ISCA Archive
http://www.isca-speech.org/archive

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

EFFECTIVE VECTOR QUANTIZATION FOR A HIGHLY COMPACT ACOUSTIC MODEL FOR LVCSR

Jielin Pan, Baosheng Yuan and Yonghong Yan
Jielin.pan@intel.com; Tel: (86-10) 8529-8800 Ext. 1810, Fax: (86-10) 8529-8717
Intel China Research Center
#601, Kerry Center, 1 GuangHua Road, ChaoYang District, Beijing, 100020, PRC

ABSTRACT

This paper introduces a method that can efficiently reduce acoustic model size and computation for LVCSR based on continuous-density hidden Mokov model (CDHMM). The method uses Bhattacharyya distance measure as a criterion to quantize the mean and variance vectors of Gaussian mixture. To minimize the quantization error, the feature vector was separated into multiple streams (such as MFCCs, delta-MFCCs and delta-delta MFCCs) and then the modified K-means clustering algorithm was applied to each stream. The key ideas of our modified K-means clustering algorithm are based on the strategy which dynamically splits and merges cluster according to its size and average distortion during each iteration for each cluster. The proposed approach can cut acoustic model size by 87% from 21.42MB to 2.75MB from a CDHMM baseline system (with 12 mixtures, 6k states) by using 256 and 8192 codewords for each stream of mean and variance vectors of Gaussian mixtures. The recognition experiment on Chinese LVCSR dictation system (of 51K words) shows that using the 87% smaller compact model, the WER increased by 5% to 10.3% from 9.8% for the CDHMM baseline system. After quantization, the Gaussian likelihood can be pre-computed only once at the beginning of every frame and their values can be stored in a lookup table, so the computation during decoding is greatly reduced as well.

1. INTRODUCTION

Continuous-density hidden Markov model (HMM) has been widely used in speaker-independent LVCSR because it outperforms discrete hidden Markov model and semi-continuous hidden Markov model [1][2]. The main idea in CDHMM is that the probability function of observations is modeled by Gaussian mixtures which can approximate the speech feature distribution more accurately.

However, time-consuming output probability computation and large memory requirement of CDHMM makes it difficult to implement a real-time LVCSR system. A lot of efforts have been made to solve this problem [3][4][5]. A very obvious way to tackle the problems is to build a smaller system by reducing both the number of mixtures per state and the number of states of the system. However, it usually introduces unacceptable increase of WER if the parameter size is reduced significantly.

Our Chinese dictation system is based on CDHMM, it has 6007 states and 72076 Gaussian mixtures. The total model size is approximately 21M bytes. Since all the model parameters need to be evaluated by processor for every frame, it needs a large memory space and it's time-consuming for likelihood computation, therefore limiting its capacity in a practical application. Therefore, how to reduce the model size significantly without degrading the system performance is the key.

In this paper we introduce a method to use vector quantization technique to quantize the mean and variance of Gaussian mixtures for reducing the model size and computation complexity. The front-end feature used in the system is a 36 dimension vector including 12 MFCCs, 12 delta-MFCCs and 12 delta-delta MFCCs. We treat these three streams separately. It is well known that the Gaussian mixtures are very sensitive to any perturbation in the value of their means, so we quantize mean of Gaussian mixtures with more codewords to minimize the quantization error. On the other hand, since the variance is less sensitive, we use fewer codewords for it.

Section 2 describes the modified K-means clustering algorithm based on Bhattacharyya distance measure. Section 3 shows the computation reduction. Section 4 reports the experiment results for the proposed method. Section 5 gives some conclusion marks.

2. MODIFIED K-MEANS CLUSTERING ALGORITHM

For quantizing mean and variance of Gaussian mixtures, we must design codebook for every stream of mean and variance. The design criteria is to minimize the total distortion between the centroid vectors of codebook and the original vectors[6]. For a given mean (variance) vector, there are three indexes which point to corresponding codeword in each stream codebook, we can combine these three codewords to represent the given mean (variance).

The key ideas of our modified K-means clustering algorithm is based on the strategy which dynamically splits and merges cluster according to its size and average distortion during each iteration for each cluster. The algorithm is described as below:

Given two Gaussian mixtures: $N(x;m_1,\Sigma_1)$ and $N(x;m_2,\Sigma_2)$, we compute the distortion between them using Bhattacharyya distance[7], defined as (1):

$$D_{bhat} = \frac{1}{8}(m_2 - m_1)^T \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1}(m_2 - m_1) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}} \qquad (1)$$

Where, m_1 and m_2 are the mean vectors of the two Gaussian mixtures, Σ_1 and Σ_2 are the variance matrix of the two Gaussian mixtures.

We quantize the mean and variance of Guassian mixtures separately. For variance quantization, we assume the mean vector is the same for all Gaussian mixtures. So the distortion measure (1) becomes:

$$D_{bhat} = \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}} \qquad (2)$$

We use (2) as the distance measure between two variance vectors.

For mean quantizaiton, we assume the variance is the same for all Gaussian mixtures, so (1) becomes:

$$D_{bhat} = G_1(m_2 - m_1)^T(m_2 - m_1) + G_2 \qquad (3)$$

where G_1 and G_2 are constants and can be ignored in distance measure as below:

$$D_{bhat} = (m_2 - m_1)^T(m_2 - m_1) \qquad (4)$$

We use (4) as the distance measure between two mean vectors.

Except for the distance measure, the clustering algorithm is the same for mean and variance of Gaussians, so we just give clustering algorithm for mean as example below.
Given a set of N Gaussians and a corresponding set of mean vectors $M = \{m_1, ..., m_N\}$, we divide it to three streams and obtain three sub-vector sets

$M_1 = \{m_{11}, ..., m_{1N}\}$,

$M_2 = \{m_{21}, ..., m_{2N}\}$,

$M_3 = \{m_{31}, ..., m_{3N}\}$

such that $m_i = \begin{pmatrix} m_{1i} \\ m_{2i} \\ m_{3i} \end{pmatrix}, i = 1, ..., N$

We use the following modified K-means algorithm to cluster each sub-vector set M_i to build K-bits codebook, there are 2^K codewords.

Step 1: Initialization: codebook bit $k = 0$, Compute the centroid of M_i. Denotes the centroid as c_j, $j = 1, ..., 2^k$

Step2: if $k = K$, k-bits codebook are obtained, end the clustering algorithm, else:

$k = k + 1$, split each cluster C_j, $j = 1, ..., 2^{k-1}$ from 1 to 2, the split criteria is as below:

(1) compute the average variance to the centroid c_j for all the vectors in this cluster, denotes it as $\boldsymbol{d}_j$

(2) create two new centroids :

$$c_j^1 = c_j + 0.5 \cdot \boldsymbol{d}_j \qquad (5)$$

$$c_j^2 = c_j - 0.5 \cdot \boldsymbol{d}_j \qquad (6)$$

(3) combine all centroids to build k-bits codebook $\{c_1, ... c_{2^k}\}$

initialize $D_1 = 1e^{-20}$.

Step 3: For each $m_{il} \in M_i$, $l = 1, ..., N$, associate it to the nearest centroid, such that

$$n(l) = \arg\min_{s=1,...,2^k} d(m_{il}, c_s) \qquad (7)$$

where $d(m_{il}, c_s)$ is the distance measure between m_{il} and c_s.

Step 4: compute the total distances for all vector as below

$$D_2 = \sum_{l=1}^{N} d(m_{il}, c_{n(l)}) \qquad (8)$$

if $|D_1 - D_2| / D_1 \leq \boldsymbol{q}$, where $\boldsymbol{q}$ is the pre-defined threshold, go to step 2.

Step 5: splitting and merging.
Resigns cluster according to (7), we can calculate the total distortion DT_i and vector number N_i for each cluster $CL_i, i = 1, ..., 2^k$, then we do the merging and splitting as below:

(1) merging

if $N_n < \boldsymbol{f}$, the cluster CL_n is merged (the centrod of the cluster is removed from the codebook and the vectors in the cluster are resigned to other clusters), $n = 1, ..., 2^k$.

Where $\boldsymbol{f}$ is pre-defined threshold.

(2) Splitting

If there is a merged cluster, then we select cluster CL_m as below:

$$m = \arg\max_{i=1,...,2^k} DT_i / N_i \qquad (9)$$

split the CL_m according to step2.

Step 6: compute new centroid for each cluster, copy into old ones , set $D_1 = D_2$, go to step 3.

This modified K-means clustering algorithm can guarantee to converge to a local minimum of the over all distortion.

3. COMPUTATION REDUCTION

In decoding process, most of computation are consumed on the computation of state observation probability. For a given Gaussian mixture with mean vector m and a variance matrix C = diag($s_1,..., s_M$), for a given observation vector x , the log likelihood of this Gaussian mixture can be computed as follow:

$$\log N(x;m,C) = G + (X - m)^T C^{-1}(x - m) \qquad (10)$$

where G is a constant. After mean and variance quantization , for every frame this log likelihood can be pre-computed only once at the beginning, and their values are stored in a lookup table, so during Viterbi decoding, the log likelihood computation is just a table lookup. The computation is greatly reduced.

4. EXPERIMENT RESULTS

The evaluation experiment was conducted on speaker-independent Chinese LVCSR dictation task with 51K words vocabulary. The baseline system uses 36 dimension feature vector consisting of 12 MFCCs, 12 delta-MFCCs and 12 delta-delta MFCCs and uses context dependent within-word tri-phone model. Each model has 3 state and each state has 12 Gaussian mixtures. The system therefore has 6007 states and 72076 Gaussian mixtures that come up with approximately 21M bytes model size. The means and variance of the models were quantized in the method described in section 2. We use 110 utterances (5 female, 6 male, 10 utterance for each) to test system performance. The followings give experiment results comparing with the baseline system.

4.1 Variance quantization

The experiment results using different size of codebooks (for each stream, the book size is the same) are show in table 1. The meanings of symbols in the table are below:

	WER	Model Size	Reduction
Baseline	9.8%	21.42M	N/A
256 codewords	9.7%	11.95M	44.2%
128 codewords	10.2%	11.93M	44.3%
64 codewords	11.0%	11.92M	44.4%

Table 1: System performance for variance quantization with different codebook sizes

From table 1, we can see that when codebook size increases, the WER decreases. When we use 256 codewords, there is no performance decrease, while the total model size reduces from 21M bytes to 11M bytes, about 44.2% reduction.

4.2 Mean quantization

Because Gaussian mixtures are very sensitive to the perturbation in the value of their means, we quantize mean of Gaussian mixtures with more codewords, so we use more codewords to quantize the 72076 means. The experiment results with different sizes of codebook (for each stream, the book size is the same) are shown in table 2.

	WER	Model Size	Reduction
Baseline	9.8%	21.42M	N/A
8192 codewords	10.7%	13.09M	38.9%
4096 codewords	11.1%	12.50M	42.3%
2048 codewords	12.4%	12.21M	43%

Table 2: System performance comparison for mean quantization with different codebook sizes

From table 2, we can see that when codebook size increases, the WER decreases. When we uses 8192 codewords, there is only 0.9% reduction in recognition rate, the total model size reduces from 21M bytes to 13.09M bytes, about 38.9% reduction.

4.3 Mean and variance quantization

For efficiently reducing the storage and computation, we must quantize the mean and variance of Gaussian mixtures simultaneously. Table 3 shows Experiment results for mean and variance quantization with different codebook sizes. From table 3 , we can see, when we quantize the mean with 8192 codewords and varince with 256 codewords, WER is 10.3%, only 5% increasing compared with the baseline system, while model size decreases from 21.42M to 2.750M, about 87% reduction.

	WER	Model size	Reduction
Baseline	9.8%	21.42M	N/A
M:8192 ,V:256	10.3%	2.750M	87.1%
M:4096, V:256	11.6%	2.160M	89.9%
M:2048, V:256	13.2%	1.865M	91.2%
M:8192, V:128	12.2%	2.732M	87.2%
M:4096, V:128	12.4%	2.143M	90.0%
M:2048, V:128	12.9%	1.847M	91.3%

Table 3: System performance comparison for mean and variance quantization with different codebook size

5.CONCLUSIONS

This paper describes a method using vector quantization to efficiently reduce the model size and computation for CDHMM based LVCSR. By dynamically quantizing each stream of mean and variance of Gaussian mixtures with proper codebook size (mean: 8192, variance: 256), our approach reduced the acoustic model size by 87% from 21.42MB to 2.75MB with WER increased only 5% from 9.8% to 10.3%. It made the system practical for application which has limited resources to run in real-time. The algorithm does not use any language-specific information, therefore it should be applied to CDCHMM LVCSR system for any languages.

6.Refferances

[1] L. Polymenakos, P. Olsen, et al. "Transcription of Broadcast News-Some Recent Improvements to IBM's LVCSR System". In Proceedings of ICASSP, vol. 2, pages 901-904,1998.

[2] Billa, J et al. "Recent Experiments in Large Vocabulary Conversational Speech Recognition". In Proceedings of ICASSP, vol. 1, pages 41-44, 1999.

[3] Brian Mak and Enrico Bocchieri, "Training of Subspace Distribution Clustering Hidden Markov Model". In Proceedings of ICASSP, vol2, pages 273-276, 1998.

[4] Jae Kim,Raziel Haimi-Cohen, and Frank Soong, "Hidden Markov Models with Divergence Based Vector Quantized Variances" In Proceedings of ICASSP, vol. 1, pages 125-128, 1999.

[5] E. Bocchieri, "Vector Quantization For The Efficient Computation of Continuous Density Likelihoods". In Proceedings of ICASSP, vol.2, pp692-695,1993.

[6] Y.Linde, A.Buzo and R.M.Gray, "An Algorithm for Vector Quantizer design". IEEE Trans. Comm., vol. COM-26, pp702-710.

[7] Brain Mak, Etienne Barnard, "Phone Clustering Using the Bhattacharyya Distance", In Proc. of ICSLP, vol. 4, pages 2005-2008,1996.