COMBINING SELECTION TREE WITH OBSERVATION REORDERING PRUNING FOR EFFICIENT SPEAKER IDENTIFICATION USING GMM-UBM

Zhenyu Xiong, Thomas Fang Zheng, Zhanjiang Song[1]*, and Wenhu Wu*

Center for Speech Technology, State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[xiongzhy, fzheng]@cst.cs.tsinghua.edu.cn, wuwh@tsinghua.edu.cn

[1]Beijing d-Ear Technologies Co., Ltd., http://www.d-Ear.com, zjsong@d-Ear.com

ABSTRACT

In this paper a new method of reducing the computational load for Gaussian Mixture Model Universal Background Model (GMM-UBM) based speaker identification is proposed. In order to speed up the selection of N-best Gaussian mixtures in a UBM, a Selection Tree (ST) structure as well as relevant operations is proposed. Combined with the existing Observation Reordering Pruning (ORP) method which was proposed for rapid pruning of unlikely speaker model candidates, the proposed method achieves a much larger computation reduction factor than any single individual method. Experimental results show that a GMM-UBM system used in a conjunction with ST and ORP can speed up the computation by a factor of about 16 with an error rate increase of only about 1% compared with a baseline GMM-UBM system.

1. INTRODUCTION

Research on speaker recognition, including with identification and verification, has been an active area for several decades. It is popular to model the speakers with the Gaussian mixture model (GMM) based on the maximum-likelihood (ML) criterion in speaker identification, which has been shown to outperform several other existing techniques [1]. The Gaussian Mixture Model Universal Background Model (GMM-UBM) method for speaker verification has also demonstrated high performance in several NIST evaluations and has become the dominant approach in text-independent speaker verification [2]. In this paper, the GMM-UBM method is introduced for speaker identification.

In many applications, accuracy and computation are two important factors. Some straightforward techniques have been investigated to speed up computation in a GMM-UBM speaker verification system while achieving an acceptable tradeoff between accuracy and computation [3]. In GMM-UBM speaker identification systems, the major computation loads are the likelihood calculation for all mixtures of the UBM to select the highest scoring mixtures (N-best mixtures) and the likelihood calculation for all speaker models in the system. In this paper, a Selection Tree (ST) method is proposed to speed up the selection of the N-best mixtures in a UBM. The existing Observation Reordering Pruning (ORP) method for rapid pruning of unlikely speaker model candidates is also introduced [4]. The proposed ST combined with ORP can speed up speaker identification to a much higher extent than either the ST or the ORP method individually.

The remainder of the paper is organized as follows. In Section 2, a brief description of GMM-UBM speaker identification is provided. In Sections 3 and 4, the ST method is given in detail and the ORP method is briefly introduced. The experimental results are shown in Section 5 while some conclusions are given in Section 6.

2. GMM-UBM METHOD

In GMM-UBM speaker identification, speakers are modeled with GMMs. A high order (usually 1,024 or 2,048) speaker-independent UBM is first built using a large speech corpus. Then each speaker model is derived from the UBM via Bayesian or Maximum *a Posteriori* (MAP) adaptation method using the corresponding speaker's speech data [2].

Since the UBM and speaker models share a correspondence, a fast scoring technique can be used as follows. For each input feature vector, all the UBM mixtures are scored to determine the top N highest scoring mixtures, in other words, the N-best mixtures or top N mixtures, and the speaker model likelihood is calculated using only the N speaker mixtures corresponding to the top N from the UBM, where N is much smaller than the order of either the UBM or the speaker model (usually N is 4 or 5).

3. SELECTION TREE

In the GMM-UBM method, all mixtures of a UBM are used to calculate likelihood for each input vector, which is a heavy computational load for the system. To effectively find the top mixtures, we propose that all Gaussian mixtures of the UBM be clustered hierarchically and organized into a tree structure. In this way, the acoustic space is partitioned into multiple regions of different levels of resolution. The top mixtures for a given vector can be found easily by searching the tree.

In order to build the tree, a distance measure between any two Gaussian components should be defined. While diagonal covariance matrices are assumed, the distance $d(m,n)$ between two Gaussian components, G_m and G_n, with distributions $N(\mu_m,\Sigma_m)$ and $N(\mu_n,\Sigma_n)$ is evaluated as follows[5]

$$d(m,n)=\sum_i \left[\frac{\sigma_m^2(i)-\sigma_n^2(i)+(\mu_m(i)-\mu_n(i))^2}{\sigma_n^2(i)} + \frac{\sigma_n^2(i)-\sigma_m^2(i)+(\mu_n(i)-\mu_m(i))^2}{\sigma_m^2(i)}\right] \quad (1)$$

where $\mu_m(i)$ is the *i*-th element of mean vector μ_m while $\sigma_m^2(i)$ the *i*-th diagonal element of the covariance matrix Σ_m for Gaussian G_m. Secondly, each non-leaf node in the tree will be approximated by a single Gaussian probability density function (pdf) with a weight. For a node c with a set of nodes R belonging to it, the pdf parameters for node c are calculated as follows [6]

$$\mu_c(i)=\frac{\sum_{k\in R} w_k\mu_k(i)}{\sum_{k\in R} w_k} \quad (2)$$

$$\sigma_c^2(i)=\frac{\sum_{k\in R} w_k\left(\sigma_k^2(i)+\mu_k^2(i)\right)}{\sum_{k\in R} w_k}-\mu_c^2(i) \quad (3)$$

$$w_c=\sum_{k\in R} w_k \quad (4)$$

where $\mu_c=\{\mu_c(i)\}^T$ is the mean vector, $\Sigma_c=Diag\{o_c^2(i)\}$ the covariance matrix, and w_c the weight.

3.1. Tree Construction

The number of layers L and the tree structure in the upper $L-1$ should be first determined before constructing the tree. A top-down Gaussian clustering algorithm is proposed to construct the tree as follows.

1) The pdf of the root node is calculated using Equ.s (2), (3), and (4) with all Gaussian components of the UBM. All the Gaussian components are regarded to belong to the root.
2) The pdfs of nodes in the next layer that belong to a node in the current layer are initiated by the *minimax* method and then interpolated with that of the current layer's node [4].
3) The K-means algorithm is applied to cluster the Gaussian mixture components belonging to the current node into several classes each of which will form a new node in the next layer. For each iteration, the mean, variance, and weight for each node (class) are updated using Equ.s (2), (3), and (4) until the distance converges.
4) The same procedures in Steps (2) and (3) are repeated for the next lower layers until the nodes in the last non-leaf layer are generated and each leaf node is assigned to its corresponding parent node.

3.2. Mixtures Selection in Selection Tree

For each test vector, all nodes in the second layer are used for calculating likelihood, and the N nodes with highest scores, are selected. All the child nodes of these top N nodes are then scored and another top N nodes of a lower layer are selected repeatedly this way down until the leaf-node layer. Finally, N leaf nodes are selected as the approximators of the top N mixtures of the UBM. The name "Selection Tree" comes from such a many-to-many selection procedure in a tree structure. The selection tree is different from the decision tree, in which there is only one two-value decision in each layer.

4. OBSERVATION REORDERING PRUNING

In speech processing, the feature vectors are extracted from overlapping windows (the so-called *frames*) of speech during which the vocal tract characteristics are assumed stationary. This results in a high degree of correlation in neighboring observations which will reduce the efficiency of the beam-search in a speaker independent system. Since the order of the observation sequence does not affect the final decision of the GMM-UBM speaker identification, an observation reordering pruning (ORP) method is considered for the applications in which the entire observation sequence is known [4].

For an observation sequence $X=\{x_1,x_2,\ldots x_T\}$, the reordered sequence Y is obtained as follows.

1) Initialize Y with a subset of observations selected at uniformly spaced intervals across the vectors in X and remove these observations from X.

2) Check the observations remaining in X from left to right, move those nearest to the midpoints of the observations in Y (in the sense of index) and append them to Y.
3) Repeat Step (2), until all observations have been reordered and moved in Y.

Then the reordered observation sequence Y is processed by the speaker identification system with the standard beam-search method. Thus unlikely speaker model candidates can be rapidly pruned. This method was proved efficient in speeding up the system.

5. EXPERIMENTS

5.1. Database and Features

The evaluation database includes a telephone speech corpus uttered by 1,031 speakers. Each speaker has 30s of speech for training and 1 or 2 utterances for test. There are 1,086 test segments in total and the duration of each test segment varies from 1s to 10s, with an average of 7s. The UBM was estimated with about 2 hours of speech from other 60 males and 60 females which were not used for the evaluations.

Silence was removed from speech using an energy based speech activity detection algorithm, and then 16-dimensional Mel-Frequency cepstral coefficients (MFCC) and 16 delta coefficients were extracted every 10 ms from 20 ms overlapping windows.

5.2. Computational Measurement

To represent the computational loads of a speaker identification system, a computation cost is defined as the total number of Gaussian likelihood calculation times for an input feature vector. And the computation reduction factor is defined as the ratio of the computation costs before and after using a certain speeding-up method. For an effective method, the factor should be much bigger than 1, and the bigger the better.

The computation cost for the GMM-UBM system is evaluated as

$$M_U + N \times S \tag{5}$$

where M_U is the number of Gaussian mixtures for the UBM, N is the number of top mixtures used for speaker model likelihood calculation and S is the number of speaker models in the system..

Since there is not an uniform expression for the computation costs for a system when the Selection Tree or the Observation Reordering Pruning is used, the computation costs and the computation reduction factors are calculated in run-time when the method is used in a speaker identification experiment.

5.3. Experimental Results

5.3.1. Baseline

A conventional GMM-UBM based system was taken as the baseline, where a UMB with 1,024 mixtures was used and the speaker models were derived form the UBM via the MAP adaptation method. Only top 4 mixtures of the UBM were used for speaker model likelihood calculation for each input feature vector. The correct rate for this baseline GMM-UBM system is 95.32%, and the computation cost is 1,024+4*1,031=5,184.

5.3.2. Selection Tree

The Selection Tree (ST) method was tested with ten tree structures, including three three-layer trees, three four-layer trees, one five-layer tree, one six-layer tree and one seven-layer tree. The effectiveness of the ST is evaluated with: (1) the correct rate of the identification system, (2) the computation reduction factor for the top mixtures selection, which is defined as

$$\frac{M_U}{M_T} \tag{6}$$

where M_T is the average number of likelihood computation times for an input vector when using the ST, and (3) the computation reduction for the whole system, which is evaluated as

$$\frac{M_U + N \times S}{M_T + N \times S}. \tag{7}$$

The results are listed in Table 1 where n_{2-7} are the numbers of nodes in the second to the seventh layers, the first row indicates the baseline GMM-UBM system, while $F_{Selection}$ denotes the computation reduction factors. Compared with the baseline GMM-UBM system, the five-layer Selection Tree achieves a biggest computation reduction factor of about 14.7 for top mixtures selection with only about 1% of error rate increase. It can be seen that as the number of layers increases, the factor increases; however much more layers will lead to either a increase of error rate or a (unstable) decrease of the factor. Both the number of layers and the tree structure should be suitable and hence need careful design. With this tree, the computation reduction factor for the whole system is 1.24.

5.3.3. Observation Reordering Pruning

The Observation Reordering Pruning (ORP) method [7] was also tested with different beam-widths. Similarly, the computation reduction factor

$$\frac{N \times S}{M_S} \tag{7}$$

n_2	n_3	n_4	n_5	n_6	n_7	$F_{Selection}$	Corr.(%)
1024						1	95.32
8	1024					1.7	95.26
16	1024					3.3	95.15
32	1024					5.4	95.32
64	1024					6.6	95.26
8	32	1024				5.4	95.04
16	64	1024				9.3	95.10
16	128	1024				11.0	95.15
16	64	256	1024			14.7	95.26
8	16	64	256	1024		14.6	95.15
8	32	64	128	256	1024	14.7	95.10

Table 1. Selection Tree.

	F_{whole}	Corr.(%)
Baseline	1	95.32
Integrated	15.8	95.26

Table 2. GMM-UBM integrated with ST and ORP.

where M_S is the average number of likelihood calculation times with all speaker models for one input vector during the beam search when using the ORP method, is used to evaluate the effectiveness of the ORP. And the computation reduction factor for the whole system is

$$\frac{M_U + N \times S}{M_U + M_S}. \qquad (8)$$

Figure 1 shows that different correct rates are achieved with different speeding-up factors when different beam widths being used. The ORP method speeds up the system by a factor of 15.2 for speaker models likelihood calculation while maintaining the same performance (95.32%) compared with the baseline. However the computation reduction factor for the whole system is only 4.0.

5.3.4. Combing ST and ORP

Any single method, either ST or ORP, does not achieve a big enough computation reduction factor for the whole system, the factor is only 1.24 for ST while 4.0 for ORP. An experiment was done to see the effectiveness when combining ST and ORP together, where the Computation reduction factor for the whole system is computed using

$$\frac{M_U + N \times S}{M_T + M_S}. \qquad (8)$$

The result is listed in Table 2.

Table 2 shows that a combination of the Selection Tree method and the Observation Reordering Pruning method integrated into the GMM-UBM system achieves a pretty big computation reduction factor of 15.8 for the whole system with only 1% of error rate increase compared with the baseline GMM-UBM system.

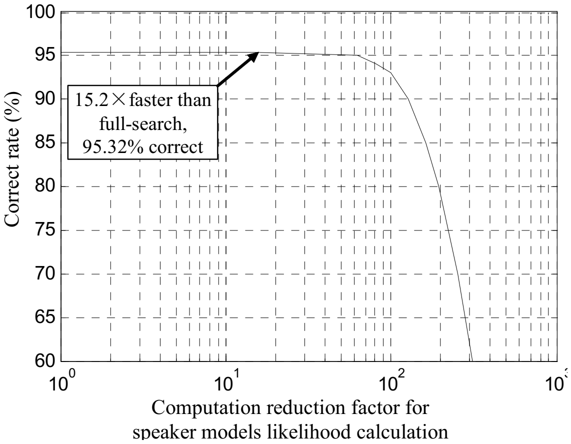

Figure 1. Observation Reordering Pruning.

6. CONCLUSIONS

In this paper, we proposed the use of a combination of the newly proposed Selection Tree method and the existing Observation Reordering Pruning method for GMM-UBM based text-independent speaker identification. The proposed method has a satisfying performance which can speed up the computation significantly without noticeable degradation in performance.

7. REFERENCES

[1] D.A. Reynolds and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing,* vol. 3, pp. 72-83, 1995.

[2] D.A. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing,* vol. 10, no. 1-3, pp.19-41, 2000.

[3] J. Mclaughlin, D.A. Reynolds and T. Gleason, "A study computation speed-ups of the GMM-UBM speaker recognition system," *Proc. Eurospeech*, pp. 1215-1218, 1999.

[4] B.L. Pellom, and J. H. L. Hansen, "An efficient scoring algorithm for Gaussian mixture model based speaker identification," *IEEE Signal Processing Letter,* vol. 5, pp. 281-284, 1998.

[5] B. Xiang and T. Berger, "Efficient text-independent speaker verification with structural Gaussian mixture models and Neural Network," *IEEE Trans. Speech Aaudio Processing,* vol. 11, pp. 447-456, 2003.

[6] K. Shinoda and C.H. Lee, "A structural bayes approach to speaker adaptation," *IEEE Trans. Speech Audio Processing,* vol. 9, pp. 276-287, 2001.