Fast GMM Computation for Speaker Verification Using Scalar Quantization and Discrete Densities

Guoli Ye and Brian Mak

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{mak,yeguoli}@cse.ust.hk

Man-Wai Mak

Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University
enmwmak@polyu.edu.hk

Abstract

Most of current state-of-the-art speaker verification (SV) systems use Gaussian mixture model (GMM) to represent the universal background model (UBM) and the speaker models (SM). For an SV system that employs log-likelihood ratio between SM and UBM to make the decision, its computational efficiency is largely determined by the GMM computation. This paper attempts to speedup GMM computation by converting a continuous-density GMM to a single or a mixture of discrete densities using scalar quantization. We investigated a spectrum of such discrete models: from high-density discrete models to discrete mixture models, and their combination called high-density discrete-mixture models. For the NIST 2002 SV task, we obtained an overall speedup by a factor of 2–100 with little loss in EER performance.

Index Terms: speaker verification, scalar quantization, high density discrete HMM, discrete mixture HMM

1. Introduction

Gaussian mixture model (GMM) is a probabilistic model commonly used in various speech areas. For example, most automatic speech recognition (ASR) systems use hidden Markov model (HMM) with GMM states; most speaker verification (SV) or speaker recognition (SR) systems use GMM to represent the universal background model (UBM) and speaker models (SM). Hence, a fast GMM computation method is of great interest to the speech community.

Fast GMM computation has been investigated in various speech-related areas such as SV [1], ASR [2], and voice conversion for text-to-speech synthesis [3]. In [1], McLaughlin *et. al* identified two orthogonal factors in GMM computation: the acoustic resolution in terms of the number of Gaussian components in a GMM, and the temporal resolution in terms of various forms of decimation (or down-sampling). In another approach, under the name of Gaussian selection [4, 5] and tree-structured UBM [6], only the most relevant Gaussian components in the UBM are selected and evaluated. In [2], Chan *et. al* defined a four-layer categorization scheme of GMM computation: two of the layers are basically the same as McLaughlin's, and the other two are speedup of Gaussian component computation and HMM state selection (which is irrelevant to SV). The speedup method proposed by En-Najjary [3] is similar to that in semi-continuous HMM or in the evaluation of speaker models in SV [7]: only evaluate the top N Gaussian components that have the highest likelihoods. For large UBM and small N, the method reduces the number of Gaussian evaluations almost by half, with negligible effect on verification accuracy.

In this paper, we investigate another way to adjust the acoustic resolution of a GMM to achieve fast GMM computation. Instead of simply reducing the number of Gaussian components (i.e., reducing the GMM model order), a combination of scalar quantization and discrete density techniques are used to convert a continuous-density GMM (GMM) to

- a *single* discrete density called high-density discrete model (HDDM) based on the techniques used in the high-density discrete hidden Markov model (HDDHMM) [8] in ASR.
- a *mixture* of discrete densities called discrete mixture model (DMM) based on the techniques used in the discrete-mixture hidden Markov model (DMHMM) [9, 10] in ASR.
- a *mixture* of high-density discrete densities called high-density discrete-mixture model (HDDMM) which is a combination of HDDM and DMM.

It has been shown that HDDHMM and DMHMM may reduce GMM computations in ASR with no or little degradation in recognition performance. Since an SV system spends almost all its computation in GMM evaluations, the speedup achieved by HDDM, DMM, or HDDMM should be even greater as it is not complicated by the decoding and pruning algorithms in ASR.

2. Discrete Models for Fast GMM Computation

Although, in theory, given sufficient training data, a discrete density may represent any distribution to any desirable precision by adjusting its resolution, in practice, we have to deal with the following issues: (1) insufficient data for estimating the probability of each bin in the discrete density; (2) increasing the resolution by having more bins will increase the size of the density which may become too big to be stored; (3) traditionally, vector quantization (VQ) is used to define the codebook. When there are many bins, the codebook will also be large and finding the codeword for a given input will be slow. These problems may be solved by using scalar quantization (SQ) with additional techniques described in this paper.

Conventional wisdom tells us that given a fixed number of bits, VQ is more efficient than SQ in quantizing the feature space. However, SQ has the advantage of requiring much less training data and it is much faster to find an SQ codeword.

This research is supported by the Research Grants Council of the Hong Kong SAR under the grant numbers DAG05/06.EG43, HKUST617406, and PolyU 1-BB9W.

There are two possible solutions to approximate the efficiency of VQ by SQ:

- simply use more SQ bits. This will increase the model size, but the availability and continually decreasing price of large solid-state memory help push further its limit.
- use a *mixture* of discrete densities instead of a *single* discrete density in which the dimensions of the feature vectors are assumed independent.

The above two solutions lead to the invention of high-density discrete hidden Markov model (HDDHMM) [8] and discrete-mixture hidden Markov model (DMHMM) [9] in ASR. Since a GMM may be considered as a 1-state HMM, the SQ techniques used in HDDHMM and DMHMM can be readily ported to speed up GMM computation for SV/SR.

In the following discussion, let d be the dimension of each acoustic vector $\mathbf{x}_t$, and each dimension i, is scalar-quantized to n_i SQ codewords with n_{max} being the maximum codebook size among all dimensions. Assume that, after per-dimension SQ, the acoustic vector $\mathbf{x}_t$ falls into the d-dimensional hypercube with bounds $\{(l_1,u_1),(l_2,u_2),\ldots,(l_d,u_d)\}$ where (l_i,u_i) represents the lower bound and upper bound of the SQ codeword of its ith dimension.

2.1. High-Density Discrete Model (HDDM)

The HDDM is a *single* discrete density. The full-space VQ codewords are constructed by the product of per-dimension SQ codewords. Thus, if each dimension is scalar-quantized to $n_i, i=1,\ldots,d$, SQ bins, then there will be $N=\prod_{i=1}^{d} n_i$ VQ bins in the full space. For instance, if $d=12$ for the typical static MFCC vectors (without the energy term), and all dimensions are scalar-quantized by 1 bit, then there will be $2^{12}=4096$ 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 the discrete density is still estimated in the acoustic *full* space.

2.1.1. Conversion of GMM to HDDM

Let the probability density function (pdf) of a continuous-density GMM with diagonal covariances be

$$p(\mathbf{x}_t)=\sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{x}_t;\mu_m,\sigma_m^2), \tag{1}$$

where M is the number of Gaussian components, and c_m, μ_m, and σ_m^2 are the mixture weight, mean vector, and variance vector of the mth component respectively. The probability mass function (pmf) of the corresponding HDDM can be pre-computed by

$$\begin{aligned}
& P(\mathbf{x}_t \text{ in } \{(l_1,u_1),(l_2,u_2),\ldots,(l_d,u_d)\}) \\
= & \sum_{m=1}^{M} c_m \int_{l_1}^{u_1}\int_{l_2}^{u_2}\cdots\int_{l_d}^{u_d} N(\mathbf{x}_t;\mu_m,\sigma_m^2)d\mathbf{x}_t \\
= & \sum_{m=1}^{M} c_m \prod_{i=1}^{d}\int_{l_i}^{u_i} N(x_{it};\mu_{im},\sigma_{im}^2)dx_{it}\ , \qquad (2)
\end{aligned}$$

where $N(x_{it};\mu_{im},\sigma_{im}^2)$ represents the univariate Gaussian density of the ith dimension of the mth component. The per-dimension integral can be computed using the $erf(\cdot)$ function.

2.1.2. Time Complexity of HDDM

Finding an equivalent VQ codeword requires $O(d\log_2 n_{max})$ time, and it takes $O(1)$ time to find the HDDM probability by a table lookup among the N bins. As a consequence, computing the GMM likelihood from its approximate HDDM can be very fast. However, the foregoing discussion does not consider the practical constraint on the model size.

- *Tradeoff between resolution and model size*
  d must be small enough that the resulting HDDM has a reasonable size. For example, for our experiments, we used 24-dimensional acoustic vectors consisting of 12 static and 12 dynamic MFCCs. If we limit it to 1 bit per dimension, the resulting HDDM will have $2^{24}=16$ million bins; but if we use 2 bits per dimension, there will be $(2^2)^{24}=256$ trillion bins! Thus, the per-dimension resolution cannot be high, and the quantization error may lead to inaccuracy in the GMM approximation.
- *Tradeoff between multiple streams and correlation loss*
  One way to get higher acoustic resolution is to split the acoustic space into multiple independent subspaces resulting in multiple-stream HDDM. To obtain a K-stream HDDM, the GMM is first approximated by a K-stream GMM. The major shortcoming is the loss of correlation among the features across the streams. Here, we limit to use two streams to achieve a higher acoustic resolution per stream and to maintain a reasonable model size.

2.2. Discrete Mixture Model

The DMM is a *mixture* of discrete densities, having the same number of mixtures as the GMM from which it is converted. In its simplest form[1] in which all dimensions in a component are assumed independent, its pmf is given by [9]:

$$\begin{aligned}
& P(\mathbf{x}_t \text{ in } \{(l_1,u_1),(l_2,u_2),\ldots,(l_d,u_d)\}) \\
= & \sum_{m=1}^{M} c_m \prod_{i=1}^{d} P_{im}(\mathbf{x}_{it}\in\{(l_i,u_i)\})\ , \qquad (3)
\end{aligned}$$

where $P_{im}(\cdot)$ is the pmf of the discrete density of the ith dimension in the mth mixture.

2.2.1. Conversion of GMM to DMM

During the conversion from GMM to DMM, each dimension of each mixture component is done independently. For instance, $P_{im}(\cdot)$ is pre-computed as follows:

$$P_{im}(x_{it} \text{ in } \{(l_i,u_i)\}) = \int_{l_i}^{u_i} N(x_{it};\mu_{im},\sigma_{im}^2)dx_{it}\ . \tag{4}$$

Notice that the DMM has a total of $d \times M$ discrete densities.

2.2.2. Time Complexity of DMM

Since finding an SQ codeword requires $O(\log_2 n_{max})$ time and there are d dimensions, the total time complexity for finding all SQ codewords is again $O(d\log_2 n_{max})$. However, since DMM is an M-mixture model, dM table lookups are required to get the per-dimension pmf values for an input $\mathbf{x}_t$. To obtain the probability of the input from these discrete density values requires dM multiplications and $M-1$ additions. Essentially, DMM replaces the dM Euclidean distance calculations

[1] In general, one may group several dimensions together and perform sub-vector quantization to get the discrete density for the sub-vectors [10].

in GMM by dM table lookups. As a consequence, it is not as fast as HDDM.

Table 1: Model size of GMM, HDDM, DMM, and HDDMM. (M = #mixture components; K = #streams; n = #SQ bits per dimension; d = dimension of acoustic vectors. Probabilities are assumed 4-byte floating point numbers.)

Model	M	K	n	d	Size (bytes)
GMM	M	—	—	d	$4M(2d+1)$
DMM	M	—	n	d	$4M(2^n d+1)$
HDDM	—	K	n	d	$4K(2^{n^{(d/K)}})$
HDDMM	M	K	n	d	$4MK(2^{n^{(d/K)}})$

2.3. Comparison between HDDM and DMM

HDDM and DMM are very similar. If HDDM has only a single stream, each Gaussian component has diagonal covariance, and no re-estimation is performed, then the HDDM and DMM converted from the same GMM will have the same accuracy but HDDM pre-computes all the probabilities in a single discrete density whereas DMM only pre-computes the discrete densities for each dimension in each component and the the probability of an input has to be computed from these pre-computed discrete densities.

For the same number of SQ bits, the model size of an DMM is generally much smaller than that of an HDDM; as a consequent, DMM may approximate the original GMM more accurately with more bits. On the other hand, since HDDM has only a single discrete density, it computes much faster than DMM. Obviously one may trade-off the speed and the size of these two discrete models by combining them in what we call *high-density discrete mixture model* (HDDMM). There are still M mixtures in an HDDMM, but each mixture component is now a multi-stream HDDM. For HDDMM, more streams in a mixture will make the model size smaller but the model runs more slowly. Table 1 shows the formulas for finding the model size of the various models.

Table 2: Resolution of various discrete models for SV. (Assume that DMM and HDDMM are converted from 512-component GMM, and probabilities are 4-byte floating-point numbers.)

Model	K	Bit Allocation	Size
HDDM-a	2	(111111111111, 111111111111)	32KB
HDDM-b	2	(211111111111, 111111111111)	48KB
HDDM-c	2	(221111111111, 111111111111)	80KB
HDDM-d	2	(222111111111, 111111111111)	144KB
HDDM-e	2	(222211111111, 111111111111)	272KB
HDDM-f	2	(222221111111, 111111111111)	528KB
HDDM-g	2	(222222111111, 111111111111)	1.04MB
HDDM-h	2	(222222211111, 111111111111)	2.064MB
HDDM-i	2	(222222221111, 111111111111)	4.112MB
DMM-a	—	111111111111111111111111	96KB
DMM-b	—	222222222222222222222222	192KB
DMM-c	—	333333333333333333333333	384KB
DMM-d	—	444444444444444444444444	768KB
HDDMM-a	12	$12 \times (3,3)$	1.536MB
HDDMM-b	8	$8 \times (3,3,3)$	4.096MB
HDDMM-c	6	$6 \times (3,3,3,3)$	49.152MB

3. Experimental Evaluation

NIST SRE 2001 and 2002 were used in this work. Specifically, all of the 1006 male utterances (from > 120 speakers) in NIST 2001 were used for creating the GMM-UBM, and speaker models were created by MAP adaptation [7] on the GMM-UBM for each of the 139 male speakers in NIST 2002. Each adaptation utterance is about 2 minutes long but half of the contents is silence. Each of the 1442 male verification utterances in NIST 2002 was scored against 11 hypothesized speakers. This amounts to 1,232 speaker trials and 14,630 impostor attempts.

The features used were 12 MFCCs plus their first derivatives, leading to 24-dimensional acoustic vectors. Cepstral mean normalization was applied to the MFCCs, followed by feature warping. Speaker verification is based on the log-likelihood ratio between a speaker model and the UBM.

GMM-UBMs with 64–2048 Gaussian components were trained and converted to the three kinds of discrete models with varied resolutions and streams. The resolution was controlled by allotting different number of SQ bits to the 24 MFCCs as shown in Table 2. All HDDMs had two streams and different numbers of streams were tried for HDDMM.

For each of the following experiments, the various models, namely, GMM, DMM, HDDM, and HDDMM are compared in terms of their execution time[2], equal error rate (EER), and minDCF. For GMM, DMM, and HDDMM, we followed the common practice and used only the top 5 Gaussians for computing the UBM and speaker model likelihoods.

Table 3: SV performance of baseline GMMs of varied order and their approximate HDDMs.

M	GMM		HDDM	
	EER	Time(sec)	EER	Time(sec)
2048	11.94	12901	13.18	105.4
1024	11.84	6488.7	**12.97**	97.96
512	**11.43**	3278.9	13.23	107.0
256	11.88	1674.2	13.71	99.28
128	12.37	868.75	13.62	105.1
64	13.20	466.35	14.62	97.70

3.1. HDDM: Effect of the GMM model orders

GMM UBMs and speaker models of different orders were converted to 2-stream HDDMs using the bit allocation scheme of HDDM-i as specified in Table 2. The SV performance of the corresponding GMMs and HDDMs are compared in Table 3. The results show that due to their lower resolution, the EERs of HDDM are lower than those of their GMM counterparts by 1–2%. However, the speed of the various HDDMs is basically constant with a running time of about 100 seconds regardless of the model order of the GMMs they are converted from since they have the same discrete density structure; it is very fast — with a speedup of 4–130 times — and the speedup is more prominent with bigger GMM models. Thus, one is free to choose the best GMM to convert to HDDM without worrying that the model order of the GMM will affect the speed of its HDDM counterpart.

[2] All experiments were run on a Linux machine that runs on the Intel CPU, Core 2 Duo E8400 @ 3.00GHz with 2GB RAM.

Table 4: Effect of different resolutions on the performance of DMM converted from a GMM-UBM with 512 components.

Bit/Dimension	EER	minDCF	Time(sec)
1 (DMM-a)	14.62	0.0601	1867.5
2 (DMM-b)	11.75	0.0522	1883.4
3 (DMM-c)	11.55	0.0528	2160.0
4 (DMM-d)	11.49	0.0536	3943.3

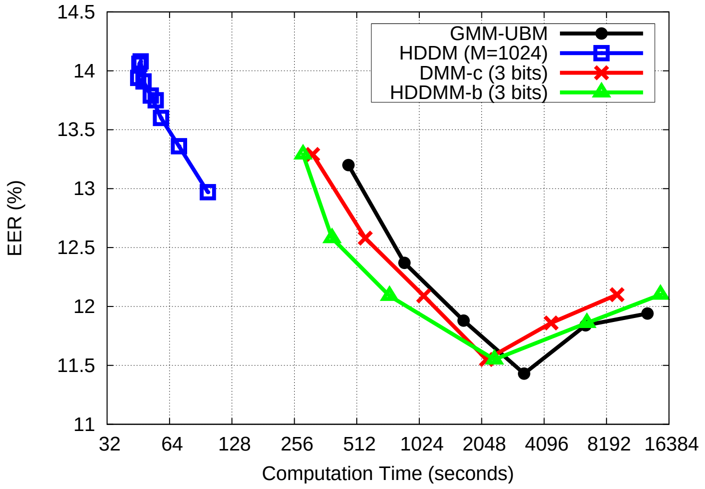

Figure 1: SV operating characteristics of different models.

3.2. DMM: Effect of different resolutions

The GMMs with 512 components were converted to DMMs of different resolutions by uniformly allotting 1–4 SQ bits for each dimension of the MFCC vector. The EER of the GMM is 11.43% and its speed is 3279s. From Table 4, the DMM with 3 bits per dimension basically has similar SV performance as its GMM counterpart but achieves a speedup of about about 33%.

3.3. Operating characteristics of different models

From the last two experiments, it is clear that HDDM and DMM represent the two ends of a spectrum of SQ-based discrete models: For the same model size, HDDM is fast but has a lower resolution, whereas DMM is not much faster than GMM but is more accurate than HDDM. Thus, it is expected that their hybrid form, HDDMM, may tradeoff speed against accuracy by dividing each Gaussian mixture component of a DMM into K streams and represents each stream using the technique in HDDM. HDDMM should have the same performance of its DMM counterpart, but its speed will increase if fewer streams are used.

In Fig. 1, the following models are compared:

- GMMs with varied number of Gaussian components.
- HDDMs with different resolutions as shown in Table 2 converted from the GMMs with 1024 Gaussian components.
- DMMs with 3 bits per dimension (DMM-c in Table 2) and varied number of mixtures.
- 8-stream HDDMMs with 3 bits per dimension (HDDMM-c in Table 2) and varied number of mixtures.

Notice that the abscissa in drawn in $\log_2$ scale.

It is found that for the same model order,

- the DMMs are about 30–50% faster than their GMM counterparts.
- the HDDMMs are only faster than their DMM counterparts when there are fewer than 512 mixture components. Detailed examination of the experiments finds that the Linux kernel has a page size of 4MB, and when the model size of HDDMMs are larger than 4MB, the runtime is significantly lower.
- after all, the HDDMMs are still faster than their GMM counterparts by 30–60% when $M \leq 512$.

4. Conclusions

The results show that HDDM is very fast and can obtain a speedup of 30 times with 1.5% drop in EER. DMM is more accurate but provides a much smaller speedup. Their hybrid model HDDMM is only faster when the GMM model order is low because of their relative large model size. However, we expect that if the hardware is probably chosen, HDDMM may provide a much better speedup; further investigation of HDDMM's dependence on the computing hardware is needed.

5. References

[1] Jack McLaughlin, Douglas A. Reynolds, and Terry Gleason, "A study of computation speed-ups of the GMM-UBM speaker recognition system," in *Proc. of Interspeech*, 1999, pp. 1215–1218.

[2] Arthur Chan et. al, "Four-layer categorization scheme of fast GMM computation techniques in large vocabulary continuous speech recognition systems," in *Proc. of Interspeech*, 2004, pp. 689–692.

[3] T. En-Najjary, O. Rosec, and T. Chonavel, "Fast GMM-based voice conversion for text-to-speech synthesis systems," in *Proc. of Interspeech*, 2004, pp. 1229–1232.

[4] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *Proc. of ICASSP*, 1993, vol. 2, pp. 692–695.

[5] R. Auckenthaler and J. S. Mason, "Gaussian selection applied to text-independent speaker verification," in *Proc. Speaker Odyssey 2001*, 2001, pp. 83–88.

[6] B. Xiang and T. Berger, "Efficient text-independent speaker verification with structural Gaussian mixture models and neural network," *IEEE Trans. SAP*, vol. 11, no. 5, pp. 447–456, Sep. 2003.

[7] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[8] Brian Mak, S. K. Au Yeung, Y. P. Lai, and M. Siu, "High-density discrete HMM with the use of scalar quantization indexing," in *Proc. of Eurospeech*, Portugal, 2005.

[9] S. Takahashi et. al, "Discrete mixture HMM," in *Proc. of ICASSP*, April 1997, vol. 2, pp. 971–974.

[10] V. Digalakis et. al, "Efficient speech recognition using subvector quantization and discrete-mixture HMMs," *Computer Speech and Language*, vol. 14, pp. 33–46, 2000.