Online Adaptive Learning of Continuous-Density Hidden Markov Models Based on Multiple-Stream Prior Evolution and Posterior Pooling

Qiang Huo, *Member, IEEE,* and Bin Ma

***Abstract*—We introduce a new adaptive Bayesian learning framework, called *multiple-stream prior evolution and posterior pooling*, for online adaptation of the continuous density hidden Markov model (CDHMM) parameters. Among three architectures we proposed for this framework, we study in detail a specific two-stream system where linear transformations are applied to the mean vectors of CDHMMs to control the evolution of their prior distribution. This new stream of prior distribution can be combined with another stream of prior distribution evolved without any constraints applied. In a series of speaker adaptation experiments on the task of continuous Mandarin speech recognition, we show that the new adaptation algorithm achieves a similar fast-adaptation performance as that of the incremental maximum likelihood linear regression (MLLR) in the case of small amount of adaptation data, while maintains the good asymptotic convergence property as that of our previously proposed quasi-Bayes adaptation algorithms.**

***Index Terms*—Bayesian approach, hidden Markov model, online adaptive learning, prior evolution, speaker adaptation.**

I. Introduction

It is now well-known that the performance of an automatic speech recognition (ASR) system often degrades drastically when there exist some acoustic mismatches between the training and testing conditions. For a Gaussian mixture continuous density hidden Markov model (CDHMM) based ASR system, adaptive learning of CDHMM parameters from adaptation/testing data provides a good way to reduce the possible acoustic mismatches between the training and testing conditions and thus to enhance the system performance robustness. Some desirable characteristics for a "good" adaptation algorithm include the following.

- *Incremental:* The model parameters can be continuously adapted to the new adaptation data (possibly derived from actual test utterances) without the requirement of storing a large set of previously used training/adaptation data. In comparison with the *batch mode* algorithm, the incremental algorithm has the advantage of the increased computational efficiency and reduced storage requirements.
- *Adaptive:* So as to continuously track the variations of the model parameters corresponding to the varying observation data, some *forgetting mechanism* is needed to reduce the effect of past observations relative to the new input data.
- *Efficient:* The adaptation algorithm will hold and/or improve performance with a small amount of adaptation data, and approach asymptotically to the matched-condition performance with the increasing amount of adaptation data.

In the past few years, there are many research efforts in constructing online adaptation algorithms for CDHMM parameters (e.g., [3], [7], [12]–[15], [18], [22], [26], [28], [30], [32], [33]). With the above considerations in mind, among many possibilities, our online adaptive learning algorithms [12]–[15], and the ones in [3] and [18], are developed consistently under a Bayesian inference framework based on a concept called *prior evolution*. The general methodology of our approach can be outlined as follows.

Suppose there are M speech units in a speech recognizer, each being modeled by a Gaussian mixture CDHMM. Consider a collection of such M CDHMMs $\Lambda = \{\lambda_q : \ q = 1, \ldots, M\}$, where $\lambda_q = (\pi^{(q)}, A^{(q)}, \theta^{(q)})$ denotes the set of parameters of the qth N-state CDHMM used to characterize the qth speech unit. In this compact notation, $\pi^{(q)}$ is the initial state distribution, $A^{(q)} = [a_{ij}^{(q)}]$ is the transition probability matrix, and $\theta^{(q)}$ is the parameter vector composed of mixture parameters $\theta_i^{(q)} = \{\omega_{ik}^{(q)}, m_{ik}^{(q)}, \Sigma_{ik}^{(q)}\}$ for each state i with the state observation probability density function (pdf) being a mixture of multivariate Gaussian pdfs

$$p(\mathbf{x}|\theta_i^{(q)}) = \sum_{k=1}^{K} \omega_{ik}^{(q)} \mathcal{N}(\mathbf{x}|m_{ik}^{(q)}, \Sigma_{ik}^{(q)})$$

where the mixture coefficients $\omega_{ik}^{(q)}$'s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik}^{(q)} = 1$, and $\mathcal{N}(\mathbf{x}|m_{ik}^{(q)}, \Sigma_{ik}^{(q)})$ is the kth normal mixand with $m_{ik}^{(q)}$ being the D-dimensional mean vector and $\Sigma_{ik}^{(q)}$ being the $D \times D$ covariance matrix with its dth diagonal element being $\sigma_{ik}^{(q)2}(d)$. For notational convenience, it is assumed that all the state observation pdfs have the same number of mixture components.

In a Bayesian framework, we intend to consider the uncertainty of the HMM parameters Λ by treating them as if they were random. Our prior knowledge about Λ is assumed to be summarized in a known joint *a priori* pdf $p(\Lambda|\varphi^{(0)})$ with parameters $\varphi^{(0)}$ (sometimes referred to as *hyperparameters*), where $\Lambda \in \Omega$, Ω denotes an admissible region of the HMM parameter space.

Manuscript received September 2, 1999; revised September 7, 2000. This work was supported by HK RGC Earmarked Grant HKU7016/97E and a HKU CRCG research initiation grant. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bryan George.

The authors are with the Department of Computer Science and Information Systems, The University of Hong Kong, Hong Kong (e-mail: qhuo@csis.hku.hk; binma@csis.hku.hk).

Publisher Item Identifier S 1063-6676(01)02990-X.

Such prior information may come from subject matter considerations. It can also be derived from previous experiences, e.g., training data $\mathcal{X}$. Let $\mathcal{X}_1^n = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$ be n independent sets of observation samples which are incrementally obtained and used to update our knowledge about Λ. Depending on different assumptions to make, constraints to apply, and knowledge sources to use, there are many ways to *evolve* $p(\Lambda)$. One way is to adopt the recursive Bayesian learning framework

$$p(\Lambda|\mathcal{X}_1^n) = \frac{p(\mathcal{X}_n|\Lambda) \cdot p(\Lambda|\mathcal{X}_1^{n-1})}{\int_\Omega p(\mathcal{X}_n|\Lambda) \cdot p(\Lambda|\mathcal{X}_1^{n-1}) d\Lambda}. \tag{1}$$

Starting the calculation of posterior pdf from $p(\Lambda|\varphi^{(0)})$, a repeated use of (1) produces a sequence of densities $p(\Lambda|\mathcal{X}_1^1)$, $p(\Lambda|\mathcal{X}_1^2)$, and so forth. Because of the *missing-data* problem of CDHMM, there are some serious computational difficulties to directly implement this learning procedure [13]. Consequently, some approximations are needed in practice.

One such approach called quasi-Bayes (QB) learning was developed in [13], [14]. Based on the concept of *density approximation*, the QB algorithm is designed to incrementally update the hyperparameters on the approximate posterior distribution. Actually, the *density approximation* also opens up the opportunity of appropriately manipulating the posterior distribution as we intend. For example, in order to make our Bayesian learning algorithms truly adaptive, we can introduce some *forgetting mechanisms*, namely *exponential forgetting* and *hyperparameter refreshing* as discussed in [13], [14] to adjust the contribution of previously observed sample utterances. Consequently, we will get a posterior distribution $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$ which is different from the true posterior distribution $p_{\text{true}}(\Lambda|\mathcal{X}_1^n)$, but includes the appropriate information we want to learn from the observation data $\mathcal{X}_1^n$. Both algorithms in [13] and [14] have the characteristics of being incremental and adaptive. The difference between them lies mainly in the facts of that different constraints on HMM parameters are applied, different forms for the prior pdf $p(\Lambda|\varphi^{(0)})$ are assumed, and thus different prior evolution algorithms are derived.

In [13], we assume λ_q's are independent, i.e., $p(\Lambda|\varphi^{(0)}) = \prod_{q=1}^M g(\lambda_q|\varphi_q^{(0)})$, where $g(\lambda_q|\varphi_q^{(0)})$ is the prior pdf of λ_q with the hyperparameters $\varphi_q^{(0)}$. Under the above independence assumption, each model can only be adapted if the corresponding speech unit has been observed in the current adaptation data. Consequently, only after all units have been observed enough times can all of the HMM parameters be effectively adapted. To enhance the efficiency and the effectiveness of the Bayes adaptive learning, in [14], we assume that the covariance matrices of HMMs, $\{\Sigma_{ik}^{(q)}\}$, are known. The initial prior pdf of Λ (excluding $\{\Sigma_{ik}^{(q)}\}$) is assumed to be $p(\Lambda|\varphi^{(0)}) = g(\mathbf{m}) \prod_{q=1}^M g(\lambda_q')$, where $g(\lambda_q')$ is the prior pdf of $\lambda_q' = (\pi_i^{(q)}, a_{ij}^{(q)}, \omega_{ik}^{(q)})$, and $g(\mathbf{m}) = \mathcal{N}(\mathbf{m}|\mu, \mathbf{U})$ has a joint normal pdf with a mean vector $\mu = \text{vec}\{\mu_{ik}^{(q)}\}$ and a covariance matrix $\mathbf{U}$. Here, we define $\mathbf{m} = \text{vec}\{m_{ik}^{(q)}\}$ to be the collection of the mean vectors of all the Gaussian mixture components of CDHMMs and denoted simply by an operator "vec." In this way, we can adapt not only the CDHMM parameters of the *observed* speech units, but also the *mean vectors* of *unseen* speech units by exploiting *correlations* between different mean vectors. In practice, in order to avoid the over-smoothing and to reduce the memory requirement for storing all of the correlation coefficients, we usually disregard the weakly correlated means. More recently, inspired by the above general QB framework, a sequential learning method of mean vectors of CDHMM based on a finite mixture approximation of their prior/posterior densities has also been investigated [18].

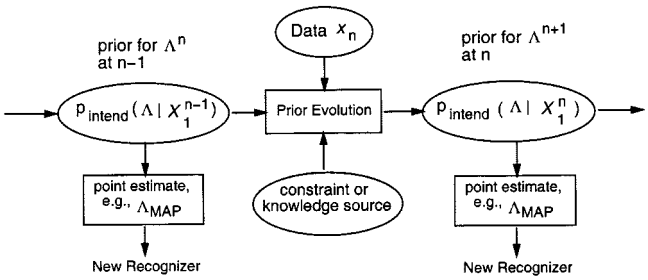

Fig. 1. Single-stream prior evolution and online adaptation.

In addition to the above method of prior evolution, we can also, for example, assume Λ to evolve in a more *constrained* way as $\Lambda^{(n)} = H_n(\Lambda^{(0)})$ where H_n represents a mapping from $\Lambda^{(0)}$ to $\Lambda^{(n)}$ and can be *incrementally* learned from the observation data $\mathcal{X}_1^n$. Then from $p(\Lambda|\varphi^{(0)})$, we can derive a new posterior distribution $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n) = p(\Lambda^{(n)})$ as the result of prior evolution.

Of course, there are other ways to evolve $p(\Lambda)$. Each leads to a different online adaptive learning algorithm. The central idea of the above approaches is that the evolving prior pdf $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$ summarizes, in a way specified by each specific prior evolution scheme, the information inherited from the prior knowledge and learned from the observation data $\mathcal{X}_1^n$. From the evolving prior distribution, the intended inference and/or decision can be made. For example, we can derive a *point estimate* $\tilde{\Lambda}$ from $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$ (e.g., MAP (maximum *a posteriori*) estimate [20], [10], [11]) and then use the conventional *plug-in MAP decision rule* (see the discussion in [16]) for recognition. This type of updating and use of $\tilde{\Lambda}$ is known as online Bayesian adaptation in speech recognition community [12]–[15]. Alternatively, $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$ can also be used directly in a Bayesian predictive classification approach [16], [17]. The concept of such type of *single-stream prior evolution* and its relation to the online adaptation of CDHMM parameters is illustrated in Fig. 1.

In this paper, we extend the concept of single-stream prior evolution to a new framework which is called *multiple-stream prior evolution and posterior pooling*. The rest of the paper is organized as follows. In Section II, we introduce the motivation of developing this new framework, the mechanism of the posterior pooling, and the architecture of the multiple-stream prior evolution. In Section III, we show an example of how to use this new framework to derive a new adaptation algorithm aimed to further increase the efficiency of the algorithms in [13], [14] in terms of both performance improvement and memory requirement. In Section IV, we apply the above new algorithm to a speaker adaptation application to illustrate its usefulness. Finally, we summarize our findings in Section V.

II. Multiple-Stream Prior Evolution and Posterior Pooling

A. Motivation

We have discussed above that there are many ways to evolve prior pdf. Each leads to a different online adaptive learning algorithm. Moreover, the prior evolution can start from either a single prior pdf, or more generally, different prior pdfs for different schemes. Depending on the specific meaning of the prior pdf and the way of prior evolution, different schemes might reflect different aspects of the learning and have their own strength and weakness. A natural way of obtaining an enhanced learning algorithm is to simultaneously maintain multiple streams of prior evolution. During the process of the prior evolution, we can design a *posterior pooling mechanism* which combines the different streams of evolved pdfs to derive an intended pdf for further inference or decision-making. This explains the first motivation of our new framework.

Another motivation of developing this new framework is related to the concepts of the decomposition of the signal variability sources, the modeling of the signal with the partial variability, and the composition of the model parameters for target signal model from individual pretrained models of signals with the partial variability. For example, the speech signal is very rich which includes the desirable linguistic information for recognition as well as many other undesirable information. A *multi-facet learning* algorithm can thus be designed to elicit from a rich set of training data $\mathcal{X}$ a set of prior distributions, $\{p^{(i)}(\Lambda|\varphi_i^{(0)}), i = 1, 2, \ldots, I\}$. Each $p^{(i)}(\Lambda|\varphi_i^{(0)})$ reflects how HMM parameters Λ varies according to one type of variability factors (e.g., speakers, speaking styles, data capturing and transmission conditions, etc.). We can treat each $p^{(i)}(\Lambda|\varphi_i^{(0)})$ as a *knowledge source* which reflects one aspect of the speech signal. After we have prepared the set of $\{p^{(i)}(\Lambda|\varphi_i^{(0)})\}$ from training data $\mathcal{X}$, we can then use them to *compose* and *derive* a condition-dependent distribution $p_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})$ guided by task specifications and a small amount of *condition-dependent* adaptation data (possibly derived from test data) $\mathcal{X}_{\text{new}}$. We can view this as a kind of *adaptive information fusion* from different knowledge sources $\{p^{(i)}(\Lambda|\varphi_i^{(0)}),\ i = 1, 2, \ldots, I\}$. Alternatively, we can first evolve $p^{(i)}(\Lambda|\varphi_i^{(0)})$ by using the adaptation data $\mathcal{X}_{\text{new}}$ and an appropriate prior evolution method to obtain a set of intended distributions $\{p^{(i)}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})\}$. Then the appropriate information fusion technique can be used to derive $p_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})$. This explains the second motivation of our new framework.

Apparently, for the above two types of applications, they request a common mathematical tool of summarizing a useful pdf from several pdfs. In the following, we propose one possible solution for this type of information fusion.

B. Fusion Mechanism

Given a set of prior pdfs $\{p^{(i)}(\Lambda|\varphi_i^{(0)}),\ i = 1, 2, \ldots, I\}$ (or evolved pdfs, i.e., posterior pdfs $\{p^{(i)}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}}), i = 1, 2, \ldots, I\}$), we can first compose an intended distribution

$$\tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}}) = \sum_i \epsilon_i \times p^{(i)}(\Lambda|\varphi_i^{(0)}) \tag{2}$$

where $\epsilon_i (0 \le \epsilon_i \le 1$ and $\sum_i \epsilon_i = 1)$ is the *fusion weight* to control the *relative importance* of the different knowledge sources $\{p^{(i)}(\Lambda|\varphi_i^{(0)})\}$. The ϵ_i's can either be automatically trained from the adaptation data $\mathcal{X}_{\text{new}}$, or just be specified according to task specifications and modeling intention. Then we can use a manageable distribution $p(\Lambda|\hat{\varphi})$ to approximate $\tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})$ by minimizing the *Kullback-Leibler directed divergence* [19] as follows:

$$\hat{\varphi} = \arg\min_{\varphi} \int \tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}}) \log \frac{\tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})}{p(\Lambda|\varphi)} \mathrm{d}\Lambda. \tag{3}$$

With $p(\Lambda|\hat{\varphi})$, we can derive a point estimate (e.g., taking a mode) of Λ and then to use the plug-in MAP decision rule to construct a speech recognizer.

Alternatively, we can first evolve $p^{(i)}(\Lambda|\varphi_i^{(0)})$ by using the adaptation data $\mathcal{X}_{\text{new}}$ and an appropriate prior evolution method to obtain a set of intended distributions $\{p^{(i)}_{\text{intend}}(\Lambda|\mathcal{X}_{\text{new}})\}$. Then the above information fusion technique can be used to derive $p(\Lambda|\hat{\varphi})$ and to construct the speech recognizer accordingly.

Apparently, if the application involves many utterances during the real use of the ASR system (i.e., $\mathcal{X}_1^n = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$ become available incrementally), the above discussed scheme can then be operated in an incremental mode. Consequently, there are several possible architectures for multiple stream prior evolution. In the following, we use two-stream prior evolution as an example to illustrate three different architectures of this new framework.

C. Architectures

The first architecture is shown in Fig. 2(a). Each stream of prior evolution starts from a single prior pdf which in the first evolution step, is the initial prior pdf, and afterwards, is the intended pdf pooled appropriately from previously evolved multiple streams of priors. The second architecture is shown in Fig. 2(b). Each stream of prior evolution starts from an independent prior pdf which in the first evolution step, is the initial prior pdf, and afterwards, is the previously evolved prior pdf in this stream. The third architecture is shown in Fig. 2(c) and is a hybrid one. One stream of prior evolves as in architecture (b) and another stream evolves as in architecture (a). All of the above three architectures can find their usages in different scenarios and applications. What is common among three architectures is the *posterior pooling* part: Given two evolved pdfs $p^{(1)}_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$ and $p^{(2)}_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$, we can define the intended posterior distribution as

$$\tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_1^n) = \epsilon \cdot p^{(1)}_{\text{intend}}(\Lambda|\mathcal{X}_1^n) + (1-\epsilon) \cdot p^{(2)}_{\text{intend}}(\Lambda|\mathcal{X}_1^n). \tag{4}$$

By using the information fusion technique discussed above, we can derive a $p(\Lambda|\varphi^{(n)})$ to approximate $\tilde{p}_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$. Then we can treat $p(\Lambda|\varphi^{(n)})$ as $p_{\text{intend}}(\Lambda|\mathcal{X}_1^n)$, and continue the prior evolution process as described in one of the above three architectures. The MAP estimate of the CDHMM parameters derived from the evolving prior distribution $p(\Lambda|\varphi^{(n)})$ can be used to update the speech recognition system. This technique

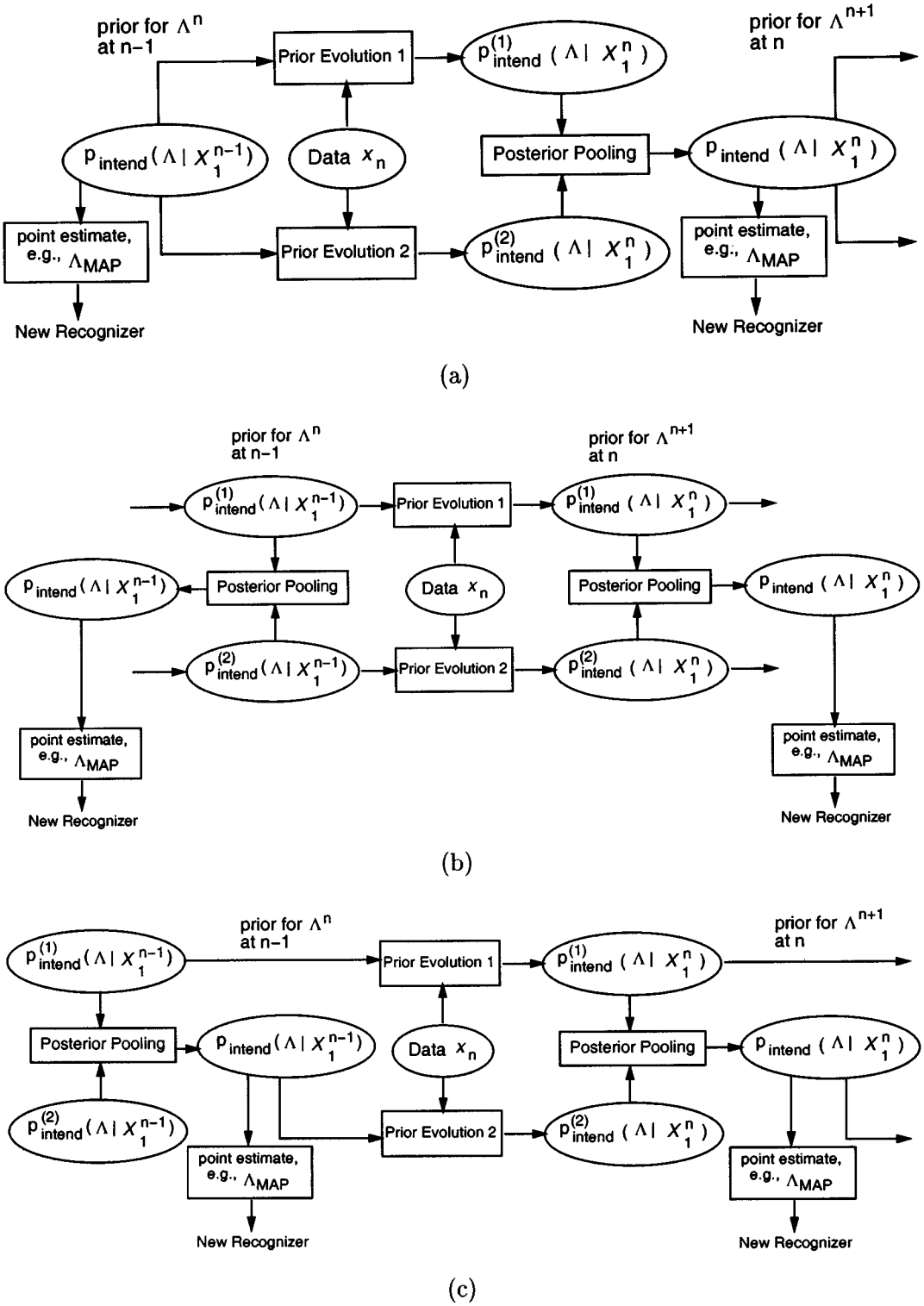

Fig. 2. Architectures for multiple-stream prior evolution and posterior pooling framework: (a) each stream evolves from a single (pooled) prior, (b) each stream evolves independently from different priors, and (c) hybrid, i.e., one stream evolves independently, and another stream evolves from the pooled prior.

of *multiple-stream prior evolution and posterior pooling* thus provides a good tool to exploit respectively the different *knowledge sources* in an appropriate way. Such information can be incorporated into the existing system so that the system can be continuously adapted to the new condition-dependent speech data and/or evolve in a desired way.

In the next section, as a case study, we consider a specific two-stream prior-evolution and posterior pooling system with the architecture (a), where one stream is the QB evolution of the CDHMM parameters as described in [13], [14], and another stream is governed by applying linear transformations to the mean vectors of CDHMM to control their evolution.

III. Case Study: QB Plus Transformation Constrained Prior Evolution

A. Method

In this study, we only consider the case of CDHMMs in which the covariance matrices are specified (i.e., fixed without adaptation). In addition to the notations described in the introduction section, we further define another operator "bdiag" to denote a block diagonal matrix, e.g., $\Xi = \text{bdiag}\{\Sigma_{ik}^{(q)}\}$, with each diagonal block element to be also a matrix, e.g., $\Sigma_{ik}^{(q)}$. The initial prior pdf of Λ is assumed to be $g(\Lambda) = g(\mathbf{m}) \prod_{q=1}^{M} g(\lambda_q')$, where $g(\lambda_q')$ takes the special form of a matrix beta pdf with sets of positive hyperparameters of $\{\eta_i^{(q)}\}, \{\eta_{ij}^{(q)}\}, \{\nu_{ik}^{(q)}\}$, and $g(\mathbf{m}) = \mathcal{N}(\mathbf{m}|\boldsymbol{\mu}(0), \mathbf{U}(0))$ has a joint normal pdf with a mean vector $\boldsymbol{\mu}(0) = \text{vec}\{\mu_{ik}^{(q)}(0)\}$ and a covariance matrix $\mathbf{U}(0)$ [13], [14]. This class of prior distributions actually constitutes a conjugate family of the complete-data density and is denoted as $\mathcal{P}$. In this study, we only consider the case of multiple stream prior evolution for mean vectors of CDHMMs. All of the other HMM parameters will evolve in the QB stream as described in [13], [14].

In the new stream of prior evolution, at time instant n, we assume that mean vectors $m_{ik}^{(q)}(n)$'s have been evolved from the original mean vectors $m_{ik}^{(q)}(0)$'s by linear transformations as follows:

$$m_{ik}^{(q)}(n) = A_{c_1(i,k,q)}^{(n)} m_{ik}^{(q)}(0) + b_{c_2(i,k,q)}^{(n)}$$

where $A_{c_1(i,k,q)}^{(n)}$ is a $D \times D$ matrix and $b_{c_2(i,k,q)}^{(n)}$ is a D-dimensional bias vector. These transformations can be shared by different mean vectors in a very flexible way. $c_1(i,k,q)$ and $c_2(i,k,q)$ represent the class indexs which are the results of two mappings from distinct mixture component labels to the shared transformation class labels. For simplicity, we only study the case of $c_1(i,k,q) = c_2(i,k,q) = c(i,k,q)$ here. Following the practice in [22], we use a hierarchical tree to define the above mappings by attaching to each node of the tree a distinct transformation. The transformation corresponding to an internal node closer to the root node represents a higher degree of sharing of the transformation among Gaussian components. These transformations can be incrementally estimated from $\mathcal{X}_1^n$ by using an approximate maximum likelihood approach described in [22], [7]. Following the practice in [22] again, in order to check if a transformation can be estimated reliably, we maintain, for each transformation, an accumulated "EM-count," count$(c(i,k,q))$, of the number of feature vectors from $\mathcal{X}_1^n$. If count$(c(i,k,q))$ exceeds a pre-specified threshold, the transformation $\{A_{c(i,k,q)}^{(n)}, b_{c(i,k,q)}^{(n)}\}$ will be viewed as being reliably estimated. In evolving $m_{ik}^{(q)}(n)$'s, the transformation $\{A_{c(i,k,q)}^{(n)}, b_{c(i,k,q)}^{(n)}\}$ will be chosen by traversing the above mapping tree to make sure the most detailed yet reliably estimated transformation be used [22]. With the above constraints applied to the $m_{ik}^{(q)}$'s, the new stream of prior pdf will evolve as follows:

$$p_{\text{new}}(\mathbf{m}|\mathcal{X}_1^n) = \mathcal{N}(\mathbf{m}|\boldsymbol{\mu}_{\text{new}}(n), \mathbf{U}_{\text{new}}(n))$$

where

$$\begin{aligned}\boldsymbol{\mu}_{\text{new}}(n) &= \mathcal{A}(n) \cdot \boldsymbol{\mu}(0) + \mathcal{B}(n) \\ \mathbf{U}_{\text{new}}(n) &= \mathcal{A}(n) \cdot \mathbf{U}(0) \cdot [\mathcal{A}(n)]^t\end{aligned}$$

with $\mathcal{A}(n) = \text{bdiag}\{A_{c(i,k,q)}^{(n)}\}$ and $\mathcal{B}(n) = \text{vec}\{b_{c(i,k,q)}^{(n)}\}$. Note that in this paper, we use $(\cdot)^t$ to denote the transpose of a vector or a matrix.

Another stream of prior pdf will evolve as described in [13], [14] as follows:

$$p_{QB}(\mathbf{m}|\mathcal{X}_1^n) = \mathcal{N}(\mathbf{m}|\boldsymbol{\mu}_{QB}(n), \mathbf{U}_{QB}(n)).$$

By pooling $p_{QB}(\mathbf{m}|\mathcal{X}_1^n)$ and $p_{\text{new}}(\mathbf{m}|\mathcal{X}_1^n)$ together as described in (4), we obtain a mixture of Gaussian pdfs

$$\tilde{p}_{\text{intend}}(\mathbf{m}|\mathcal{X}_1^n) = \epsilon \cdot p_{QB}(\mathbf{m}|\mathcal{X}_1^n) + (1-\epsilon) \cdot p_{\text{new}}(\mathbf{m}|\mathcal{X}_1^n).$$

We can now use another Gaussian pdf $\mathcal{N}(\mathbf{m}|\boldsymbol{\mu}(n), \mathbf{U}(n))$ to approximate the above pdf $\tilde{p}_{\text{intend}}(\mathbf{m}|\mathcal{X}_1^n)$ under the criterion of minimizing the Kullback-Leibler directed divergence of $\mathcal{N}(\mathbf{m}|\boldsymbol{\mu}(n), \mathbf{U}(n))$ from $\tilde{p}_{\text{intend}}(\mathbf{m}|\mathcal{X}_1^n)$. It can be derived that

$$\begin{aligned}\boldsymbol{\mu}(n) &= \epsilon\boldsymbol{\mu}_{QB}(n) + (1-\epsilon)\boldsymbol{\mu}_{\text{new}}(n) \\ \mathbf{U}(n) &= \epsilon\mathbf{U}_{QB}(n) + (1-\epsilon)\mathbf{U}_{\text{new}}(n) \\ &\quad + \epsilon(1-\epsilon)(\boldsymbol{\mu}_{QB}(n) \\ &\quad - \boldsymbol{\mu}_{\text{new}}(n))(\boldsymbol{\mu}_{QB}(n) - \boldsymbol{\mu}_{\text{new}}(n))^t.\end{aligned}$$

A block diagram of the above two-stream prior-evolution and posterior pooling system is shown in Fig. 3. In the following subsection, we discuss two implementation issues, namely 1) how to construct the mapping tree for CDHMM Gaussian components to share the linear transformations and 2) how to specify the *fusion weight* ϵ.

B. Implementation Issues

Construction of Mapping Tree: For notational simplicity, in this subsection, let us use $\mathcal{N}(\mathbf{x}|m_1, \Sigma_1), \mathcal{N}(\mathbf{x}|m_2, \Sigma_2), \ldots, \mathcal{N}(\mathbf{x}|m_L, \Sigma_L)$ to denote L Gaussian components of the initial CDHMMs, where $L = M \cdot N \cdot K$. We intend to build a binary tree with a specified *height* such that the set of *leaf nodes* of the tree represents a partition of the set of the above Gaussians. We adopt a *divisive clustering* method similar to the so-called LBG algorithm described in [23] to construct such a tree with the required modifications detailed as follows.

- We use the following symmetric *divergence* measure between two Gaussians, say, $\mathcal{N}(\mathbf{x}|m_i, \Sigma_i)$ and $\mathcal{N}(\mathbf{x}|m_j, \Sigma_j)$, to serve as the distortion measure [19] (this distortion measure is also used in, e.g., [22], [3])

$$\begin{aligned}J(i,j) = &\frac{1}{2}\text{tr}[(\Sigma_i - \Sigma_j)(\Sigma_j^{-1} - \Sigma_i^{-1})] + \\ &\frac{1}{2}\text{tr}[(\Sigma_i^{-1} + \Sigma_j^{-1})(m_i - m_j)(m_i - m_j)^t]\end{aligned}$$

where tr$[\cdot]$ denotes the trace of a matrix.

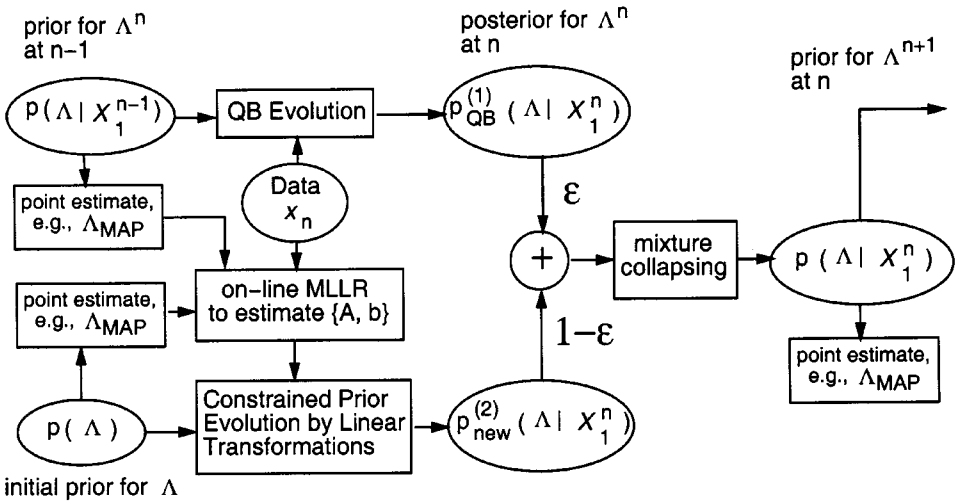

Fig. 3. Two-stream prior-evolution and posterior pooling system: QB plus linear transformation constrained prior evolution.

- The *centroid* Gaussian pdf, $\mathcal{N}(\mathbf{x}|m_c, \Sigma_c)$, for a set of Gaussians, say, $\{\mathcal{N}(\mathbf{x}|m_i, \Sigma_i),\ i = 1, 2, \ldots, L\}$, is calculated as follows:

$$m_c = \sum_{i=1}^{L} \frac{1}{L} m_i$$

$$\Sigma_c = \sum_{i=1}^{L} \frac{1}{L} [\Sigma_i + (m_c - m_i)(m_c - m_i)^t].$$

  This can be derived by minimizing the Kullback-Leibler directed divergence of any approximating normal pdf from the Gaussian mixture $\sum_{i=1}^{L}(1/L)\mathcal{N}(\mathbf{x}|m_i, \Sigma_i)$.
- The binary "splitting" of a centroid Gaussian pdf $\mathcal{N}(\mathbf{x}|m_c, \Sigma_c)$ is conducted by splitting the mean vector m_c into two new mean vectors $m_c^{(1)} = m_c(1+\vartheta), m_c^{(2)} = m_c(1-\vartheta)$, where ϑ denotes a small perturbation factor.

The above constructed tree defines a one-to-many mapping from each distinct Gaussian component in CDHMMs to a set of nodes on the path from the leaf node to which the Gaussian component belongs, to the root of the tree. So, this tree can be used to supply the required mapping as discussed in the previous subsection. It is noted that the above *divisive clustering* method is different from the hierarchical regression class tree construction method in [22] which adopts an *agglomerative clustering* procedure. Both methods have been found work well in practice. The above *divisive clustering* procedure is more computationally efficient than the *agglomerative clustering* procedure in [22]. This issue becomes important when one want to dynamically change the tree structure based on adapted CDHMM means in the online adaptation process.

Specification of Fusion Weight: In the current two-stream prior evolution scheme, one is the QB evolution which is known to have a good asympototic convergence property [13], [14]. Another is the linear transformation constrained prior evolution which because of the sharing of the transformations, is more efficient when the amount of adaptation data $\mathcal{X}_1^n$ is small. To take advantage of the strengths of the both streams, the *fusion weight* ϵ $(0 \leq \epsilon \leq 1)$ can be designed to be a monotonically increasing function of the amount of available adaptation data.

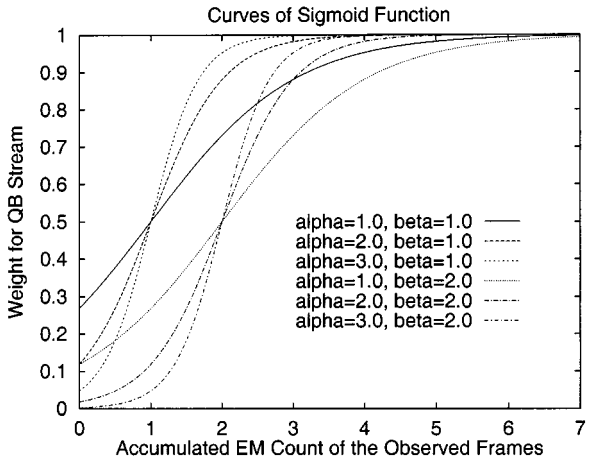

Fig. 4. Plots of the sigmoid functions for different values of α and β.

In this way, when the amount of adaptation data is small, the constrained prior evolution stream will have a bigger influence, while a good asymptotic convergence can be achieved with an increasing influence from QB evolution stream.

In the following experiments, as a first step, we ignore the correlations between $m_{ik}^{(q)}$'s, i.e., $\mathbf{U}(0)$ is assumed to be a diagonal covariance matrix. Then, the formulas in the previous subsections can be greatly simplified by treating the evolution of the individual $m_{ik}^{(q)}$ separately. After $u_{ik}^{(q)}(n)$ is calculated, we set all off-diagonal elements to be 0 so that the QB evolution will be started again from a new prior in a consistent way. In this case, we can also use a different $\epsilon_{ik}^{(q)}$ for each $m_{ik}^{(q)}$. In this study, the following sigmoid function is adopted:

$$\epsilon_{ik}^{(q)} = \begin{cases} \frac{1}{1+\exp[-\alpha(c_{ik}^{(q)} - \beta)]} & \text{if } c_{ik}^{(q)} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $c_{ik}^{(q)}$ is the related accumulated "EM count" of the number of feature vectors from $\mathcal{X}_1^n$, $\alpha > 0$ and $\beta > 0$ are two parameters to control the shape and the location of the sigmoid function. In Fig. 4, we plot the above sigmoid functions for $\alpha = 1.0, 2.0, 3.0$ and $\beta = 1.0, 2.0$. The value of β (denoted as 'beta' in Fig. 4) determines the boundary, $\epsilon_{ik}^{(q)} = 0.5$, for the

fusion weight. If we set $\beta = 1.0$, this means that once a frame of feature vector is observed for a CDHMM mixture component, the QB stream starts playing a bigger role in the posterior pooling. The value of α (denoted as "alpha" in Fig. 4) indicates the slope of the sigmoid function near the boundary $c_{ik}^{(q)} - \beta = 0$. Bigger the value of α, faster the QB stream takes over the linear transformation constrained stream. In our experiments reported in the next section, we choose, after some preliminary experiments, $\alpha = 1.0$, $\beta = 1.0$. We found that the value of β is relatively less sensitive for the performance improvement (we tried $\beta = 1.0$ and $\beta = 2.0$, both work well), while the value of α should not be set too big.

C. Discussion

Our choice of using linear transformation constraint to control the prior evolution for CDHMM means is apparently inspired by the success of using the linear transformations for the adaptation of CDHMM parameters in a maximum likelihood (ML) estimation framework, now known as the MLLR (ML linear regression) in speech recognition community (e.g., [2], [5], [8], [9], [21], and [25]). In literature, there are also other efforts (e.g., [1], [4], [6], [27]–[29]) in combining different adaptation algorithms to derive an enhanced algorithm. Most of them are developed in a heuristic way and/or only for batch-mode adaptation. The first motivation of our new framework as we described in Section II-A is similar to the aforementioned works. However, our approach has been developed consistently under a unified Bayesian framework. Each step of approximation can be theoretically justified. Our prior evolution framework is also flexible enough to include the batch-mode MAP estimation as a special case, which can be viewed as a one-step prior evolution, followed by a point estimate (taking a mode) from the evolved prior. Furthermore, as we described in Section II-A as the second motivation, the same framework can also be used to address other robust speech recognition problems which will be reported elsewhere. In the following section, we apply the above new algorithm to a speaker adaptation application to illustrate its usefulness.

IV. Speaker Adaptation Experiments

A. Experimental Setup and Baseline System

To examine the viability and the efficacy of the proposed method, a series of experiments for continuous speech recognition of Putonghua (Mandarin Chinese) are performed. The first database we used is the HKU96 Putonghua Corpus developed in our laboratory [34]. The HKU96 corpus consists of a total of 20 native Putonghua speakers, ten females and ten males, each speaking

1) all Putonghua syllables in all tones at least once;
2) 11 words of two to four syllables;
3) 16 digit strings of four to seven digits;
4) three sentences of seven rhymed syllables with /a/, /i/, and /u/ endings, respectively;
5) hundreds of sentences with verbalized punctuation from newspaper text.

All speech recordings were made in a quite room with a single National Cardioid Dynamic microphone. Speech was digitized using a Sound Blaster 16 ASP A/D card plugged into a 486 PC at 16-bit accuracy and with a sampling rate of 16 KHz. We used 18 224 sentences (about 15.5 h of raw speech) from 18 speakers (nine females and nine males) for training. Other two speakers (one female and one male) are used for speaker-independent (SI) testing and speaker adaptation. For testing data, we randomly choose 378 sentences (about 25 min of raw speech which includes 4122 syllables or 10 351 phones) from the female speaker, and 215 sentences (about 12 min of raw speech which includes 2362 syllables or 5788 phones) from the male speaker. The remaining sentences from those two speakers are used for adaptation.

The second database we used is the 863 Putonghua Corpus [35] acquired from mainland China. We randomly choose 12 speakers (six males and six females) from this corpus to serve as another set of SI testing speakers. All speech recordings were made in a quite office environment with several close-talking microphones of the same type by asking speaker to read a preprepared script of sentences from newspaper text. Speech was also digitized using Sound Blaster cards at 16-bit accuracy and with a sampling rate of 16 KHz. For each speaker, there are in total 519 sentences. Among them, 100 sentences are reserved for testing, and the remaining ones for adaptation.

Input speech was first pre-emphasized by a fixed first-order system, $1 - 0.97z^{-1}$, and then grouped into frames of 25 ms with a frame shift of 10 ms. For each frame, a Hamming window was applied followed by the computation of 12 MFCCs. The 39-dimensional feature vector used in this study consists of 12 MFCCs and log-scaled energy normalized by the peak of the individual sentence, plus their first and second-order derivatives. Sentence-based cepstral mean subtraction is applied for acoustic normalization both in training and testing.

The baseline system is a speaker independent, decision-tree-based mixture-Gaussian tied-state HMM system. The basic speech unit is the triphones considering both the within-syllable and cross-syllable contextual dependencies. The acoustic models are trained by using the HTK2.1 toolkit [31]. The adopted context-independent (CI) phone set consists of 36 phones plus silence. With this phone set definition, there are 8022 triphones in Putonghua by assuming that each syllable can be followed by any syllables. Among them, 5594 triphones are observed in our training data set, with only 4760 triphones each appearing at least three times. Each phone is modeled by a left-to-right three-emitting-state Gaussian-mixture CDHMM without state skipping. Each state has four Gaussian mixture components with each component having a diagonal covariance matrix. A special three-state CDHMM is also used for silence modeling.

The recognition task is the recognition of 410 Putonghua *base syllables* disregarding tones. The recognition network enforces silence at the start and end of sentences and allows optional silences between syllables. As for syllable language model, a uniform grammar with a syllable perplexity of 411 (i.e., each syllable can be followed by any of the 410 base syllables and silence) is used. All the recognition experiments are performed with the search engine provided by HTK2.1 toolkit.

In building the baseline system, about 150 linguistic questions are used in decision-tree construction and the relevant

thresholds for stopping criterion are adjusted to generate 3019 tied states. For this system, the averaged syllable accuracies of 75.8% and 60.7% are achieved respectively over two testing speakers on HKU96 corpus, and over 12 speakers on 863 corpus. The big performance difference indicates clearly the existence of mismatches between the two corpora.

B. Comparison of Speaker Adaptation Results

Starting from the above baseline system, we performed supervised incremental speaker adaptation experiments on 14 testing speakers by using four different methods as follows:

1) incremental MLLR adaptation method in [22], [7];
2) incremental QB adaptation method without correlation in [13];
3) incremental QB adaptation method with correlation in [14];
4) new adaptation method which includes the evolution of two streams of prior pdfs, i.e., QB without correlation and linear transformation constrained prior.

In the experiments for MLLR and new hybrid adaptation methods, the required mapping tree is built and then fixed for all of the Gaussian mixture components of the CDHMMs in the baseline recognition system by using the tree construction method described in the previous section. During the adaptation process, different number (utmost 256) of linear transformations are adaptively chosen based on the amount of available adaptation data. In a preliminary study reported in [15], full affine transformations are used. In this study, an improved performance is achieved by using block-diagonal transformations of each having three blocks corresponding to static features, their delta, and delta-delta versions. This is consistent with the findings in many MLLR-based adaptation results (e.g., [9] and [24]). In QB adaptation, the prior is evolved sentence by sentence. However, in MLLR estimation of the linear transformations, an updating interval of 30 s of speech is used. This means that in the new hybrid approach, for each given block of adaptation data, although the QB stream evolves sentence by sentence, the posterior pooling occurs every 30 s. As for the QB adaptation of correlated CDHMMs, the correlation neighborhood size is chosen to be eight (see explanation in [14]). Readers are referred to [13], [14] for more details about how to specify the initial prior from SI training data. In all of the above four adaptation algorithms, one EM iteration is performed.

Fig. 5 shows the performance (syllable accuracy in percent) comparison averaged over two testing speakers on HKU96 corpus as a function of amount of available adaptation data (in terms of minutes of raw speech) among the above four adaptation methods. We also list the averaged performance for speaker-dependent (SD) testing with the means of CDHMMs trained from 40 min of raw speech for each speaker while other CDHMM parameters kept the same as the SI-trained ones in our baseline system. The experimental results confirm our expectation, i.e., the new hybrid algorithm achieves a similar fast-adaptation performance as that of the incremental MLLR in the case of small amount of adaptation data, while maintains the good asymptotic convergence property as that of the original QB algorithm. In this set of experiments, we observed

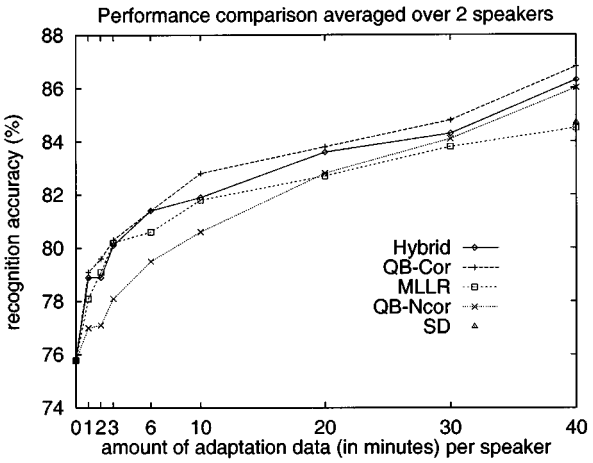

Fig. 5. Performance (syllable accuracy in percent) comparison averaged over two testing speakers on HKU96 Corpus as a function of amount (in minutes) of available adaptation data per speaker for four online adaptation methods: new hybrid method (Hybrid), QB with correlation (QB-Cor), MLLR, QB without correlation (QB-Ncor). SD recognition accuracy is 84.7%.

TABLE I
COVERAGE (IN %) OF TRIPHONES AND TIED STATES AS A FUNCTION OF THE AMOUNT OF ADAPTATION DATA FOR TWO TESTING SPEAKERS ON HKU96 CORPUS

Speaker & Units	Amount of Adaptation Data in Minutes							
	1	2	3	6	10	20	30	40
Male, Triphone	3.5	6.3	8.5	12.9	16.9	24.2	29.2	32.5
Male, Tied-State	24.3	39.6	50.0	67.8	77.4	87.9	92.3	94.1
Female, Triphone	4.1	6.7	8.6	13.5	17.5	24.4	28.4	31.5
Female, Tied-State	28.3	43.1	51.3	68.4	77.8	88.3	92.4	94.4

that the QB adaptation method with correlation achieves the best overall performance. The adaptation performance will also depend on the richness of the basic units (here the triphones and more accurately tied states) in the adaptation data. To put the above performance comparison in perspective, we list in Table I the coverage of the above units as a function of the amount of adaptation data for two testing speakers on HKU96 corpus.

Fig. 6 shows a similar performance comparison as in Fig. 5 by running the above four adaptation algorithms on 12 testing speakers from 863 corpus. Note that the SD performance is based on the models trained from 20 min raw speech for each speaker. This time, our new hybrid algorithm achieves consistently the best performance among the algorithms compared and over the different amount of adaptation data. It is interesting to recall that the QB algorithm with correlation performs better than the new hybrid algorithm on HKU96 corpus. One possible explanation is that the correlation coefficients estimated from 18 training speakers on HKU96 corpus are not reliable enough to generalize well for an acoustically mismatched corpus, namely 863 corpus. This also suggests that in order to make the QB algorithm with correlation work well, we need to have a rich set of training data to reliably estimate the correlation coefficients. Otherwise, their effectiveness for the prediction of the correlated means will be reduced.

Fig. 7 shows a similar performance comparison as in Fig. 6 by running the above four adaptation algorithms in batch mode.

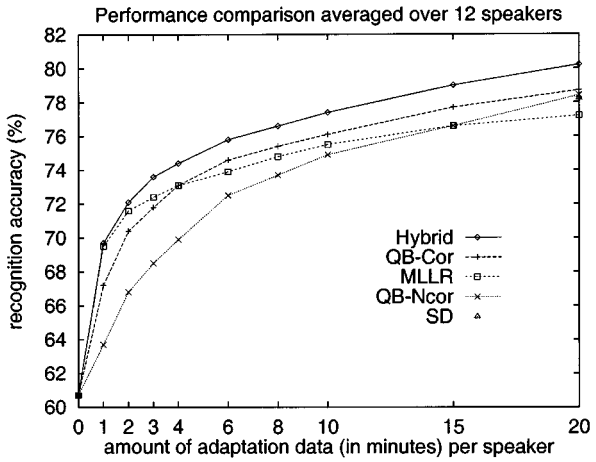

Fig. 6. Similar performance comparison as in Fig. 5 by running the online adaptation algorithms on 12 testing speakers from 863 Corpus. SD recognition accuracy is 78.2%.

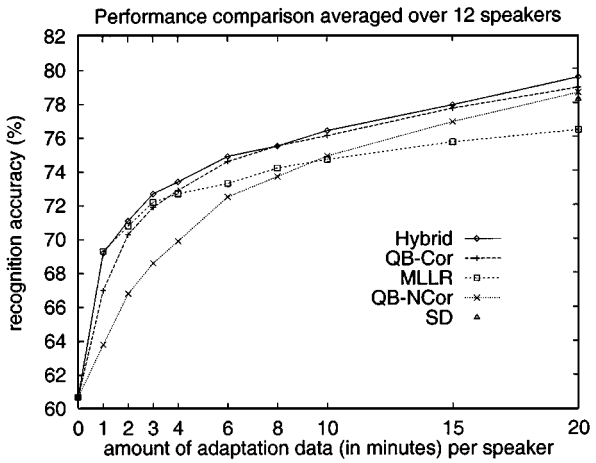

Fig. 7. Similar performance comparison as in Fig. 6 by running the adaptation algorithms in batch mode. SD recognition accuracy is 78.2%.

Three EM iterations are performed in all the experiments. The similar observations are made and the same conclusion as above can be drawn from the results. Furthermore, by comparing the results in Fig. 7 with that in Fig. 6, we observed that for QB algorithms (both with and without correlation), online and batch-mode adaptation performs equally well. However, for MLLR, we observed that online adaptation performs better than batch-mode adaptation. The benefit of online MLLR is maintained in the hybrid online adaptation algorithm too. This can be clearly observed by reproducing the results for MLLR and Hybrid algorithms as shown in Fig. 8.

C. Discussion

In addition to the desired characteristics we discussed in the introduction section, another two aspects, namely memory requirement and computational complexity are also important for comparing different adaptation algorithms. For the above four adaptation algorithms we compared, the additional memory requirement is roughly as follows:

1) for QB adaptation algorithm without correlation, it is similar to that of storing the CDHMM parameters;

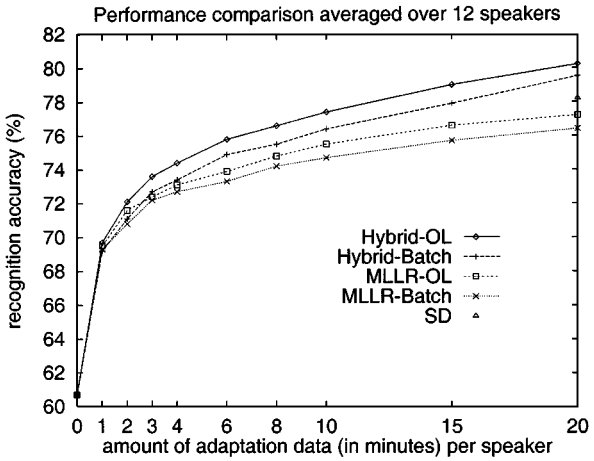

Fig. 8. Performance comparison between online and batch-mode adaptation for hybrid and MLLR algorithms.

TABLE II
COMPARISON OF USER CPU TIME (IN SECONDS) BY RUNNING FOUR ONLINE ADAPTATION ALGORITHMS ON A BLOCK OF 30 s SPEECH (IN MLLR AND HYBRID ALGORITHMS, 115 TRANSFORMATIONS ARE USED. IN QB-COR, THE CORRELATION NEIGHBORHOOD SIZE IS EIGHT)

Algorithm	Hybrid	QB-Cor	MLLR	QB-NCor
User CPU Time	25.52	67.35	21.85	13.58

2) for QB adaptation algorithm with correlation, in addition to what is required as in the above QB without correlation algorithm, storing the correlation coefficients accounts for a big memory overhead which is the main drawback of this algorithm;
3) for MLLR, the memory overhead comes from storing the linear transformations and the relevant statistics needed to derive these transformations. It depends on the number of transformations used;
4) for the new hybrid algorithm, the memory required is roughly the sum of those in 1) and 3).

For a detailed analysis of the memory requirement of the relevant algorithms, readers are referred to [8], [9], and [21] for the MLLR algorithm, and to [13] and [14] for the QB algorithms.

As for the comparison of the computational complexity of the above four adaptation algorithms, we tabulate in Table II the *user CPU time* required for running a single online adaptation step with a block of 30 s speech. The timing results are obtained on a Sun UltraSPARC-II with a 248 MHz clock. In the MLLR and Hybrid algorithms, 115 transformations are used. In QB-Cor, the correlation neighborhood size is eight. Except for QB-Cor, all of the other three algorithms can complete the online adaptation step in real time (i.e., within 30 s) on this machine. Apparently, the better performance of the new hybrid algorithm is achieved with the cost of a moderate increase of memory requirement and a slight computational overhead in comparison with either MLLR or QB without correlation.

One of the important research topics concerns about the efficient adaptation with only a couple of minutes of adaptation speech data. In this case, the form of linear transformations we used in the above hybrid algorithm is still too complicated to

estimate reliably the required transformation parameters. Recently, by adopting a simple transformation (i.e., bias for mean vector and scaling for variance of CDHMM [24], [25]) and assuming a specific prior pdf for these transformation parameters, such a "transformation-based" QB adaptation algorithm has been developed in [3] by using the general QB framework in [13]. This algorithm can be viewed as another way of prior evolution with the abovementioned linear constraints imposed. By combining this stream of prior evolution with the one in [13], we have developed another powerful algorithm which is expected to achieve a better performance for small amount of adaptation data. We will report this result elsewhere.

V. Summary

In this study, we propose a new incremental adaptive Bayesian learning framework for online adaptation of the CDHMM parameters. In a series of comparative experiments, we show that the new method has a better learning behavior as desired for a good adaptation algorithm than the methods of online MLLR and QB adaptation without correlation. The QB adaptation algorithm with correlation and the new adaptation method proposed in this paper are good candidates for both short-term and long-term adaptation of CDHMM parameters. The former usually requires more memory than the latter and is computationally more expensive. In conclusion, we recommend the user to use the new hybrid algorithm for adaptation purpose. If short-term adaptation is the only concern of the application, then MLLR is also a good tool to use. For many real-world applications, unsupervised online adaptation (OLA) is usually more realistic and desirable. One of the remaining research issues is how to guide the unsupervised OLA when the recognition rate is initially low. Different degrees of parameter tying and/or smoothing might be helpful. Incorporating some data validation mechanism will also be useful. More theoretical works are needed to develop a better verification paradigm. The new framework of *multiple-stream prior evolution and posterior pooling* opens up many research opportunities. By using multiple-stream framework, we can always exploit multiple sources of knowledge and/or apply different kinds of constraints to facilitate learning. The key to the success of these approaches depends on whether the imposed constraints really exist in the entities under investigation. It is believed that the best setup will depend on the purpose of modeling and learning as well as the nature of the specific applications. Intelligent use of these flexible tools for different purposes in different applications will be an important part of the future research.

References

[1] S. M. Ahadi and P. C. Woodland, "Combined Bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 11, pp. 187–206, 1997.

[2] J.-T. Chien, C.-H. Lee, and H.-C. Wang, "A hybrid algorithm for speaker adaptation using MAP transformation and adaptation," *IEEE Signal Processing Lett.*, vol. 4, pp. 167–168, June 1997.

[3] J.-T. Chien, "On-line hierarchical transformation of hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 656–667, Dec. 1999.

[4] S. Cox, "Predictive speaker adaptation in speech recognition," *Comput. Speech Lang.*, vol. 9, pp. 1–17, 1995.

[5] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 357–366, Sept. 1995.

[6] V. V. Digalakis and L. G. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 294–300, July 1996.

[7] V. V. Digalakis, "Online adaptation of hidden Markov models using incremental estimation algorithms," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 253–261, May 1999.

[8] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Comput. Speech Lang.*, vol. 10, pp. 249–264, 1996.

[9] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, 1998.

[10] 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, Mar. 1994.

[11] Q. Huo, C. Chan, and C.-H. Lee, "Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 334–345, Sept. 1995.

[12] ——, "On-line adaptation of the SCHMM parameters based on the segmental quasi-Bayes learning for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 141–144, Mar. 1996.

[13] Q. Huo and C.-H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 161–172, Mar. 1997.

[14] ——, "On-line adaptive learning of the correlated continuous density hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 386–397, July 1998.

[15] Q. Huo and B. Ma, "A new CDHMM adaptation method: Being incremental, adaptive and more efficient," in *Proc. 1998 Int. Symp. Chinese Spoken Language Processing*, Singapore, 1998, pp. 71–74.

[16] Q. Huo and C.-H. Lee, "A Bayesian predictive classification approach to robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 200–204, Mar. 2000.

[17] H. Jiang, K. Hirose, and Q. Huo, "Robust speech recognition based on a Bayesian prediction approach," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 426–440, July 1999.

[18] ——, "Improving Viterbi Bayesian predictive classification via sequential Bayesian learning in robust speech recognition," *Speech Commun.*, vol. 28, no. 4, pp. 313–326, 1999.

[19] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.

[20] C.-H. Lee, C.-H. Lin, and B.-H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," *IEEE Trans. Signal Processing*, vol. 39, pp. 806–814, Apr. 1991.

[21] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, pp. 171–185, 1995.

[22] ——, "Flexible speaker adaptation for large vocabulary speech recognition," in *Eurospeech'95*, Madrid, Spain, 1995, pp. 1155–1158.

[23] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, 1980.

[24] L. Neumeyer, A. Sankar, and V. Digalakis, "A comparative study of speaker adaptation techniques," in *Proc. Eurospeech'95*, Madrid, Spain, 1995, pp. 1127–1130.

[25] A. Sankar and C.-H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 190–202, May 1996.

[26] K. Shinoda and C.-H. Lee, "Unsupervised adaptation using structural Bayes approach," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing'98*, Seattle, WA, May 1998, pp. II-793–II-796.

[27] O. Siohan, C. Chesta, and C.-H. Lee, "Joint maximum *a posteriori* estimation of transformation and hidden Markov model parameters," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing'2000*, 2000, pp. II-965–II-968.

[28] J. Takahashi and S. Sagayama, "Vector-field-smoothed Bayesian learning for fast and incremental speaker/telephone-channel adaptation," *Comput. Speech Lang.*, vol. 11, pp. 127–146, 1997.

[29] M. Tonomura, T. Kosaka, and S. Matsunaga, "Speaker adaptation based on transfer vector field smoothing using maximum *a posteriori* probability estimation," *Comput. Speech Lang.*, vol. 10, pp. 117–132, 1996.

[30] S.-J. Wang and Y.-X. Zhao, "On-line Bayesian tree-structured transformation of hidden Markov models for speaker adaptation," in *Proc. IEEE Automatic Speech Recognition Understanding Workshop (CD-ROM Version)*, Keystone, CO, 1999.

[31] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 2.1)*. Cambridge, MA: Cambridge Univ. Press, 1997.
[32] G. Zavaliagkos, R. Schwartz, and J. Makhoul, "Batch, incremental and instantaneous adaptation techniques for speech recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing'95*, Detroit, MI, May 1995, pp. I-676–I-679.
[33] Y.-X. Zhao, "Self-learning speaker and channel adaptation based on spectral variation source decomposition," *Speech Commun.*, vol. 18, pp. 65–77, 1996.
[34] Y.-Q. Zu, W.-X. Li, M.-C. Ho, and C. Chan, *HKU96 – a Putonghua corpus (CD-ROM version)*: Speech Lab., Dept. Comput. Sci., Univ. Hong Kong, 1996.
[35] Y.-Q. Zu, "Sentence design for speech synthesis and speech recognition database by phonetic rules," in *Proc. Eurospeech'97*, 1997, pp. 743–746.

Qiang Huo (M'95) received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, in 1989, and the Ph.D. degree from USTC in 1994, all in electrical engineering.

From 1986 to 1990, his research work focused on the hardware design and development for real-time digital signal processing, image processing and computer vision, and speech and speaker recognition. From 1991 to 1994, he was with the Department of Computer Science, University of Hong Kong (HKU), Hong Kong, where he conducted research on speech recognition. From 1995 to 1997, he was with ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan, where he engaged in research in speech recognition. He rejoined the Department of Computer Science and Information Systems, HKU, in January 1998 as an Assistant Professor. His current major research interests include automatic recognition and/or verification of speech, speaker, and Chinese character; computational model for spoken dialogue processing; biometric authentication; adaptive signal modeling and processing; artificial neural network algorithms; machine learning; and general pattern recognition theory.

Bin Ma received the B.S. degree from the University of Shandong, China, in 1990, the M.S. degree from Institute of Automation (IA), Chinese Academy of Sciences (CAS), Beijing, in 1993. Since September 1996, he has been pursuing the Ph.D. degree in computer science in the Department of Computer Science and Information Systems, University of Hong Kong.

From 1990 to 1996, he was with the National Laboratory of Pattern Recognition, IA-CAS, where he conducted research on speech recognition. His research interests include acoustic modeling, task-independent training and adaptive training in speech recognition.