Regression-class Tree based Method for Efficient Speaker Identification

Gang Wang, Xiaojun Wu, Thomas Fang Zheng, Linlin Wang and Chenhao Zhang
Center for Speech and Language Technologies, Division of Technical Innovation and Development,
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
E-mail: {Wang-g07, Wangll07, Zhangch09}@mails.tsinghua.edu.cn, {Xjwu, Fzheng}@tsinghua.edu.cn
Tel/Fax: +86-10-62796393

***Abstract*— In this paper an efficient speaker identification (SpkID) method is proposed. In GMM-based speaker recognition, the model training and the likelihood score computations are very time-consuming and therefore have been bottlenecks of SpkID applications under the requirement of fast recognition especially in case of a large population of target speakers. A method, based on regression-class tree (RCT) structural UBM which is similar to a kind of sorted tree, is proposed and can apparently speed up the training and recognition of GMM-UBM based SpkID system. A number of components of the UBM can be pruned and their likelihood scores can be easily calculated using a kind of regression method. Experimental results show that the proposed method can improve the computational efficiency by 3.5 times for training and 15.3 times for recognition with only slight identification performance degradation.**

I. INTRODUCTION

Speaker identification (SpkID) [1] is a sub-task of speaker recognition, whose object is to automatically identify whether the speaker of a speaker-unknown utterance is in a pre-specified set of known target speakers and which target speaker he/she best matches on the acoustical characteristics. SpkID has been an active research area for several decades which has been tried in wide applications including access control, forensic evidence and telephone-based account transactions, etc. SpkID is a two-stage procedure containing training and recognition. In the training stage, a speaker model is built using each speaker's feature vectors extracted from the training utterance based on some modeling algorithm. For examples, GMM-UBM [2] and GMM-SVM [3] are two powerful and popular approaches to model a speaker's characteristics for theirs flexibility to approximate the underlying probability distribution in a high dimensional space. Mel frequency cepstral coefficient (MFCC) [4] is one of the commonly used features. In the recognition stage, feature vectors extracted from a test utterance (speaker-unknown) are used to calculate the likelihood scores [2] against all target speaker models and these scores are used to judge which speaker is the most likely one.

Although accuracy is always the first consideration in SpkID, efficient identification is another important factor in many practical systems. The common methods cannot be satisfied with in some systems such as speaker indexing, telephone banking and forensic intelligence because of the strict real-time requirement.

In SpkID systems the main computation load consists of three parts. First, in a GMM-UBM based SpkID system generally all components of the UBM need to be scored to find the *N-best* components for each frame of test feature vectors [2], which is a heavy computation load especially when the number of the test feature vectors is large. Second, the likelihood scores against all target speaker models need to be calculated to use the MFCC or GMM supervector in GMM-UBM or GMM-SVM system, respectively. The more known target speakers there are in the pre-specified set, the larger the computation load would be. Third, in a GMM-SVM based SpkID system the speaker's GMM model corresponding to the test utterance needs to be built to obtain the GMM supervector [3] as the input feature vector of SVM. Furthermore, in many GMM-UBM based SpkID systems the speaker's GMM model of the test utterance needs to be built to compensate the channel variability [5, 6] or normalize the likelihood score in D-Norm [7] to improve the SpkID performance. As is well known, a GMM model is usually adapted from the UBM using the maximum a posterior (MAP) [8] method that will occupy much computation time.

Now, there have been several approaches to speed up SpkID. To find the *N-best* components more quickly, hash GMM [9], structural GMM-structured background model (SGMM-SBM) [10] and tree-based kernel selection (TBKS) [11] were proposed. In these methods, the UBM is reconstructed as some structure such as a hash table or a tree so that it is possible to reduce the computation load by pruning. To speed up the calculation of likelihood score, the observation reordering based pruning (ORBP) [12], hierarchical speaker identification (HSI) [13] and GMM-based known speaker models clustering algorithm (SMC) [14] were proposed. ORBP reorders the test feature vectors and reduces the number of the test feature vectors by pruning some of them from computation. In HSI and SMC methods, the known speaker models are clustered into K classes. Firstly, the likelihood scores are calculated against the K putative speaker models corresponding to the cluster centroids. Secondly, those classes with small scores are pruned and the likelihood score against the remainder known target speakers are calculated.

Proceedings of the Second APSIPA Annual Summit and Conference, pages 462–465,
Biopolis, Singapore, 14-17 December 2010.

However, there is not a method to improve efficiency of GMM model training. In this paper, a training method based on regression-class tree (RCT) is proposed. The UBM is reconstructed as a RCT structure. When using the MAP algorithm based on the UBM to train a GMM model, pruning and regression can be easily adopted to reduce the computation load while guaranteeing the score accuracy. In other words, each frame of the training feature vectors does not need to be calculated against all components of the UBM because of the pruning based on the RCT. Meanwhile, the scores of the pruned components can be easily obtained from the scores of their parent nodes with a regression method. Furthermore, the RCT can be also used to find the *N-best* components during the recognition stage by pruning in a short time.

This paper is organized as follows. In Section II, the RCT construction and its usages are described in details. The experimental setup, results and analysis are given in Section III. Conclusions are drawn in Section IV.

II. RCT BASED TRAINING AND RECOGNITION

Generally, the UBM is a large Gaussian mixture model containing many components and modeling the whole acoustic space. Usually the UBM is a sequential structure. The RCT UBM is different from the common UBM because it is a hierarchical structure. Pruning and regression can be easily applied on this structure.

A. The Construction of Regression Class Tree

The components of the UBM are used to build RCT. The RCT structure is shown in Fig. 1.

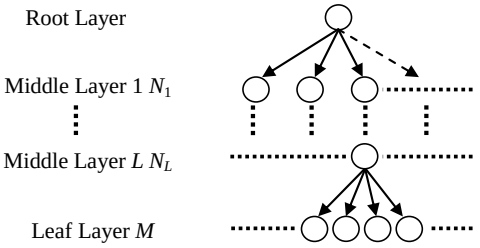

Fig. 1 Regression class tree structure.

where L denotes the number of the middle layers. N_k denotes the number of the nodes in Middle Layer k ($1 \le k \le L$). Each node is corresponding to a single weighted Gaussian probability density function (PDF). M denotes the number of leaf nodes which equals to the number of the components of the UBM. Each leaf node corresponds to a component of the UBM. The nodes in Middle Layer k are the N_k cluster centroids that are obtained by clustering all the nodes in Middle Layer k+1 into N_k classes. In this way, the RCT structure can model the whole acoustic space with different levels of acoustic resolution. The bottom-up strategy and the *K*-means clustering method are adopted to construct the RCT as follows:

(1) Denoted as $g(\mu_r, \sigma_r^2)$, the PDF of the root node is calculated using equations (1) – (3), which are a kind of maximum likelihood estimation under the consideration of component weight.

$$\mu_n^i = \sum \omega_m \mu_m^i / \omega_n, \; m \in C_n \tag{1}$$

$$\sigma_{ni}^2 = \sum \omega_m \left(\sigma_{mi}^2 + (\mu_m^i)^2 \right) / \omega_n - (\mu_n^i)^2, \; m \in C_n \tag{2}$$

$$\omega_n = \sum \omega_m, \; m \in C_n \tag{3}$$

C_n is a set containing all the components belonging to the n-th cluster ($1 \le n \le N_k$). ω_m is the weight of the m-th component in C_n. ω_n is the sum of all component weights in C_n. μ_n^i and μ_m^i are the i-th dimension element of the n-th cluster centroids means μ_n and the m-th component means μ_m in the n-th cluster, respectively ($1 \le i \le D$, D is the dimension of the feature vector). σ_{ni}^2 and σ_{mi}^2 are the i-th diagonal element of the n-th cluster centroids diagonal covariance matrix σ_n^2 and the m-th component diagonal covariance matrix σ_m^2 in the n-th cluster. When calculating $g(\mu_r, \sigma_r^2)$, C_n contains all the leaf nodes and N_k is 1.

(2) The *K*-means algorithm [15, 16] is applied to cluster all the nodes in Middle Layer k+1 into N_k classes. The PDFs are calculated corresponding to the N_k cluster centroids using equations (1) – (3). The Kullback-Leibler (KL) divergence [10] is chosen as the distortion measure between two Gaussian components in the *K*-means algorithm reference to equation (4).

$$d(m,n) = \sum_i \left[\frac{\sigma_{mi}^2 - \sigma_{ni}^2 + (\mu_m^i - \mu_n^i)^2}{\sigma_{ni}^2} + \frac{\sigma_{ni}^2 - \sigma_{mi}^2 + (\mu_n^i - \mu_m^i)^2}{\sigma_{mi}^2} \right] \tag{4}$$

(3) Use the PDFs of the N_k cluster centroids to interpolate with the root node PDF to obtain the PDFs of the nodes in Middle Layer k [10]. Interpolation equations are (5)–(6).

$$\mu_{nc} = (1-\alpha)\mu_n + \alpha\mu_r \tag{5}$$

$$\sigma_{nc}^2 = (1-\alpha)\left[\sigma_n^2 + \mu_n^2\right] + \alpha\left[\sigma_r^2 + \mu_r^2\right] - \mu_{nc}^2 \tag{6}$$

where μ_{nc} is the n-th node's PDF means ($1 \le n \le N_k$). σ_{nc}^2 is the n-th node's PDF diagonal covariance matrix. α is the fusion coefficient between the root node PDF $g(\mu_r, \sigma_r^2)$ and cluster centroid PDF $g(\mu_n, \sigma_n^2)$. α is chosen according to the experimental results.

(4) Repeat Steps (2) - (3) until the nodes in the middle layers are generated. The initial value of k is L and decrease k to 1 with step 1. Finally, the RCT would be built from bottom to up.

(5) For each node N_p in Middle Layer k ($1 \le k \le L, 1 \le p \le N_k$), calculate the distortions between N_p and N_p's child nodes using equation (7). Equation (7) is the weighted KL distance measure. Find the child node N_{max} which is the farthest away from N_p and denote the largest distance as d_{max}. Calculate the weighted KL distances between N_p and any one of the child nodes of N_p's sibling nodes. If the distance between one child node and N_p is smaller than d_{max}, insert the node as a redundancy child node (RCN) of N_p.

$$d_\omega(m,p) = d(m,p) / \omega_m \tag{7}$$

B. Training method based on RCT

Each frame of the training feature vectors is used to calculate the likelihood scores against all the components in the UBM when building a speaker GMM [2]. However, the UBM is so large a GMM representing a distribution over almost the full acoustic space that one frame of the feature vectors is close to only a few components of it. For those components which are far away from the training feature vector, the likelihood scores are so small that there is little or even no impact to the resolution of the GMM model, even if the scores are not accurate.

According to the above analysis, the RCT and pruning can be used to reduce the computation load. First, the search width b is defined to find the best compromise between efficiency and accuracy. The training steps are as follows:

(1) Set the nodes in Middle Layer 1 as the initial Computation Set (CS).

(2) For each frame of the training feature vectors, calculate the weighted likelihood scores against the PDFs corresponding to the nodes in CS.

(3) Sort the nodes in CS in a descending order according to the corresponding likelihood scores.

(4) Select top b scored nodes and set their all child nodes as the new CS and the other nodes are pruned. If the number of nodes in CS is smaller than b, set all the child nodes of all the nodes in CS as the new CS.

(5) For those pruned nodes, the scores of their leaf nodes can be calculated by Equation (8) because the difference between $g(i)$ and $g(p)$ is small.

$$\begin{aligned} S_p &= \log\left(\omega_p.g(p)\right) = \log\left(g(p)\right) + \log\left(\omega_p\right), \\ S_i &= \log\left(\omega_i . g(i)\right) = \log\left(g(i)\right) + \log\left(\omega_i\right), \\ S_i &= S_p - \log\left(\omega_p\right) + \log\left(\omega_i\right) \end{aligned} \quad (8)$$

N_p denotes a pruned node and N_i denotes the i-th leaf node of the sub-tree rooting from N_p. $g(p)$ is the PDF of the node p and ω_p is weight of the $g(p)$. $g(i)$ is the PDF of the node i and ω_i is weight of the $g(i)$. S_i is the score of the node N_i and S_p is the score of the node N_p..

(6) Go to the Step (2) unless the new CS is empty.

After all the training feature vectors have been processed, use the MAP adaption algorithm to obtain the speaker model or the GMM supervector.

The larger b is, the more components are calculated. In the worst case that b is larger than or equal to the node number in Middle Layer L, all the nodes in the RCT will be scored and the algorithm is inefficient because more components (middle layer nodes) are calculated. If b is too smaller, the accuracy would be poor. Furthermore, the construction of the RCT can also affect the accuracy and efficiency of the algorithm.

C. Recognition method based on RCT

The recognition procedure is similar to the training. For one frame of the test feature vectors, only a few mixtures contribute significantly to the likelihood score. Generally, likelihood score value can be approximated very well using only the top N scored components [2]. The searching strategy based on the RCT is similar to the training except that Step (5) is omitted because the scores of the pruned nodes are of no use. Obviously, the top N components selected in the RCT are not guaranteed to be the exact top. Nevertheless, the impact can be ignored.

III. Experimental Results and Analysis

A. Experimental data and set up

The experimental system was based on GMM-UBM. The SpkID experiments were conducted on the speaking style database of Chinese Corpus Consortium (CCC) [17], containing 64 target speakers and 20 out-of-set speakers. Each target speaker has a 20-second utterance for training and 9 10-second utterances for identification. Each out-of-set speaker has 9 10-second utterances for identification. All the speeches were sampled at 8 kHz with 8-bit width.

Feature extraction was performed on a 20ms frame every 10ms. The pre-emphasis coefficient was 0.97 and hamming windowing was applied. An energy-based voice activity detection (VAD) was performed with each frame labeled either valid or invalid. 16-dimensional MFCC features were extracted from the utterances only for those valid frames with 30 triangular Mel filters used in the MFCC calculation. For each frame, the MFCC coefficients and their first derivative formed a 32-dimentional feature vector. The cepstral mean subtraction [18] in the feature-domain and session variability subspace projection (SVSP) [6] in the model-domain were applied to reduce the affect of channel. SVSP algorithm needs to build the GMM model of the test utterance to compensate the channel variability. The UBM consisted of M = 1,024 Gaussian mixture components, where the value of M was chosen empirically.

The MAP adaptation was used in the baseline system to train the speaker models from the UBM. During the training stage, all mixture components of the UBM were calculated. During the recognition stage, only top N = 4 mixture components of the UBM were used to compute the speaker model likelihoods score. All the experiments were performed on the same computer and the experimental application is single threaded.

B. Experimental Results and Analysis

TABLE I
THE STRUCTURE OF RCT

L	M_1, M_2, M_3, M_4	γ/ *Corr*(%)		
		b=2	b=4	b=6
2	16, 256, X, X	18.3/86.9	10.7/96.9	7.5/97.2
3	4, 32, 256, X	23.3/88.5	12.2/98.0	8.3/98.2
3	4, 16, 256, X	19.7/89.7	10.2/98.2	6.9/98.3
4	4, 16, 64, 256	28.4/90.3	15.1/98.5	10.2/98.6

In Table I, 4 different structures of RCT were defined. γ is the speed up factor of the RCT. Equation (9) was used to calculate γ. M is component number of the UBM and P is the calculated component number in the RCT. *Corr%* is the top three choice accuracy rate of SpkID. According to the experimental results, structure 4 and b = 4 were chosen.

$$\gamma = M / P \quad (9)$$

TABLE II

EFFECT OF THE REDUNDANCY CHILD NODES

A_g	RCN		
	Training Time(Sec)	Rcognition Time(Sec)	*Corr*(%)
0	99.2	898.4	98.5
1	105.6	902.3	98.7
2	110.5	918.6	98.7
3	127.3	952.7	98.8
Max	185.6	1538.3	98.8

In Table II, RCN denotes the redundancy child node algorithm corresponding to step (5) when constructing the RCT. A_g is the number of RCNs. A_g=0 indicates that no node was inserted as N_p's RCN, while A_g=1, 2, 3 indicates that the A_g nodes were inserted as RCNs of N_p, which were the A_g nodes closest to N_p and the distance between any one of them and N_p was smaller than d_{max}. *Max* denotes that all the child nodes closer to N_p than N_{max} were inserted as N_p's RCNs. RCNs improved the SpkID performance yet increased the computation time. The balance between efficiency and performance needs to be considered and in the followed experiments A_g was 1.

TABLE III

THE COMPARISON OF EFFICIENT AND PERFORMANCE

Method	Training Time(Sec)	Rcognition Time(Sec)	*Corr*(%)
Baseline	376.8	14079.1	99.1
TBKS	376.8	1796.4	98.3
ORBP	376.8	5776.2	95.2
HSI	376.8	2132.6	96.8
RCT	110.5	918.6	98.7
RCT+ HSI	110.5	283.9	96.4

RCT can reduce the computation time of SpkID. RCT can also improve the computational efficiency of the GMM-SVM system according to the analysis in Section I. The fusion between RCT and HSI can further improve the efficiency although the degradation of SpkID performance is larger than that when only RCT or HSI is used.

IV. CONCLUSIONS

In this paper we propose a fast training and recognition method based on RCT structural UBM for SpkID. The pruning and regression technology based on RCT structure can improve the computational efficiency with slight degradation of identification performance. The experiments show that the proposed method outperforms the other methods in computational efficiency and identification performance. In future we will further research the affection of the component weights to clustering and searching. Furthermore, how to better fusion the RCT and HSI should be deeply studied.

REFERENCES

[1] NIST Speaker Recognition Evaluation Plan, Online Available http://www.nist.gov/speech/tests/sre/

[2] D. A. Reynolds, T. Quatieri, R. Dunn. Speaker verification using adapted Gaussian Mixture Models [J]. Digital Signal Processing, 2000, Vol. 10, pp: 19-41

[3] W. M. Campbell, D. E. Sturim, D. A. Reynolds and A. Solomonoff. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. ICASSP 2006, pp: 97-100

[4] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. on ASSP. 1980, Vol. 28, pp: 357-366

[5] R. Vogt, S. Sridharan, "Experiments in session variability modeling for speaker verification," ICASSP, 2006. Vol. 1, pp: 897-900.

[6] J. Deng, T. F. Zheng, W. H. Wu. Session variability subspace projection based model compensation for speaker verification. ICASSP 2007, Vol. IV, pp: 57-60

[7] M. Ben, R. Blouet and F. Bimbot. A Monte-Carlomethod for score normalization in automatic speaker verification using Kullback-Leibler distances. ICASSP, 2002, Vol 1, pp: 689-692

[8] J. L. Gauvain, and C. H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. Speech Audio Process, Vol. 2, 1994, pp: 291–298

[9] R. Auckenthaler, J. Mason, Gaussian selection applied to text-independent speaker verification. In Proc. A Speaker Odyssey—Speaker Recognition Workshop, 2001

[10] B. Xiang, T. Berger, Efficient text-independent speaker verification with structural Gaussian mixture models and neural network. IEEE Trans. Speech Audio Process. 2003. Vol. 11 (5), 447–456.

[11] Z. Y. Xiong, T. F. Zheng, Z. J. Song, F. Soong, W. H. Wu, A tree-based kernel selection approach to efficient Gaussian mixture model–universal background model based speaker identification, Speech Communication, 2006, Vol. 48, pp: 1273-1282

[12] Z. Y. Xiong, T. F. Zheng, Z. j. Song, W. h. Wu. Combining Selection Tree with Observation Reordering Pruning for Efficient Speaker Identification Using GMM-UBM. ICASSP, 2005, pp: 625-628.

[13] B. Sun, W. Liu, and Q. Zhong, "Hierarchical speaker identification using speaker clustering," in Int. Conf. Natural Lang. Process. Knowledge Eng., 2003, pp. 299–304.

[14] V. R. Apsingekar and P. L. De Leon, Speaker Model Clustering for Efficient Speaker Identification in Large Population Applications, IEEE Trans. on ASLP, Vol. 17, NO. 4, 2009, pp: 848-853

[15] A. V. Hall. Methods for demonstrating resemblance in taxonomy and ecology, Nature, Vol. 214, pp. 830-831, 1967

[16] X. D. Huang, A. Acero, H. Hon, 2001. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice-Hall.

[17] Chinese Corpus Consortium, Online available, http://www.CCCForum.org/

[18] S. Furui. Cepstral analysis technique for automatic speaker verification. IEEE Trans. on Acoustics, Speech and Signal Processing, 1981. Vol. 29(2), pp: 254-272