On the Use of Different Speech Representations for Speaker Modeling

Ke Chen, *Senior Member, IEEE*

***Abstract*—Numerous speech representations have been reported to be useful in speaker recognition. However, there is much less agreement on which speech representation provides a perfect representation of speaker-specific information conveyed in a speech signal. Unlike previous work, we propose an alternative approach to speaker modeling by the simultaneous use of different speech representations in an optimal way. Inspired by our previous empirical studies, we present a soft competition scheme on different speech representations to exploit different speech representations in encoding speaker-specific information. On the basis of this soft competition scheme, we present a parametric statistical model, *generalized Gaussian mixture model* (GGMM), to characterize a speaker identity based on different speech representations. Moreover, we develop an expectation-maximization algorithm for parameter estimation in the GGMM. The proposed speaker modeling approach has been applied to text-independent speaker recognition and comparative results on the KING speech corpus demonstrate its effectiveness.**

***Index Terms*—Different speech representations, expectation-maximazation (EM) algorithm, generalized Gaussian mixture model (GGMM), KING speech corpus, speaker modeling, speaker-specific information, soft competition, speaker recognition.**

I. Introduction

As ONE OF the most important topics in biometrics, speaker recognition is a process that automatically determines a speaker's identity based on his or her voice. There are widespread applications ranging from confidential data access to audio indexing in multimedia [5], [18], [40]. Although speaker recognition includes miscellaneous tasks, most of studies focus on speaker identification and verification. Speaker identification is to classify an unknown voice token as one of reference speakers, whereas speaker verification is to accept or reject an identity claim. Moreover, a speaker recognition system often works in either of two operating modes: text dependent, where the same text is required for both training and test; and text independent, where arbitrary text is allowed to utter without restriction.

From a pattern recognition perspective, speaker recognition is a difficult problem because speech always changes over time and is affected by numerous factors ranging from environments to speaker's healthy status. Due to the nature of speech, speaker recognition becomes a special pattern recognition problem in that there are large intraspeaker variabilities along with relatively small interspeaker variabilities [43]. In contrast, a speech signal is recognized as a nonstationary stochastic process in general, which implies the necessity of a short-term spectral analysis. Unfortunately, such an analysis leads to a huge number of high-dimensional data or speech frames even for a problem of moderate size so all existing classification techniques cannot cope with speaker recognition without the help of feature extraction techniques [21]. It is well known that an effective feature extraction technique can generate a parsimonious, yet more distinguishing, representation for raw data. As a consequence, it is inevitable for speaker recognition to demand such a low-dimensional representation to encode speaker-specific information conveyed in speech.

For speech signal processing, numerous speech spectral representations have been developed by parameterizing the speech spectrum. Common spectrum representations are linear predictive coefficients and their various transformations (e.g., reflection coefficients, PARCOR coefficients, cepstral coefficients, filter-bank energies, and their cepstral representations). These representations have been widely used in both speech and speaker recognition. The earlier studies uncover that a speech signal contains mainly three types of information (i.e., linguistic information, speaker-specific information, and information on a specific environment) and so does any speech spectral representation extracted from the speech signal [21]. An ideal speaker recognition system should be able to discriminate speakers regardless of linguistic information, whereas a general speech recognition system would rather decode speech to distill only linguistic information without considering speaker identities. Thus, the use of the same spectral representations in both speaker and speech recognition has become an obstacle hindering either of two systems from producing high performance. Many efforts have been made toward better extracting and representing speaker-specific information from speech signals. These endeavors can be roughly classified into two categories: *exploration of a single representation*, where an effective single speech representation is sought to characterize speaker-specific information; and *exploitation of different representations*, where multiple speech representations of speech signals are considered to encode speaker-specific information.

The basic idea underlying exploration of a single representation is finding a speech representation for better encoding speaker-specific information than its common spectral counterparts. Such methods include feature selection, feature transformation, and novel speech representations alternative to a spectral representation. By feature selection, one attempts to discover a subset of features likely carrying speaker-specific information (less associated with linguistic information reciprocally) from a common spectral representation [19],

Manuscript received March 20, 2003; revised September 10, 2003. This paper was recommended by Guest Editor D. Zhang.

The author is with the School of Informatics, The University of Manchester, Manchester M60 1QD, U.K. (e-mail: Ke.Chen@manchester.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2005.848166

[24], [34]. Apart from some text-dependent cases [24], [34], however, most such fail to generate a proper subset due to the lack of an appropriate mathematical expression methodology [5] and numberless varieties of speaker-specific information. In general, generic text-independent speaker feature selection is still an open problem. Instead, more researchers focus on feature transformation where the original spectral representations are somehow manipulated to produce the effect that highlights speaker-specific information and/or suppresses linguistic information, as well as other parts of speech irrelevant to speaker-specific information. Such techniques include adaptive feature weighting [3], [29] and other transformations (e.g., wavelet-based transformation) [20]. Finally, some perceptual factors have also been considered to compensate for spectral representations, which turn out to be helpful for speaker recognition [1], [22], [25], [26].

Exploitation of different representations tends to overcome the difficulty that no existing single representation perfectly expresses speaker-specific information. In this methodology, multiple feature vectors, corresponding to different representations, are always extracted from a speech frame, respectively, based on different feature extraction methods. As a consequence, different speech representations are available, and each can be individually used to replace the original speech data in automatic speaker recognition. Because different representations tend to encode different facets of speaker-specific information, the joint use of different representations more likely captures a diversity of speaker-specific information intertwined with linguistic information in speech. Obviously, such a methodology always attempts to take advantage of the latest progress made by exploration of a single representation. So far, two approaches have been developed for use of different representations in speaker recognition. One is simply lumping different representations together to constitute a composite representation [2], [7], [28], [33]. To some extent, such an approach improves the performance of speaker recognition but leads to a higher dimensional feature vector of redundancy, given that any representation can individually represent the original speech data. Thus, the use of a composite representation unavoidably suffers from the curse of dimensionality. In addition, different representations get involved in different measure criteria, and so it is a nontrivial issue on how to normalize different representations to form a composite representation. Unlike the use of a composite representation, the other is combining speaker models created separately on different representations [10], [35], [41], [42], which results in a two-stage speaker modeling process (i.e., creation of speaker models based on individual representations, respectively, and then a combination of speaker models on different representations). Apparently, such a methodology is suboptimal because different facets reflected by different representations are indirectly considered only at the decision-making stage rather than the speaker modeling stage.

Unlike all the approaches mentioned previously, we propose an alternative approach to speaker modeling by the simultaneous use of different representations in an optimal way. Inspired by our previous empirical studies, we present a soft competition scheme on different representations toward better encoding speaker-specific information. In the soft competition scheme, different representations corresponding to a speech frame compete with each other for the right of representing speaker-specific information. Unlike the winner-take-all scheme, our soft competition scheme allows different feature vectors to work together in an optimal way, where winners play a more important role than losers. On the basis of this soft competition scheme, we propose a parametric statistical model, generalized Gaussian mixture model (GGMM), to characterize a speaker identity based on different representations. For parameter estimation in the GGMM, we develop an expectation-maximization (EM) algorithm based on the general EM framework [16]. The proposed speaker modeling approach has been applied to text-independent speaker recognition, and comparative results on the KING database, a benchmark speech corpus designed especially for text-independent speaker recognition, demonstrate its effectiveness.

The remainder of this article proceeds as follows. We first describe the motivation and the basic idea of soft competition on different speech representation for speaker modeling. After presenting the GGMM to carry out the soft competition scheme, we develop an EM learning algorithm for parameter estimation. We then employ the GGMM as a speaker model for speaker recognition and report simulation results on the KING speech corpus. We further discuss some issues relevant to the proposed approach.

II. Methodology

In this section, we first describe an inspiration coming from our earlier empirical studies, which draws on our basic idea of soft competition on different speech representations, and then propose a probabilistic model for the optimal use of different speech representations.

A. Motivation

In our earlier work [8]–[14], [44], we empirically investigated how the performance of speaker recognition is influenced by different speech representations through the use of modular neural networks, the mixture-of-experts (ME) model, [23] and its extensions. Among those investigations, there is always a common outcome (i.e., for a classifier separately trained on different representations, its performance on a whole testing set is often *inconsistent* with those on its subsets when we compare the influence of different representations on speaker recognition based on an identical classifier).

Here, we present the finding of such inconsistency by means of a text-independent speaker identification experiment. We used a speech corpus in Mandarin Chinese, where there were 20 male speakers and all the utterances were recorded in three sessions spaced apart by at least 1 month. After a conventional short-term acoustic analysis, we extracted three different representations or feature vectors from each speech frame [21] (i.e., linear predictive coefficients (LPC), perceptual linear prediction (PLP), and Mel-scaled cepstral coefficients (MCC) vectors, respectively). The utterances recorded in session 1 were employed as a training set. An ME classifier of the identical architecture was trained on different representations, respectively. The utterances in the corpus were used as testing data. Testing results were observed in two ways: the error rate or misidentification rate on the whole testing set and those on subsets formed by randomly partitioning the testing set into seven mutually exclusive subsets of nearly equal length of utterances.

As illustrated in Fig. 1, each plot indicates the performance of this ME classifier on different representations as the aforementioned testing set or one of its subsets is used for test. Fig. 1(a) depicts the overall performance on the whole testing set, where the performance of MCC is the best among three representations and the performance of PLP is better than that of LPC. By comparison, the performance on different representations can be ranked in light of error rates, denoted by e_X, where X is one of three representations as follows: $e_{\mathrm{MCC}} < e_{\mathrm{PLP}} < e_{\mathrm{LPC}}$. Fig. 1(b)–(h) illustrate the testing results on seven mutually exclusive subsets of the whole testing set, respectively. In contrast, the same ranking as shown in Fig. 1(a) occurs only once in Fig. 1(b), whereas testing results on other six subsets are inconsistent with that ranking as illustrated in Fig. 1(c)–(h). The further observation indicates that any of three representations yields the lowest error rate on some certain subsets, but none of them leads to the lowest error rate on all the subsets. For instance, the performance of LPC is superior to others on three subsets as shown in Fig. 1(d), (e), and (h), although its overall performance is inferior to that of others. Similarly, the performance of MCC becomes the poorest on a subset as depicted in Fig. 1(e), although its overall performance is ranked as the top shown in Fig. 1(a). As a consequence, the performance of different representations dynamically varies on different subsets, and none of them overwhelmingly dominates the performance of speaker identification on all subsets. We have further investigated the phenomenon by the extensive use of subsets of different sizes and alternative speech representations. All the experimental results consistently support our findings, as exemplified above.

Consistent with earlier studies, the previous empirical studies lend support to the argument that for speaker modeling none of the existing speech representations is perfect for encoding speaker-specific information. Our finding further suggests that the various speaker-specific information conveyed in different parts of speech should be dynamically specified by different kinds of speech representations, which provides an alternative insight into exploitation of different representations. Inspired by the aforementioned empirical studies, we propose a soft competition scheme for the joint use of different representations for speaker modeling in an optimal way.

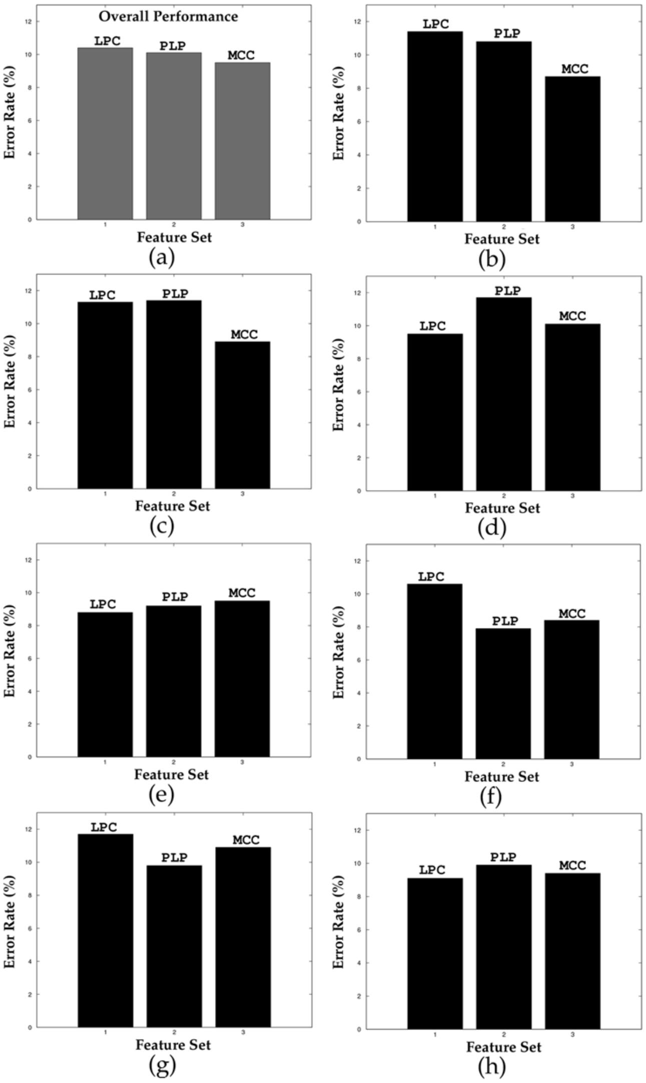

Fig. 1. Testing results of text-independent speaker identification as different speech representations are used for training the same classifier, respectively. (a) Results on the whole testing set. (b) Results on subset 1, $e_{\mathrm{MCC}} < e_{\mathrm{PLP}} < e_{\mathrm{LPC}}$. (c) Results on subset 2, $e_{\mathrm{MCC}} < e_{\mathrm{PLP}} < e_{\mathrm{LPC}}$. (d) Results on subset 3, $e_{\mathrm{LPC}} < e_{\mathrm{MCC}} < e_{\mathrm{PLP}}$. (e) Results on subset 4, $e_{\mathrm{LPC}} < e_{\mathrm{PLP}} < e_{\mathrm{MCC}}$. (f) Results on subset 5, $e_{\mathrm{PLP}} < e_{\mathrm{MCC}} < e_{\mathrm{LPC}}$. (g) Results on subset 6, $e_{\mathrm{PLP}} < e_{\mathrm{MCC}} < e_{\mathrm{LPC}}$. (h) Results on subset 7, $e_{\mathrm{LPC}} < e_{\mathrm{MCC}} < e_{\mathrm{PLP}}$.

B. Soft Competition on Different Speech Representations

For representing a speech data set, we assume there are $K(K > 1)$ different feature extraction methods so K different speech representations can be extracted from this speech data set at a speech frame level after short-term acoustic processing. To facilitate the presentation, we let $\mathcal{D} = \{D^{(t)}\}_{t=1}^{T}$ denote a data set containing T speech frames. Thus, the notation $\mathcal{X} = \{\mathcal{X}_1(\mathcal{D}); \ldots ; \mathcal{X}_K(\mathcal{D})\}$ stands for a collection of K different representations extracted from the data set $\mathcal{D}$, where $\mathcal{X}_k = \{\boldsymbol{x}_k(D^{(t)})\}_{t=1}^{T}(k = 1, \ldots, K)$ and $\boldsymbol{x}_k(D^{(t)})$ is the speech feature vector, corresponding to representation k, obtained from the speech frame $D^{(t)}$ by applying the kth feature extraction method. For a speech frame $D^{(t)}$, therefore, we can always obtain a K feature vectors[1] collectively denoted as $\{\boldsymbol{x}_1^{(t)}(D^{(t)}); \ldots ; \boldsymbol{x}_K^{(t)}(D^{(t)})\}$. To simplify our presentation, we hereinafter drop the obvious data terms, $\mathcal{D}$ and $D^{(t)}$, from the collective notation of different speech representations.

It is well known that a speech representation mainly contains linguistic and speaker-specific information. Our empirical studies mentioned previously indicate that different speech representations may convey various amount of speaker-specific information in a certain circumstance. For speaker modeling, an ideal speech representation is the one that contains the maximum amount of speaker-specific information or the minimum amount of linguistic information reciprocally, which we refer to

[1] These feature vectors may have different dimensions.

as an *optimal feature vector* in this article. For denoting such an optimal feature vector in context of different representations, we introduce a set of binary indicator variables, $z_1, \ldots, z_K$. The indicator, $z_k (k = 1, \ldots, K)$, corresponds to the feature vector $\boldsymbol{x}_k$ and is defined as $z_k = 1$ if $\boldsymbol{x}_k$ is the optimal feature vector of a speech frame D; otherwise, $z_k = 0$. According to our definition on the optimal feature vector, $\sum_{k=1}^{K} z_k = 1$ is always guaranteed.

For a speech frame, its $K(K > 1)$ feature vectors or representations are always mutually in the position of rivals to compete for being its optimal feature vector in context of speaker modeling. If we adopt a winner-take-all scheme for competition, the optimal feature vector would be solely used to represent the speech frame and, as a result, others should be abandoned. Given that an indicator of the optimal feature vector is a random variable, the use of the winner-take-all scheme leads to a probabilistic relationship between the speech frame and its feature vectors or representations via indicators of optimality as follows:

$$P(\boldsymbol{x}_k) = P(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\} \,|\, z_k = 1), \quad k = 1, \ldots, K. \tag{1}$$

For speaker modeling, (1) implies that a speech frame is represented only by its optimal feature vector in a stochastic sense. Apparently, the aforementioned winner-take-all competition idea would always work if such indicators are known or can be evaluated. In practice, however, the indicators remain unknown unless there is an explicit measure of speaker-specific information. Unfortunately, to our knowledge, such a measure has not been discovered so far, and it is more likely that there is no simple measure in reality for such a purpose.

Unlike the winner-take-all scheme, the principle of *soft competition* provides an alternative perspective for competition. Instead of eliminating losers, the soft competition scheme allows all the rivals to work together for a task. Nevertheless, a winner would play a more important role than a loser. In other words, the nature of a soft competition scheme leads to a niche for maximum synergy from all the rivals. As a generic principle, soft competition has been extensively applied to data clustering (e.g., [17], [31], which yields better performance). Motivated by the principle of soft competition and its applications in data clustering, we propose a mixture model for the optimal use of all the different speech representations simultaneously

$$\begin{aligned} &P(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}) \\ &\quad = \sum_{k=1}^{K} P(z_k = 1 | \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}) P(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\} | z_k = 1) \\ &\quad = \sum_{k=1}^{K} P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}) P(\boldsymbol{x}_k). \end{aligned} \tag{2}$$

Here, the last step is obtained by applying (1), and the notation $P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$ is adopted to highlight the fact that a mixing proportion is an input-dependent probability given the ensemble of different representations. Thus, (2) provides a soft competition scheme for different speech representations. For each speech frame, its feature vectors or different representations, extracted by different feature extraction methods, compete each other for the right to represent the speaker-specific information conveyed in the speech frame. The competition leads to a set of credits for different representations, and all the representations are used by a weighted combination. When (2) is employed as a statistical speaker model, $P(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$ would yield the probability that a speech frame encoded by its K different representations belongs to the speaker. Accordingly, $P(\boldsymbol{x}_k)$ is the probability that the speech frame encoded by only the individual representation $\boldsymbol{x}_k$ belongs to the speaker. $P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$ tends to dynamically generate credits for the optimal combination of K statistical speaker models individually working on K different speech representations. To build such a speaker model, we specify a parametric finite mixture model, where its parameters are estimated by learning from data. Here, we emphasize that neither $P(\boldsymbol{x}_k)$ nor $P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$ in (2) is created separately for characterizing a speaker identity, and all the elements in (2) must be treated as a whole for speaker modeling.

III. Model Description

In this section, we present a parametric model, GGMM, to carry out our soft competition scheme for speaker modeling. Moreover, we develop a learning algorithm for this parametric model based on the EM framework [16]. The proposed statistical model will be applied as a statistical speaker model by exploiting different speech representations.

A. Generalized Gaussian Mixture Model

As specified in (2), our soft competition scheme specifies a finite mixture model consisting of component densities based on individual representations and input-dependent mixing proportions given the ensemble of different representations.

As a speaker modeling technique, Gaussian mixture model (GMM) has been applied for speaker recognition and has turned out to be an effective statistic model for encoding the speaker-specific information conveyed in a speech representation [36], [37], [39]. As pointed out by Reynolds and Rose [39], each unimodal component of a Gaussian mixture density characterizes an acoustic class underlying a phonetic event (e.g., vowels, nasals, or fricatives), and therefore can specify a speaker-dependent vocal tract configuration, whereas the multimodal GMM is modeling a set of speaker-dependent acoustic classes. Moreover, a Gaussian mixture density also provides a smooth approximation to the underlying long-term sample distribution of observations achieved from utterances of a specific speaker [39]. Inspired by previous investigations [36], [37], [39], we adopt a GMM as a component density based on an individual representation in our parametric model. For a representation $\mathcal{X}_k$, the component density function is

$$p(\boldsymbol{x}_k \,|\, \Phi_k) = \sum_{j=1}^{N_k} \beta_{kj} G(\boldsymbol{x}_k, \boldsymbol{\mu}_{kj}, \Sigma_{kj}) \tag{3}$$

where $G(\boldsymbol{x}_k, \boldsymbol{\mu}_{kj}, \Sigma_{kj})$ is a Gaussian density function defined by

$$G(\boldsymbol{x}_k, \boldsymbol{\mu}_{kj}, \Sigma_{kj}) = \frac{1}{(2\pi)^{\frac{m_k}{2}} |\Sigma_{kj}|^{(1/2)}} \times \exp\left[-\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu}_{kj})^T \Sigma_{kj}^{-1} (\boldsymbol{x}_k - \boldsymbol{\mu}_{kj})\right].$$

Here, $\boldsymbol{x}_k$ is a m_k-dimensional feature vector in $\mathcal{X}_k, \boldsymbol{\mu}_{kj}$, and $\Sigma_{kj} (j = 1, 2, \ldots, N_k)$ are mean vectors and covariance matrices of Gaussian densities in (3), and β_{kj} $(j = 1, 2, \ldots, N_k)$ are input-independent mixing proportions, where all β_{kj} satisfy $\sum_{j=1}^{N_k} \beta_{kj} = 1$ and $\beta_{kj} \geq 0$. Given N_k Gaussian components in (3), all the parameters are denoted by

$$\Phi_k = \{\beta_{kj}, \boldsymbol{\mu}_{kj}, \Sigma_{kj}\} \quad j = 1, 2, \ldots, N_k.$$

The mixing proportions in our finite mixture model in (2) follows the multinomial distribution [30]. Now, we encounter a distribution specified on the ensemble of different representations rather than the original data set. Therefore, we need to develop a suitable model for describing such a distribution. Due to the nature of different representations, each speech representation can be independently used, as done in most speaker recognition systems. When different representations are considered together, we specify $P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$ by a linear combination of multinomial distributions on different representations:

$$P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}) = \sum_{i=1}^{K} \alpha_i P(z_k = 1 \,|\, \boldsymbol{x}_i), \quad k = 1, \ldots, K. \tag{4}$$

Equation (4) suggests another soft competition scheme: each representation can be individually used to specify the multinomial distribution, but they may not play equal roles while they are considered. That is, α_i reflects the importance of the ith representation in the overall multinomial distribution by considering all the different representations, where all α_i are subject to $\sum_{i=1}^{K} \alpha_i = 1$ and $\alpha_i \geq 0$.

For parameterizing the model in (4), we specify a parametric model for the multinomial distribution $P(z_k = 1 \,|\, \boldsymbol{x}_i) (i = 1, \ldots, K)$ by a generalized linear multinomial logit model [15], [32]

$$p(z_k = 1 \,|\, \Omega_i, \boldsymbol{x}_i) = b(\boldsymbol{x}_i, \boldsymbol{\omega}_{ik}) = \frac{\exp\left(\boldsymbol{\omega}_{ik}^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\boldsymbol{\omega}_{ik}^T \boldsymbol{x}_i\right)} \quad k = 1, \ldots, K \tag{5}$$

where $\Omega_i = \{\boldsymbol{\omega}_{i1}, \ldots, \boldsymbol{\omega}_{iK}\}$ is the set of all parameters in the generalized linear model. Inserting (5) into (4), we obtain a parametric model, $\lambda_k(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}, \boldsymbol{\Omega})$, for $P(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\})$:

$$\begin{aligned} p(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}, \boldsymbol{\Omega}) &= \lambda_k(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}, \boldsymbol{\Omega}) \\ &= \sum_{i=1}^{K} \alpha_i b(\boldsymbol{x}_i, \boldsymbol{\omega}_{ik}) \quad k = 1, \ldots, K \end{aligned} \tag{6}$$

where $\boldsymbol{\Omega}$ is the set of parameters $\{\Omega_1, \ldots, \Omega_K\}$ and $\{\alpha_1, \ldots, \alpha_K\}$ in the parametric model for generating mixing proportions.

Assembling (3) and (6) based on (2), we immediately obtain a parametric form of our finite mixture model for soft competition on different speech representations

$$\begin{aligned} &p(\{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\} \,|\, \boldsymbol{\Phi}) \\ &\quad = \sum_{k=1}^{K} p(z_k = 1 \,|\, \{\boldsymbol{x}_1; \ldots; \boldsymbol{x}_K\}, \boldsymbol{\Omega}) p(\boldsymbol{x}_k \,|\, \Phi_k) \\ &\quad = \sum_{k=1}^{K} \left[\sum_{i=1}^{K} \alpha_i b(\boldsymbol{x}_i, \boldsymbol{\omega}_{ik})\right] \left[\sum_{j=1}^{N_k} \beta_{kj} G(\boldsymbol{x}_k, \boldsymbol{\mu}_{kj}, \Sigma_{kj})\right] \\ &\quad = \sum_{k=1}^{K} \sum_{j=1}^{N_k} \sum_{i=1}^{K} \alpha_i \beta_{kj} b(\boldsymbol{x}_i, \boldsymbol{\omega}_{ik}) G(\boldsymbol{x}_k, \boldsymbol{\mu}_{kj}, \Sigma_{kj}). \end{aligned} \tag{7}$$

In (7), all the parameters are collectively expressed by the notation

$$\boldsymbol{\Phi} = \{\Phi_k, \boldsymbol{\Omega}\} \quad k = 1, \ldots, K.$$

Apparently, our statistical model includes several GMMs corresponding to different representations; hence, we refer to (7) as a GGMM. As applied in speaker modeling, the GGMM characterizes a speaker identity based on different speech representations through the use of parameters, $\boldsymbol{\Phi}$.

To better understand the proposed GGMM model, we illustrate the schematic structure of our GGMM in Fig. 2. As depicted in Fig. 2, K GMMs, where a GMM receiving the input $\boldsymbol{x}_k$ contains N_k component Gaussian densities, are employed to model a speaker's characteristics based on K different speech representations, whereas mixing proportions are produced by a linear combination of K proportion generators working on different speech representations. As a consequence, the overall output of the GGMM is the convex weighted combination of outputs of K GMMs. Note that the mixing proportions in the GGMM are input dependent, which significantly distinguishes itself from a classical GMM of input-independent mixing proportions [30].

B. Expectation-Maximization Learning Algorithm

As pointed out in Section III-A, a speaker identity can be characterized by a GGMM via its parametric model $\boldsymbol{\Phi}$ in the context of different speech representations. Therefore, the creation of a speaker model is a parameter estimation or learning process from speaker's utterances. For a finite mixture model, there are numerous methodologies for parameter estimation [30]. So far, the most popular and well-established methodology is the maximum likelihood estimation based on the EM framework [16]. In this section, we develop an EM learning algorithm to fit our GGMM structure.

Suppose a training set corresponding to a specific speaker is $\mathcal{D} = \{D^{(t)}\}_{t=1}^{T}$ and $K (K > 1)$ different feature extraction methods are available. For the training set, the application of K feature extraction methods leads to K different speech representations collectively denoted by $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$ where $\mathcal{X}_k = \{\boldsymbol{x}_k^{(t)}\}_{t=1}^{T}$ is a collection of all the feature vectors of the

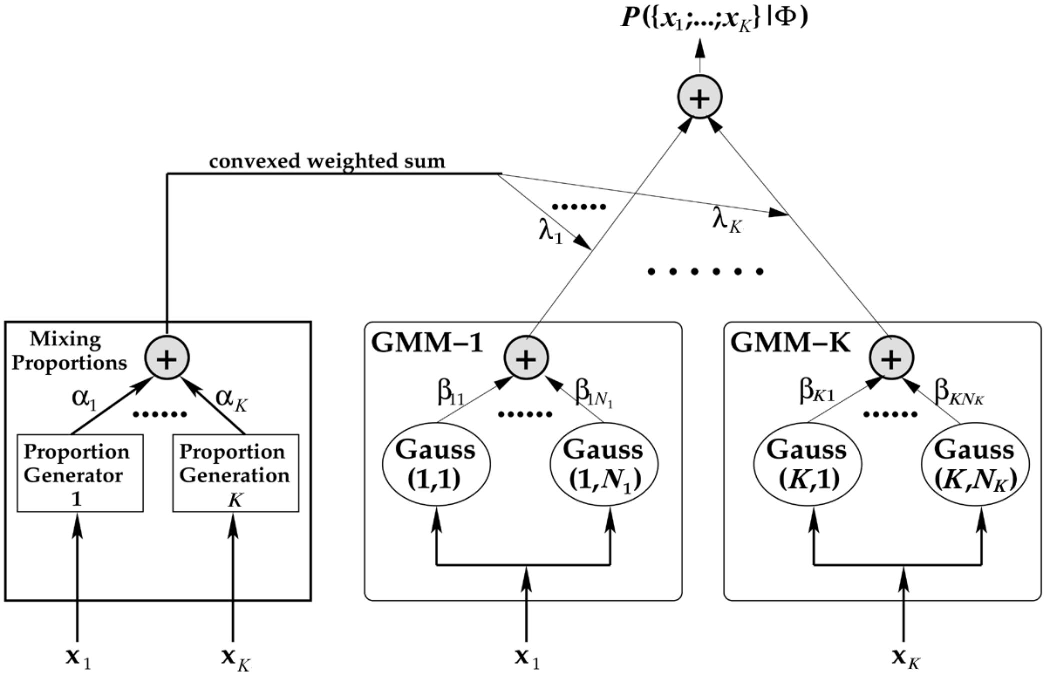

Fig. 2. The schematic diagram of our generalized Gaussian mixture model.

same dimensionality corresponding to the kth speech representation. We refer to all speech representations in $\mathcal{X}$ as *observable data* hereafter because all of them can be obtained by a collection process in reality.

The basic idea underlying the EM framework is introducing a set of *missing data* (unobservable data) to the original likelihood function to simplify its optimization. In the context of our GGMM, we introduce a set of missing data $\mathcal{Z}$, corresponding to observable data, collectively denoted by

$$\mathcal{Z}=\left\{\left\{zz_i^{(t)}\right\}_{i=1}^{K},\left\{\left\{z_{j\,|\,k}^{(t)}\right\}_{j=1}^{N_k}\right\}_{k=1}^{K},\left\{z_k^{(t)}\right\}_{k=1}^{K}\right\}_{t=1}^{T}. \tag{8}$$

$zz_i^{(t)}$ is a binary indicator associated with mixing proportion generators (see Fig. 2) defined as (9) shown at the bottom of the page. According to the definition in (9), $\sum_{i=1}^{K} zz_i^{(t)}=1$. Similarly, $z_{j\,|\,k}^{(t)}$ is another binary indicator for Gaussian component densities in the kth GMM receiving $\boldsymbol{x}_k$ (see Fig. 2) shown in (10) at the bottom of the page. Here $\sum_{j=1}^{N_k} z_{j\,|\,k}^{(t)}=1$ is ensured in (10). With respect to missing data $z_k^{(t)}$, it is actually the same as defined in Section II-B for indicating whether the representation or feature vector $\boldsymbol{x}_k^{(t)}$ in $\mathcal{X}_k$ is the optimal feature vector of a speech frame $D^{(t)}$. Now, we explicitly write $z_k^{(t)}$ by (11) shown at the bottom of the page.

Once we define missing indicator variables, we have a complete data set consisting of observable and missing data by $\{\boldsymbol{X}^{(t)},\boldsymbol{Z}^{(t)}\}_{t=1}^{T}$ where $\boldsymbol{X}^{(t)}=\{\boldsymbol{x}_1^{(t)};\ldots;\boldsymbol{x}_K^{(t)}\}$ and $\boldsymbol{Z}^{(t)}=\{\{zz_i^{(t)}\}_{i=1}^{K},\{\{z_{j\,|\,k}^{(t)}\}_{k=1}^{K}\}_{j=1}^{N_k},\{z_k^{(t)}\}\}$. Thus, we can specify a probabilistic model that relates the missing data to the observable data. In light of $\boldsymbol{Z}^{(t)}$, this probabilistic model is written based on $p(\boldsymbol{X}^{(t)}\,|\,\boldsymbol{\Phi})$ in (7) as follows:

$$\begin{aligned}
&p\left(\boldsymbol{Z}^{(t)}\,\middle|\,\boldsymbol{X}^{(t)},\boldsymbol{\Phi}\right)\\
&\quad=p\left(zz_i^{(t)}=1,z_k^{(t)}=1\,\middle|\,\boldsymbol{X}^{(t)},\boldsymbol{\Omega}\right)\\
&\qquad\times p\left(z_{j\,|\,k}^{(t)}=1,\,z_k^{(t)}=1\,\middle|\,\boldsymbol{x}_k^{(t)},\Phi_k\right)\\
&\quad=\left[\alpha_i b\left(\boldsymbol{x}_i^{(t)},\boldsymbol{\omega}_{ik}\right)\right]\left[\beta_{kj}G\left(\boldsymbol{x}_k^{(t)},\boldsymbol{\mu}_{kj},\Sigma_{kj}\right)\right]
\end{aligned}$$

$$zz_i^{(t)}=\begin{cases}1, & \text{if } \boldsymbol{x}_i^{(t)} \text{ in } \mathcal{X}_i \text{ is the most suitable for generating mixing proportions}\\ 0, & \text{otherwise.}\end{cases} \tag{9}$$

$$z_{j\,|\,k}^{(t)}=\begin{cases}1, & \text{if the } j\text{th Gaussian density is the best speaker model for } D^{(t)};\\ 0, & \text{otherwise.}\end{cases} \tag{10}$$

$$z_k^{(t)}=\begin{cases}1, & \text{if the } \boldsymbol{x}_k^{(t)} \text{ in } \mathcal{X}_k \text{ is the optimal feature vector of } D^{(t)};\\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

$$= \prod_{k=1}^{K}\prod_{j=1}^{N_k}\prod_{i=1}^{K}\left\{\alpha_i\beta_{kj}b\left(\boldsymbol{x}_i^{(t)},\boldsymbol{\omega}_{ik}\right)\right.$$
$$\left.\times\, G\left(\boldsymbol{x}_k^{(t)},\boldsymbol{\mu}_{kj},\Sigma_{kj}\right)\right\}^{zz_i^{(t)}z_{j\,|\,k}^{(t)}z_k^{(t)}}. \qquad (12)$$

In (12), we use the fact that the indicator variables tell which elements in the GGMM are in operation. Given the complete data $\{\boldsymbol{X}^{(t)},\boldsymbol{Z}^{(t)}\}_{t=1}^{T}$, the probabilistic model in (12) yields a complete log-likelihood function by taking the logarithm

$$\begin{aligned} L_C(\boldsymbol{\Phi};\{\mathcal{X},\mathcal{Z}\}) &= \log\prod_{t=1}^{T}p\left(\boldsymbol{Z}^{(t)}\,\middle|\,\boldsymbol{X}^{(t)},\boldsymbol{\Phi}\right)\\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{i=1}^{K}zz_i^{(t)}z_{j\,|\,k}^{(t)}z_k^{(t)}\\ &\quad\times\left[\log\alpha_i+\log\beta_{kj}+\log b\left(\boldsymbol{x}_i^{(t)}\boldsymbol{\omega}_{ik}\right)\right.\\ &\quad\left.+\log G\left(\boldsymbol{x}_k^{(t)},\boldsymbol{\mu}_{kj},\Sigma_{kj}\right)\right] \end{aligned} \qquad (13)$$

which allows us to consider the optimization problem in a simpler way.

Based on the complete log-likelihood function, our EM algorithm is a recursive learning process where each iteration consists of two steps (i.e., an E step and an M step). In general, the E step tends to decompose the complicated maximum likelihood problem into a set of simpler subproblems, whereas the M step works for solving those subproblems on the basis of simplification by the E step. Based on modified parameters, a next iteration of EM learning takes place, which results in a loop of the E step and the M step.

In the E step, we need to estimate the expectation values of random missing indicator variables, $E[L_C(\boldsymbol{\Phi};\{\mathcal{X},\mathcal{Z}\})\,|\,\mathcal{X}]$, based on the observable data $\mathcal{X}$ and the current parameter values of the GGMM, $\boldsymbol{\Phi}^{(s)}$. Due to the nature of binary indicator variables, their expectation values are actually the posterior probabilities given $\mathcal{X}$ and $\boldsymbol{\Phi}^{(s)}$ as the value of indicators equals one. In the context of the GGMM, the following posterior probabilities, denoted by $h_i^{(t)}(\boldsymbol{\Phi}^{(s)})$, $h_{ik}^{(t)}(\boldsymbol{\Phi}^{(s)})$, and $h_{kj}^{(t)}(\boldsymbol{\Phi}^{(s)})$, are estimated in the E step

$$\begin{aligned} h_i^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) &= E\left[zz_i^{(t)}\,\middle|\,\mathcal{X},\boldsymbol{\Phi}^{(s)}\right]\\ &= P\left(zz_i^{(t)}=1\,\middle|\,\boldsymbol{X}^{(t)}\,,\boldsymbol{\Phi}^{(s)}\right) \end{aligned} \qquad (14a)$$

$$\begin{aligned} h_{ik}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) &= E\left[zz_i^{(t)},z_k^{(t)}\,\middle|\,\mathcal{X},\boldsymbol{\Phi}^{(s)}\right]\\ &= P\left(zz_i^{(t)}=1,z_k^{(t)}=1\,\middle|\,\boldsymbol{X}^{(t)}\,,\boldsymbol{\Phi}^{(s)}\right) \end{aligned} \qquad (14b)$$

$$\begin{aligned} h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) &= E\left[z_{j\,|\,k}^{(t)},z_k^{(t)}\,\middle|\,\mathcal{X},\boldsymbol{\Phi}^{(s)}\right]\\ &= P\left(z_{j\,|\,k}^{(t)}=1,z_k^{(t)}=1\,\middle|\,\boldsymbol{X}^{(t)}\,,\boldsymbol{\Phi}^{(s)}\right). \end{aligned} \qquad (14c)$$

Formulae for estimating the above posterior probabilities are derived in the Appendix.

In the M step, we require maximizing $E[L_C(\boldsymbol{\Phi};\{\mathcal{X},\mathcal{Z}\})\,|\,\mathcal{X}]$ for all the parameters in $\boldsymbol{\Phi}$. By examining (13), we see that parameters in GMMs influence $E[L_C(\boldsymbol{\Phi};\{\mathcal{X},\mathcal{Z}\})\,|\,\mathcal{X}]$ by only terms $h_{kj}^{(t)}(\boldsymbol{\Phi}^{(s)})\log G(\boldsymbol{x}_k^{(t)},\boldsymbol{\mu}_{kj},\Sigma_{kj})$ and $h_{kj}^{(t)}(\boldsymbol{\Phi}^{(s)})\log\beta_{kj}$, whereas the parameters on mixing proportions are influencing $E[L_C(\boldsymbol{\Phi};\{\mathcal{X},\mathcal{Z}\})\,|\,\mathcal{X}]$ through only terms $h_i^{(t)}(\boldsymbol{\Phi}^{(s)})\log\alpha_i$ and $h_{ik}^{(t)}(\boldsymbol{\Phi}^{(s)})\log b(\boldsymbol{x}_i^{(t)},\boldsymbol{\omega}_{ik}^{(t)})$. Thus, the M step decomposes the original optimization problem into the following separate maximization subproblems.

For parameters in the GMM receiving the input $\boldsymbol{x}_k^{(t)}$, two optimization subproblems are

$$\Theta_{kj}^{(s+1)} = \arg\max_{\Theta_{kj}}\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\log G\left(\boldsymbol{x}_k^{(t)},\boldsymbol{\mu}_{kj},\Sigma_{kj}\right) \qquad (15)$$

where $\Theta_{kj}=(\boldsymbol{\mu}_{kj},\Sigma_{kj})$ is a collective notation of parameters of the jth Gaussian density in the GMM receiving $\boldsymbol{x}_k^{(t)}$, and

$$\beta_{kj}^{(s+1)} = \arg\max_{\beta_{kj}}\sum_{t=1}^{T}\sum_{j=1}^{N_k}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\log\beta_{kj}$$
$$\text{s.t. }\sum_{j=1}^{N_k}\beta_{kj}=1,\quad\beta_{kj}\geq 0. \quad (16)$$

Similarly, two optimization subproblems with respect to mixing proportions are

$$\Omega_i^{(s+1)} = \arg\max_{\Omega_i}\sum_{t=1}^{T}\sum_{k=1}^{K}h_{ik}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\log b\left(\boldsymbol{x}_i^{(t)},\boldsymbol{\omega}_{ik}\right) \ (17)$$

and

$$\alpha_i^{(s+1)} = \arg\max_{\alpha_i}\sum_{t=1}^{T}\sum_{i=1}^{K}h_i^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\log\alpha_i$$
$$\text{s.t. }\sum_{i=1}^{K}\alpha_i=1,\quad\alpha_i\geq 0. \quad (18)$$

The parameter estimation in a GMM has been well studied [30], and analytic solutions to optimization subproblems in (15) and (16) are available as follows:

$$\boldsymbol{\mu}_{kj}^{(s+1)} = \frac{1}{\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)}\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\boldsymbol{x}_k^{(t)} \qquad (19)$$

$$\begin{aligned} \Sigma_{kj}^{(s+1)} &= \frac{1}{\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)}\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\\ &\quad\times\left[\boldsymbol{x}_k^{(t)}-\boldsymbol{\mu}_{kj}^{(s+1)}\right]\left[\boldsymbol{x}_k^{(t)}-\boldsymbol{\mu}_{kj}^{(s+1)}\right]^T. \end{aligned} \qquad (20)$$

If the covariance matrix Σ_{kj} is diagonal where diagonal elements are $\sigma_{kj}^{(s+1)}(1),\ldots,\sigma_{kj}^{(s+1)}(m_k)$, the analytic solution in (20) is simplified by

$$\begin{aligned} \left[\boldsymbol{\sigma}_{kj}^{(s+1)}(m)\right]^2 &= \frac{1}{\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)}\sum_{t=1}^{T}h_{kj}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right)\\ &\quad\times\left[\left[\boldsymbol{x}_k^{(t)}(m)\right]-\left[\boldsymbol{\mu}_{kj}^{(s+1)}(m)\right]\right]^2 \end{aligned} \qquad (21)$$

where $\boldsymbol{\sigma}_{kj}^{(s+1)}(m), \boldsymbol{x}_k^{(t)}(m)$, and $\boldsymbol{\mu}_{kj}^{(s+1)}(m)$ are the mth element of the vectors, $\boldsymbol{\sigma}_{kj}^{(s+1)}, \boldsymbol{x}_k^{(t)}$, and $\boldsymbol{\mu}_{kj}^{(s+1)}$, respectively. As suggested by Reynolds and Rose [39], we use only such a diagonal covariance matrix in our simulations. Finally, the analytic solution to the optimization subproblem in (16) is also available by

$$\beta_{kj}^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_{kj}^{(t)} \left(\boldsymbol{\Phi}^{(s)}\right). \tag{22}$$

For mixing proportions, the optimization subproblem in (17) is analytically insolvable, whereas the analytic solution to the optimization subproblem in (18) is

$$\alpha_i^{(s+1)} = \frac{1}{T} \sum_{t=1}^{T} h_i^{(t)} \left(\boldsymbol{\Phi}^{(s)}\right). \tag{23}$$

Due to the nature of a multinomial logit model underlying each mixing proportion generator, the optimization subproblem in (17) is an iteratively reweighted least squares (IRLS) problem. In our earlier work [15], we developed a Newton–Raphason algorithm for solving the IRLS problem, and the iterative solution is

$$\Omega_i^{(s,n)} = \Omega_i^{(s,n-1)} - \eta \boldsymbol{H}^{-1}\left(\Omega_i^{(s,n-1)}, \mathcal{X}\right) \boldsymbol{J}\left(\Omega_i^{(s,n-1)}, \mathcal{X}\right) \tag{24}$$

where $\eta (0 < \eta \leq 1)$ is a learning rate. $\boldsymbol{H}(\Omega_i^{(s,n-1)}, \mathcal{X})$ is the Hessian matrix consisting of $(K-1) \times (K-1)$ block $\boldsymbol{H}_{qr}(q, r = 1, \ldots, K-1)$ where

$$\boldsymbol{H}_{qr} = -\sum_{t=1}^{T} h_i^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{iq}\right) \times \left[\delta_{qr} - b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ir}\right)\right] \boldsymbol{x}_i^{(t)} \left[\boldsymbol{x}_i^{(t)}\right]^T.$$

Here δ_{qr}, is the Kronecker delta defined as follows: $\delta_{qr} = 1$, if $q = r$; otherwise, $\delta_{qr} = 0$. $\boldsymbol{J}(\Omega_i^{(s,n-1)}, \mathcal{X})$ is the Jacobian matrix consisting of $(K-1)$ vectors $\boldsymbol{J}_q (q = 1, \ldots, K-1)$ where

$$\boldsymbol{J}_q = \sum_{t=1}^{T} h_i^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) \left[h_{iq}^{(t)}\left(\boldsymbol{\Phi}^{(s)}\right) - b\left(\boldsymbol{x}_i^{(t)}, \omega_{iq}\right)\right] \boldsymbol{x}_i^{(t)}.$$

Note that the iteration in (24) forms an inner loop with respect to n in the M step of the sth epoch of the EM algorithm and $\Omega_i^{(s+1)} = \Omega_i^{(s,n^*)}$ where n^* is the prespecified number of iterations, as a stopping criterion, in the inner loop.

As proved by Dempster *et al.* [16], our original likelihood function based on only observable data increases monotonically along the sequence of parameter estimation generated by an EM algorithm and converges to a local maximum. Thus, the parameter values in $\boldsymbol{\Phi}$ obtained by the EM algorithm establish a speaker model based on different speech representations.

IV. Simulations

In this section, we present simulation results by applying our GGMM proposed in Section III to text-independent speaker recognition by means of the KING speech corpus [6]. To demonstrate the effectiveness of our approach, we also apply traditional approaches to the same problem for comparison. These approaches include the GMM trained on individual speech representations, the GMM trained on a composite speech representation, and a linear combination of GMMs trained on individual speech representations.

In the sequel, we first present a brief description of our simulations, including database, acoustic signal preprocessing, and speech representation extraction. To facilitate the presentation of comparative results, we give a brief review of a linear opinion pool for combination of multiple GMMs used in our simulations. Finally, we report simulation results on speaker identification and verification.

A. Brief Description

The KING corpus is a benchmark English acoustic corpus collected by Higgins at ITT around 1987 and resampled in 1992, which is designed especially for text-independent speaker recognition. It consists of wideband (WB) and narrow-band (NB) sets. The WB set was collected with a high-quality microphone in a quiet room, whereas the NB set was collected by telephone handsets through various long distance telephone channels. In each set, all speakers are male, and ten sessions for each speaker were recorded from a week to a month apart. It is reported in [6] that some data in the WB set were unfortunately missing, which results in different populations in two sets.

Similar to specifications used in [36], our system adopts the following preprocessing for text-independent speaker recognition: 1) preemphasizing with the filter response $H(z) = 1 - 0.95z^{-1}$; 2) 32-ms Hamming windowing without overlapping; 3) removing the silence and unvoiced part of speech in terms of short-term average energy; and 4) extracting different feature vectors or representations from each short-term frame. In our simulations, three different speech representation methods are employed. Thus, three different feature vectors or representations can be extracted from a short-term frame (i.e., 19-order LPC, 16-order PLP, and 19-order MCC vectors). To form a composite representation, we simply lump these three vectors together, which leads to a 54-order composite feature vector for the short-term frame. In simulations on the NB set, the mean subtraction technique [36] is also used in preprocessing for robustness.

In our simulations, the GGMM is employed as a speaker model to characterize a speaker identity based on different speech representations. For the same task, the ordinary GMM is also used for comparison, where a GMM is trained on either an individual representation or a composite representation. In addition, the combination of GMMs by the linear opinion pool described later is performed for further comparison.

For measuring the performance of a text-independent speaker recognition system, a sequence of speech frames is divided into overlapping segments of T frames, as suggested by Reynolds [36]

$$\overbrace{D^{(l)}, D^{(l+1)}, \ldots, D^{(l+T-1)}}^{\text{segment } l}, D^{(l+T)}, \ldots\ldots$$

$$D^{(l)}, \overbrace{D^{(l+1)}, \ldots, D^{(l+T-1)}, D^{(l+T)}}^{\text{segment } l+1}, D^{(l+T+1)}, \ldots\ldots$$

For the lth testing segment, the score of a speaker model is calculated as follows:

$$S(l) = \sum_{t=l}^{l+T-1} \log P\left(D^{(t)}\right). \tag{25}$$

Here, $P(D^{(t)})$ is the probability produced by a statistical speaker model (i.e., GGMM or GMM) for a short-term frame $D^{(t)}$ through its representation(s) or feature vector(s). Thus, the smoothed likelihood value corresponding to a segment $S(l)$ becomes a score for decision making. Note that the length of a segment T in the previous testing method provides a practical way to investigate the influence of utterances of different lengths on the performance of a speaker recognition system in simulations. In other words, the use of this tunable parameter may simulate utterances of arbitrary lengths by changing its value.

B. Linear Opinion Pool

The linear opinion pool is a commonly used framework for combination of different sources of information and has been used for combination of scores produced by various speaker models for speaker recognition [35]. The basic idea is the combination by a linearly weighted sum of outputs of the component models. For K component models

$$P_{\text{LOP}}\left(D^{(t)}\right) = \sum_{k=1}^{K} w_k P_k\left(\boldsymbol{x}_k^{(t)}\right) \tag{26}$$

where $P_{\text{LOP}}(D^{(t)})$ is the output of the previous combination system, w_k is proportional weights subject to $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$, and $P_k(\boldsymbol{x}_k^{(t)})$ is the probability output by the kth component model for a short-term frame through its representation $\boldsymbol{x}_k^{(t)}$. In our simulations, each component model is a GMM trained on an individual representation, and the combination system indirectly takes advantage of different representations through decision fusion for speaker recognition.

There are several methods for selecting weights in a linear opinion pool. In our simulations, we adopt a score normalization method to determine the weights. By a validation set of L testing segments, we normalize the scores produced by each component model for the testing segments. Thus, the weight for the kth component model is obtained by

$$w_k = \frac{\sum_{l=1}^{L} S_k(l)}{\sum_{k=1}^{K} \sum_{l=1}^{L} S_k(l)} \quad k = 1, \ldots, K \tag{27}$$

where $S_k(l)$ is the score of the lth segment produced by the kth component model. Apparently, such a weight selection method provides a rough measure for the contribution of each component model in terms of the whole validation set. In our simulations, additional speech of 10 s is appended to the training set for constituting a validation set.

TABLE I
THE NUMBER OF GAUSSIAN DENSITIES IN GMMs AND OUR GGMM FOR DIFFERENT TRAINING DURATIONS

Duration	GMM	GGMM
40 s	32	$12\times3 = 36$
70 s	64	$20\times3 = 60$

C. Results of Speaker Recognition

Our simulations adopt multiple trials for obtaining the reliable performance. In each trial, two sessions are randomly selected from ten recording sessions for training, and the remaining eight sessions are employed for testing. For the WB set, we further investigate the performance of different training speech durations. Along with a long training duration (70 s) from two sessions, a short training duration (40 s) from a single session is also employed to train each speaker model. Given that a short time duration likely suffers from severe mismatch, we expect that such simulation results manifest how well a speaker modeling approach performs against mismatch. For various training durations, we use different numbers of component densities in speaker models. Table I specifies structures of GMMs and the GGMM used in simulations. In total, ten trials have been performed, and the overall performance is reported here.

Equal error rate (EER), where the false rejection rate is equal to the false acceptance rate, is a common posterior measurement for testing the discriminant capability of a speaker model in speaker verification. As shown in Table II, the statistics (mean and standard deviation) indicate the overall EERs produced by different speaker models in multiple trials. It is evident from Table II that the joint use of different speech representations results in better performance on both the WB and the NB sets, and our GGMM consistently yields the best performance, regardless of training durations. In particular, our approach significantly outperforms others in the presence of severe mismatch (i.e., on the NB set) and limited training data (i.e., training by a 40-s duration), which demonstrates the effectiveness of the proposed soft competition scheme by exploiting both competition and cooperation among different speech representations for robustness.

Unlike speaker verification, speaker identification needs to consult all speaker models registered in a system for decision making, which provides a direct way to test how accurate a speaker model characterizes the speaker identity. The training of speaker models, as described previously, is the same as done in speaker verification. For evaluating performance thoroughly, we use two testing methods; that is, an unknown voice token is identified by either the *one best* testing procedure, where the identity is inferred based on the top candidate, or the *three best* testing procedure, where the identity is determined by means of top three candidates. The results of multiple trials with various testing methods are shown in Tables III and IV. It is observed from Tables III and IV that higher identification rates are obtained as different speech representations are used in speaker

TABLE II
EERs (%) of GMMs Trained on Individual Speech Representations (GMM-LPC, GMM-PLP, GMM-MCC), and on a Composite Representation (GMM-C), the Linear Opinion Pool of GMMs (LOP/GMM), and the GGMM as the Length of Testing Segments Is 8 s

Training	Data	Speaker Model					
Duration	Set	GMM-LPC	GMM-PLP	GMM-MCC	GMM-C	LOP/GMM	GGMM
40 s	WB	11.6±2.5	11.8±4.1	11.1±3.1	9.6±2.8	8.7±2.5	7.6±2.7
70 s	WB	6.8±2.2	6.7±2.3	6.2±2.8	5.8±3.3	5.1±2.3	4.6±2.1
70 s	NB	17.5±3.7	16.6±3.4	16.9±4.7	15.9±4.1	14.1±3.7	12.6±3.5

TABLE III
One Best Identification Rates (%) of GMMs Trained on Individual Speech Representations (GMM-LPC, GMM-PLP, GMM-MCC), and on a Composite Representation (GMM-C), the Linear Opinion Pool of GMMs (LOP/GMM), and the GGMM as the Length of Testing Segments Is 8 s

Training	Data	Speaker Model					
Duration	Set	GMM-LPC	GMM-PLP	GMM-MCC	GMM-C	LOP/GMM	GGMM
40 s	WB	80.6±4.2	78.9±6.1	80.3±4.7	81.4±6.8	83.1±4.7	84.9±4.3
70 s	WB	90.1±4.5	89.7±4.2	90.2±4.9	90.8±4.3	91.4±4.7	92.3±3.9
70 s	NB	63.9±6.8	64.8±7.5	65.2±7.3	66.1±6.2	67.2±6.7	73.6±6.8

TABLE IV
Three Best Identification Rates (%) of GMMs Trained on Individual Speech Representations (GMM-LPC, GMM-PLP, GMM-MCC), and on a Composite Representation (GMM-C), the Linear Opinion Pool of GMMs (LOP/GMM), and the GGMM as the Length of Testing Segments Is 8 s

Training	Data	Speaker Model					
Duration	Set	GMM-LPC	GMM-PLP	GMM-MCC	GMM-C	LOP/GMM	GGMM
40 s	WB	83.7±2.3	82.9±4.9	84.0±3.8	84.8±2.4	86.7±3.3	88.9±3.1
70 s	WB	92.3±4.0	92.1±3.1	92.7±3.2	93.9±2.7	95.2±3.3	96.8±2.5
70 s	NB	66.7±4.1	68.1±4.5	68.6±3.6	70.5±4.3	74.9±4.9	82.3±4.1

models in comparison with those models trained on an individual representation, and our GGMM is superior to others used for comparison, which is highly consistent with the performance of speaker verification. In particular, three best results of the GGMM are significantly better than others on the NB set, which further demonstrates the robustness resulting from soft competition on different speech representations.

For learning in a parametric model, training time is another index for performance evaluation. Fig. 3 illustrates CPU times taken by different speaker models when they are trained on the training set corresponding to the duration of 40 s. Note that the CPU time taken by the linear opinion pool (LOP/GMM) simply refers to the sum of the longest time for training an individual model and the time for determining weights in (27), given that individual speaker models might be trained in a parallel way. From Fig. 3, it is seen that the use of different representations, including our approach, incurs more or less higher computational loads in general. Fortunately, the training of our GGMM does not take significantly longer time in comparison with the GMMs trained on individual training sets (i.e., GMM-LPC, GMM-PLP, GMM-MCC). In contrast, the training time taken by our GGMM based on the global optimization is significantly shorter than that of the GMM trained on a composite representation, GMM-Composite, due to the curse of dimensionality, as well as that of LOP/GMM without parallel processing. All the simulations are performed on a PC of the Linux platform (Pentium III 800).

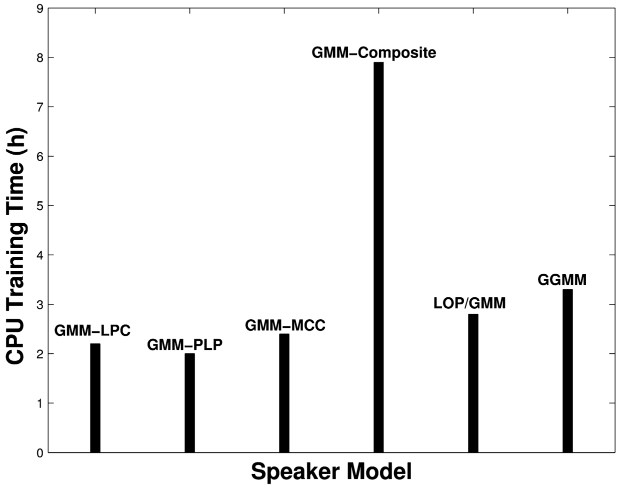

Fig. 3. Training (CPU) time taken by different speaker modeling approaches corresponding to a training set of 40-s duration.

In summary, the previous results of speaker verification and identification demonstrate the usefulness of modeling speaker characteristics with different speech representations, although doing so may incur higher computational costs. By the optimal use of different speech representations, our GGMM yields the favorite performance, especially in the presence of severe mismatch and limited training data.

V. Discussion

As pointed out previously, the proposed GGMM is a generalized finite mixture model. Although an EM algorithm has been proposed to fit its structure in Section III-B, the initialization of training is a nontrivial issue in practice. Our simulations indicate that by a random initialization, the GGMM sometimes fails to reach a local maximum quickly. To remedy this problem, we adopt a two-stage training procedure in our simulations (i.e., *local* and *global* learning). Recall that in our GGMM (see Fig. 2), there are several GMM modules receiving different input vectors. In the local learning stage, we treat the GMMs modules as independent models and employ the EM algorithm to train those GMMs separately, exactly as done in [37]. Such a local learning stage proceeds until values of their likelihood functions are large enough. Then the global learning stage starts, and the parameter values obtained in the local learning stage are used as initial values in the GGMM. As a result, the proposed EM algorithm in Section III-B is used for training the GGMM until a stopping condition is satisfied. Our empirical studies show that such a two-stage training procedure yields faster learning in contrast to the direct application of our EM algorithm to the GGMM without the local learning.

There are a number of approaches to speaker recognition through the use of different speech representations [35]. Among those approaches reviewed in [35], the data fusion through a linear opinion pool seems to resemble the proposed approach in this article due to the nature of linear combination of component models trained on different speech representations. In essence, however, they are totally distinct in terms of weight selection for combination. In the GGMM, there is a soft competition scheme on different feature vectors or representations on the basis of each speech frame, which leads to an adaptive input-dependent weight selection scheme for linear combination. Such a weight selection scheme is motivated by our previous empirical studies described in Section II-A; it has been shown that a single speech representation may not always produce the best representation for different speech frames in terms of speaker recognition. In contrast, the linear opinion pool adopts an input-independent weight selection scheme, where the global effect of different speech representations is only considered in an indirect and less accurate way. Furthermore, a global optimization procedure is used in our GGMM, whereas a suboptimal learning, training component models, and weight selection separately, takes place in the existing data fusion techniques [35]. Thus, the salient features of our GGMM yield an improvement in performance and also significantly distinguish itself from other existing data fusion techniques under the linear opinion pool framework.

In comparison with speaker modeling techniques based on a single speech representation, a weakness of the proposed approach lies in a relatively higher computational cost during training, although it appears logic due to the simultaneous use of multiple information sources. Furthermore, learning in our approach is passive; all speech frames available are treated equally important and useful for speaker recognition. As argued in [12] and [44], speaker-specific information may not uniformly distribute in every piece of an utterance. Based on this argument, two existing techniques could be applied to our GGMM to overcome the weakness of a higher computational cost and improve its performance. Frame pruning [4] can be used to discard those speech frames of little speaker-specific information, whereas adaptive weighting [3], [12], [44] would be used for active learning so a speech frame conveying more speaker-specific information can play a more important role in decision making for speaker recognition. In the future study, we would introduce such techniques to our GGMM speaker modeling approach.

To conclude, we have presented a novel approach to speaker modeling, which exploits different speech representations in an optimal way based on a soft competition scheme. Our studies demonstrate that the favorite and robust performance can be obtained by the optimal use of different speech representations, which paves a new path for exploring and exploiting potential speaker-specific information sources for speaker recognition. The proposed GGMM architecture can be generalized by the use of other component models rather than GMMs and have been extended to supervised pattern classification tasks based on different feature representations [8]. In biometrics, there are other biometric features (e.g., signature) whose nature resembles that of speech; affected by various factors, such biometric patterns always change over time. As a consequence, their interclass variabilities gradually become smaller, whereas their intraclass variabilities appear larger. Here, we would name such a class of tasks *dynamic pattern recognition*. By using various representations of a raw data set, we anticipate that our GGMM likely yields favorite and robust performance as applied to such a class of biometric authentication tasks. Furthermore, the basic idea underlying our approach would provide an alternative insight into solving any dynamic pattern recognition problems.

Appendix

In the Appendix, we derive formulas for estimating posterior probabilities with respect to those random indicators used in the E step. Prior to the development, we first explicitly express the probabilities in our statistical model given values of indicator variables. According to the definition of indicator variables, we have

$$P\left(zz_i^{(t)} = 1 \,\middle|\, \boldsymbol{\Phi}^{(s)}\right) = \alpha_i^{(k)} \tag{A.1a}$$

$$P\left(z_k^{(t)} = 1 \,\middle|\, zz_i^{(t)} = 1, \boldsymbol{\Phi}^{(s)}\right) = b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) \tag{A.1b}$$

$$P\left(z_{j\,|\,k}^{(t)} = 1 \,\middle|\, z_k^{(t)} = 1, \boldsymbol{\Phi}^{(s)}\right) = \beta_{kj}^{(s)} \tag{A.1c}$$

$$P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_k^{(t)} = 1, \boldsymbol{\Phi}^{(s)}\right) = \sum_{j=1}^{N_k} \beta_{kj}^{(s)} G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \boldsymbol{\Sigma}_{kj}^{(s)}\right) \tag{A.1d}$$

$$P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_{j\,|\,k}^{(t)} = 1, z_k^{(t)} = 1, \boldsymbol{\Phi}^{(s)}\right) = G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \boldsymbol{\Sigma}_{kj}^{(s)}\right). \tag{A.1e}$$

The application of the Bayesian rule to (14a) yields

$$\begin{aligned} h_i^{(t)}\left(\mathbf{\Phi}^{(s)}\right) &= P\left(zz_i^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) \\ &= \frac{P\left(\boldsymbol{X}^{(t)} \,\middle|\, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) P\left(zz_i^{(t)}=1 \,\middle|\, \mathbf{\Phi}^{(s)}\right)}{P\left(\boldsymbol{X}^{(t)} \,\middle|\, \mathbf{\Phi}^{(s)}\right)}. \end{aligned} \tag{A.2}$$

The mixing proportions are used for combining the output of those input-dependent GMMs, and the estimation of $P(zz_i^{(t)} = 1 \,|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)})$ needs to consider the current state of GMMs. Therefore, $P(\boldsymbol{X}^{(t)} \,|\, zz_i^{(t)} = 1, \mathbf{\Phi}^{(s)})$ can be estimated through the use of the total probability rule as follows:

$$\begin{aligned} P\left(\boldsymbol{X}^{(t)} \,\middle|\, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) &= \sum_{k=1}^{K} P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_k^{(t)}=1, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) \\ &\quad \times P\left(z_k^{(t)}=1 \,\middle|\, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) \\ &= \sum_{k=1}^{K} P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_k^{(t)}=1, \mathbf{\Phi}^{(s)}\right) \\ &\quad \times P\left(z_k^{(t)}=1 \,\middle|\, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) \\ &= \sum_{k=1}^{K}\sum_{j=1}^{N_k} \beta_{kj}^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right). \end{aligned} \tag{A.3}$$

Note that in the second step $zz_i^{(t)} = 1$ is dropped from $P(\boldsymbol{X}^{(t)} \,|\, z_k^{(t)} = 1, zz_i^{(t)} = 1, \mathbf{\Phi}^{(s)})$ because it is only associated with mixing proportions, and therefore, independent of any GMM. The last step results from (A.1b) and (A.1d). Inserting (7), (A.1a), and (A.3) into (A.2) yields (A.4), shown at the bottom of the page.

Similarly, the use of the Bayesian rule in (14b) leads to (A.5), shown at the bottom of the page, where the application of the product rule based on (A.1a) and (A.1b) yields

$$\begin{aligned} P\left(zz_i^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \mathbf{\Phi}^{(s)}\right) &= P\left(z_k^{(t)}=1 \,\middle|\, zz_i^{(t)}=1, \mathbf{\Phi}^{(s)}\right) P\left(zz_i^{(t)}=1 \,\middle|\, \mathbf{\Phi}^{(s)}\right) \\ &= \alpha_i^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) \end{aligned} \tag{A.6a}$$

and for the same reason on the independence in (A.3), we have

$$\begin{aligned} P\left(\boldsymbol{X}^{(t)} \,\middle|\, zz_i^{(t)}=1, z_k^{(t)}=1, \mathbf{\Phi}^{(s)}\right) &= P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_k^{(t)}=1, \mathbf{\Phi}^{(s)}\right) \\ &= \sum_{j=1}^{N_k} \beta_{kj}^{(s)} G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right). \end{aligned} \tag{A.6b}$$

Inserting (7) and (A.6) into (A.5) yields (A.7), shown at the bottom of the page.

Finally, the application of the Bayesian rule to (14c) results in (A.8), shown at the bottom of the page. The further use of

$$h_i^{(t)}\left(\mathbf{\Phi}^{(s)}\right) = P\left(zz_i^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) = \frac{\alpha_i^{(s)} \sum_{k=1}^{K}\sum_{j=1}^{N_k} \beta_{kj}^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right)}{\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{i=1}^{K} \alpha_i^{(s)} \beta_{kj}^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right)}. \tag{A.4}$$

$$h_{ik}^{(t)}\left(\mathbf{\Phi}^{(s)}\right) = P\left(zz_i^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) = \frac{P\left(\boldsymbol{X}^{(t)} \,\middle|\, zz_i^{(t)}=1, z_k^{(t)}=1, \mathbf{\Phi}^{(s)}\right) P\left(zz_i^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \mathbf{\Phi}^{(s)}\right)}{P\left(\boldsymbol{X}^{(t)} \,\middle|\, \mathbf{\Phi}^{(s)}\right)} \tag{A.5}$$

$$h_{ik}^{(t)}\left(\mathbf{\Phi}^{(s)}\right) = P\left(zz_i^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) = \frac{\alpha_i^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) \sum_{j=1}^{N_k} \beta_{kj}^{(s)} G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right)}{\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{i=1}^{K} \alpha_i^{(s)} \beta_{kj}^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right)}. \tag{A.7}$$

$$h_{kj}^{(t)}\left(\mathbf{\Phi}^{(s)}\right) = P\left(z_{j\,|\,k}^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) = \frac{P\left(\boldsymbol{X}^{(t)} \,\middle|\, z_{j\,|\,k}^{(t)}=1, z_k^{(t)}=1, \mathbf{\Phi}^{(s)}\right) P\left(z_{j\,|\,k}^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \mathbf{\Phi}^{(s)}\right)}{P\left(\boldsymbol{X}^{(t)} \,\middle|\, \mathbf{\Phi}^{(s)}\right)}. \tag{A.8}$$

$$h_{kj}^{(t)}\left(\mathbf{\Phi}^{(s)}\right) = P\left(z_{j\,|\,k}^{(t)}=1, z_k^{(t)}=1 \,\middle|\, \boldsymbol{X}^{(t)}, \mathbf{\Phi}^{(s)}\right) = \frac{\beta_{kj}^{(s)} G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right) \sum_{i=1}^{K} \alpha_i^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right)}{\sum_{k=1}^{K}\sum_{j=1}^{N_k}\sum_{i=1}^{K} \alpha_i^{(s)} \beta_{kj}^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right) G\left(\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_{kj}^{(s)}, \mathbf{\Sigma}_{kj}^{(s)}\right)}. \tag{A.10}$$

the product and total probability rules in $P(z_{j\,|\,k}^{(t)} = 1, z_k^{(t)} = 1\,|\,\mathbf{\Phi}^{(s)})$ yields

$$P\left(z_{j\,|\,k}^{(t)} = 1, z_k^{(t)} = 1 \middle| \mathbf{\Phi}^{(s)}\right) = P\left(z_{j|k}^{(t)} = 1 \middle| z_k^{(t)} = 1, \mathbf{\Phi}^{(s)}\right) P\left(z_k^{(t)} = 1 \middle| \mathbf{\Phi}^{(s)}\right) \quad \text{(A.9a)}$$

and

$$P\left(z_k^{(t)} = 1 \middle| \mathbf{\Phi}^{(s)}\right) = \sum_{i=1}^{K} P\left(z_k^{(t)} = 1 \middle| zz_i^{(t)} = 1, \mathbf{\Phi}^{(s)}\right) P\left(zz_i^{(t)} = 1 \middle| \mathbf{\Phi}^{(s)}\right) = \sum_{i=1}^{K} \alpha_i^{(s)} b\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{\omega}_{ik}^{(s)}\right). \quad \text{(A.9b)}$$

Inserting (7), (A.1c), (A.1e), and (A.9) into (A.8) yields (A.10), shown at the bottom of the previous page.

Acknowledgment

Part of the work presented in this article was done while the author was with The University of Birmingham. The author would like to thank anonymous reviewers for their comments that improved the presentation.

References

[1] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Amer.*, vol. 52, no. 6, pp. 1687–1697, 1972.

[2] J. Attlli, M. Savic, and J. Campbell, "A TMS32020-based real time, text-independent, automatic speaker verification system," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1988, pp. 599–602.

[3] Y. Bennani, "Adaptive weighting of pattern features during learning," in *Proc. Int. Joint Conf. Neural Networks*, 1999, pp. 3008–3013.

[4] L. Besacier and J. F. Bonastre, "Frame pruning for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, 1998, pp. 765–768.

[5] J. P. Campbell, "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[6] A. D. Carlo, M. Falcone, and A. Paoloni, "Corpus design for speaker recognition," in *Proc. ESCA Workshop on Auto. Speaker Recog. Identifi. Verification*, Martigny, Switzerland, 1994, pp. 47–50.

[7] J. Colombi, D. Ruck, S. Rogers, M. Oxley, and T. Anderson, "Cohort selection and word grammar effects for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1996, pp. 85–88.

[8] K. Chen, "A connectionist method for pattern classification with diverse features," *Pattern Recognit. Lett.*, vol. 19, no. 7, pp. 545–558, 1998.

[9] K. Chen and H. Chi, "A modular neural network architecture for pattern classification based on different feature sets," *Int. J. Neural Syst.*, vol. 9, no. 6, pp. 563–581, 1999.

[10] K. Chen, L. Wang, and H. Chi, "Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification," *Int. J. Pattern Recognit. Artific. Intell.*, vol. 11, no. 3, pp. 417–445, 1997.

[11] K. Chen, D. Xie, and H. Chi, "Speaker identification based on hierarchical mixture of experts," in *Proc. World Cong. Neural Networks*, Washington, DC, 1995, pp. 1493–1496.

[12] ——, "A modified HME architecture for text-dependent speaker identification," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1309–1313, Sep. 1996.

[13] ——, "Text-dependent speaker identification based upon input/output HMMs: An empirical study," in *Neural Proc. Lett.*, vol. 3, 1996, pp. 81–89.

[14] ——, "Speaker identification using time-delay HMEs," *Int. J. Neural Syst.*, vol. 7, no. 1, pp. 29–43, 1996.

[15] K. Chen, L. Xu, and H. Chi, "Improved learning algorithm for mixture of experts in multiclass classification," *Neural Netw.*, vol. 12, no. 9, pp. 1229–1252, 1999.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[17] B. Fritzke, "Growing cell structures: A self-organizing network for unsupervised and supervised learning," *Neural Netw.*, vol. 7, no. 9, pp. 1441–1660, 1994.

[18] S. Furui, "Recent advances in speaker identification," *Pattern Recognit. Lett.*, vol. 18, no. 9, pp. 859–872, 1997.

[19] A. Haydar, M. Demirekler, and M. K. Yurtseven, "Speaker identification through use of features selected using genetic algorithm," *Electron. Lett.*, vol. 34, no. 1, pp. 39–40, 1998.

[20] C. T. Hsieh, E. Lai, and Y. C. Wang, "Robust speech features based on wavelet transform with application to speaker identification," in *Proc. Inst. Elect. Eng. Vis., Image, Signal Process.*, vol. 149, 2002, pp. 108–114.

[21] X. D. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing*. New York, : Wiley, 2000.

[22] B. Imperl, Z. Kacic, and B. Horvat, "The use of harmonic features for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1997, pp. 1131–1134.

[23] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[24] G. J. Jang, T. W. Lee, and Y. H. Oh, "Learning statistically efficient features for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 2001, pp. 437–440.

[25] C. R. Jankowski, T. F. Quatieri, and D. A. Reynolds, "Formants AM-FM for speaker identification," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1994, pp. 608–611.

[26] ——, "Fine structure features for speaker identification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1996, pp. 689–612.

[27] J. Kitter, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, no. 2, pp. 226–239, Feb., 1998.

[28] K. P. Li and E. H. Wrench, "Text-independent speaker identification with short utterances," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1983, pp. 555–558.

[29] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust speaker recognition: A feature-based approach," *IEEE Signal Process. Mag.*, pp. 56–71, Sept. 1996.

[30] G. McLanchlan and D. Peel, *Finite Mixture Models*. New York, : Wiley, 2000.

[31] T. Martinetz, S. Berkovich, and K. Schulten, "Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 558–569, Jul. 1993.

[32] P. McLanchlan and J. A. Nelder, *Generalized Linear Models*. London, U.K.: Chapman & Hall, 1983.

[33] M. Pandit and J. Kittler, "A comparison of composite features under degraded speech in speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, 1993, pp. 371–374.

[34] ——, "Feature selection for a DTW-based speaker verification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1998, pp. 769–772.

[35] R. P. Ramachandran, K. R. Farrell, R. Ramachandran, and R. J. , "Speaker recognition—General classifier approaches and data fusion methods," *Pattern Recognit.*, vol. 35, no. 12, pp. 2801–2821, 2002.

[36] D. A. Reynolds, "A Gaussian mixture modeling approach to text-independent speaker identification," Ph.D. dissertation, Elect. Eng., Georgia Inst. Technol., Atlanta, 1992.

[37] ——, "Speaker identification and verification using Gaussian mixture models," in *Proc. ESCA Workshop on Automatic Speaker Recog., Identifi. Verification*, 1994, pp. 27–30.

[38] ——, "Experimental evaluation of features for robust speaker identification," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 639–643, Oct., 1994.

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

[40] A. E. Rosenberg and F. Soong, "Recent research in automatic speaker recognition," in *Advances in Speech, Signal Processing*, S. Furui and M. M. Sondhi, Eds. Norwell, MA, : Kluwer, 1992, pp. 701–738.

[41] C. Sanderson and K. K. Paliwal, "Joint cohort normalization in a multi-feature speaker verification system," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 2001, pp. 232–235.

[42] F. Soong and A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 6, pp. 871–879, Jun. 1988.

[43] L. Wang, K. Chen, and H. Chi, "Capture interspeaker information with a neural network for speaker identification," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 436–445, Mar. 2002.

[44] K. Chen, D. Xie, and H. Chi, " Errata to 'A modified HME architecture for text-dependent speaker identification'," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, p. 455, Mar., 1997. for errata see.

Ke Chen (M'97–SM'00) received the B.S. and M.S. degrees from Nanjing University, China, in 1984 and 1987, respectively, and the Ph.D. degree from Harbin Institute of Technology, China, in 1990, all in computer science.

He has been with The University of Manchester, Manchester, U.K., since 2003, where he is now a Senior Lecturer in computer science. He was with University, Peking University, Ohio State University, Columbus, Kyushu Institute University, and Tsinghua University. He was a Professor at Microsoft Research Asia in 2000 and Hong Kong Polytechnic University in 2001. His current research interests include pattern recognition, machine learning, and machine perception.

Dr. Chen serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and *International Journal of Image and Graphics*, a Technical Program Co-Chair of 2005 International Conference on Natural Computation, and has been a member of technical program committee of numerous international conferences. At present, he chairs IEEE Computational Intelligence Society NNTC Audio & Speech Processing Task Force and ETTC Multimedia & Biometrics Task Force, and is also the Vice Chair of IEEE Computational Intelligence Society UKRI Chapter. He is a recipient of several academic awards, including NSFC Distinguished Principal Young Investigator Award and JSPS Research Award. He is a member of the IEEE Computational Intelligence Society.