EFFICIENT SPEECH RECOGNITION USING SUBVECTOR QUANTIZATION AND DISCRETE-MIXTURE HMMS

S. Tsakalidis[1], V. Digalakis[1,2] and L. Neumeyer[2]

(1) Dept. of Electronics and Computer Engineering
Technical University of Crete
Hania, 73100, GREECE

(2) SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, USA

ABSTRACT

This paper introduces a new form of observation distributions for hidden Markov models (HMMs), combining subvector quantization and mixtures of discrete distributions. We present efficient training and decoding algorithms for the discrete-mixture HMMs (DMHMMs). Our experimental results in the air-travel information domain show that the high-level of recognition accuracy of continuous mixture-density HMMs (CDHMMs) can be maintained at significantly faster decoding speeds. Moreover, we show that when the same number of mixture components is used in DMHMMs and CDHMMs, the new models exhibit superior recognition performance.

1. INTRODUCTION

In [1],[2] we developed a novel encoding scheme for the transmission of the mel-warped cepstral coefficients (MFCCs) in a client-server architecture for speech-enabled applications over the World Wide Web (WWW) and wireless channels. MFCCs are the parameters used by most state-of-the-art speech recognition systems today. By using subvector quantization and a bit-allocation algorithm that was driven by speech recognition performance, we were able to encode the 13 MFCCs using as little as 20 bits in noise-free environments, while maintaining the recognition performance of a high-quality front end. This is a rather surprising result, given that all hidden Markov Model (HMM)-based state-of-the-art recognition systems today represent the MFCCs using floating-point arithmetic and model their distributions with Gaussian mixtures.

The possibility of representing the MFCCs with a small number of bits, instead of the 416 (= 3 coefficients x 32 bits per coefficient) that are traditionally used in continuous-density Gaussian-mixture HMMs (CDHMMs), in addition to being advantageous for transmission and storage, has serious implications in acoustic modeling. Using Gaussian mixtures to model a set of coefficients that can be represented with 20 bits is clearly overkill. In this paper, we demonstrate that the high level of recognition performance of CDHMMs can be maintained with a far more efficient type of HMM, the discrete-mixture HMM (DMHMMs) with subvector quantization of the coefficient parameters.

Before CDHMMs became the model of choice in most state-of-the-art recognizers used in speech laboratories and speech-application companies worldwide, the first generation of HMM-based speech recognition systems was using discrete-distribution HMMs. The advantage of discrete HMMs over CDHMMs is the faster computation of the output HMM probabilities. Until now, however, it was generally believed that the recognition error rates that can be achieved with discrete distribution HMMs are a factor of one-and-a-half to two times higher than their continuous-density counterparts. There are many reasons why previous attempts with discrete HMMs did not achieve the level of performance of CDHMMs:

- In previous work, discrete HMMs did not quantize the acoustic space in sufficient detail. The 13 MFCCs were typically quantized using 8 bits, or 256 centroids. In contrast, in one of our implementations we quantize the 13 MFCCs with five subvectors and product-code vector quantization (VQ), assigning 5, 5, 4, 4 and 2 bits, respectively, to each subvector. This corresponds to a total of 32*32*16*16*4 = 1,048,576 centroids for the full vector.
- In our work we use a novel quantization scheme that is driven by speech-recognition performance, that optimally assigns bits to the subvectors that are more important for recognition.
- Subvector quantization by itself is not sufficient – it must be combined with mixtures of discrete distributions. An HMM system that uses subvector quantization and models the full vector with a product of discrete distributions will not capture the correlation between the different subvectors. In contrast, an HMM system that uses subvector quantization and models the full vector with a mixture of discrete distributions, each being the product of conditionally independent discrete subvector distributions, does model the correlation between the different subvectors.

In the remainder of this paper we shall show how a discrete HMM can be constructed that will achieve the level of performance of CDHMMs at much faster decoding speeds.

Related to the model that we propose in this paper is the work by Takahashi *et al.*[3], where scalar quantization was combined with mixtures of discrete distributions. However, as we shall see in Section 4, discretization of a CDHMM using scalar quantization results to a system that is actually slower than the original CDHMM. Bocchieri [4] has partitioned the observation vector into subvectors and used it with a continuous Gaussian HMM, clustering separately the Gaussians of the different subvectors.

2. SUBVECTOR QUANTIZATION

A client-server architecture for speech-enabled applications was presented in [1], [2]. The MFCCs were encoded at the client, the codebook indices were transmittted, then mapped at their corresponding centroids at the server side, and recognition was performed at the server, using CDHMMs.

In [1] we encoded the MFCCs using nonuniform scalar quantization of the MFCCs, and in [2] we presented an improved subspace quantization scheme of the cepstral coefficients. The MFCCs are partitioned into subvectors, and then the subvectors are encoded by using separate codebooks. The total number of codewords that represent the acoustic space is the product of the

number of codewords used for the representation of each subvector.

The feature vector can be partitioned into subvectors using automatic methods based on the estimated pairwise correlation coefficients of its elements. We found, however, that a simple approach, where the vector of MFCCs is partitioned into subvectors that contain consecutive coefficients, performs well.

Having formed the subvectors of the product code, one must allocate the bits among the respective codebooks. In [2] we introduced a bit-allocation algorithm that uses the word-error rate (WER) as a metric to optimally assign bits to the different subvectors. Specifically, we start with an initial bit allocation and then increase the bit rate by adding bits to the subvectors that yield the maximal incremental increase in recognition performance. The algorithm terminates once the maximum bit rate or the desired recognition performance has been reached.

3. DISCRETE-MIXTURE HMMS

3.1 Definition of Output Distribution

The encoding scheme that we described in Section 2 first partitions the vector of MFCCs into L subvectors, $x_t = [x_{1t}, \cdots, x_{Lt}]$, and then quantizes each subvector using a separate VQ codebook, $vq(x_t) = [vq(x_{1t}), \cdots, vq(x_{Lt})]$.

In the client-server architecture presented in [2], we found that by transmitting the codebook indices, mapping them at their corresponding centroids at the server, and using a CDHMM to perform recognition, the performance of the unquantized MFCCs was maintained.

Using the computationally expensive Gaussian distributions to model subvectors of a few MFCCs that can be encoded with two to three bits, however, is very inefficient. In this paper we propose to use a new form of discrete-mixture output distribution, $b_j(x_t)$, which has the following form:

$$b_j(x_t) = \mathrm{P}(vq(x_{1t}), vq(x_{2t}), \cdots, vq(x_{Lt})) = \sum_{k=1}^{M} c_{jk} \prod_{i=1}^{L} \mathrm{P}_{jki}(vq(x_{it})),$$

where c_{jk} is the mixture coefficient for the k-th mixture in state j, and $\mathrm{P}_{jki}(vq(x_{it}))$ is the probability of observing the discrete symbol $vq(x_{it})$ for the i-th subvector. The mixture coefficients are nonnegative, and sum to one for each HMM state j.

The output distribution introduced above assumes that the indices of different subvectors are conditionally independent given the state and mixture index. Dependencies between the different subvectors for a given state are modeled through the mixture components. When compared to the conventional CDHMMs, the discrete mixture HMMs replace a multivariate Gaussian density with the product of L discrete distributions, one for each subvector, as is shown in Figure 1.

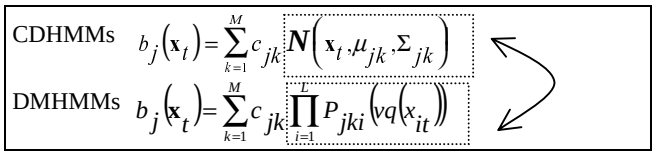

Figure 1: Comparison between Continuous Gaussian-mixture HMMs and Discrete-mixture HMMs.

3.2 Training of DMHMMs

Training of DMHMMs can be done using the Baum-Welch algorithm. The initial models can be obtained by discretizing a corresponding set of CDHMMs.

Initialization

Using the correspondence shown in Figure 1, the discrete distributions of a DMHMM can be initialized from the corresponding terms of a CDHMM. If the i-th subvector is quantized using B bits, then the probability of observing the l-th centroid $\mathrm{P}_{jki}(vq(x_{it}) = l)$ can be initialized to the value:

$$\mathrm{P}_{jki}(l) = \frac{N(v_{il1}; \mu_{jk1}, \sigma^2_{jk1}), \cdots, N(v_{ild}; \mu_{jkd}, \sigma^2_{jkd})}{\sum_{l} N(v_{il1}; \mu_{jk1}, \sigma^2_{jk1}), \cdots, N(v_{ild}; \mu_{jkd}, \sigma^2_{jkd})},$$

where the summation in the denominator is over all 2^B centroids, the l-th centroid of the i-th subvector is the d-dimensional vector $[v_{il1}, \cdots, v_{ild}]$, and $\mu_{jk1}, \cdots, \mu_{jkd}$ are the corresponding elements of the mean for the k-th Gaussian distribution of state j.

Reestimation Equations

The reestimation formulae can be derived directly by maximizing Baum's auxiliary function. It can be shown that the new estimate for the probability of observing the l-th centroid for the i-th subvector of the k-th mixture in state j is given by

$$\overline{\mathrm{P}}_{jki}(l) = \frac{\sum_{t:vq(x_{it})=l} \gamma_t(j,k)}{\sum_{t} \gamma_t(j,k)}$$

where $\gamma_t(j,k)$ is the posterior probability of being in state j at time t with the k-th mixture component accounting for the observation. This probability can be computed using the forward-backward algorithm and the previous estimates of the output probabilities, using

$$\gamma_t(j,k) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j} \alpha_t(j)\beta_t(j)} \cdot \frac{c_{jk} \prod_{i=1}^{L} \mathrm{P}_{jki}(vq(x_{it}))}{\sum_{k=1}^{M} c_{jk} \prod_{i=1}^{L} \mathrm{P}_{jki}(vq(x_{it}))}.$$

3.3 Smoothing DMHMMs

For large-vocabulary systems with a large number of context-dependent states, many of the discrete-mixture components will be estimated from a small number of samples. It is, therefore, important to smooth the discrete mixtures. Although more elaborate schemes, like deleted interpolation, can be used, we experimented with two simple schemes of linear interpolation.

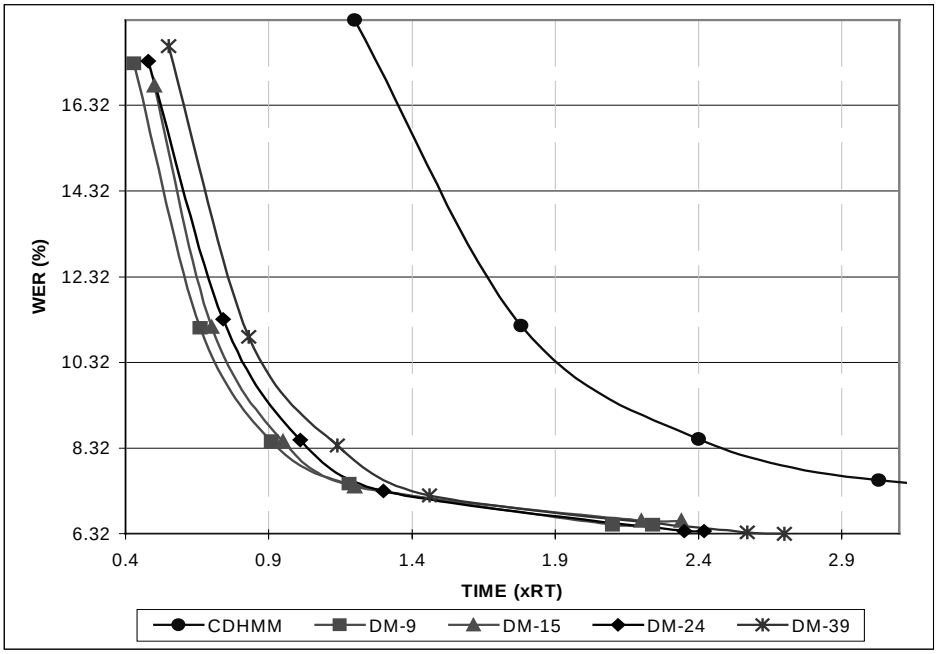

Figure 2: Various operating points indicating the word-error rate vs. speed trade-off for the baseline continuous CDHMM and discrete-mixture HMMs with 9, 15, 24 and 39 subvectors. The discrete-mixture HMMs clearly outperform the baseline CDHMM, achieving similar recognition error rates at a fraction of the decoding time required by the CDHMM.

In the first method, each mixture distribution for a particular state-subvector combination is linearly interpolated with the average distribution for that state and subvector with a weight that is empirically determined. The average distribution is computed by linearly combining the *M* mixture distributions for that state using their mixture weights.

The second method smooths each distribution by linearly combining its newly estimated values with the values of the same distribution from the previous iteration.

3.4 Pruning of DMHMM probabilities

To speed up the Gaussian computation in CDHMM systems with diagonal covariances, an incremental pruning method is often used. In this technique, the log-likelihood of a new Gaussian is incrementally computed by summing the contribution of the different MFCCs, and the accumulated value is periodically compared (for example, every five dimensions) to the total log-likelihood of the best Gaussian computed up to this point. If its value is less than the best likelihood by more than a pre-specified margin, then the Gaussian is pruned and the contribution from the remaining MFCCs is not calculated.

Since table look-up is an operation similar to a one-dimensional Gaussian computation, a similar pruning algorithm can be applied to the discrete DMHMM system. However, experimenting with the previous algorithm, we observed that it is not satisfactory: the mixtures were pruned typically during the last check, since we compared the partial values of the likelihood with the total value of the best mixture component. Hence, we modified the previous pruning algorithm so that the accumulated partial log-likelihood of a new mixture distribution is compared to the best partial log-likelihood computed up to this point for the same subvectors. This pruning algorithm can be made even more efficient, by ordering the subvectors so that the most discriminating subvectors are computed first, by doing the comparisons early on and by experimentally finding the optimum number of comparisons, since increasing the number of checks above a certain point will introduce delays.

4. EXPERIMENTAL RESULTS

We used SRI's DECIPHER continuous speech-recognition system, configured with a six-feature front end that outputs 12 MFCCs, cepstral energy, and their first- and second-order differences. We performed experiments in the ATIS domain [5], and we used a bigram language model throughout our experiments. The training data consisted of 20,000 sentences, and the test set consisted of 400 sentences from 34 male and female speakers. The CDHMM system was a genonic HMM [6], with state-clustered tied mixtures.

For the DMHMM systems, the feature vector is split into a specified number of subvectors, which are then processed by the quantizer, and a vector of discrete values with the same length as the number of subvectors is the input to the discrete recognizer. We experimented with 9, 15, 24 and 39 subvectors. In the latter case, each subvector consists of a single feature element. We did not experiment with fewer than 9 subvectors because, in the process, size increased significantly.

In our experiments we were mainly interested in reducing the computation time, while maintaining the WER of the baseline CDHMM system. All decoding times reported in this paper were measured on the same machine with an Intel 266-MHz Pentium-II processor and 256-MB main memory. A Viterbi beam search was used, and the number of active hypotheses was maintained at the same level while decoding times were measured.

4.1 Comparison of Decoding Times

We first performed a series of experiments comparing the decoding times of the baseline CDHMM system and the discrete DMHMMs that were obtained by discretizing the baseline system for the various configurations mentioned above. The results are summarized in Table 1. Differences in WER are statistically insignificant. Without pruning, we can see that the decoding time decreases for DMHMMs as the number of subvectors decreases. Except for the case of scalar quantization (39 subvectors), the DMHMMs are always faster than the CDHMM system, speeding up the computation by as much as 38%. The 39-subvector result is consistent with the findings of [3], where discrete HMMs were combined with scalar quantization and it was found that the decoding was slower than the corresponding continuous system. The use of the improved incremental pruning algorithm with the DMHMMs increases the speed-up over the baseline system even more, with a total computation saving of 59.5% for the case of 9 subvectors.

	No Pruning		With Pruning		
System	WER (%)	TIME (xRT)	WER (%)	TIME (xRT)	Speed-up over CDHMM (%)
CDHMM	6.60	5.69	6.60	5.25	-
DM-39	6.25	6.33	6.32	2.61	50.3
DM-24	6.60	5.23	6.38	2.36	55.0
DM-15	6.58	4.14	6.63	2.21	58.0
DM-9	6.25	3.53	6.53	2.13	59.5

Table 1: Word-error rates and decoding times with and without incremental pruning for continuous- and discrete-mixture HMMs.

The superior performance of the new DMHMM systems over the conventional CDHMMs in terms of decoding efficiency is clearly demonstrated in Figure 2, where we show the systems of Table 1 operating at different points in terms of WER/decoding time trade-offs. The various operating points were obtained by changing the beam-width of the Viterbi beam search. The closer the curves to the lower left point, the better, since we achieve low error rates at faster speeds. We see that the DMHMM systems are two to three times faster than the CDHMM for a fixed WER. In addition, for a fixed decoding time, the DMHMMs perform far better for areas in the graph where pruning errors occur. For instance, the DMHMM systems achieve real-time performance at a WER of about 7.5%, whereas the CDHMM system suffers from too many pruning errors at that operating point.

4.2 Number of Mixture Components

The DMHMM system models the output probability distribution with a mixture of nonparametric discrete distributions. In contrast, the CDHMM system has the constraint that its mixture components have a specific form, that of a Gaussian distribution. It is, therefore, reasonable to expect the same level of performance by DMHMMs using a smaller number of mixtures than the baseline CDHMM system. This implies that additional speed-ups are possible by decreasing the number of mixture components. From a different viewpoint, the recognition performance of a DMHMM system should exceed that of a CDHMM when the same number of mixture components is used. These hypotheses are verified in Table 2. There, we show the WERs, the decoding time, and the required memory (process size in Mbytes) for CDHMMs and the 15-subvector DMHMM system with 8, 16 and 32 mixture components. The DMHMM systems in these experiments were retrained, and the second smoothing algorithm of Section 3.3 was used. We see that the WER of the DMHMM system degrades far more gracefully than that of the CDHMM system, as the number of mixture components is decreased. For 16 and 8 mixture components the DMHMM system significantly outperforms the baseline CDHMM systems in terms of WER.

System	# of mixtures	WER (%)	TIME (xRT)	Memory (MB)
CDHMM	32	6.60	5.25	27.0
CDHMM	16	8.10	2.75	18.0
CDHMM	8	9.04	2.47	15.0
DM-15	32	6.90	2.83	57.7
DM-15	16	7.32	1.83	34.8
DM-15	8	7.57	1.48	23.0

Table 2: Word-error rates, decoding times, and process size for continuous- and discrete-mixture HMMs with different numbers of mixture components.

In conclusion, we have shown that DMHMMs with subvector quantization are far more efficient in decoding time than similarly configured conventional CDHMMs. Since DMHMMs use nonparametric distributions, they need fewer mixtures for the same accuracy level than CDHMMs, and this reduces further the recognition time.

ACKNOWLEDGMENTS

This work was accomplished under contract to Telia Research of Sweden and by SRI research and development funds.

5. REFERENCES

[1] V. Digalakis, L. Neumeyer and M. Perakakis, "Quantization of Cepstral Parameters for Speech Recognition over the WWW," *Proceedings ICASSP'98*, Seattle, WA, May 1998.

[2] V. Digalakis, L. Neumeyer and M. Perakakis, "Product-Code Vector Quantization of Cepstral Parameters," *Proceedings ICSLP'98, Sydney, Australia*, December 1998.

[3] S. Takahashi, K. Aikawa and S. Sagayama, "Discrete Mixture HMM," *Proceedings ICASSP'97*, Munich, Germany, April 1997.

[4] E. Bocchieri and B. Mak, "Subspace Distribution Clustering for Continuous Observation Density HMMs," *Proceedings Eurospeech'97*, Rhodes, Greece, September 1997.

[5] P. Price. "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. of the 3rd DARPA Workshop*, Hidden Valley, PA, June 1990.

[6] V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," *IEEE Trans. Speech Audio*, pp. 281-289, July 1996.