Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Efficient Text-Independent Speaker Verification with Structural Gaussian Mixture Models and Neural Network

Bing Xiang, *Member, IEEE,* and Toby Berger, *Fellow, IEEE*

(2013)

***Abstract*—We present an integrated system with structural Gaussian mixture models (SGMMs) and a neural network for purposes of achieving both computational efficiency and high accuracy in text-independent speaker verification. A structural background model (SBM) is constructed first by hierarchically clustering all Gaussian mixture components in a universal background model (UBM). In this way the acoustic space is partitioned into multiple regions in different levels of resolution. For each target speaker, a SGMM can be generated through multilevel maximum a posteriori (MAP) adaptation from the SBM. During test, only a small subset of Gaussian mixture components are scored for each feature vector in order to reduce the computational cost significantly. Furthermore, the scores obtained in different layers of the tree-structured models are combined via a neural network for final decision. Different configurations are compared in the experiments conducted on the telephony speech data used in the NIST speaker verification evaluation. The experimental results show that computational reduction by a factor of 17 can be achieved with 5% relative reduction in equal error rate (EER) compared with the baseline. The SGMM-SBM also shows some advantages over the recently proposed hash GMM, including higher speed and better verification performance.**

EDICS: 1-SPEA

***Index Terms*—Gaussian clustering, neural network, speaker verification, structural Gaussian mixture model.**

I. Introduction

RESEARCH on speaker recognition [1], including identification and verification, has been an active area for several decades. The goal is to have a machine automatically identify a particular person or verify a person's claimed identity from his/her voice. As one of the techniques in biometrics, speaker recognition can be used in many access control applications, such as network security, phone transactions, room access, etc. The speakers are divided into two groups, the enrolled target speakers and the nontarget speakers or background speakers. Both identification and verification can be classified into text-independent and text-dependent applications based on whether or not the person is required to speak pre-determined words or sentences. The focus of this paper is text-independent speaker verification.

Gaussian mixture models (GMMs) [2], [3] recently have become the dominant approach in text-independent speaker recognition. One of the powerful attributes of GMMs is their capability to form smooth approximations to arbitrarily shaped densities [4]. Although for text-dependent applications, hidden Markov models (HMMs) can have some advantages when incorporating temporal knowledge, GMMs still show the best performance to date for text-independent speaker recognition with high accuracy. However, to have a detailed description of the acoustic space and also achieve good verification performance, the number of Gaussian mixture components in each model is usually large, especially when diagonal covariance matrices are used. In a GMM-based text-independent speaker verification system, generally a universal background model (UBM) with a large number of Gaussian mixture components is created based on hours of speech data from nontarget speakers. Then a target speaker GMM is created by maximum a posteriori (MAP) adaptation [5] of the UBM. Although the computational cost nearly can be halved based on the correspondence between the UBM and the target GMM, scoring all components within the UBM for each test feature vector still dominates the processing time during verification.

There have been some approaches proposed for speech recognition before to reduce the computation of Gaussian mixture components in HMMs, such as vector quantization (VQ) [6] or Gaussian selection [7]. The feature space is partitioned with a vector quantizer to associate a shortlist of Gaussians with each codeword. Only the Gaussians in the shortlist are computed precisely, instead of all Gaussians. The likelihood computation can be reduced significantly with small loss of recognition accuracy. Several methods using different thresholding schemes to compute shortlists of Gaussians that are most likely to have a significant contribution to the mixture sum were devised in [8]. In [9] and [10], approaches based on tree-structured Gaussian densities were proposed to achieve computational efficiency in speech recognition. A tree structure with bottom-up clustering was also proposed in [11] for purpose of pruning the aggregated Gaussian models. In [12], a decision-tree technique was proposed to partition the feature space hierarchically. Each node in the tree represents a linear hyperplane, which is obtained based on maximum mutual information. The computation required for evaluating Gaussians is reduced by a factor of 25 with 4% relative degradation in word error rate.

Manuscript received April 10, 2002; revised April 8, 2003. This work was supported in part by NSF Grant CCR-9980616. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ramesh A. Gopinath.

B. Xiang was with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA. He is now with the Speech and Language Processing Department, BBN Technologies, Cambridge, MA 02138 USA (e-mail: bxiang@bbn.com).

T. Berger is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: berger@ece.cornell.edu).

Recently, inspired by Gaussian selection in speech recognition, a hash GMM was proposed and applied to text-independent speaker verification [13]. In this method a small hash GMM is trained first with entire training data. Then a shortlist of the indices of mixture components in the large UBM is generated for each hash GMM component based on the occurring frequencies of the best scoring components in the hash GMM and the UBM. For each test feature vector, only those components in the shortlist of the highest-scoring hash component are evaluated. Computational reduction by a factor of ten was achieved with minor degradation in verification performance. Some other approaches to reducing the computational cost in speaker verification systems have also been proposed. It was shown that by lowering model order or decreasing frame rate [14], [15], computational cost can be reduced by a factor of four with negligible degradation of verification performance. An efficient scoring algorithm was proposed in [16] for speaker identification. Rapid pruning of unlikely speaker model candidates was achieved by reordering the sequence of observation vectors. It improved the speed by a factor of six compared to the conventional sequential sampling.

In addition to GMM-based system, artificial neural networks (ANN) also have drawn much attention in both the speech recognition and the speaker recognition communities. The application of neural networks to speaker recognition dates back more than a decade [17]. Being nonlinear classifiers, neural networks possess the ability to discriminate the characteristics of different speakers. But when the dimensionality of the input feature space is high and the network structure becomes sophisticated, the convergence during training is slow. Moreover the performance obtained when using a neural network alone is inferior to that of GMM. Some hybrids of neural networks and HMMs were introduced before in speech recognition with the neural network acting as a pre-processor [18] or posterior evaluator [19] for HMMs. Recently a neural network was used in speaker verification as a post-processor in the score level to combine the scores from three GMM-based systems with different features [20]. The performance from multisystem fusion is better than that from any of the three systems. There are also some other applications of neural networks in speaker recognition, such as to design discriminative features for robust speaker recognition to compensate for telephone handset distortion [21].

For purposes of achieving both computational efficiency and verification performance, we propose an integrated system with structural Gaussian mixture models (SGMMs) and a multilayer feed-forward neural network. A structural background model (SBM) is created first based on a well-trained UBM. During the construction, all Gaussian mixture components in the UBM are clustered hierarchically. In this way the acoustic space is partitioned into multiple regions in different levels of resolution. Each node in the tree-structured SBM is represented by a Gaussian. For each target speaker, a SGMM is generated through multilevel MAP adaptation on the SBM. During test, the computational cost can be reduced significantly by searching down the SBM tree and scoring only a small subset of the Gaussian mixture components in the SBM and SGMM. Furthermore, multiple scores from different layers of the SGMM-SBM are combined via a neural network to form the final decision. This integrated system shows larger discriminative ability than the baseline system GMM-UBM. Variable configurations on the tree structures, distance measures, cluster centroid estimations and neural networks are compared in this paper. Experiments are conducted on the data used in the NIST 1999 speaker verification evaluation. The results show that computational reduction by a factor of 17 can be achieved with 5% relative reduction in equal error rate (EER) compared with the GMM-UBM. The SGMM-SBM also shows advantages over the hash GMM one of which is better verification performance.

This paper is organized as follows. In Section II we briefly review GMM-based speaker verification systems. Then we introduce the construction of the SBM and SGMM in detail in Section III. The verification procedure and the score fusion via neural network are presented in Section IV. In Section V our experimental results are reported. The last section summarizes the principal conclusions.

II. Gaussian Mixture Models

Since the construction of a SBM is based on the well-trained UBM and also the GMM-UBM acts as a baseline system in this paper, we briefly review the GMM-UBM speaker recognition system [2]. The UBM with M Gaussian components is trained by the EM algorithm [22] using hours of speech data from non-target speakers. The likelihood function of a K-dimensional feature vector $\vec{x}_t$ is

$$p(\vec{x}_t|\lambda_{UBM}) = \sum_{i=1}^{M} w_i G_i(\vec{x}_t) \tag{1}$$

where w_i is the weight of the i-th component, and $G_i(.)$ is the Gaussian probability density function (pdf) with mean $\vec{\mu}_i$ and covariance matrix Σ_i

$$G_i(\vec{x}_t) = \frac{1}{(2\pi)^{\frac{K}{2}}|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{x}_t-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}_t\vec{\mu}_i)}. \tag{2}$$

For each target speaker, a GMM can be created through MAP adaptation of the UBM. Based on the experimental results, adapting only the mean for each Gaussian component yields the best performance in speaker verification. The mean $\vec{\nu}_i$ of the ith component of the GMM is obtained by

$$\vec{\nu}_i = \frac{\eta_i}{\eta_i + r} E_i(\vec{x}) + \frac{r}{\eta_i + r}\vec{\mu}_i \tag{3}$$

where $\vec{x}$ is the set of feature vectors and $\vec{\mu}_i$ is the mean of the i-th component in the UBM. Here η_i and $E_i(\vec{x})$ are computed from the expectation step of the EM algorithm as

$$\eta_i = \sum_{t=1}^{T} P(i|\vec{x}_t), \tag{4}$$

and

$$E_i(\vec{x}) = \frac{1}{\eta_i}\sum_{t=1}^{T} P(i|\vec{x}_t)\vec{x}_t. \tag{5}$$

r is a relevance factor and $P(i|\vec{x}_t)$ is the a posteriori probability of the i-th component given feature vector $\vec{x}_t$

$$P(i|\vec{x}_t) = \frac{w_i G_i(\vec{x}_t)}{\sum_{j=1}^{M} w_j G_j(\vec{x}_t)}. \tag{6}$$

The speaker verification task is a hypothesis testing problem. The decision between whether the $\vec{x}$ is generated by the target speaker or by someone else is based on the average frame log-likelihood ratio between the target GMM and the UBM. With the general assumption of independence of the feature vectors of $\vec{x}$, it is calculated as

$$\begin{aligned}&\frac{1}{T}\left(\log p(\vec{x}|\lambda_{TGT}) - \log p(\vec{x}|\lambda_{UBM})\right)\\ &\quad = \frac{1}{T}\left(\sum_{t=1}^{T} \log p(\vec{x}_t|\lambda_{TGT}) - \sum_{t=1}^{T} \log p(\vec{x}_t|\lambda_{UBM})\right)\end{aligned} \tag{7}$$

where λ_{TGT} is the parameter set of the target GMM. By thresholding the average frame log-likelihood ratio, a decision can be made to either accept or reject the target speaker hypothesis. It is observed that only a few of the mixtures contribute significantly to the likelihood value when a large GMM is evaluated. Based on this fact and also the correspondence between the target GMM and the UBM, a fast scoring approach was proposed [2]. For each feature vector, the top C scoring Gaussian components among the M components in the UBM are found first. Then only the corresponding C components in the target GMM are evaluated. Since C is usually far less than M, the computational cost is reduced by almost a half.

III. Construction of SBM and SGMM

The training of SBM and SGMM is shown in Fig. 1.

Based on the UBM, an L-layer tree-structured SBM, as shown in Fig. 2, can be generated to model the structure of acoustic space. Through a top-down hierarchical clustering, each node in the upper $L-1$ layers represents a cluster of the Gaussian components in the UBM and is modeled by a single Gaussian pdf. Each leaf node corresponds to a separate Gaussian mixture component in the UBM. In this way, the SBM models the acoustic space in different levels of resolution. Once the SBM is constructed, a target SGMM with the same tree structure can be created by multilevel MAP adaptation of the SBM. Each layer in the SGMM is generated via MAP adaptation of the corresponding layer in the SBM so that the Gaussian components in the SGMM and the SBM retain a close correspondence to each other, which is useful during verification.

A. Distance Measure

Before the construction of the SBM, a distance measure between two Gaussians must be defined. Several distance measures have been proposed in the literature. Kullback–Leibler (KL) divergence [23] and Bhattacharyya distance [24] are two of the widely used measures. Consider two Gaussian mixture components G_i and G_j with distributions $N(\vec{\mu}_i, \Sigma_i)$ and $N(\vec{\mu}_j, \Sigma_j)$. The symmetric KL divergence is defined as the sum of the relative entropy between G_i and G_j and that

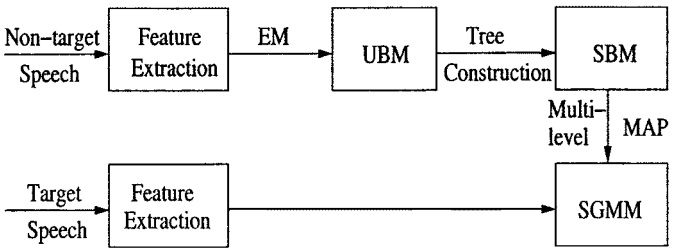

Fig. 1. Diagram of training.

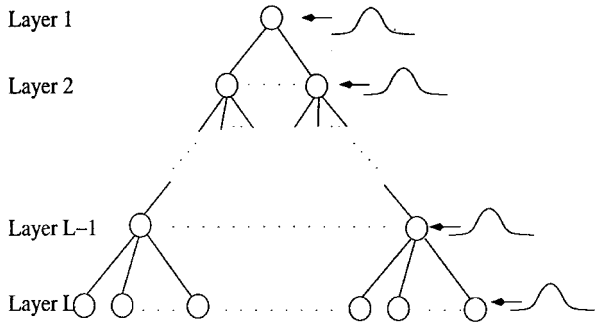

Fig. 2. Tree structure.

between G_j and G_i. It quantifies the information for discriminating between the two Gaussian components. When diagonal covariance matrices are assumed, it can be represented as

$$\begin{aligned}d_{KL}(i,j) &= \int G_i(x) \ln \frac{G_i(x)}{G_j x} dx + \int G_j(x) \ln \frac{G_j(x)}{G_i(x)} dx\\ &= \frac{1}{2}\sum_{k=1}^{K}\left[\frac{\sigma_i^2(k) + (\mu_i(k) - \mu_j(k))^2}{\sigma_j^2(k)}\right.\\ &\qquad \left. + \frac{\sigma_j^2(k) + (\mu_i(k) - \mu_j(k))^2}{\sigma_i^2(k)} - 2\right]\end{aligned} \tag{8}$$

where $\mu_i(k)$ is the kth element of the mean vector $\vec{\mu}_i$, and $\sigma_i^2(k)$ is the kth diagonal element of the covariance matrix Σ_i.

We also consider the Bhattacharyya distance [24]

$$\begin{aligned}d_{BH}(i,j) &= \frac{1}{2}\ln \frac{\left|\frac{\Sigma_i + \Sigma_j}{2}\right|}{|\Sigma_i|^{\frac{1}{2}}|\Sigma_j|^{\frac{1}{2}}}\\ &\quad + \frac{1}{8}(\vec{\mu}_i - \vec{\mu}_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (\vec{\mu}_i - \vec{\mu}_j)\end{aligned} \tag{9}$$

which is closely tied to upper bounding the Bayes classification error. The second term of the Bhattacharyya distance is the Mahalanobis distance using an average covariance matrix.

B. Cluster Centroid Estimation

Cluster centroid estimation is another important issue for Gaussian clustering during the tree construction. The centroid of a cluster of Gaussian components can be estimated using maximum likelihood (ML) estimation. In both [6] and [9], it was assumed that the numbers of data samples from each mixture component are equal. However, this may cause some performance degradation because the acoustic space is modeled inappropriately. With mixture weights considered, which

represent the amount of data corresponding to each mixture component, the ML estimation can be approximated by

$$\mu_c(k) = \frac{\sum_{i\in R} w_i \mu_i(k)}{\sum_{i\in R} w_i} \tag{10}$$

$$\sigma_c^2(k) = \frac{\sum_{i\in R} w_i \left(\sigma_i^2(k) + \mu_2^i(k)\right)}{\sum_{i\in R} w_i} - \mu_c^2(k) \tag{11}$$

where $\mu_c(k)$ is the kth element of the mean vector for the centroid of cluster R and $\sigma_c^2(k)$ is the kth diagonal element of the centroid covariance matrix. The weight of R can be calculated as the sum of all weights w_i for $i \in R$

$$w_c = \sum_{i\in R} w_i. \tag{12}$$

When KL divergence is used, we also propose a KL approach for cluster centroid estimation. An iterative algorithm is used to minimize the sum of the KL divergence between the cluster centroid and all the Gaussians within the cluster, i.e.,

$$D_{KL} = \sum_{i\in R} d_{KL}(c,i). \tag{13}$$

By setting the derivatives of D_{KL} with respect to the mean $\mu_c(k)$ and variance $\sigma_c^2(k)$ to zero separately, the formulae for updating the mean and variance are

$$\mu_c(k) = \frac{\sum_{i\in R} \left(\frac{1}{\sigma_c^2(k)} + \frac{1}{\sigma_i^2(k)}\right) \mu_i(k)}{\sum_{i\in R} \left(\frac{1}{\sigma_c^2(k)} + \frac{1}{\sigma_i^2(k)}\right)}, \tag{14}$$

and

$$\sigma_c^2(k) = \left(\frac{\sum_{i\in R} \left[\sigma_i^2(k) + (\mu_c(k) - \mu_i(k))^2\right]}{\sum_{i\in R} \frac{1}{\sigma_i^2(k)}} \right)^{\frac{1}{2}}. \tag{15}$$

With the initial values generated by the ML approach, the final mean and variance can be obtained after several iterations with the mean updated first in each iteration.

Similarly, to minimize the within-cluster Bhattacharyya distance, the mean and variance can be updated iteratively [25] through the BH approach by

$$\mu_c(k) = \frac{\sum_{i\in R} \left(\sigma_c^2(k) + \sigma_i^2(k)\right)^{-1} \mu_i(k)}{\sum_{i\in R} \left(\sigma_c^2(k) + \sigma_i^2(k)\right)^{-1}}, \tag{16}$$

$$\sigma_c^2(k) = \frac{|R|}{\sum_{i\in R} \left[\frac{2}{\sigma_c^2(k) + \sigma_i^2(k)} - \left(\frac{\mu_c(k) - \mu_i(k)}{\sigma_c^2(k) + \sigma_i^2(k)} \right)^2 \right]} \tag{17}$$

where $|R|$ is the number of components in cluster R.

C. Tree Construction

During the SBM tree construction process, the number of layers L and the structure in the upper $L-1$ layers are determined first before clustering. The algorithm for the Gaussian clustering is described as follows.

1) The mean and variance of the first layer, i.e., the root node, are calculated by (10) and (11) with all Gaussian components of the UBM in one cluster.
2) Then the nodes in the next layer are initiated by the *minimax* method based on the KL divergence or Bhattacharyya distance. The pdf of each node is obtained by linear interpolation between the parameters of the original pdf and that of the parent node [23].
3) A k-means procedure is applied to cluster Gaussian mixture components into each node. In each iteration, when KL divergence is used, the mean and variance can be updated either by the ML approach [(10), (11)] or by the KL approach [(14), (15)]. Similarly, the ML or BH approach [(16), (17)] can be applied when Bhattacharyya distance is chosen. The initial values in both KL and BH approaches are generated by the ML approach as mentioned before. The values of mean, variance and weight are stored in each node once the distance converges.
4) The same procedures in step (2) and (3) are repeated for other layers until the clusters in layer $L-1$ are generated.
5) Each Gaussian component in each of the clusters in layer $L-1$ is stored in a separate child node, the node in the leaf layer.

Due to the variable cluster size, the number of branches for those nodes in layer $L-1$ varies from node to node and depends on the different approaches to tree construction. However, the total number of leaf nodes is always equal to M.

It should be pointed out that a tree structure was applied in a multigrained model [26] for speaker recognition before. Each node in the tree is modeled by a GMM and represents a phone class or a phone. So all the training data must be labeled first. However, no transcription is needed for training the SBM because the Gaussian mixture components are clustered in an unsupervised way. It has been shown in [23] that such an unsupervised mixture grouping is phonologically meaningful. We also notice that a VQ-based GMM was proposed in [27] for text-independent speaker identification. The whole acoustic space is clustered into several subspaces via vector quantization. Then each subspace is characterized by a GMM. However, for each test feature vector the scores are computed against all GMMs to find the maximum. Therefore no computational efficiency was considered there.

D. Multilevel Adaptation

After construction of the SBM, a separate model needs to be trained for each target speaker. Since all nodes in the same layer of the SBM compose a Gaussian mixture model, with the sum of weights being 1, a tree-structured target SGMM can be obtained by adapting the SBM with MAP adaptation [5] in each layer separately. The mean $\vec{\nu}_{l,i}$ of the i-th component in layer l of the SGMM is obtained by

$$\vec{\nu}_{l,i} = \frac{\eta_{l,i}}{\eta_{l,i} + r_{l,m}} E_{l,i}(\vec{x}) + \frac{r_{l,m}}{\eta_{l,i} + r_{l,m}} \vec{\mu}_{l,i} \tag{18}$$

where $r_{l,m}$ is a relevance factor for adaptation of mean in layer l. $E_{l,i}(\vec{x})$ and $\eta_{l,i}$ are obtained by

$$E_{l,i}(\vec{x}) = \frac{1}{\eta_{l,i}} \sum_{t=1}^{T} P_l(i|\vec{x}_t)\vec{x}_t, \tag{19}$$

$$\eta_{l,i} = \sum_{t=1}^{T} P_l(i|\vec{x}_t). \tag{20}$$

$P_l(i|\vec{x}_t)$ is the a posteriori probability of i-th component in layer l given the feature vector $\vec{x}_t$

$$P_l(i|\vec{x}_t) = \frac{w_{l,i}G_{l,i}(\vec{x}_t)}{\sum_{j=1}^{M_l} w_{l,j}G_{l,j}(\vec{x}_t)} \tag{21}$$

where M_l is the number of nodes in layer l. Similarly, the variance $\overrightarrow{\xi^2}_{l,i}$ of the i-th component in layer l of the SGMM can be obtained by (see (22) at the bottom of the page) where $\vec{h}_{l,i}$ is a parameter vector from the prior density [5] and $r_{l,v}$ is a relevance factor for adaptation of variance in layer l. When $r_{l,m}$ is equal to $r_{l,v}$ and

$$\vec{h}_{l,i} = r_{l,v}\overrightarrow{\sigma^2}_{l,i} \tag{23}$$

(22) can be rearranged as (see (24) at the bottom of the page). It can be further simplified to

$$\overrightarrow{\xi^2}_{l,i} = \frac{\eta_{l,i}}{\eta_{l,i}+r_{l,v}}E_{l,i}(\overrightarrow{x^2}) + \frac{r_{l,v}}{\eta_{l,i}+r_{l,v}} \times \left(\overrightarrow{\sigma^2}_{l,i} + \overrightarrow{\mu^2}_{l,i}\right) - \overrightarrow{\nu^2}_{l,i}, \tag{25}$$

where

$$E_{l,i}(\overrightarrow{x^2}) = \frac{1}{\eta_{l,i}}\sum_{t=1}^{T} P_l(i|\vec{x}_t)\overrightarrow{x_t^2}. \tag{26}$$

The weight $\varpi_{l,i}$ of the i-th component in layer l of the SGMM can be obtained by [2]

$$\varpi_{l,i} = \frac{\rho_{l,i}}{\sum_{j=1}^{M_l}\rho_{l,j}}, \tag{27}$$

where

$$\rho_{l,i} = \frac{\eta_{l,i}}{\eta_{l,i}+r_{l,w}}\cdot\frac{\eta_{l,i}}{T} + \frac{r_{l,w}}{\eta_{l,i}+r_{l,w}}w_{l,i}. \tag{28}$$

$r_{l,w}$ is a relevance factor for adaptation of weight in layer l. In this way the Gaussian components in the target SGMM retain correspondence to those in the SBM, which is similar with the correspondence between the target GMM and the UBM in a GMM-UBM system.

IV. Verification

Fig. 3 is a block diagram of our speaker verification system. After features are extracted from the test speech, SBM and SGMM both are scored to generate the background and target scores. These scores are combined via a multilayer perceptron (MLP) to produce the final decision.

A. Scoring SBM and SGMM

For each test feature vector $\vec{x}_t$, based on the likelihood in each node, the highest-scoring node in the second layer of the SBM, node $g^*_{2,t}$, is found first. Then all its child nodes in the third layer are scored. The tree is searched in this way down to a certain node $g^*_{L-1,t}$ in layer $L-1$. The scores obtained at the nodes along the path become the background scores for the corresponding layers. The background score in layer l is

$$s_{b_l} = \frac{1}{T}\sum_{t=1}^{T}\log \max_{i_{l,t,b}\le i\le i_{l,t,e}} w_{l,i}G_{l,i}(\vec{x}_t),\ 2\le l\le L-1 \tag{29}$$

where $i_{l,t,b} \le i \le i_{l,t,e}$ are the indices of those scored components in layer l for $\vec{x}_t$, i.e. the child nodes of $g^*_{l-1,t}$. The root node is in fact $g^*_{1,t}$. All child nodes of node $g^*_{L-1,t}$ are evaluated, comprising a subset of the leaf nodes. Similar with the GMM-UBM baseline, the logarithm of the sum of the top C likelihoods in the subset is used as the background score of the leaf layer

$$s_{b_L} = \frac{1}{T}\sum_{t=1}^{T}\log\sum_{i=1}^{C} w_{L,m_{t,i}}G_{L,m_{t,i}}(\vec{x}_t) \tag{30}$$

where $m_{t,i}$, $1\le i\le C$ are the indices of the top C components in the subset. This also makes the comparison with the baseline GMM-UBM fair, where the top C likelihoods among all Gaussian components are counted. Based on the indices of those nodes in the SBM which contribute to different layers' background scores, the corresponding nodes in the upper $L-2$ layers of the target SGMM and the corresponding C leaf nodes are evaluated to generate the multiple target scores s_{tl}, with $2\le l\le L$.

B. Score Combination via Multilayer Perceptron

It would be straightforward to combine the multiple scores from the SGMM and SBM via linear combination or choose the maximum of the multiple likelihood ratios. However, these approaches cannot bring obvious improvement over the performance obtained from the leaf layer scores only, as shown later in the experimental results section. The verification performance obtained by using the scores from the lower layer is better than its counterpart from the upper layers since it describes the acoustic space with higher resolution. However, it is believed that the scores from the upper layers still catch some useful information for speaker verification. Especially in the situation of sparse training data, the upper layers with smaller number

$$\overrightarrow{\xi^2}_{l,i} = \frac{\vec{h}_{l,i} + \sum_{t=1}^{T} P_l(i|\vec{x}_t)diag\left[(\vec{x}_t-\vec{\nu}_{l,i})(\vec{x}_t-\vec{\nu}_{l,i})^T\right] + r_{l,m}diag\left[(\vec{\mu}_{l,i}-\vec{\nu}_{l,i})(\vec{\mu}_{l,i}-\vec{\nu}_{l,i})^T\right]}{\eta_{l,i}+r_{l,v}} \tag{22}$$

$$\overrightarrow{\xi^2}_{l,i} = \frac{\sum_{t=1}^{T} P_l(i|\vec{x}_t)diag\left[(\vec{x}_t-\vec{\nu}_{l,i})(\vec{x}_t-\vec{\nu}_{l,i})^T\right] + r_{l,v}\left(\overrightarrow{\sigma^2}_{l,i} + diag\left[(\vec{\mu}_{l,i}-\vec{\nu}_{l,i})(\vec{\mu}_{l,i}-\vec{\nu}_{l,i})^T\right]\right)}{\eta_{l,i}+r_{l,v}} \tag{24}$$

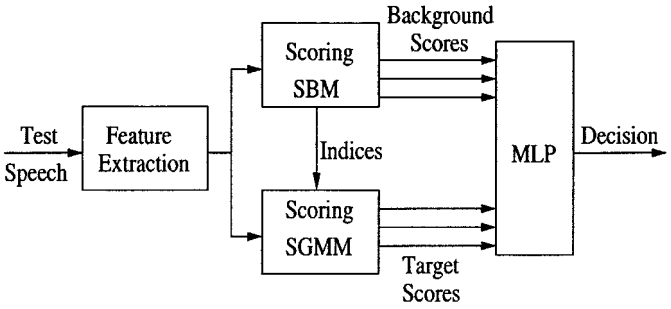

Fig. 3. Diagram of test.

of Gaussians may be trained more sufficiently than the lower layers. To reflect the nonlinear dependencies among the scores from different levels, it is desirable to use a neural network for score combination [20]. Theory shows that the standard multilayer perceptron (MLP) with one hidden layer and smooth nonlinear activation functions can realize an arbitrary nonlinear decision boundary [28]. The three-layer MLP combines the scores in different layers of the SBM and SGMM to obtain the final score for each trial. As shown in Fig. 4, the scores act as the inputs of the MLP.

The output y_j at the j-th node of the hidden layer is obtained by a sigmoid function

$$y_j = \frac{1}{1 + e^{-\left(\vec{w}_{1,j}^T \vec{s} - \delta_{1,j}\right)}} \tag{31}$$

where $\vec{w}_{1,j}$ and $\delta_{1,j}$ are the weights connecting the input and hidden layer. The elements in $\vec{s}$ are the separate scores from the different layers of the SGMM and SBM for each trial and act as the inputs of the MLP. Similarly, the final output z is computed as

$$z = \frac{1}{1 + e^{-\left(\vec{w}_2^T \vec{y} - \delta_2\right)}} \tag{32}$$

where $\vec{w}_2$ and δ_2 are the weights between the hidden layer and the single output neuron. This type of MLP has $2(L-1)$ input nodes with the background scores and target scores as separate inputs so that z is related with $\vec{s}$ by a function $f_1(.)$

$$z = f_1\left(s_{t_2}, s_{b_2}, \ldots, s_{t_L}, s_{b_L}\right). \tag{33}$$

Another kind of MLP is also considered in this paper. It employs $L-1$ input nodes which use the $L-1$ log-likelihood ratios obtained from layer 2 to layer L of the SGMM and the SBM, with the background scores subtracted from the corresponding target scores. In this case z can be represented by

$$z = f_2\left(s_{t_2} - s_{b_2}, \ldots, s_{t_L} - s_{b_L}\right). \tag{34}$$

Then a final decision can be made based on the value of the MLP output, z.

The MLP is trained with the widely used back-propagation (BP) algorithm [29], a gradient-descent approach. With scores from some known target trials and impostor trials, the desired output is set to 1 for target trials and 0 for impostor trials. Through such supervised training, the MLP increases the ability of discriminating target and impostor speakers. During test, when a similar set of scores are obtained from the SGMM-SBM and put into the MLP, it is possible to correct some errors, i.e., to convert low log-likelihood ratios of target trials to a higher output, and vice versa for the high log-likelihood ratios from some impostor trials. A higher verification performance can be expected in the integrated system. Also, since the dimension of input is almost an order of magnitude lower than the dimension of feature vector, it is easy to converge so the training is fast. Different sizes of the hidden layer are tested in the experiments section.

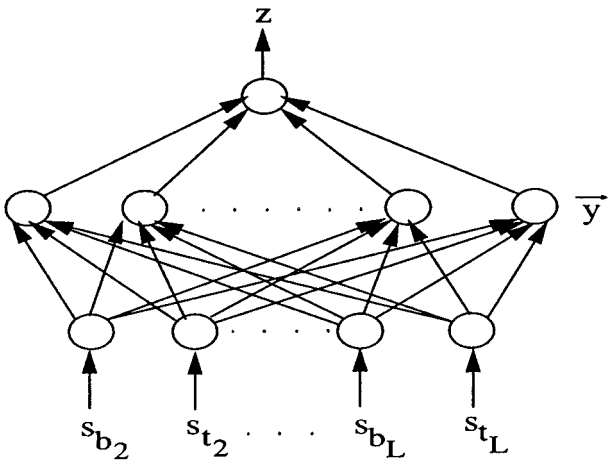

Fig. 4. Multilayer perceptron

V. Experiments

A. Database

The SGMM-SBM were evaluated on the telephony speech data (Switchboard II, phase 3) used in the NIST 1999 speaker verification evaluation. Four hours of speech from 240 male and 240 female test segments in the NIST 1998 evaluation (Switchboard II, phase 2), with handsets balanced, are used for training the gender-dependent SBM. Each of 230 male target speakers and 309 female target speakers has two minutes of speech for training. Among them 60 male and 60 female speakers are held out for training the MLP. 29 241 gender-matched verification trials from the other 170 male and 249 female speakers are used for the test. So there is no speaker overlap between the MLP training data and test data. The duration of each test segment varies from a few seconds to one minute, with the majority of tests falling into a range between 15 to 45 s. The 13 626 verification trials from 87 male and 114 female speakers in the NIST 1997 evaluation (Switchboard II, phase 1) are used as development data for tuning the structure of the MLP. The ratios between target and impostor trials in both evaluations are roughly 1:10. More details about the NIST evaluation can be found at [30].

B. Front End Processing

The frame rate is set to 16 ms. Nineteen-dimensional Mel-frequency cepstral coefficients (MFCC) are extracted from silence-removed and bandlimited data first. Then the channel effect is compensated by transforming the MFCCs with feature warping [31], which recently was formulated within the framework of short-time Gaussianization [32]. The 19 delta coefficients are

calculated based on the warped MFCCs and appended to form a 38-dimensional feature vector which is used in all the following experiments.

C. Evaluation Measure

The evaluation of the speaker verification system is based on Detection Error Tradeoff (DET) curves, which show the tradeoff between false alarm (FA) and false rejection (FR) errors. In addition to the equal error rate (EER), there is also a detection cost function (DCF) defined for the NIST evaluation [33]

$$DCF = C_{FA}P_{FA|N}P_N + C_{FR}P_{FR|T}P_T \tag{35}$$

where P_N and P_T are the a priori probability of nontarget and target tests with $P_N = 0.99$ and $P_T = 0.01$; the specific cost factors are $C_{FA} = 1$ and $C_{FR} = 10$, which shifts the point of interest toward low FA rates.

D. Computational Reduction Factor

To represent the computational requirements of the GMM-UBM relative to the SGMM-SBM, a computational reduction factor is defined as

$$F = \frac{M + C}{M_s + C_s} \tag{36}$$

where M, C and C_s are the numbers of Gaussian components evaluated in the UBM, target GMM and SGMM, respectively. M_s is the average number of Gaussian likelihoods computed in the SBM for each feature vector. In all experiments, M is 1024 and C is 5. C_s can be obtained by

$$C_s = C + L - 2 \tag{37}$$

since we evaluate one node in each layer from layer 2 to layer $L-1$ and C nodes in the leaf layer of the SGMM. The value of M_s is approximated by

$$M_s = \frac{1}{T}\sum_{t=1}^{T}\sum_{l=2}^{L} n_{l,t} \tag{38}$$

where $n_{l,t}$ is the number of nodes evaluated in layer l of the SBM for test feature vector $\vec{x}_t$ during the scoring as described before.

E. Experimental Results

1) Variable Tree Structures With KL-ML Approach: Several tree structures were tested first with KL-ML approach, i.e., with KL divergence and ML estimation. The log-likelihood ratio from the leaf layers only is used in the first three sections of experimental results. It will be denoted as GMM-SBM to distinguish it from the score fusion of SGMM-SBM discussed later. Only mean was adapted for the leaf layer of the SGMM. The relevance factor $r_{L,m}$ was set to 1 based on some initial experiments. Five three-layer trees and one four-layer tree were examined separately, with F increased as shown in Table I. n_{2-4} are the numbers of nodes in the second, third and fourth layer. The first row corresponds to the baseline GMM-UBM with 1024 components. Each three-layer tree had 2, 4, 8, 16, or 32 nodes in the second layer, and the four-layer tree had four nodes in the second layer, with eight child nodes each, so that 32 nodes exist in the third layer. Compared to the GMM-UBM, the four-layer KL-ML GMM-SBM achieves computational reduction by a factor of about 17, and causes only around 5% relative degradation of verification performance in terms of both EER (from 12.9% to 13.5%) and minimum DCF (from 0.0470 to 0.0495). Thus, although the number of Gaussians computed precisely is small, these Gaussians can still approximate the likelihood of test feature vectors when the tree is constructed based on the acoustic similarities between all Gaussian components.

TABLE I
VARIABLE TREE STRUCTURES WITH KL-ML APPROACH FOR GMM-SBM

n_2	n_3	n_4	F	EER(%)	min DCF(10^{-3})
1024			1	12.9	47.0
2	1024		2	13.1	47.1
4	1024		3	13.3	47.7
8	1024		6	13.4	48.6
16	1024		9	13.4	48.7
32	1024		13	13.3	48.7
4	32	1024	17	13.5	49.5

2) Variable Distance Measures and Cluster Estimations: For each tree structure mentioned above, the other three different combinations of distance measures and centroid estimations are also tested. The KL-KL approach constructs the SBM based on the KL divergence and KL estimation. The BH-ML and BH-BH approaches use Bhattacharyya distance and ML estimation or BH estimation, respectively. We also compare these approaches with the reduced-size GMM-UBM. To make the comparison fair, we keep the resolution of the 1024-component GMM-UBM intact. For each test feature vector, a subset of Gaussian components is chosen randomly without any knowledge. The size of the subset is set as 512, 256, 128, or 64, respectively. It is used as M_s to calculate the computational reduction factor F. The minimum DCFs from multiple systems are shown in Fig. 5. The four different GMM-SBMs show similar performances when the computational reduction factor F is below 12. When F exceeds 12, the KL-ML approach becomes slightly better than the other three.

To achieve computational reduction similar to that of the four-layer SBM, the size of the chosen subset of the GMM-UBM has to be reduced to 64. However, the DCF obtained from the randomly chosen 64-component GMM-UBM increases by 40% relatively compared to the 1024-component GMM-UBM. Furthermore, compared to the 512-component GMM-UBM, the four-layer KL-ML GMM-SBM achieves computational reduction of around 8.5 with almost no performance loss.

The EERs and minimum DCFs obtained from all four-layer GMM-SBMs using the KL-ML, KL-KL, BH-ML, and BH-BH approaches are shown in Table II. The four-layer KL-ML SBM has a relatively larger computational reduction factor and also a better verification performance than the other three approaches in terms of both EER and minimum DCF. The spread of the values of F among the four different four-layer GMM-SBMs is larger than those among the three-layer trees. It is due to the larger differences between the cluster sizes in the four-layer trees constructed with different approaches than those in the

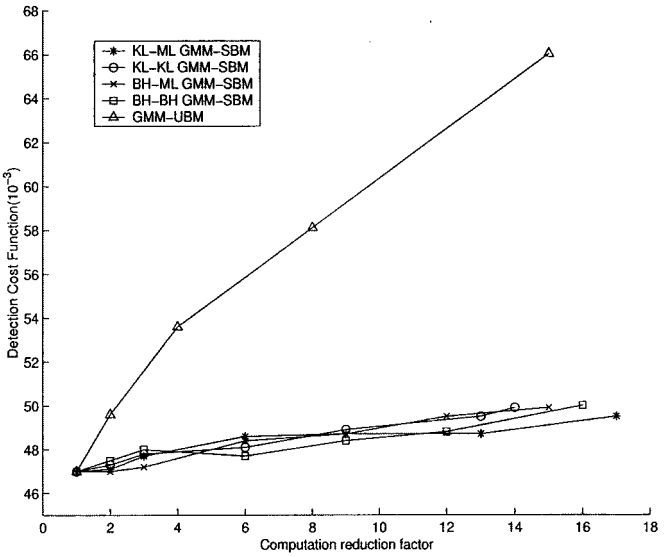

Fig. 5. Comparison between different systems (e.g. KL-ML means KL divergence and ML estimation).

TABLE II
COMPARISON BETWEEN MULTIPLE SYSTEMS

System	F	EER(%)	min DCF(10^{-3})
KL-ML SBM	17	13.5	49.5
KL-KL SBM	14	13.6	49.9
BH-ML SBM	15	13.6	49.9
BH-BH SBM	16	13.5	50.0
64-comp UBM	15	16.8	66.0
128-comp UBM	8	14.9	58.1
256-comp UBM	4	13.8	53.6
512-comp UBM	2	13.2	49.6

three-layer trees. It was found that the variance of the cluster size from KL-ML approach is smaller than the others. This means the Gaussian components are distributed more evenly among the tree nodes, which may contribute to its highest performance.

3) Comparison With Hash GMM: The four-layer KL-ML GMM-SBM was also compared with the hash GMM [13], as shown in Table III. Two 32-component hash GMMs were tested with shortlist size 32 or 64, which achieve $F = 15$ and $F = 10$, respectively. The GMM-SBM is slightly better than both hash GMMs, achieving a larger computational reduction and slightly better performance. This is especially so in the area of low false alarm rates because the GMM-SBM has lower optimal DCF. Since the training of SBM is based on the UBM only, it is also much faster than the generation of a hash GMM with all the shortlists, since the latter involves the training of a 32-component GMM with hours of speech data.

4) Multilevel MAP Adaptation of SBM: As described before, the mean, variance and weight of each node in the SGMM can be obtained via multilevel MAP adaptation of the SBM. In this section, the effects of adapting the different sets of parameters are evaluated. The results of adapting mean only, adapting mean and variance and adapting all three parameters are shown in Table IV. In current experiments, all relevance factors are set to 1. The likelihood ratios between each layer of the SGMM and

TABLE III
COMPARISON WITH HASH GMM

System	F	EER(%)	min DCF(10^{-3})
KL-ML GMM-SBM	17	13.5	49.5
hash GMM (32-list)	15	13.9	51.0
hash GMM (64-list)	10	13.7	49.9

TABLE IV
MULTI-LEVEL MAP ADAPTATION OF SBM

	Adapt μ		Adapt μ,σ		Adapt μ,σ,w	
Layer	EER(%)	DCF(10^{-3})	EER(%)	DCF(10^{-3})	EER(%)	DCF(10^{-3})
2	24.1	98.1	22.9	94.8	22.9	96.0
3	19.5	72.6	20.8	74.8	20.8	77.9
4	13.5	49.5	16.8	53.9	17.1	56.2

the corresponding layer in the SBM are used for the decisions. As expected, the performance obtained from the lower layer is better than its counterpart from the upper layers. For layer 2 with only four nodes, i.e., four Gaussian components, adapting both mean and variance achieves better performance than the other two adaptation approaches. For layer 3 and layer 4 with 32 and 1024 components respectively, the best performance is achieved by adapting mean only. This is consistent with the results of the high-order GMM-UBM reported in [2]. The reason why adapting mean and variance works well for layer 2 is related to the small number of Gaussian components and relatively large amount of training data. The best ways for the adaptation in each layer will be used for the score fusion described in the next section.

5) Score Combination With MLP in SGMM-SBM: Recall that to combine the multiple scores from the four-layer SGMM and SBM, we proposed two kinds of MLPs, both with one hidden layer. The MLPs are trained with the held out speakers from the NIST 1999 evaluation as mentioned before. Different sizes of hidden layer were tested with the development data from the NIST 1997 evaluation by changing the number of neurons from 10 to 60. As shown in Fig. 6, the optimal number of neurons in the hidden layer is 30 for the three-input MLP and 50 for the six-input MLP in terms of minimum DCF. The performance from the six-input MLP is better than that from the three-input one. This is not surprising since, with all background scores and target scores as separate inputs, the MLP learns more from the training data, making it easier to distinguish the target trials from the impostor trials.

With 30 hidden nodes in the three-input MLP and 50 hidden nodes in the six-input MLP, the results from the test data is shown in Table V. They are also compared to the performances from another two score fusion approaches, i.e., taking maximum or linearly combining the three likelihood ratios. Furthermore, to make the comparison with the baseline fair, the target and background scores from the GMM-UBM are used to act as two inputs of a three-layer MLP with 10 nodes in the hidden layer. The two-input MLP is trained with the same training data as that for the three-input and six-input MLP. By taking advantage of the same training data, the SGMM-SBM with the six-input MLP is better than that with the three-input MLP, same as the results from the development data. Its EER achieves 10% relative

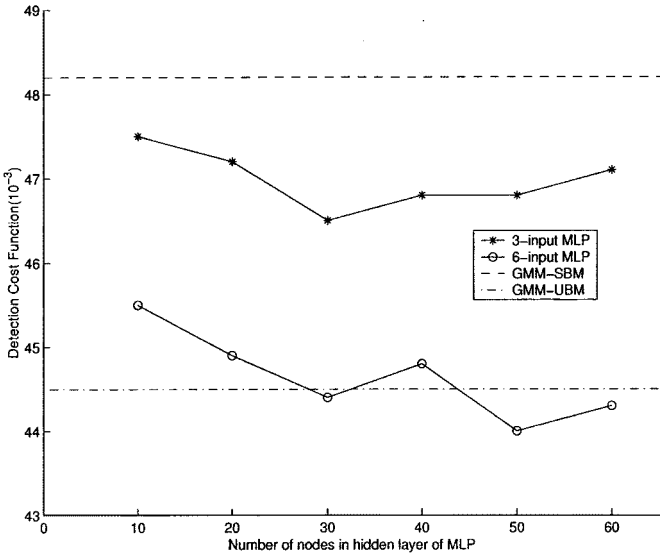

Fig. 6. Results of SGMM-SBM/MLP with development data.

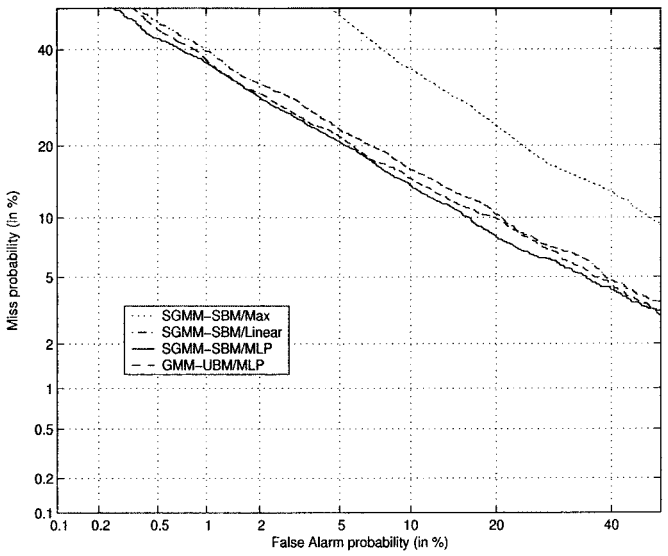

Fig. 7. Comparison of score fusion approaches.

TABLE V
COMBINATION OF MULTIPLE SCORES

System	EER(%)	min DCF(10^{-3})
SGMM-SBM/Max	21.9	83.9
SGMM-SBM/Linear	13.5	49.1
SGMM-SBM/3-input-MLP	13.3	48.2
SGMM-SBM/6-input-MLP	12.1	46.6
GMM-UBM/MLP	12.7	46.7

reduction compared to that of the linear combination and 5% relative reduction compared to that of the GMM-UBM. Also, the DCF is decreased to almost the same value as that of the GMM-UBM.

The DET curves of the different systems are shown in Fig. 7. The SGMM-SBM/MLP represents the score fusion with the six-input MLP.

Since the MLP based score fusion needs to be computed only once for each trial and the computational cost is negligible compared with that from the computation of Gaussian likelihoods, the impact on the computational reduction factor can be ignored. Thus, with neural score combination effected by the neural network, the SGMM-SBM not only achieves computational reduction by a factor of 17 but also reduces the EER by 5% relatively, compared with the GMM-UBM. Both computational efficiency and high verification performance are achieved in this integrated system. Utilizing a hierarchical structure to model the acoustic space and a multilevel MAP adaptation to relate the target SGMM to the SBM makes the fast scoring feasible. The MLP further discriminates the target and impostor scores by nonlinear combination of the multiple scores from the target and background models to achieve high performance.

VI. CONCLUSIONS

An integrated system with structural Gaussian mixture models and a neural network was presented in this paper for efficient text-independent speaker verification. All Gaussian mixture components in the UBM are clustered hierarchically during the construction of a tree-structured SBM. An SGMM is created through multilevel MAP adaptation for each target speaker. In this way the acoustic space is partitioned into multiple regions in different levels of resolution. The computational cost can be reduced significantly by scoring only a subset of the Gaussian components during verification. Multiple background scores and target scores obtained in different layers of the SBM and the SGMM are combined via an MLP. The experiments on the NIST speaker verification data show that a computational reduction of factor 17 can be achieved with a relative reduction of 5% in EER when using neural-network-based score combination. The SGMM-SBM also shows some advantages over the recently proposed hash GMM. In the future other tree construction approaches may be investigated. Also, more research is needed on the combination of scores from multiple layers in order to further improve verification performance. For example, other types of neural networks also could be the alternatives to MLP to take more advantage of the scores from different levels.

We also point out that although SGMM and MLP score combinations were proposed and evaluated for speaker verification, they can also be applied in the text-independent speaker identification to reduce computational cost. An SGMM can be created for each enrolled speakers and a similar MLP can be designed with all scores from different layers of the SGMMs as input. The number of output neurons will be equal to the number of speakers in the system, instead of just being one as used in this paper for speaker verification. Finally, we point out that combining multiple likelihoods from several levels of the hierarchy using neural networks is also applicable to the high-speed speech recognition with tree-structured models.

ACKNOWLEDGMENT

The authors would like to thank the people of Conversational Biometrics group in IBM T. J. Watson Research Center for valuable suggestions at an early stage of this work. They would also

like to thank the unknown reviewers and the associate editor Dr. R. A. Gopinath for their comments and suggestions.

REFERENCES

[1] J. Campbell, "Speaker recognition: a tutorial," *Proc. IEEE*, vol. 85, pp. 1437–1462, Sept. 1997.

[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] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proc. Eurospeech*, 1997.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[5] J. L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.

[6] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1993, pp. 692–695.

[7] K. M. Knill, M. J. F. Gales, and S. J. Young, "Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs," in *Proc. Int. Conf. Spoken Language Processing*, 1996.

[8] D. B. Paul, "An investigation of Gaussian shortlists," in *Proc. Automatic Speech Recognition and Understanding Workshop*, 1999.

[9] T. Watanabe, K. Shinoda, K. Takagi, and K.-I. Iso, "High speed speech recognition using tree-structured probability density function," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1995.

[10] J. Simonin, L. Delphin-Poulat, and G. Damnati, "Gaussian density tree structure in a multi-Gaussian HMM-based speech recognition system," in *Proc. Int. Conf. Spoken Language Processing*, 1998.

[11] T. J. Hanzen and A. K. Halberstadt, "Using aggregation to improve the performance of mixture Gaussian acoustic models," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1998.

[12] M. Padmanabhan, L. R. ahl, and D. Nahamoo, "Partitioning the feature space of a classifier with linear hyperplanes," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 3, pp. 282–288, 1999.

[13] R. Auckenthaler and J. Mason, "Gaussian selection applied to text-independent speaker verification," in *Proc. A Speaker Odyssey—Speaker Recognition Workshop*, 2001.

[14] J. McLaughlin, D. Reynolds, and T. Gleason, "A study of computation speed-ups of the GMM-UBM speaker recognition system," in *Proc. Eurospeech*, 1999.

[15] S. van Vuuren and H. Hermansky, "On the importance of components of the modulation spectrum of speaker verification," in *Proc. Int. Conf. Spoken Language Processing*, 1998.

[16] B. L. Pellom and J. H. L. Hansen, "An efficient scoring algorithm for Gaussian mixture model based speaker identification," *IEEE Signal Processing Lett.*, vol. 5, no. 11, pp. 281–284, 1998.

[17] J. Oglesby and J. S. Mason, "Optimization of neural models for speaker identification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1990, pp. 261–264.

[18] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network—hidden Markov model hybrid," *IEEE Trans. Neural Networks*, vol. 3, no. 2, pp. 252–259, 1992.

[19] H. Bourlard and C. J. Wellekins, "Links between Markov models and multilayer perceptrons," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1167–1178, Dec. 1990.

[20] J. Navrátil, U. V. Chaudhari, and G. N. Ramaswamy, "Speaker verification using target and background dependent linear transforms and multi-system fusion," in *Proc. Eurospeech*, 2001.

[21] L. P. Heck, Y. Konig, M. K. Sonmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," *Speech Commun.*, vol. 31, pp. 181–192, 2000.

[22] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, pp. 1–38, 1977.

[23] K. Shinoda and C. H. Lee, "A structural Bayes approach to speaker adaptation," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 276–287, 2001.

[24] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.

[25] J. C. Junqua, *Robust Speech Recogntion in Embedded Systems and PC Application*. Norwell, MA: Kluwer, 2000.

[26] U. V. Chaudhari, J. Navrátil, S. H. Maes, and R. A. Gopinath, "Transformation enhanced multi-grained modeling for text-independent speaker recognition," in *Proc. Int. Conf. Spoken Language Processing*, 2000.

[27] Q. Lin, E.-E. Jan, C. W. Che, D.-S. Yuk, and J. Flanagan, "Selective use of the speech spectrum and a VQGMM method for speaker identification," in *Proc. Int. Conf. Spoken Language Processing*, 1996.

[28] S. Raudys, *Statistical and Neural Classifiers: An Integrated Approach to Design*. New York: Springer, 2001.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, pp. 318–364.

[30] [Online] Available: http://www.nist.gov/speech/tests/spk/index.htm.

[31] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. A Speaker Odyssey—Speaker Recognition Workshop*, 2001.

[32] B. Xiang, U. V. Chaudhari, J. Navrátil, N. Ramaswamy, and R. A. Gopinath, "Short-time Gaussianization for robust speaker verification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 2002.

[33] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation—overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, pp. 225–254, 2000.

Bing Xiang (M'03) was born in 1973 in China. He received the B.S. degree in radio and electronics and M.E. degree in signal and information processing from Peking University in 1995 and 1998, respectively. In January, 2003, he received the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY.

From 1995 to 1998, he worked on speaker recognition and auditory modeling in National Laboratory on Machine Perception, Peking University. Then he entered Cornell University and worked on speaker recognition and speech recognition in DISCOVER Lab as a Research Assistant. He also worked in the Human Language Technology Department of IBM Thomas J. Watson Research Center as a summer intern in both 2000 and 2001. He was a selected remote member of the SuperSID Group in the 2002 Johns Hopkins CLSP summer workshop in which he worked on speaker verification with high-lelvel information. In January, 2003, he joined the Speech and Language Processing Department of BBN Technologies where he is presently a Senior Staff Consultant-Technology. His research interests include large vocabulary speech recognition, speaker recognition, speech synthesis, keyword spotting, neural networks and statistical pattern recognition.

Toby Berger (S'60–M'66–SM'74–F'78) was born in New York, NY, on September 4, 1940. He received the B.E. degree in electrical engineering from Yale University, New Haven, CT in 1962, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA in 1964 and 1966, respectively.

From 1962 to 1968 he was a Senior Scientist at Raytheon Company, Wayland, MA, specializing in communication theory, information theory, and coherent signal processing. In 1968 he joined the faculty of Cornell University, Ithaca, NY where he is presently the Irwin and Joan Jacobs Professor of Engineering. His research interests include information theory, random fields, communication networks, wireless communications, video compression, voice and signature compression and verification, neuroinformation theory, quantum information theory, and coherent signal processing. He is the author/co-author of Rate Distortion Theory: A Mathematical Basis for Data Compression, Digital Compression for Multimedia: Principles and Standards, and Information Measures for Discrete Random Fields.

Dr. Berger has served as editor-in-chief of the IEEE TRANSACTIONS ON INFORMATION THEORY and as president of the IEEE Information Theory Group. He has been a Fellow of the Guggenheim Foundation, the Japan Society for Promotion of Science, the Ministry of Education of the People's Republic of China and the Fulbright Foundation. In 1982 he received the Frederick E. Terman Award of the American Society for Engineering Education, and in 2002 he received the Shannon Award from the IEEE Information Theory Society, its highest honor. He is a life member of Tau Beta Pi.