM states—that is, $Q(s) \cap Q(s') = \emptyset$ if $s \neq s'$. We shall refer to HMM's that use no sharing of mixture components as *fully continuous* HMM's.

To overcome the robustness and computation issues, the other extreme has also appeared in the literature: all HMM states share the same set of mixture components—that is, $Q(s) = Q$ is independent of the state s. HMM's with this degree of sharing were proposed in [1]–[3] under the names *Semi-Continuous* and *Tied-Mixture* HMM's. Tied-mixture distributions have also been used with segment-based models, and a good review is given in [8]. The relative performance of tied-mixture and fully continuous HMM's usually depends on the amount of the available training data. Until recently, it was believed that with small to moderate amounts of training data, tied-mixture HMM's outperform fully continuous ones, but that with larger amounts of training data and appropriate smoothing fully continuous HMM's perform better [2],[4]. However, as we shall see in the remainder of this paper, continuous HMM's with appropriate tying and smoothing mechanisms can outperform tied-mixture ones even with small to moderate amounts of training.

Intermediate degrees of tying have also been examined. In phone-based tying, described in [9]–[11], only HMM states that belong to allophones of the same phone share the same mixture components—that is, $Q(s) = Q(s')$ if s and s' are states of context-dependent HMM's with the same center phone. We will use the term *phonetically tied* to describe this kind of tying. Of course, for context-independent models, phonetically tied and fully continuous HMM's are equivalent. However, phonetically tied mixtures (PTM's) did not significantly improve recognition performance in previous work.

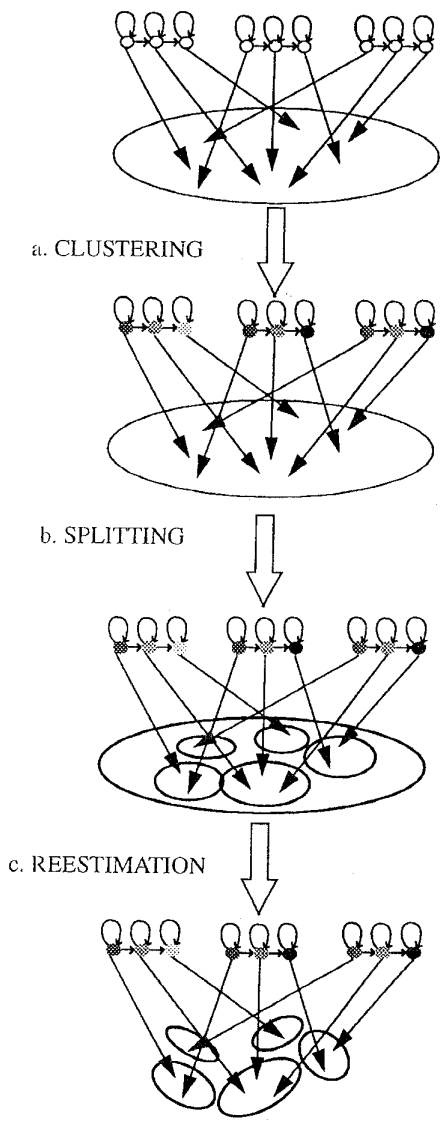

Fig. 1. Construction of genonic mixtures. Arrows represent the stochastic mappings from state to mixture component. Ellipses represent the sets of Gaussians in a single genone.

III. Genonic Mixtures

The continuum between fully continuous and tied-mixture HMM's can be sampled at any point. The choice of PTM's, although linguistically motivated, is somewhat arbitrary and may not achieve the optimum tradeoff between resolution and trainability. We prefer to optimize performance by using an automatic procedure to identify subsets of HMM states that will share mixtures. The algorithm that we propose follows a bootstrap approach from a system that has a higher degree of tying (i.e., a TM or a PTM system), and progressively unties the mixtures using three steps: clustering, splitting and reestimation (Fig. 1).

A. Clustering

In the first step of the algorithm (see Fig. 1(a)), the HMM states of all allophones of a phone are clustered following an agglomerative hierarchical clustering procedure [12]. The states are clustered based on the similarity of their mixture-weight distributions. Any measure of dissimilarity between two discrete probability distributions can be used as the distortion measure during clustering. In [1], Huang pointed out that the probability density functions of HMM states should be shared using information-theoretic clustering. Following Lee [14] and Hwang [15], we use the increase in the weighted-by-counts entropy of the mixture-weight distributions that is caused by the merging of the two states. Let $H(s)$ denote the entropy of the discrete distribution $[\boldsymbol{p}(q|s), q \in Q(s)]$,

$$H(s) = -\sum_{q \in Q(s)} \boldsymbol{p}(q|s) \log \boldsymbol{p}(q|s). \tag{2}$$

Then, the distortion that occurs when two states s_1 and s_2 with $Q(s_1) = Q(s_2)$ are clustered together into the clustered

state s is defined as

$$d(s_1, s_2) = (n_1 + n_2)H(s) - n_1 H(s_1) - n_2 H(s_2) \quad (3)$$

where n_1, n_2 represent the number of observations used to estimate the mixture-weight distributions of the states s_1, s_2, respectively. The mixture-weight distribution of the clustered state s is

$$\boldsymbol{p}(q|s) = \frac{n_1}{n_1 + n_2}\boldsymbol{p}(q|s_1) + \frac{n_2}{n_1 + n + 2}\boldsymbol{p}(q|s_2), \quad (4)$$

and the clustered state uses the same set of mixture components as the original states, $Q(s) = Q(s_1) = Q(s_2)$. This distortion measure can be easily shown to be nonnegative, and in addition, $d(s, s) = 0$.

The clustering procedure partitions the set of HMM states S into disjoint sets of states

$$S = S_1 \cup S_2 \cup \cdots \cup S_n \quad (5)$$

where n, the number of clusters, is determined empirically.

The same codebook will be used for all HMM states belonging to a particular cluster S_i. Each state in the cluster will, however, retain its own set of mixture weights.

B. Splitting

Once the sets of HMM states that will share the same codebook are determined, seed codebooks for each set of states that will be used by the next reestimation phase are constructed (see Fig. 1(b)). These seed codebooks can have a smaller number of component densities, since they are shared by fewer HMM states than the original codebook. They can be constructed by either one or a combination of two procedures:

- identifying the most likely subset $Q(S_i) \subset Q(S)$ of mixture components for each cluster of HMM states S_i, and using a copy of that subset in the next phase as the seed codebook for states in S_i;
- using a copy of the original codebook for each cluster of states. The number of component densities in each codebook can then be clustered down (see Section V-A) after performing one iteration of the Baum-Welch algorithm over the training data with the new relaxed tying scheme.

The clustering and splitting steps of the algorithm define a mapping from HMM state to cluster index

$$g = \gamma(s) \quad (6)$$

as well as the set of mixture components that will be used by each state, $Q(s) = Q(g)$.

C. Reestimation

The parameters are reestimated using the Baum-Welch algorithm. This step allows the codebooks to deviate from the initial values (see Fig. 1(c)) and achieve a better approximation of the distributions.

We shall refer to the Gaussian codebooks as *genones*[1] and to the HMM's with arbitrary tying of Gaussian mixtures as *genonic* HMM's. The group in CMU was the first that succeeded in using state clustering for large-vocabulary speech recognition [13]. Clustering of either phone or subphone units in HMM's has also been used in [14]–[17]. Mixture-weight clustering of different HMM states can reduce the number of free parameters in the system and, potentially, improve recognition performance because of the more robust estimation. It cannot, however, improve the resolution with which the acoustic space is represented, since the total number of component densities in the system remains the same. In our approach, we use clustering to identify sets of subphonetic regions that will share mixture components. The subsequent steps of the algorithm increase the number of distinct densities in the system and provide the desired detail in the resolution.

Reestimation of the parameters can be achieved using the standard Baum–Welch reestimation formulae (see, e.g., [3] for the case of tied-mixture HMM's). For arbitrary tying of mixture components and Gaussian component densities, the observation distributions become

$$\boldsymbol{p}(x_t|s) = \sum_{q \in Q(g)} \boldsymbol{p}(q|s) N_{gq}(x_t; \mu_{gq}, \Sigma_{gq}) \quad (7)$$

where $N_{gq}(x_t; \mu_{gq}, \Sigma_{gq})$ is the q-th Gaussian of genone g. It can be easily verified that the Baum–Welch reestimation formulae for the means and the covariances become

$$\hat{\mu}_{gq} = \frac{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \boldsymbol{n}_t(j, q) x_t}{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \boldsymbol{n}_t(j, q)} \quad (8)$$

and

$$\hat{\Sigma}_{gq} = \frac{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \boldsymbol{n}_t(j, q)(x_t - \hat{\mu}_{gq})(x_t - \hat{\mu}_{gq})^T}{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \boldsymbol{n}_t(j, q)} \quad (9)$$

where the first summation is over all states s_j in the inverse image $\gamma^{-1}(g)$ of the genonic index g. The accumulation weights in the equations above are

$$\boldsymbol{n}_t(j, q) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_j \alpha_t(j)\beta_t(j)}\right] \cdot \left[\frac{\boldsymbol{p}(q|s_j) N_{gq}(x_t; \mu_{gq}, \Sigma_{gq})}{\sum_q \boldsymbol{p}(q|s_j) N_{gq}(x_t; \mu_{gq}, \Sigma_{gq})}\right] \quad (10)$$

where μ_{gq}, Σ_{gq} are the initial mean and covariance, the summations in the denominator are over all HMM states and all mixture components in a particular genone, respectively, and the quantities $\alpha_t(j), \beta_t(j)$ are obtained using the familiar forward and backward recursions of the Baum–Welch algorithm [18]. The reestimation formulae for the remaining HMM parameters—i.e., mixture weights, transition probabilities, and initial probabilities—are the same as those presented in [3].

[1] This term should be partially attributed to IBM's fenones and CMU's senones. A genone is a set of Gaussians shared by a set of states and should not be confused with the word genome.

To reduce the large amount of computation involved in evaluating Gaussian likelihoods during recognition, we have developed fast computation schemes that are described in Section V.

IV. Word Recognition Experiments

We evaluated genonic HMM's on the Wall Street Journal (WSJ) corpus [19]. We used SRI's DECIPHER™ continuous speech recognition system, configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from an FFT filterbank. Context-dependent phonetic models were used, and the inventory of the triphones was determined by the number of occurrences of the triphones in the training data. The corresponding biphone or context-independent models were used for triphones that did not appear in the training data. In all of our experiments we used Gaussian distributions with diagonal covariance matrices as the mixture component densities. For fast experimentation, we used the progressive-search framework [20]. With this approach, an initial fast recognition pass creates word lattices for all sentences in the development set. These word lattices are used to constrain the search space in all subsequent experiments. To avoid errors due to decisions in the early stages of the decoding process, the lattice error rate[2] was less than 2% in all experiments. In both the lattice-construction and lattice-rescoring phases, context-dependent models across word boundaries were used only for these words and contexts that occured in the training data a number of times that exceeded a prespecified threshold.

In our development we used both the WSJ0 5000-word and the WSJ1 64 000-word portions of the database. The systems used in the WSJ0 experiments had 2200 context-dependent three-state left-to-right phonetic models with a total of 6600 state distributions. Due to the larger amount of training, the corresponding numbers for the WSJ1 experiments increased to 7000 phonetic models and 21 000 state distributions. Each one of these state distributions was associated to a different set of mixture weights. The TM and PTM systems were trained using two context-independent followed by two context-dependent iterations of the Baum-Welch algorithm. The genonic systems were booted using the context-dependent PTM system and were trained following the procedure described in Section III.

We used the baseline bigram and trigram language models provided by Lincoln Laboratory—5000-word closed-vocabulary,[3] and 20 000-word open-vocabulary language models were used for the WSJ0 and WSJ1 experiments, respectively. The trigram language model was implemented using the N-best rescoring paradigm [21], by rescoring the list of the N-best sentence hypotheses generated using the bigram language model.

In the remainder of this section, we present results that show how mixture tying affects recognition performance. We also present experiments that investigate other modeling aspects of continuous HMM's, including modeling multiple vs. single observation streams and modeling time-correlation using linear discriminant analysis.

[2] The lattice error rate is defined as the word error rate of the path through the word lattice that provides the best match to the reference string.

[3] A closed-vocabulary language model is intended for recognizing speech that does not include words outside of the vocabulary.

TABLE I
Comparison of Various Degrees of Tying on a 5,000-Word WSJ0 Development Set

System	Number of Genones	Gaussians per genone	Total parameters (thousands)	Word Error (%)
TM	1	256	5,126	14.1
PTM	40	100	2,096	11.6
Genones	495	48	1,530	10.6

A. Degree of Mixture Tying

To determine the effect of mixture tying on the recognition performance, we evaluated a number of different systems on both WSJ0 and WSJ1. Table I compares the performance and the number of free parameters of tied mixtures, PTM's, and genonic mixtures on a development set that consists of 18 male speakers and 360 sentences of the 5000-word WSJ0 task. The training data for this experiment included 3500 sentences from 42 speakers. We can see that systems with a smaller degree of tying outperform the conventional tied mixtures by 25%, and at the same time have a smaller number of free parameters because of the reduction in the codebook size.

The difference in recognition performance between PTM and genonic HMM's is, however, much more dramatic in the WSJ1 portion of the database. There, the training data consisted of 37 000 sentences from 280 speakers, and gender-dependent models were built. The male subset of the 20 000-word, November 1992 evaluation set was used, with a bigram language model. Table II compares various degrees of tying by varying the number of genones used in the system. We can see that because of the larger amount of available training data, the improvement in performance of genonic systems over PTM systems is much larger (20%) than in our 5000-word experiments. Moreover, the best performance is achieved for a larger number of genones—1700 instead of the 495 used in the 5000-word experiments. These results are depicted in Fig. 2.

In Table III we explore the additional degree of freedom that genonic HMM's have over fully continuous HMM's, namely that states mapped to the same genone can have different mixture weights. We can see that tying the mixture weights in addition to the Gaussians introduces a significant degradation in recognition performance. This degradation increases when the features are modeled using multiple observation streams (see Section IV-B) and as the amount of training data and the number of genones decrease. When all states using the same genone have tied (i.e., the same) mixture weights, then this genonic system is effectively a tied-state system, and the number of clustered states is equal to the number of genones.

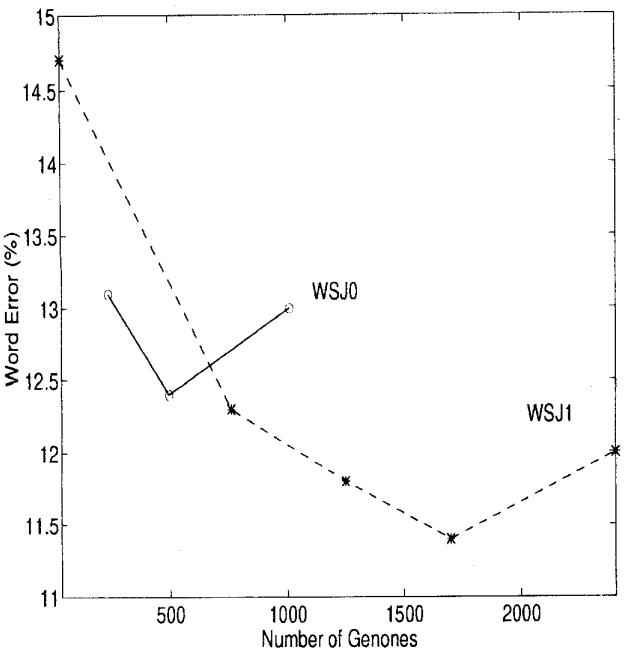

Fig. 2. Recognition performance for different degrees of tying on the 5000-word WSJ0 and 20 000-word WSJ1 tasks of the WSJ corpus.

TABLE II
RECOGNITION PERFORMANCE ON THE MALE SUBSET OF 20,000-WORD WSJ NOVEMBER 1992 ARPA EVALUATION SET FOR VARIOUS NUMBERS OF CODEBOOKS USING A BIGRAM LANGUAGE MODEL

	PTM	Genonic HMMs			
Number of genones	40	760	1250	1700	2400
Word error rate (%)	14.7	12.3	11.8	11.4	12.0

TABLE III
COMPARISON OF STATE-SPECIFIC VSERSUS GENONE-SPECIFIC MIXTURE WEIGHTS FOR DIFFERENT RECOGNITION TASKS

Recognition Task	Number of Genones	Number of Streams	Word Error (%)	
			Tied	Untied
5K WSJ0	495	6	9.7	7.7
20K WSJ1	1,700	1	12.2	11.4

B. Multiple versus Single Observation Streams

Another traditional difference between fully continuous and tied-mixture systems is the independence assumption of the latter when modeling multiple speech features. Tied-mixture systems typically model static and dynamic spectral and energy features as conditionally independent observation streams, given the HMM state. The reason is that tied-mixture systems provide a very coarse representation of the acoustic space, which makes it necessary to quantize each feature separately and artificially increase the resolution by modeling the features as independent. Then, the number of bins of the augmented feature is equal to the product of the number of bins of all individual features. The disadvantage is, of course, the independence assumption. When, however, the degree of tying is smaller, the finer representation of the acoustic space makes it unnecessary to improve the resolution accuracy by modeling the features as independent, and the feature-independence assumption can be removed. This claim is verified experimentally in Table IV. The first row in Table IV shows the recognition performance of a system that models the six static and dynamic spectral and energy features as independent observation streams. The second row shows the performance of a system that models the six features in a single stream. We can see that the performance of the two systems is similar, with the single-stream system performing insignificantly better.

TABLE IV
COMPARISON OF MODELING USING SIX VERSUS ONE OBSERVATION STREAMS FOR THE SIX UNDERLYING FEATURES ON THE MALE SUBSET OF 20 000-WORD WSJ NOVEMBER 1992 EVALUATION SET WITH A BIGRAM LANGUAGE MODEL

System	Sub (%)	Del (%)	Ins (%)	Word Error (%)
6 streams	9.0	0.8	2.5	12.3
1 stream	8.7	0.8	2.3	11.8

C. Linear Discriminant Features

For a given HMM state sequence, the observed features at nearby frames are highly correlated. HMM's, however, model these observations as conditionally independent, given the underlying state sequence. To capture local time correlation, we used a technique similar to the one described in [11]. Specifically, we used a linear discriminant feature extracted using a linear transformation of the vector consisting of the cepstral and energy features within a window centered around the current analysis frame. The discriminant transformation was obtained using linear discriminant analysis [12] with classes defined as the HMM state of the context-independent phone. The state index assigned to each frame was determined using the maximum *a-posteriori* criterion and the forward-backward algorithm. Gender-specific transformations were used, with a window size of three frames. The original 3×39-dimensional vector consisting of the cepstral coefficients, the energy and their first and second order derivatives over the three-frame window was transformed using the discriminant transformation to a lower-dimensional vector. We experimented with various sizes of the transformed vector, and found that for our experimental conditions a 35-D vector performed better.

We found that the performance of the linear discriminant feature was similar to that of the original features, and that performance improves if the discriminant feature vector is used in parallel with the original cepstral features as a separate observation stream. The two streams were assigned equal weights during decoding. From Table V, we can see that the linear discriminant feature reduced the error rate on the WSJ1 20 000-word open-vocabulary male development set by approximately 7% using either a bigram or a trigram language model.

TABLE V
WORD ERROR RATES (%) ON THE 20,000-WORD OPEN-VOCABULARY MALE DEVELOPMENT SET OF THE WSJ1 CORPUS WITH AND WITHOUT LINEAR DISCRIMINANT TRANSFORMATIONS

System	Bigram LM	Trigram LM
1,700 Genones	20.5	17.0
+ Linear Discriminants	19.1	15.8

TABLE VI
WORD ERROR RATES ON THE NOVEMBER 1992 EVALUATION, THE WSJ1 DEVELOPMENT, AND THE NOVEMBER 1993 EVALUATION SETS USING 20,000-WORD OPEN-VOCABULARY BIGRAM AND TRIGRAM LANGUAGE MODELS

	Test set		
Grammar	Nov92	WSJ1 Dev	Nov93
Bigram	11.2	16.6	16.2
Trigram	9.3	13.6	13.6

The best-performing system with 1700 genones and the linear discriminant feature was then evaluated on various test and development sets of the WSJ database using bigram and trigram language models. Our word recognition results, summarized in Table VI, are comparable to the best reported results to date on these test sets [4]–[6].

V. REDUCING GAUSSIAN COMPUTATIONS

Genonic HMM recognition systems require evaluation of very large numbers of Gaussian distributions, and can be very slow during recognition. In this section, we will show how to reduce this computation while maintaining recognition accuracy. For simplicity, we use a baseline system in this section that has 589 genones, each with 48 Gaussian distributions, for a total of 28 272 39-D Gaussians. This system has a smaller number of genones than the best-performing system of Section IV and no context-dependent modeling across words. It runs much faster than our most accurate system, but its performance of 13.4% word error on ARPA's November 1992, 20 000-word evaluation test set using a bigram language model is slightly worse than our best result of 11.4% on this test set when the linear discriminant feature is not used (Table II). Decoding time from word lattices is 12.2 times slower than real time on an R4400 processor. The term "real time" may at first appear mis-leading when used for word-lattice decoding, since we are only performing a subset of the search. In a conventional Viterbi-decoding system, actual, full-grammar recognition times could be from a factor of three to an order of magnitude higher. We can, however, follow a multipass approach and use a discrete-density system for the first (i.e., the lattice-building pass) with a grammar organized as a lexicon tree [22]: this first pass can be faster than real time, in which case the full decoding is dominated by the subsequent, lattice-decoding pass with the more expensive computationally genonic system. Moreover, we have found in our experiments that in both lattice and full-grammar decoding using genonic HMM's, the computation is dominated by the evaluation of the Gaussian likelihoods. Hence, reducing the number of Gaussians that require evaluation at each frame is critical for both fast experimentation and practical applications of the technology. We have explored two methods of reducing Gaussian computation: Gaussian clustering and Gaussian shortlists, and we have used word lattices to evaluate our algorithms.

TABLE VII
IMPROVED TRAINING OF SYSTEMS WITH FEWER GAUSSIANS BY CLUSTERING FROM A LARGER NUMBER OF GAUSSIANS

System	Gaussians per Genone	Word Error (%)
Baseline 1	48	13.4
Baseline 1 + clustering	18	14.2
Above + retraining	18	13.6
Baseline 2	25	14.4

A. Gaussian Clustering

The number of Gaussians per genone can be reduced using clustering. Specifically, we used an agglomerative procedure to cluster the component densities within each genone to a smaller number. We considered several criteria that were used in [23], like an entropy-based and a generalized likelihood-based distortion measure. We found that the entropy-based measure worked better. This criterion is the continuous-density analog of the increase in weighted-by-counts entropy of the discrete HMM mixture-weight distributions that we used in the agglomerative clustering step of the genonic HMM system construction. Specifically, the cost of pooling two Gaussian densities—$N_i(x_t; \mu_i, \Sigma_i)$ and $N_j(x_t; \mu_j, \Sigma_j)$—is the difference between the entropy of the pooled Gaussian and the sum of the entropies of the initial densities, all weighted by the number of samples used to estimate each density:

$$d(i,j) = \frac{n_i + n_j}{2} \log|\Sigma_{i\cup j}| - \frac{n_i}{2} \log|\Sigma_i| - \frac{n_j}{2} \log|\Sigma_j| \tag{11}$$

where n_i, n_j are the number of samples used to estimate the initial densities and $N_{i\cup j}(x_t; \mu_{i\cup j}, \Sigma_{i\cup j})$ is the pooled density.

In Table VII we can see that the number of Gaussians per genone can be reduced by a factor of three by first clustering and then performing one additional iteration of the Baum-Welch algorithm. The table also shows that clustering followed by additional training iterations gives better accuracy than directly training a system with a smaller number of Gaussians (Table VII, baseline 2). This is especially true as the number of Gaussians per genone decreases.

B. Gaussian Shortlists

Although clustering reduces the total number of Gaussians significantly, all the Gaussians belonging to genones used by

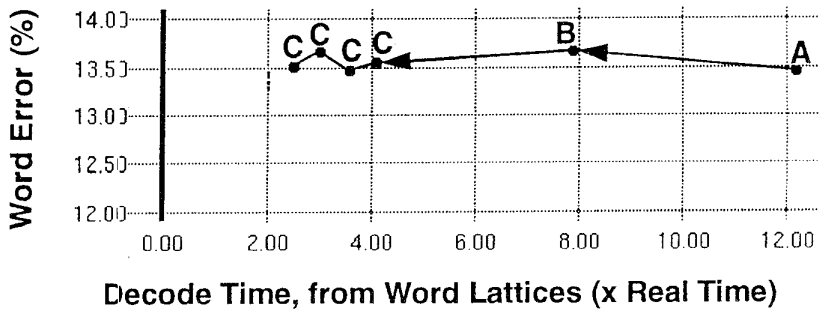

Fig. 3. Word error rate as a function of the decoding time for the baseline system (a) and systems with fast Gaussian evaluation schemes (b and c). (a) Unclustered system. (b) Clustered system. (c) Clustered system using various shortlists.

HMM states that are in the Viterbi beam search must be evaluated at each frame during recognition. This evaluation includes a large amount of redundant computation; we have verified experimentally that the majority of the Gaussians will yield negligible probabilities. As a result, after reducing the Gaussians by a factor of three using clustering, the decoding time from word lattices is still 7.9 times slower than real-time.

We have developed a method similar to the one introduced by Bocchieri [24] for preventing a large number of unnecessary Gaussian computations. Our method is to partition the acoustic space and for each partition to build a *Gaussian shortlist,* a list which specifies the subset of the Gaussian distributions expected to have high likelihood values in a given region of the acoustic space. First, vector quantization (VQ) is used to subdivide the acoustic space into VQ regions. Then, one list of Gaussians is created for each combination of VQ region and genone. The lists are created empirically, by considering a sufficiently large amount of speech data. For each acoustic observation, each Gaussian distribution is evaluated. Those distributions whose likelihoods are within a predetermined fraction of the most likely Gaussian are added to the list for that VQ region and genone. This is the main difference from the algorithm proposed by Bocchieri, where the groups of Gaussians for each VQ region are determined by only looking at the centroid of that region. Our scheme will result in some empty or too short lists. We have found that empty lists can cause a degradation in recognition performance, which can be avoided by enforcing a minimum shortlist size—we add to empty shortlists those Gaussians of the genone that achieve the highest likelihood for some observations quantized to the VQ region.

When recognizing speech, each observation is vector quantized, and only those Gaussians which are found in the shortlist are evaluated. This technique has allowed us to reduce by more than a factor of five the number of Gaussians considered each frame when applied to unclustered genonic recognition systems. Here we apply Gaussian shortlists to the clustered system described in Section V-A. Several methods for generating improved, smaller Gaussian shortlists are discussed and applied to the same system.

Table VIII shows the word error rates for shortlists generated by a variety of methods. Through these methods, we reduced the average number of Gaussian distributions evaluated for each genone from 18 to 2.48 without compromising accuracy. In contrast, the original method proposed by Bocchieri introduces a small degradation in recognition performance [24]. This improvement is achieved at the cost of a more expensive computationally shortlist-building phase, where we evaluate the Gaussian likelihoods for all genones and all observations in a training set. The various shortlists tested were generated in the following ways.

TABLE VIII
WORD ERROR RATES AND GAUSSIANS EVALUATED, FOR A VARIETY OF GAUSSIAN SHORTLISTS

Shortlist Type	Shortlist Length	Gaussians evaluated per frame	Word Error (%)
none	18	5459	13.6
12D-256	6.08	1964	13.5
39D-256	4.93	1449	13.5
39D-4096-min3	3.68	1088	13.6
39D-4096-min1	2.48	732	13.5

None: No shortlist was used. This is the baseline case from the clustered system described above. All 18 Gaussians are evaluated whenever a genone is active.

12D-256: To partition the acoustic space, the vector of 12 cepstral coefficients is quantized using a VQ codebook with 256 codewords. With unclustered systems, this method generally achieves a 5 : 1 reduction in Gaussian computation. In this clustered system, only a 3 : 1 reduction was achieved, most likely because the savings from clustering and Gaussian shortlists overlap. The average shortlist length was 6.08.

39D-256: The cepstral codebook that partitions the acoustic space in the previous method ignores 27 of the 39 feature dimensions. By using a 39-D 256-codeword VQ codebook, we created better-differentiated acoustic regions and reduced the average shortlist length to 4.93.

39D-4096-min3: We further decreased the number of Gaussians per region by shrinking the size of the regions. Here we used a single-feature VQ codebook with 4096 codewords, and reduced the average shortlist size to 3.68. For such a large codebook, vector quantization can be accelerated using a binary tree VQ fastmatch [25]. The minimum shortlist size was 3.

39D-4096-min1: In our experiments with 48 Gaussians/genone, we found it important to ensure that each list contained a minimum of three Gaussian densities. With our current clustered systems we found that we can achieve similar recognition accuracy with a minimum shortlist size of one. As shown in Table VIII, this technique results in lists with an average of 2.48 Gaussians per genone, without degradation in recognition accuracy.

Our results on the computational reduction on the evaluation of Gaussian likelihoods are summarized in Fig. 3. We started with a speech recognition system with 48 Gaussians per genone (a total of 28 272 Gaussian distributions) that evaluated 14 538 Gaussian likelihood scores per frame and achieved a 13.4% word error rate performing the lattice decoding phase 12.2 times slower than real-time. Combining the clustering and Gaussian shortlist techniques described in Section V, we managed to decrease the average number of Gaussians contained in each list to 2.48. As a result, the system's computational requirements were reduced to 732 Gaussian evaluations per frame, resulting in a system with word error of 13.5%, performing the lattice decoding phase at 2.5 times real-time.

VI. Conclusions

An algorithm has been developed that balances the tradeoff between resolution and trainability. Our method generalizes the tying of mixture components in continuous HMM's and achieves the degree of tying that is best suited to the available training data and the size of the recognition problem that we have in hand. We demonstrated in the large-vocabulary WSJ database that by selecting the appropriate degree of tying, the word-error rate can be decreased by 25% over conventional tied-mixture HMM's. To cope with the increase in computational requirements compared to tied-mixture HMM's, we have presented fast algorithms for evaluating the likelihoods of Gaussian mixtures. The number of Gaussians evaluated per frame was reduced by a factor of 20 and the decoding time by a factor of 6.

References

[1] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," in *Readings in Speech Recognition,* A. Waibel and K. F. Lee, Ed. New York: Morgan Kaufmann, 1990, pp. 340–346.

[2] X. D. Huang and M. A. Jack, "Performance comparison between semi-continuous and discrete hidden Markov models," *Electron. Lett.*, vol. 24, no. 3, pp. 149–150, 1988.

[3] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 2033–2045, Dec. 1990.

[4] D. Pallet, J. G. Fiscus, W. M. Fisher, and J. S. Garofolo, "1993 benchmark tests for the ARPA spoken language program," in *Proc. ARPA Workshop on Human Language Technology*, Princeton, NJ, Mar. 1994.

[5] J. L. Gauvain, L. F. Lamel, G. Adda, and M. Adda-Decker, "The LIMSI continuous speech dictation system: Evaluation on the ARPA Wall Street Journal Task," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1994, pp. I-125–I-128.

[6] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using HTK," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1994, pp. II-125–II-128.

[7] 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," *Bell Systems Tech. J.*, vol. 64, no. 6, , pp. 1211–34, 1985.

[8] O. Kimball and M. Ostendorf, "On the use of tied-mixture distributions," in *Proc. ARPA Workshop on Human Language Technology*, Mar. 1993.

[9] D. B. Paul, "The Lincoln robust continuous speech recognizer," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, May 1989, pp. 449–452.

[10] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic modeling for large vocabulary speech recognition," *Comput. Speech Lang.*, pp. 127–165, Apr. 1990.

[11] X. Aubert, R. Haeb-Umbach and H. Ney, "Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1993, pp. 648–651.

[12] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[13] X. D. Huang, K. F. Lee, H. W. Hon, and M. Y. Hwang, "Improved acoustic modeling with the SPHINX speech recognition system," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, May 1991, pp. 345–348.

[14] K. F. Lee, "Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 599–609, Apr. 1990.

[15] M.-Y. Hwang and X. D. Huang, "Subphonetic Modeling with Markov States–Senone," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Mar. 1992, pp. I-33–I-36..

[16] D. B. Paul and E. A. Martin, "Speaker stress-resistant continuous speech recognition," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1988, pp. 283–286.

[17] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Context dependent modeling of phones in continuous speech using decision trees," in *Proc. ARPA Workshop on Speech & Natural Language*, pp. 264–269, Feb. 1991.

[18] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[19] G. Doddington, "CSR corpus development," in *Proc. ARPA Workshop on Spoken Language Technology*, Feb 1992.

[20] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large vocabulary dictation using SRI's DECIPHER^{TM} speech recognition system: Progressive search techniques," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1993, pp. II-319–II-322.

[21] R. Schwartz and Y.-L. Chow, "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, pp. 701–704, May 1991.

[22] H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger, "Techniques to achieve an accurate real-time large-vocabulary speech recognition system," in *Proc. ARPA Workshop on Human Language Technology*, Mar. 1994, pp. 393–398.

[23] A. Kannan, M. Ostendorf, and J. R. Rohlicek, "Maximum likelihood clustering of gaussians for speech recognition," *IEEE Trans. Speech, Audio Processing*, vol. 2, pp. 453–455, July 1994.

[24] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *IEEE Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Apr. 1993, pp. II-692–II-695.

[25] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.

Vassilios V. Digalakis was born in Hania, Greece, on February 2, 1963. He received the Diploma in electrical engineering from the National Technical University of Athens, Greece, in 1986, the M.S. degree in electrical engineering from Northeastern University, Boston, MA, in 1988, and the Ph.D. degree in electrical and systems engineering from Boston University, Boston, MA, in 1992.

From 1986 to 1988 he was a Teaching and Research Assistant at Northeastern University. From 1988 to 1991 he served as a Research Assistant at Boston University, Boston, MA. From January 1992 to February 1995 he was with the Speech Technology and Research Laboratory, SRI International in Menlo Park, CA. At SRI, he was a principal investigator for SRI/ARPA research contracts and he developed new speech recognition and speaker adaptation algorithms for the DECIPHER speech recognition system. He also developed language education algorithms using speech recognition techniques. He is currently with the Department of Electronic and Computer engineering of the Technical University of Crete, Hania, where he holds an assistant professor position. His research interests are in pattern and speech recognition, information theory and digital communications.

Peter Monaco was born in Mountain View, CA, in 1967. He received the B.A. degree in engineering sciences in 1989 from Dartmouth College and the M.Phil. degree in computer speech processing in 1990 from Cambridge University, Cambridge, U.K.

From 1991 to 1994 he was employed at SRI International, Menlo Park, CA. He was the primary designer of SRI's Telephone Banking System, developing a DSP implementation of SRI's signal processing front end, tools for compilation and optimization of recognition grammars, echo-cancellation algorithms, telephone interface hardware, and an interpreter for a dialog scripting language. In addition he performed the necessary systems integration work to connect the system to SRI's Federal Credit Union computer, where it remains in service. Beginning in 1993, Peter focused on speed and memory optimizations within SRI's DECIPHER system. Concentrating on ARPA's Wall Street Journal task, Peter developed a real-time implementation of a 20,000 word recognition system which ran on industry standard hardware. In late 1994, along with Ron Croen, Hy Murveit, and Mike Cohen, he founded Corona Corporation, now known as Nuance Communications, where he serves as Director of Engineering. Nuance focuses on developing speech-recognition-based products for over-the-phone applications.

Hy Murveit (S'79–M'82–A'83) was born in Baltimore, MD in 1957. He received the B.Eng. degree in 1977 from SUNY Stony Brook in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1979 and 1983. His thesis focused on speech recognition algorithms used in special-purpose integrated circuits.

He was employed at SRI international, Menlo Park, CA, from 1983–1994 where he was a co-developer of the DECIPHER, MISTRI, and SRI/ATIS speech recognition/understanding systems. He was principal investigator for several SRI/ARPA research contracts and participated in several ARPA speech recognition benchmark tests. At SRI he investigated algorithms for speech recognition systems, integration of speech recognition and natural language processing, effective man-machine interfaces using speech, and architectures for real-time implementation of large speech-recognition systems. In late 1994, along with R. Croen, P. Monaco, and M. Cohen, he founded Corona Corporation, now known as Nuance Communications, where he serves as Vice President of Research and Development. Nuance focuses on developing speech-recognition-based products for over-the-phone applications.