High-density Discrete HMM with the Use of Scalar Quantization Indexing

Brian Mak†, Siu-Kei Au Yeung‡, Yiu-Pong Lai‡, Manhung Siu‡

†Department of Computer Science
‡Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
mak@cs.ust.hk, jeffay@ust.hk, harry@ust.hk, eemsiu@ee.ust.hk

Abstract

With the advance in semiconductor memory and the availability of very large speech corpora (of hundreds to thousands of hours of speech), we would like to revisit the use of *discrete hidden Markov model* (DHMM) in automatic speech recognition. To estimate the discrete density in a DHMM state, the acoustic space is divided into bins and one simply count the relative amount of observations falling into each bin. With a very large speech corpus, we believe that the number of bins may be greatly increased to get a much higher density than before, and we will call the new models, the *high-density discrete hidden Markov model* (HDDHMM). Our HDDHMM is different from traditional DHMM in two aspects: firstly, the codebook will have a size in thousands or even tens of thousands; secondly, we propose a method based on scalar quantization indexing so that for a d-dimensional acoustic vector, the discrete codeword can be determined in $O(d)$ time. During recognition, the state probability is reduced to an $O(1)$ table look-up. The new HDDHMM was tested on WSJ0 with 5K vocabulary. Compared with a baseline 4-stream continuous density HMM system which has a WER of 9.71%, a 4-stream HDDHMM system converted from the former achieves a WER of 11.60%, with no distance or Gaussian computation.

1. Introduction

The recent success of many automatic speech recognition systems owes much to the advances in hidden Markov modeling (HMM) [1]. HMM may be roughly categorized into 3 types:

- *discrete HMM* (DHMM) [2] in which state observation statistics is modeled by discrete density;
- *semi-continuous or tied-mixture HMM* (SCHMM) [3, 4] in which a pool of Gaussians are shared by all states, and state probability densities are different only in their mixture weights of these globally shared Gaussians; and,
- *continuous density HMM* (CDHMM) [2] in which each state is modeled by a separate mixture of Gaussian densities.

Each of the 3 types of HMMs has many variants that differ mainly in their parameter sharing details, and the way the acoustic space is partitioned into multiple subspaces and/or multiple streams. Examples are phone-tied mixture HMM, state-clustered tied-mixture HMM [5], and subspace distribution clustering HMM (SDCHMM) [6], etc. In general, the model complexity increases from DHMM, SCHMM, to CDHMM.

With the advance in semiconductor memory and thus its falling price, and the availability of very large speech corpora (of hundreds to thousands of hours of speech), we would like to revisit the use of *discrete hidden Markov model* (DHMM) in large-vocabulary continuous speech recognition (LVCSR). DHMM is attractive for the following reasons:

- it is simple: the density estimation is merely a bin counting process, and its state probability can be found by a table lookup.
- the state distribution is non-parametric; in theory, it can model any distribution if there are sufficient training data.
- the state likelihoods can be handled more easily since they are really probabilities which has a smaller and predictable dynamic range between 0.0 and 1.0.

In the past, DHMM is only used for simple task. There are at least three reasons for that. Firstly, to implement DHMM, all acoustic vectors are vector-quantized (VQ) but the VQ codebook cannot be too large, otherwise it will take a long time to find the codeword for a new acoustic vector. Secondly, a lot of memory will be required to store a large discrete density for each HMM state. Thirdly, the discrete density of a large codebook requires a lot of training data which are lacking in the past. As a result, usually 256 to 1024 codewords are used. However, a small codebook will induce larger quantization errors that are unacceptable for complicated tasks which require high accuracy. The use of multiple-stream codebooks may alleviate the problems but only with the additional assumption that the streams are independent.

Some of the limitations of a large codebook are fading with the advances in semiconductor technologies and the release of very large speech corpora. In this paper, we propose the use of DHMM in which the discrete densities have a very high density involving thousands to tens of thousands of codewords; we call our new HMM, the "*high-density discrete hidden Markov model*" (HDDHMM). There are two major problems to solve for such HDDHMM:

- How to find the codeword for an acoustic vector fast?
- Data scarcity problem: How to train such high-density discrete densities for each HMM state?

For fast determination of a codeword, we propose representing a full-space codeword by the combinatorial product of the scalar quantization codewords from each dimension. As a result, a codeword can be determined in $O(d)$ time for a d-dimensional acoustic vector, independent of the number of bins. To solve the data scarcity problem, we suggest a simple and quick conversion of CDHMM to HDDHMM.

There is a relevant attempt before called the *discrete mixture HMM* (DMHMM) [7]. DMHMM replaces the one-dimensional Gaussian density in each mixture component of a

CDHMM state with diagonal covariances by a discrete density, but otherwise keeps the mixture density structure of CDHMM intact. [7] also suggested 20–40 codewords for each dimension of DMHMM. On the other hand, our new HDDHMM has only one large discrete density for each stream of an HMM state, and only a few codewords (e.g. 2 or 4) are used per dimension.

2. High-density Discrete HMM (HDDHMM)

2.1. Construction of Full-space codewords from Per-dimension SQ codewords

Let d be the dimension of each acoustic vector. In HDDHMM, each dimension is scalar-quantized (SQ) to $n_i, i = 1, \ldots, d$ SQ codewords. The full-space codewords are represented by the multiplicative combination of the per-dimension SQ codewords. As a result, the full d-dimensional acoustic space is divided into $N = \prod_{i=1}^{d} n_i$ bins. Each HDDHMM state probability distribution becomes a discrete density consisting of the probabilities of an acoustic observation falling into each of the N bins. For instance, if $d = 13$ for the typical static MFCC vectors, and all dimensions are scalar-quantized to $n_1 = n_2 = \cdots = n_d = 2$ codewords, then there will be $2^{13} = 8192$ bins. Although scalar quantization (SQ) is employed, it is only used to efficiently index different regions in the original d-dimensional acoustic space through the combinatorial effect of per-dimension SQ codewords, and discrete density is still estimated in the acoustic *full* space. This is different from DMHMM in which a discrete density is estimated in each 1-dimensional acoustic *subspace*.

2.2. Finding a codeword

To find the codeword for a new acoustic vector, each of its components will be compared with its corresponding SQ codebook for at most $\log_2 n_i$ time. Continuing with our example of using a uniform 1-bit SQ codebook for each dimension of a d-dimensional acoustic vector, finding a codeword is reduced to d arithmetic comparison operations.

2.3. Multiple Streams

For the typical 39-dimensional MFCC acoustic vector including the delta and delta delta features, even a 1-bit SQ for each dimension will lead to a codebook size of $2^{39} = 549,755,813,888$. Here we adopt the common solution of using multiple independent streams. Each stream of an HMM state has its own high-density discrete density.

2.4. State Likelihood

Given a new observation $\mathbf{x}_t$, for a K-stream system, it is broken up into K sub-vectors as $\mathbf{x}_t = \{\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \ldots, \mathbf{x}_t^{(K)}\}$, where $\mathbf{x}_t^{(k)}$ represents the sub-vector of the kth stream. Its probability at state s is computed as

$$P(\mathbf{x}_t|s) = \prod_{k=1}^{K} P(\mathbf{x}_t^{(k)}|s)^{\eta_k} ,$$

where $\eta_k, k = 1, 2, \ldots, K$ are the stream exponents used to weight the contributions of the various streams.

With a discrete density, the computation of the state likelihood of each stream is reduced to a simple table lookup. Each HMM state will have a table of the size of $N = \prod_{i=1}^{d} n_i$. For our running example, $N = 8192$. Since now the table entries are real probabilities, 1 or 2 bytes are probably enough to represent them.

3. Conversion of CDHMM to HDDHMM

Except for small systems with a few hundreds of (tied) HMM states, even hundreds or thousands of hours of training speech may not be sufficient to train an HDDHMM. While the problem may be perhaps solved by smoothing or interpolation techniques, in this paper, we investigate a simple algorithm to convert continuous density of a CDHMM state to a reliable estimate of the discrete density of an HDDHMM.

3.1. Conversion Procedure

The conversion procedure is summarized as follows:

STEP 1 : To create a K-stream HDDHMM, a K-stream CDHMM is first created with appropriate parameter tying.

STEP 2 : Scalar quantization (SQ) is performed on each dimension to the give the desirable number of codewords per dimension.

STEP 3 : For each SQ codeword, determine its lower bound and upper bound. Let's denote these bounds for the ith dimension by (l_i, u_i).

STEP 4 : Each bin in a d-dimensional stream is represented by a multiplicative combination of the per-dimension SQ codewords from its d dimensions. Thus, each bin is a hypercube with bounds $\{(l_1, u_1), (l_2, u_2), \ldots, (l_d, u_d)\}$ for its d sides.

STEP 5 : The corresponding K-stream HDDHMM will have the same topology as that of the K-stream CDHMM, except that, for each stream, the CDHMM state probability density functions are now replaced by HDDHMM's state distributions computed by integration. That is, if the pdf for the kth stream of the jth CDHMM state with a mixture density of M Gaussian components is,

$$b_j^{(k)}(\mathbf{x}_t) = \sum_{m=1}^{M} c_{jm}^{(k)} N(\mathbf{x}_t^{(k)}; \mu_{jm}^{(k)}, \Sigma_{jm}^{(k)}),$$

the bin with bounds $\{(l_1, u_1), (l_2, u_2), \ldots, (l_d, u_d)\}$ on the corresponding HDDHMM state j will have the following probability: $p_j^{(k)}(\mathbf{x}_t^{(k)} \text{ in the bin})$

$$\begin{aligned} &= \sum_{m=1}^{M} c_{jm}^{(k)} \int_{l_1}^{u_1} \int_{l_2}^{u_2} \cdots \int_{l_d}^{u_d} N(\mathbf{x}_t^{(k)}; \mu_{jm}^{(k)}, \Sigma_{jm}^{(k)}) d\mathbf{x}_t^{(k)} \\ &= \sum_{m=1}^{M} c_{jm}^{(k)} \int_{l_1}^{u_1} N(x_{1t}^{(k)}; \mu_{1jm}^{(k)}, \sigma_{1jm}^{(k)}) dx_{1t}^{(k)} \\ &\quad \int_{l_2}^{u_2} N(x_{2t}^{(k)}; \mu_{2jm}^{(k)}, \sigma_{2jm}^{(k)}) dx_{2t}^{(k)} \\ &\quad \cdots \int_{l_d}^{u_d} N(x_{dt}^{(k)}; \mu_{djm}^{(k)}, \sigma_{djm}^{(k)}) dx_{dt}^{(k)} , \end{aligned}$$

where $N(x_{it}^{(k)}; \mu_{ijm}^{(k)}, \sigma_{ijm}^{(k)})$ represents the univariate Gaussian density of the ith dimension of the mth component of the kth stream in the jth CDHMM state. The per-dimension integral can be computed using the $erf(\cdot)$ function as follows:

$$\int_{l_i}^{u_i} N(x_{it}^{(k)}; \mu_{ijm}^{(k)}, \sigma_{ijm}^{(k)}) dx_{it}^{(k)}$$
$$= \quad 0.5 \left[erf\left(\frac{u_i - \mu_{ijm}^{(k)}}{\sqrt{2}\sigma_{ijm}^{(k)}} \right) - erf\left(\frac{l_i - \mu_{ijm}^{(k)}}{\sqrt{2}\sigma_{ijm}^{(k)}} \right) \right] .$$

- After the conversion, HDDHMM may be re-estimated using the EM algorithm.

4. Experimental Evaluation

We carried out a preliminary study of the proposed HDDHMM on the Wall Street Journal speech corpus WSJ0 [8]. We used the standard SI-84 training set for training the acoustic models. It consists of 83 speakers and 7138 utterances for a total of about 14 hours of training speech.

Table 1: 4 Stream definitions.

Stream	MFCCs	Per-dimension Bit Allocation	# Bins
1	1–12	{2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}	16384
2	14–25	{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}	4096
3	27–38	{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}	4096
4	13,26,39	{3, 3, 3}	512

4.1. Feature Extraction and Scalar Quantization (SQ)

The traditional 39-dimensional MFCC vector was extracted at every 10ms over a window of 25ms. It was split into 4 streams as shown in Table 1.

SQ was carried out using a subset of the training data. The number of bits for each dimension is pre-determined as shown in Table 1. SQ was performed per dimension using LBG quantization algorithm [9], and the bounds $\{(l_i, u_i)\}$ for each SQ codeword were recorded.

4.2. Acoustic Modeling

The baseline system consists of (1-stream) continuous density HMMs (CDHMM) of 15,449 cross-word triphones (derived from 39 base phonemes). There are totally 3131 tied states, and each triphone HMM has a maximum number of 16 Gaussian mixture components.

Then a 4-stream CDHMM system was trained as follows: based on the baseline system, a 1-stream CDHMM with only 1 mixture per state was developed so that it had the same HMM topology (and thus same tied states) as the original baseline 1-stream CDHMM. The 1-stream 1-mixture CDHMM system was then converted to a 4-stream 1-mixture CDHMM system, and the number of mixtures was grown to 16 for each stream until we had a 4-stream 16-mixture CDHMM system.

Finally, the newly proposed HDDHMM system was converted from the 4-stream 16-mixture CDHMM system using the procedure described in Section 3.1. Basically, the conversion procedure turns CDHMM state probability density functions into discrete state densities of HDDHMM. Probability of each bin p, was represented by a 2-byte short integer x, with a floor probability of 10^{-12} using the following formula,

$$p = \exp\left(-x/1185.9\right) .$$

No extra smoothing or interpolation was done at this moment. In addition, all stream weights were set to 1.0.

Moreover, all acoustic model training were carried out by the HTK software.

4.3. Recognition Performance

The performance of various CDHMM and HDDHMM systems were tested on the standard nov'92 5K non-verbalized test set using a bigram language model of perplexity 147. The test set consists of 330 utterances from 8 speakers. Since the HDDHMM systems use true probabilities but the CDHMM systems use probability densities, the likelihoods computed during recognition have very different dynamic ranges. All decoding parameters such as the grammar factors and pruning thresholds were first set according to the CDHMM systems. They were then proportionally scaled for testing the HDDHMM systems according to the likelihood ratios of the two kinds of HMMs.

4.3.1. Based on 1-Mixture Systems

We firstly compare the following context-dependent systems which were derived from 1-mixture CDHMMs:

- CDHMM-1stream-1mix: 1-stream 1-mixture cross-word triphone CDHMM.
- HDDHMM-4stream-1: 4-stream cross-word triphone HDDHMM converted from the CDHMM-1stream-1mix system.

In this experiment, we would like to find out how well the high-density discrete distribution can represent a state probability distribution while factoring out the stream independence assumption. In this case, since each state density has only one multivariate Gaussian with diagonal covariance, the sub-vectors in the 4 streams are implicitly independent of each other in both the CDHMMs as well as in the HDDHMMs. The recognition results are shown in Table 2. We observe $\sim$2% (absolute) drop in the recognition accuracy which indicates that there is room for our conversion algorithm to improve.

Table 2: Word accuracies of context-dependent 1-mixture CDHMM and its converted HDDHMM on WSJ0.

System	Word Accuracy
CDHMM-1stream-1mix	85.22%
HDDHMM-4stream-1	83.19%

4.3.2. Based on 16-Mixture Systems

We then trained a 16-mixture cross-word triphone CDHMM system, converted it to a 4-stream CDHMM system, and then converted the latter again to a 4-stream HDDHMM system. In Table 3, the following three systems are compared:

- CDHMM-1stream-16mix: 1-stream 16-mixture cross-word triphone CDHMM.
- CDHMM-4stream-16mix: 4-stream 16-mixture cross-word triphone CDHMM.
- HDDHMM-4stream-16: 4-stream cross-word triphone HDDHMM converted from the CDHMM-4stream-16mix system.

Table 3: Word accuracies of context-dependent 16-mixture CDHMM and its converted HDDHMM on WSJ0.

System	Word Accuracy
CD-CDHMM-1stream-16mix	92.25%
CD-CDHMM-4stream-16mix	90.29%
CD-HDDHMM-4stream-16	88.40%

It is found that there is a 2% (absolute) loss in recognition accuracy when a 1-stream CDHMM system is converted to a 4-stream CDHMM system. When we move from a 4-stream CDHMM system to a 4-stream HDDHMM system, there is an additional 2% (absolute) performance loss — and a similar loss is also observed in the first experiment.

4.4. Memory and Speed Comparison

HDDHMM requires much more memory space to store its state densities than CDHMM. For instance, for our 16-mixture-based systems, the CDHMM system requires $3131 \times 16 \times (1 + 39 \times 2) \times 4 = 15.8$MB to store all its Gaussian means and variances and mixture weights in 4-byte floats, while the HDDHMM system requires $3131 \times (16384 + 4096 + 4096 + 512) \times 2 = 157$MB to store all its discrete densities in 2-byte short integers.

In Fig. 1, the operating characteristics of three HMM systems are compared: 4-stream CDHMM; 4-stream HDDHMM converted from the 4-stream CDHMM counterpart; and a conventional DHMM with 256 codewords for each MFCC stream and 64 codewords for the energy stream. All experiments were run on a P4 3.2GHz PC with 1GB RAM. It can be seen that our new HDDHMM performs better than the conventional discrete HMM, and can run faster than the CDHMM system if the runtime is expected to be less than 5x real time.

5. Conclusions

In this paper, we propose a new HMM called "*high-density discrete HMM*" (HDDHMM). Each HDDHMM state is represented by a high-density discrete probability distribution. The bins in the distribution are indexed by the multiplicative combination of scalar-quantized codewords from each dimension so that the bins can be identified in $O(d)$ time for a d-dimensional acoustic vector. The HDDHMM state probabilities can also be retrieved in $O(1)$ time at the expense of more memory space for the model densities. However, we believe that with the falling price of memory, this is not a problem — the gain in speed is of more interest to us than the greater memory requirement.

The results shown in this paper are very preliminary, and the system parameters of the new 4-stream systems were not yet tuned. Thus, we believe there are a lot of rooms for improvement. For instance, SRI reported [10] a 6-stream system with 7.7% WER on WSJ0, but our 4-stream CDHMM system gives an WER of 9.7% while our 1-stream CDHMM system matches with SRI's. This indicates that our 4-stream CDHMM system is not optimal. Since our 4-stream HDDHMM is converted from the 4-stream CDHMM, we have to first ensure that the latter has good performance.

6. Acknowledgments

This research is partially supported by the Research Grants Council of the Hong Kong SAR under the grant numbers HKUST6201/02E, and CA02/03.EG04.

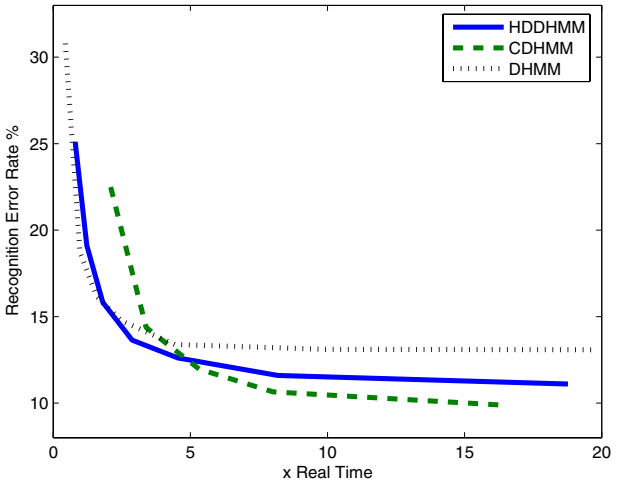

Figure 1: The operating characteristics of three 4-stream systems: CDHMM, HDDHMM, and DHMM.

7. References

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE Proceedings*, vol. 77, no. 2, pp. 257–285, 1989.

[2] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, January 1986.

[3] X. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Journal of CSL*, vol. 3, no. 3, pp. 239–251, July 1989.

[4] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Trans. on ASSP*, vol. 38, no. 12, pp. 2033–2045, Dec. 1990.

[5] G. Zavaliagkos, J. McDonough, D. Miller, A. El-Jaroudi, J. Billa, F. Richardson, K. Ma, M. Siu, and H. Gish, "The BBN Byblos 1997 large vocabulary conversational speech recognition system," in *Proc. of ICASSP*, 1998, vol. 2, pp. 905–908.

[6] E. Bocchieri and B. Mak, "Subspace distribution clustering hidden Markov model," *IEEE Trans. on SAP*, vol. 9, no. 3, pp. 264–275, March 2001.

[7] Satoshi Takahashi, Kiyoaki Aikawa, and Shigeki Sagayama, "Discrete mixture HMM," in *Proc. of ICASSP*, April 1997, vol. 2, pp. 971–974.

[8] D. B. Paul and J. M. Baker, "The design of the Wall Street Journal-based CSR corpus," in *Proc. of the DARPA Speech and Natural Language Workshop*, Feb. 1992.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[10] V. Digalakis and H. Murveit, "Genones: Optimizing the degree of mixture tying in a large vocabulary hidden Markov model based speech recognizer," in *Proc. of ICASSP*, 1994, vol. 1.