Online Bayesian Tree-Structured Transformation of HMMs With Optimal Model Selection for Speaker Adaptation

Shaojun Wang and Yunxin Zhao, *Senior Member, IEEE*

Abstract—This paper presents a new recursive Bayesian learning approach for transformation parameter estimation in speaker adaptation. Our goal is to incrementally transform or adapt a set of hidden Markov model (HMM) parameters for a new speaker and gain large performance improvement from a small amount of adaptation data. By constructing a clustering tree of HMM Gaussian mixture components, the linear regression (LR) or affine transformation parameters for HMM Gaussian mixture components are dynamically searched. An online Bayesian learning technique is proposed for recursive maximum *a posteriori* (MAP) estimation of LR and affine transformation parameters. This technique has the advantages of being able to accommodate flexible forms of transformation functions as well as *a priori* probability density functions (pdfs). To balance between model complexity and goodness of fit to adaptation data, a dynamic programming algorithm is developed for selecting models using a Bayesian variant of the "minimum description length" (MDL) principle. Speaker adaptation experiments with a 26-letter English alphabet vocabulary were conducted, and the results confirmed effectiveness of the online learning framework.

Index Terms—Affine transformation, Bayesian model selection, hidden Markov models (HMMs), linear regression (LR), model complexity, recursive Bayesian learning, robust priors, speaker adaptation, tree-structure.

I. INTRODUCTION

IN THE last two decades, significant advances have been made in statistical-modeling-based automatic speech recognition (ASR). However, due to complex interspeaker variabilities, the performance of speaker-independent (SI) large vocabulary continuous speech recognition (LVCSR) systems still lags behind that of speaker-dependent systems. The interspeaker variabilities may be attributed to speaker voice characteristics, dialect accents, education or social backgrounds, etc. A widely adopted approach to improve the performance of SI-LVCSR systems for new users is speaker adaptation, where the parameters of SI acoustic models are adjusted to better fit a user by using a certain amount of enrollment speech from the user. In practical applications, it is desirable that speaker adaptation be able to adjust a large set of model parameters by using a very small amount of enrollment speech, which in general requires exploiting relationship among acoustic-phonetic units.

Speaker adaptation techniques can be categorized into the approaches of Bayesian estimation [22], [32] and parameter transformation [16], [34]. Bayesian estimation has the asymptotic property that by using a sufficiently large amount of adaptation data from a speaker, SI acoustic models will be converged to speaker-dependent acoustic models. On the other hand, the adaptation effect of Bayesian estimation is limited when only a small amount of enrollment speech is available. By exploiting transformation tying, parameter transformation can achieve a large adaptation effect even when the amount of enrollment speech is small. However, parameter transformation may not lead to convergence to speaker-dependent models. Adaptation algorithms have been proposed to exploit the advantages of both approaches [6], [8], [17], [44]. These algorithms can achieve a large adaptation effect when using a small amount of data and maintain the asymptotic property when using a large amount of data. Furthermore, speaker adaptation may operate in batch or online modes [32], [33]. In batch mode, adaptation is performed over a set of enrollment speech data. In online mode, adaptation is performed incrementally and data are discarded after usage. As a consequence, online speaker adaptation in general requires less computation and memory as compared with batch adaptation.

Several approaches appeared in the literature for online adaptation [7], [18], [23], [26], [27], [48], [50]. One approach [48], [50] applied expectation-maximization (EM) algorithm or segmental k-means algorithm sequentially to online test speech to accomplish unsupervised learning of model parameters. Since the accumulated sufficient statistics are computed from each utterance using the model parameters updated at that time, the parameter estimates are not as accurate as batch training. Another approach [20], [21], [23] used an *incremental* version of the EM algorithm proposed in [39]. In incremental EM approach, the conditional sufficient statistics of the kth observation are computed using the $(k-1)$th model estimates, and the sufficient statistics of the previous observations are unchanged, as opposed to the batch version: after each M-step, the conditional sufficient statistics of all the training data are recalculated using the latest parameter estimates. Even though likelihood may not be monotonically increased as in the batch EM algorithm [4], convergence of this incremental EM algorithm has
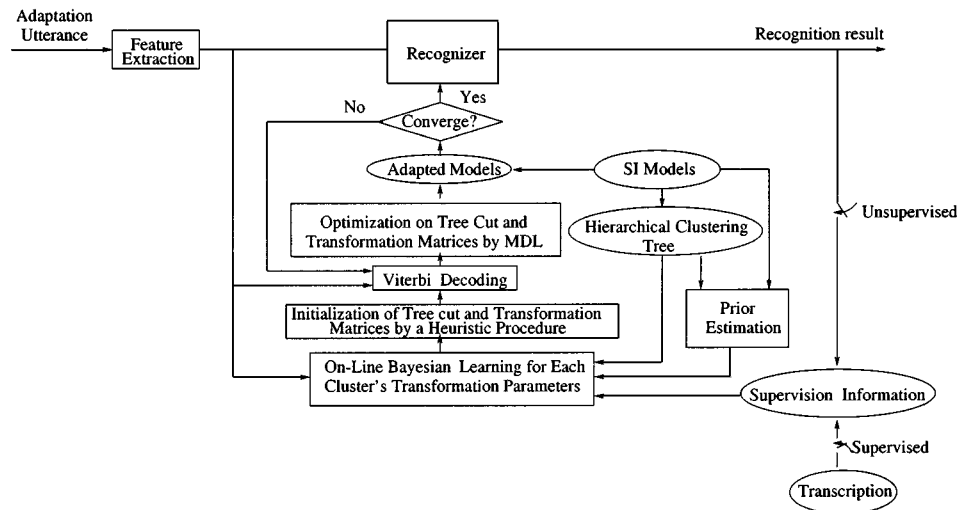
Fig. 1. Block-diagram of online Bayesian tree-structured transformation of HMM parameters.

been proved recently [24] by Csiszar's alternating minimization procedure. However, as pointed out by Digalakis [18], this incremental EM algorithm is not an online algorithm since multiple passes through the data are performed and storage of current estimate of the conditional sufficient statistics from each past observation is required. To make the algorithm an online one, Digalakis [18] suggested choosing the initial values of the conditional sufficient statistics to be zero and using a single pass through the data. The convergence property of this modified incremental EM is still open. Online quasi-Bayes learning [26], [27] first approximates the successive *a posteriori* distributions by the "closest" tractable distribution within a given class $\mathcal{P}$, under the criterion that both distributions have the same mode. The EM algorithm is next applied and the hyperparameters of the approximate *a posteriori* distribution and model parameters are incrementally updated. Empirical evidence showed that the quasi-Bayes algorithm does converge to a good solution that improves recognition rate and it has a similar behavior with the batch maximum *a posteriori* (MAP) algorithm [26], [27].

In recent years, modeling correlations among speech sounds for speaker adaptation has been widely studied. Since given a small adaptation data set, it is unlikely to have sufficient speech data for all hidden Markov model (HMM) units, certain parameter *correlation* [1], [13] and *tying* [2], [25] are introduced so that the model parameters can be consistently and fully adjusted. *Correlations* among Gaussian mean parameter vectors have been used in HMM parameter adaptation [31] and *tying* has been widely used in transformation-based adaptation [34]. Certain techniques also relate model parameters across all classes by making Markovian assumptions on the dependency structure, where joint correlation is represented by a low-order conditional distribution and hence a relatively small number of parameters are used to characterize the dependency. Examples include Markov random fields [41], multi-scale tree processes [28], tree-structural MAP adaptation [43], buried HMMs [3], and dynamic Bayesian networks [51]. Among these methods, Digalakis *et al.* [19] compared the first three and concluded that significant gain in accuracy can be obtained by exploiting dependency among acoustic model parameters.

Chien [7] recently developed an online Bayesian transformation adaptation scheme that uses the online quasi-Bayes algorithm [26] to estimate the model transformation parameters. The HMM Gaussian parameters were incrementally adapted through a set of transformation functions. A hierarchical tree of HMM parameters was built, with each HMM Gaussian component related to a set of nodes, one at each layer of the tree. For each Gaussian component, the node at the lowest layer of the tree that contains adaptation data tokens was determined and its transformation parameters were used for adaptation of the Gaussian component. With a small amount of adaptation data, this approach is capable of transforming all HMM Gaussian components and is proven superior to direct online quasi-Bayes adaptation of HMM parameters for various update interval lengths and data amounts. Nevertheless, due to the use of quasi-Bayes estimation, the forms of transformation functions were limited to those having reproducible *a priori/a posteriori* probability density function (pdf) pairs, which were either conjugate [22] or elliptically symmetric [5], [7], and were unfortunately few. Recognition performance was shown sensitive to the parameter update interval lengths, the longer the better.

In this paper, we propose applying a recursive Bayesian learning technique that we developed recently for online Bayesian transformation of HMM parameters and use the hierarchical tree-structure to control the degree of transformation tying. In recursive learning, parameter estimation is based on gradient ascent rather than closed-form solution, and therefore the constraint of reproducible *a priori/a posteriori* pdfs can be relaxed. This flexibility allows us to use affine transformations and linear regressions (LRs) in recursive online Bayesian learning, and it also allows us to choose robust *a priori* pdfs for the transformation parameters. In addition, parameter updates can be made on each utterance, overcoming the sensitivity to updating interval lengths in quasi-Bayes methods.

A block-diagram of the proposed online Bayesian learning tree-structured transformation of HMM parameters is shown in Fig. 1, where a hierarchical tree is preconstructed from the Gaussian densities of all HMMs of an SI continuous speech

recognition system. Given a new utterance, feature extraction is performed to derive a feature vector sequence that characterizes the speech input. The feature vectors are assigned to the cluster nodes in the hierarchical tree according to the state and mixture occupancy probabilities. If the adaptation is supervised, the node assignment is made by using the transcription information. Otherwise, the feature vector sequence is first recognized using the current set of HMM parameters, and the feature vectors are then assigned to the cluster nodes by referencing the decoded transcription. For each node, the transformation parameters are adaptively learnt by the proposed online Bayesian learning algorithm. An initial tree cut that determines the tree size is heuristicly searched and initial transformation parameters are obtained and applied to the Gaussian density parameters. Using the transformed models, Viterbi decoding is performed to obtain the optimal state and mixture sequence, and the feature vectors are again assigned into the tree nodes for estimation of transformation parameters. A Bayesian variant of the "mininum description length" (MDL) principle [30] is then used to determine optimal tree size and transformation matrix forms and the HMM parameters are adapted by the transformation functions. The steps from Viterbi decoding through model parameter transformation are iterated until convergence.

This paper is organized as follows. In Section II, a theoretical formulation for online Bayesian learning of transformation parameters is presented. Section III describes the use of tree-structured LR and affine transformations in the proposed online adaptation. A *bottom-up top-down* procedure is proposed in Section IV for online Bayesian learning and for initialization of tree-structured transformation parameters. In Section V, a Bayesian variant of the MDL principle is used as an information-theoretic criterion to balance the model complexity with the goodness of fit to the adaptation data, and an efficient dynamic programming algorithm is developed for selecting models. Experimental results from the proposed recursive Bayesian learning are provided and compared with those of quasi-Bayes methods in Section VI. Finally, our findings, together with future research directions and open problems, are summarized in Section VII.

II. RECURSIVE BAYESIAN LEARNING

Let $\underline{o}'_k$s represent independent blocks of speech feature vectors with probability density function $p(\underline{o}_k|\eta)$. Assume that the $\underline{o}'_k$s are received sequentially. Applying Bayes theorem, we obtain a recursive expression for the *a posteriori* pdf of η, given $\underline{o}^k = \{\underline{o}_1, \ldots, \underline{o}_k\}$, as

$$p(\eta|\underline{o}^k) = \frac{p(\underline{o}_k|\eta)p(\eta|\underline{o}^{k-1})}{\int p(\underline{o}_k|\eta)p(\eta|\underline{o}^{k-1})\,d\eta}. \qquad (1)$$

Successive computation of (1) for $k = 1, 2, \ldots$ introduces an ever-expanding combination of the previously obtained *a posteriori* pdfs and thus quickly leads to a combinatorial explosion of product terms.

In this paper, we propose a new approach for recursive Bayesian learning. Define the auxiliary function of log *a posteriori* likelihood as $R_{\underline{o}^{k+1}}(\eta, \eta^{(k)}) = Q_{\underline{o}^{k+1}}(\eta, \eta^{(k)})$

$+ \log p(\eta|\phi)$, where $Q_{\underline{o}^{k+1}}(\eta, \eta^{(k)})$ denotes the auxiliary function of log likelihood as defined in EM algorithm [15], [37] and $p(\eta|\phi)$ is the *a priori* pdf of η with a hyperparameter ϕ. It follows that maximizing $R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})$ leads to improvements in $p(\eta|\underline{o}^k)$ [15], [37]. Inspired by Titterington's work on recursive estimation using incomplete data [46], a recursive estimation formula can be derived for η by taking the normalized auxiliary function $(1/(k+1))R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})$ as the objective function. Maximizing the second-order Taylor series expansion of $(1/(k+1))R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})$ with respect to η and denoting the maximizing point by $\eta^{(k+1)}$, we have

$$\eta^{(k+1)} = \eta^{(k)} + \left[H\left(\underline{o}^{k+1}, \eta^{(k)}\right) \right]^{-1}$$
$$\cdot \frac{1}{k+1} \left. \frac{\partial R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})}{\partial \eta} \right|_{\eta=\eta^{(k)}} \qquad (2)$$

and

$$H\left(\underline{o}^{k+1}, \eta^{(k)}\right) = -\frac{1}{k+1} \frac{\partial^2 R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})}{\partial \eta \partial \eta^T}$$
$$= \frac{1}{k+1} \left[I_C\left(\underline{o}^{k+1} \middle| \eta^{(k)}\right) + I_p\left(\eta^{(k)}\right) \right] \qquad (3)$$

where $I_C(\underline{o}^{k+1}|\eta^{(k)})$ is the conditional expectation of the complete-data information matrix given $\underline{o}^{k+1}$ [37, p. 101] and due to the independency among $\underline{o}'_k$s, $I_C(\underline{o}^{k+1}|\eta^{(k)}) = \sum_{i=1}^{k+1} I_C(\underline{o}_i|\eta^{(k)})$, $I_p(\eta)$ is the *a priori* information matrix, i.e., negative Hessian matrix of log $p(\eta|\phi)$, and $H(\underline{o}^{k+1}, \eta^{(k)})$ can be called *complete-data Bayesian information matrix.* In certain cases, $I_C(\underline{o}^{k+1}|\eta^{(k)})$ can be replaced by its expectation, i.e., the complete-data Fisher information matrix $I_{CF}^{k+1}(\eta^{(k)}) = E[I_C(\underline{o}^{k+1}|\eta^{(k)})]$. In this paper, however, we consider only the case of using $I_C(\underline{o}^{k+1}|\eta^{(k)})$, and more details on I_{CF} can be found in [47]. From (2) and (3), we can see that the effect of *a priori* information decreases as the number of observations becomes large.

The batch algorithm of (2) and (3) is next converted into a recursive estimation algorithm. Define

$$\ell_k\left(\eta, \eta^{(k)}\right) = R_{\underline{o}^{k+1}}\left(\eta, \eta^{(k)}\right) - R_{\underline{o}^k}\left(\eta, \eta^{(k)}\right)$$
$$= Q_{\underline{o}_{k+1}}\left(\eta, \eta^{(k)}\right).$$

Then,

$$\frac{\partial R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})}{\partial \eta} = \frac{\partial R_{\underline{o}^k}(\eta, \eta^{(k)})}{\partial \eta} + \frac{\partial \ell_k(\eta, \eta^{(k)})}{\partial \eta}.$$

Assuming that $\eta^{(k)}$ maximizes $R_{\underline{o}^k}(\eta, \eta^{(k)})$ such that

$$\left. \frac{\partial R_{\underline{o}^k}(\eta, \eta^{(k)})}{\partial \eta} \right|_{\eta=\eta^{(k)}} = 0$$

we have

$$\left. \frac{\partial R_{\underline{o}^{k+1}}(\eta, \eta^{(k)})}{\partial \eta} \right|_{\eta=\eta^{(k)}} = \left. \frac{\partial \ell_k(\eta, \eta^{(k)})}{\partial \eta} \right|_{\eta=\eta^{(k)}}$$

and

$$\eta^{(k+1)} = \eta^{(k)} + \left[H\left(\underline{\varrho}^{k+1}, \eta^{(k)}\right) \right]^{-1}$$
$$\cdot \frac{1}{k+1} \left. \frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial \eta} \right|_{\eta=\eta^{(k)}} \qquad (4)$$

where recursive computation of $H(\underline{\varrho}^{k+1}, \eta^{(k)})$ is approximated as

$$\frac{1}{k+1} \left[\sum_{i=1}^{k+1} I_C\left(\underline{\varrho}_i \,\middle|\, \eta^{(i-1)}\right) + I_p\left(\eta^{(k)}\right) \right].$$

In order to satisfy

$$\left. \frac{\partial R_{\underline{\varrho}^{k+1}}\left(\eta, \eta^{(k+1)}\right)}{\partial \eta} \right|_{\eta=\eta^{(k+1)}} = 0$$

we iterate (4) several times for each $\underline{\varrho}_{k+1}$ and denote the resulting estimate as $\eta^{(k+1)}$.

In most cases, the matrix $H(\underline{\varrho}^{k+1}, \eta^{(k)})$ will be positive definite. This implies strict concavity of $R_{\underline{\varrho}^{k+1}}(\eta, \eta^{(k)})$ and hence the unique maximizing point for $R_{\underline{\varrho}^{k+1}}(\eta, \eta^{(k)})$. It also implies that

$$H\left(\underline{\varrho}^{k+1}, \eta^{(k)}\right)^{-1} \frac{1}{k+1} \left. \frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial \eta} \right|_{\eta=\eta^{(k)}}$$

is an ascent direction, and therefore taking a fractional step ε in the direction will lead to an increase in $(1/(k+1))R_{\underline{\varrho}^{k+1}}(\underline{\varrho}^{k+1}, \eta)$, that is,

$$\frac{1}{k+1} R_{\underline{\varrho}^{k+1}}\left(\underline{\varrho}^{k+1}, \eta^{(k+1)}\right) > \frac{1}{k+1} R_{\underline{\varrho}^{k+1}}\left(\underline{\varrho}^{k+1}, \eta^{(k)}\right).$$

Thus, we can modify (4) to be

$$\eta^{(k+1)} = \eta^{(k)} + \varepsilon_k H\left(\underline{\varrho}^{k+1}, \eta^{(k)}\right)^{-1}$$
$$\cdot \frac{1}{k+1} \left. \frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial \eta} \right]\right|_{\eta=\eta^{(k)}} \qquad (5)$$

and the modified algorithm has the desirable property of being locally monotonic when $0 < \varepsilon_k < 2$ [37]. The optimal choice of ε_k at each step is determined by a line search [36] to maximize $(1/(k+1))R_{\underline{\varrho}^{k+1}}(\eta, \eta^{(k)})$, where $(1/(k+1))R_{\underline{\varrho}^{k+1}}(\eta, \eta^{(k)})$ is approximated by

$$\frac{1}{k+1} \sum_{i=1}^{k+1} \left[Q_{\underline{\varrho}_i}\left(\eta, \eta^{(i-1)}\right) + I_p\left(\eta^{(k)}\right) \right].$$

In the choice of *a priori* pdf $p(\eta|\phi)$, we adopt the generalized Gaussian density (GGD), which has the form

$$p(\eta|\phi) = \frac{\gamma \Gamma^{1/2}(3/\gamma)}{2\Gamma^{3/2}(1/\gamma)} |\Sigma_\eta|^{-1/2} \exp\left[-(\rho(\gamma)f(\eta|\phi))^{\gamma/2} \right] \quad (6)$$

where

$f(\eta|\phi) \quad = (\eta - \mu_\eta)^T \Sigma_\eta^{-1}(\eta - \mu_\eta);$
$\rho(\gamma) \quad = \Gamma(3/\gamma)/\Gamma(1/\gamma);$
$\gamma \qquad$ shape parameter;
$\Gamma(\cdot) \qquad$ Gamma function.

The pdf is also known as the power exponential distribution, or α-Gaussian. The GGD model contains the Gaussian and Laplacian density functions as special cases when setting $\gamma = 2$ and $\gamma = 1$, respectively. For decreasing values of γ, the tails of the distribution become increasingly flat. For $0 < \gamma < 1$, the pdf exhibits an algebraic singularity at $\eta = \mu_\eta$, and as γ goes to zero, $p(\mu_\eta)$ goes to infinity. For the GGD prior, the *a priori* information matrix is given as

$$I_p(\eta) = -\partial^2 \log p(\eta|\phi)/\partial\eta\partial\eta^T$$
$$= \rho(\gamma)^{\gamma/2} \left[\gamma(f(\eta|\phi)^{\gamma/2-1}\Sigma_\eta^{-1} + 2\gamma\left(\frac{\gamma}{2}-1\right)f(\eta|\phi)^{\gamma/2-2} \right.$$
$$\left. \cdot \left(\Sigma_\eta^{-1}(\eta-\mu_\eta)\right)\left(\Sigma_\eta^{-1}(\eta-\mu_\eta)\right)^T \right]. \qquad (7)$$

The empirical Bayes approach is adopted to estimate the parameters ϕ of $p(\eta|\phi)$ [22]. A speaker independent training data set O is divided into subsets $O_1, \ldots, O_j, \ldots, O_L$ that correspond to L different speakers. The feature vectors O_j assigned to the cluster node g are assumed to have the transformation parameter $\eta_{g,j}$. The parameter $\eta_{g,j}$ is accounted as random observations generated by a prior distribution $p(\eta_g|\phi_g)$ that is common to all speakers. The marginal distribution of the training data O can then be written as

$$p(O|\phi_g) = \int \prod_{j=1}^{L} p(O_j|\eta_{g,j})p(\eta_{g,j}|\phi_g)\, d\eta_{g,j}. \qquad (8)$$

To alleviate the difficulty in integration, an approximation of the integral in (8) is maximized instead. We assume that for any ϕ_g, $p(O, \eta_{g,j}|\phi_g)$ is sharply peaked at $\hat{\eta}_{g,j} = \arg\max_{(\eta_{g,j})} p(O, \eta_{g,j}|\phi_g)$, and we try to find $\hat{\phi}_g$ to maximize $p(O, \hat{\eta}_{g,j}|\phi_g)$. This leads to an alternating maximization procedure over η_g and ϕ_g, as suggested in [22], i.e.,

$$\eta_{g,j}^{(k)} = \arg\max_{(\eta_{g,j})} p\left(O_j, \eta_{g,j} \,\middle|\, \phi_g^{(k)}\right) \qquad j = 1, \ldots, L \quad (9)$$

$$\phi_g^{(k+1)} = \arg\max_{(\phi_g)} \prod_{j=1}^{L} p\left(O_j, \eta_{g,j}^{(k)} \,\middle|\, \phi_g\right). \qquad (10)$$

In (9), the *a posteriori* estimate of $\eta_{g,j}$ can be solved by the batch EM gradient method of (2). In (10), there is no closed form solution except for the case of $\gamma = 2$ (Gaussian). In the case of GGD, for each given $\eta_{g,j}^{(k)}$, $\phi_g^{(k+1)}$ can be iteratively solved, as shown in

$$\mu_{g,\eta}^{(k+1)}(i+1) = \frac{\displaystyle\sum_{j=1}^{L} f\left(\eta_{g,j}^{(k)} \,\middle|\, \phi_g^{(k+1)}(i)\right)^{\gamma/2-1} \eta_{g,j}^{(k)}}{\displaystyle\sum_{j=1}^{L} f\left(\eta_{g,j}^{(k)} \,\middle|\, \phi_g^{(k+1)}(i)\right)^{\gamma/2-1}} \qquad (11)$$

and (12), shown at the bottom of the page, where the index i denotes iteration number. Note that when $\gamma = 2$, $f(\eta_{g,j}|\phi_g)^{\gamma/2-1} = 1$ and the above two equations provide the explicit solution of maximum likelihood estimation (MLE)

III. RECURSIVE BAYESIAN LEARNING FOR TRANSFORMATION PARAMETERS

We consider modeling isolated words by N-state continuous density HMMs, where each data block $\underline{o}_k$ corresponds to a word, with the understanding that the proposed adaptation method applies to continuous speech recognition as well. The pdf of an observation $y_t \in \Re^n$ at time t given state i is assumed to be a mixture of M multivariate Gaussian distributions

$$p(y_t|x_t = i; \lambda_i) = \sum_{m=1}^{M} \omega_{i,m} N(y_t|\mu_{i,m}, \Sigma_{i,m})$$

where $N(y_t|\mu_{i,m}, \Sigma_{i,m})$ denotes a Gaussian density with mean vector $\mu_{i,m}$ and covariance matrix $\Sigma_{i,m}$, $\omega_{i,m}$ denotes mixture weight, and the parameters of the mixture densities are denoted by $\lambda = \{\lambda_{i,m}\} = \{\omega_{i,m}, \mu_{i,m}, \Sigma_{i,m}, i = 1, \ldots, N, m = 1, \ldots, M\}$.

In recursive Bayesian learning of transformation parameters, either the HMM parameters λ are transformed by a function $F_\eta^1(\cdot)$ in the model space, or the observations are transformed by a function $F_\eta^2(\cdot)$ in the feature space, where $\eta \in \Re^D$ represents the *nuisance* parameters to be estimated. Several transformation functions have been introduced in literature for compensating mismatch between test speech and trained speech models. Although in many cases the mismatch is nonlinear and the functional form is unknown, extensive studies show that LR and affine transformation give rather large improvement to speech recognition performance [33], [45], and LR and affine transformation can be viewed as first-order approximations to nonlinear mismatch functions in both model and feature spaces. However, previous efforts in online Bayesian estimation [7], [28] could only consider bias transformation in mean parameters and scaling transformation in variance parameters due to their limitation in reproducible *a priori/a posteriori* pairs. In this paper, we are able to consider online Bayesian learning for parameters of both LR and affine transformations, since in the framework of recursive learning the form of *priors* are relaxed and hence the transformation functions are not as restricted.

A. Linear Regression Transformation

In this subsection, LR transformation [34] is applied to the mean vectors of Gaussian densities in the model space.

The transformation function $F_\eta^1(\cdot)$ has a total of $\mathcal{G}$ clusters with $\eta^{(k)} = \{A_g^{(k)}, b_g^{(k)}, g = 1, \ldots, \mathcal{G}\}$. Assume that each Gaussian density $\lambda_{i,m} = (\omega_{i,m}, \mu_{i,m}, \Sigma_{i,m})$ belongs to a cluster Ω_g. Then $\hat{\lambda}_{i,m}$, the transformed $\lambda_{i,m}$, has the form of

$$\hat{\lambda}_{i,m} = \left(\omega_{i,m}, A_g^{(k)}\mu_{i,m} + b_g^{(k)}, \Sigma_{i,m}\right). \quad (13)$$

The recursive Bayesian learning algorithm as discussed in Section II is used for estimation of the transformation parameters. Assume that the length of the $(k+1)$th data block is T_{k+1}, i.e., $\underline{o}_{k+1} = (o_{k+1,1}, \ldots, o_{k+1,T_{k+1}})$. Then

$$
\begin{aligned}
\ell_k\left(\eta, \eta^{(k)}\right) &= Q_{\underline{o}_{k+1}}\left(\eta, \eta^{(k)}\right) \\
&= -\frac{1}{2} \sum_{t=1}^{T_{k+1}} \sum_{g=1}^{\mathcal{G}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i, m) \\
&\quad \cdot \left(\left(o_{k+1,t} - A_g^{(k)}\mu_{i,m} - b_g^{(k)}\right)^T \Sigma_{i,m}^{-1}\right. \\
&\quad \left. \cdot \left(o_{k+1,t} - A_g^{(k)}\mu_{i,m} - b_g^{(k)}\right)\right) + C
\end{aligned} \quad (14)
$$

where

$$\xi_t^k(i, m) = \Pr\left(x_t^{(k)} = i, z_t^{(k)} = m \left| \underline{o}_{k+1}, A_g^{(k)}, b_g^{(k)}\right.\right)$$

for $(i, m) \in \Omega_g$, and C is a constant that absorbs the terms independent of η. For notational simplicity, the word indexes are droped, with the understanding that under each node g, the state and mixture indexes (i, m) correspond to the appropriate word HMMs with their Gaussian components tied into the cluster. The score statistic can be derived as

$$
\begin{aligned}
\frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial A_g^{(k)}} &= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i, m) \\
&\quad \cdot \left(\Sigma_{i,m}^{-1}\left(o_{k+1,t} - A_g^{(k)}\mu_{i,m} - b_g^{(k)}\right)\mu_{i,m}^T\right) \\
\frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial b_g^{(k)}} &= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i, m) \\
&\quad \cdot \left(\Sigma_{i,m}^{-1}\left(o_{k+1,t} - A_g^{(k)}\mu_{i,m} - b_g^{(k)}\right)\right).
\end{aligned}
$$

Define the precision matrix $r_{i,m} = \Sigma_{i,m}^{-1}$ with the (j, l)th element of $r_{i,m}$ denoted by $r_{i,m,j,l}$. Also define $a_{g,p,q}$ to be the

$$\Sigma_{g,\eta}^{(k+1)}(i+1) = \frac{\sum_{j=1}^{L} f\left(\eta_{g,j}^{(k)}\middle|\phi_g^{(k+1)}(i)\right)^{\gamma/2-1}\left(\eta_{g,j}^{(k)} - \mu_{g,\eta}^{(k+1)}(i)\right)\left(\eta_{g,j}^{(k)} - \mu_{g,\eta}^{(k+1)}(i)\right)^T}{\sum_{j=1}^{L} f\left(\eta_{g,j}^{(k)}\middle|\phi_g^{(k+1)}(i)\right)^{\gamma/2-1}} \quad (12)$$

(p, q)th element of A_g and $b_{g,q}$ the qth element of b_g. The entries of the information matrix $I_C(\varrho_{k+1}|\eta^{(k)})$ which is the negative of the second derivative of (14) are given as

$$
\begin{aligned}
&I_{C,\varrho_{k+1}}(a_{g,j,l}, a_{g,p,q}) \\
&= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i,m)\mu_{i,m,l}r_{i,m,j,p}\mu_{i,m,q} \\
&\quad \text{for } j, l, p, q = 1, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
&I_{C,\varrho_{k+1}}(a_{g,j,l}, b_{g,p}) \\
&= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i,m)\mu_{i,m,l}r_{i,m,j,p} \\
&\quad \text{for } j, l, p = 1, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
&I_{C,\varrho_{k+1}}(b_{g,p}, b_{g,q}) \\
&= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i,m)r_{i,m,p,q} \\
&\quad \text{for } p, q = 1, \ldots, n.
\end{aligned}
$$

The score statistic and the information matrices are used in (5) for estimation of $\eta^{(k+1)} = \{A_g^{(k+1)}, b_g^{(k+1)}, g = 1, \ldots, \mathcal{G}\}$, and the HMM parameters are transformed according to (13). This estimation and transformation procedure repeats for $k = 1, 2, \ldots$.

B. Affine Transformation

In this subsection, LR is applied to the observations in the feature space, i.e., $\underline{O}_t = A\underline{Y}_t + b$, which is often termed as affine transformation [16]. It is equivalent to a constrained model space transformation on both the mean vectors and covariance matrices. The transformation on $\lambda_{i,m} = (\omega_{i,m}, \mu_{i,m}, \Sigma_{i,m})$ has the form of

$$
\hat{\lambda}_{i,m} = \left(\omega_{i,m}, A_g^{(k)}\mu_{i,m} + b_g^{(k)}, A_g^{(k)}\Sigma_{i,m}(A_g^{(k)})^T\right). \quad (15)
$$

As in the case of LR, $\eta^{(k)} = \{A_g^{(k)}, b_g^{(k)}, g = 1, \ldots, \mathcal{G}\}$.

Recursive Bayesian learning of (5) is used for estimation of the transformation parameters by making the same assumption on data blocks. Denoting $\hat{A}_g^{(k)} = (A_g^{(k)})^{-1}$, then

$$
\begin{aligned}
\ell_k\left(\eta, \eta^{(k)}\right) &= Q_{\varrho_{k+1}}\left(\eta, \eta^{(k)}\right) \\
&= -\frac{1}{2}\sum_{t=1}^{T_{k+1}}\sum_{g=1}^{\mathcal{G}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m) \\
&\quad \cdot \left(\log|\Sigma_{i,m}| - 2\log\left|\hat{A}_g^{(k)}\right| \right. \\
&\quad + \left(\hat{A}_g^{(k)}o_{k+1,t} - \mu_{i,m} - \hat{A}_g^{(k)}b_g^{(k)}\right)^T\Sigma_{i,m}^{-1} \\
&\quad \left. \cdot \left(\hat{A}_g^{(k)}o_{k+1,t} - \mu_{i,m} - \hat{A}_g^{(k)}b_g^{(k)}\right)\right) + C \\
&\quad (16)
\end{aligned}
$$

where

$$
\xi_t^k(i,m) = \Pr\left(x_t^{(k)} = i, z_t^{(k)} = m \,\middle|\, \varrho_{k+1}, A_g^{(k)}, b_g^{(k)}\right)
$$

for $(i,m) \in \Omega_g$ and C is a term independent of η.

The score statistic can be derived as

$$
\begin{aligned}
&\frac{\partial \ell_k(\eta, \eta^{(k)})}{\partial \hat{A}_g^{(k)}} \\
&= -\sum_{t=1}^{T_{k+1}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m)\left(-\left[\left(\hat{A}_g^{(k)}\right)^{-1}\right]^T + \Sigma_{i,m}^{-1} \right. \\
&\quad \left. \cdot \left(\hat{A}_g^{(k)}o_{k+1,t} - \mu_{i,m} - \hat{A}_g^{(k)}b_g^{(k)}\right)\left(o_{k+1,t} - b_g^{(k)}\right)^T\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell_k\left(\eta, \eta^{(k)}\right)}{\partial b_g^{(k)}} &= \sum_{t=1}^{T_{k+1}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m) \\
&\quad \cdot \left(\left(\hat{A}_g^{(k)}\right)^T\Sigma_{i,m}^{-1}\left(\hat{A}_g^{(k)}o_{k+1,t} - \mu_{i,m} - \hat{A}_g^{(k)}b_g^{(k)}\right)\right).
\end{aligned}
$$

Define $\hat{a}_{g,p,q}$ and $a_{g,p,q}$ to be the (p,q)th element of $\hat{A}_g$ and A_g, and define $b_{g,p}$ to be the pth element of b_g, respectively. The entries of the matrix $I_C(\varrho_{k+1}|\eta^{(k)})$, which is the negative of the second derivative of (16), is given as

$$
\begin{aligned}
&I_{C,\varrho_{k+1}}(\hat{a}_{g,j,l}, \hat{a}_{g,p,q}) \\
&= \sum_{t=1}^{T_{k+1}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m) \\
&\quad \cdot (-a_{g,l,p}a_{g,q,j} + (o_{k+1,t,l} - b_{g,l}) \\
&\quad \cdot r_{i,m,j,p}(o_{k+1,t,q} - b_{g,q}))
\end{aligned}
$$

$$
I_{C,\varrho_{k+1}}(\hat{a}_{g,j,l}, b_{g,p})
$$

$$
= \begin{cases}
\displaystyle\sum_{t=1}^{T_{k+1}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m) \\
\quad \left(-\displaystyle\sum_{v=1}^{n}(o_{k+1,t,l} - b_{g,l})\right. \\
\qquad \left. r_{i,m,j,v}\hat{a}_{g,v,p}\right), & \text{for } l \neq p \\
\displaystyle\sum_{t=1}^{T_{k+1}}\sum_{(i,m)\in\Omega_g}\xi_t^k(i,m) \\
\quad \left(-\displaystyle\sum_{v=1}^{n}r_{i,m,j,v}\left[\displaystyle\sum_{w=1}^{n}\hat{a}_{g,v,w}\right.\right. \\
\qquad (o_{k+1,t,w} - b_{g,w}) - \mu_{i,m,v} \\
\qquad \left.\left.+\hat{a}_{g,v,l}(o_{k+1,t,l} - b_{g,l})\right]\right), & \text{for } l = p
\end{cases}
$$

$$I_{C,\varrho_{k+1}}(b_{g,p}, b_{g,q})$$

$$= \sum_{t=1}^{T_{k+1}} \sum_{(i,m)\in\Omega_g} \xi_t^k(i,m)$$

$$\cdot \left(\sum_{w=1}^{n} \sum_{v=1}^{n} \hat{a}_{g,v,p} r_{i,m,v,w} \hat{a}_{g,w,q} \right)$$

where $j, l, p, q = 1, \ldots, n$.

As noted by Gales [21], the affine transform can be implemented as a transformation of the speech features and a simple addition of the term $-\log(|\hat{A}_g|^2)$ in the likelihood score. Thus, during recognition the log-likelihood scores are calculated as

$$\log p \left(o_t | x_t = i, z_t = m; \mu_{i,m}, \Sigma_{i,m}, \hat{A}_g, b_g \right)$$

$$= \log \left(N \left(\hat{A}_g(o_t - b_g) | \mu_{i,m}, \Sigma_{i,m} \right) \right) - \tfrac{1}{2} \log \left(|\hat{A}_g|^2 \right).$$

IV. TREE-STRUCTURED TRANSFORMATION

When LR or affine transformation functions are tied across Gaussian mixture components, each transformation function is associated with a number of mixture components. This is achieved by defining a set of transformation clusters where each cluster covers the mixture components associated with the same transformation function. An assumption is made that mixture components with similar parameter values will change in a similar manner in each variation condition and these mixture components should therefore be assigned to the same transformation cluster. The tree-structured clustering technique provides a hierarchical way of defining transformation clusters.

To construct a hierarchical tree, we follow the procedure of Chien [7], [42] where the Gaussian mixture components of HMMs are clustered by using the binary split K-means algorithm with a divergence measure, i.e.,

$$d = \tfrac{1}{2} \operatorname{tr} \left[(\Sigma_{i,m} - \Sigma_{g,b})(\Sigma_{g,b}^{-1} - \Sigma_{i,m}^{-1}) \right]$$

$$+ \tfrac{1}{2} \operatorname{tr} \left[(\Sigma_{g,b}^{-1} + \Sigma_{i,m}^{-1})(\mu_{i,m} - \mu_{g,b})(\mu_{i,m} - \mu_{g,b})^T \right]$$

where $\lambda_{g,b} = (\mu_{g,b}, \Sigma_{g,b})$ was obtained by merging the Gaussian pdfs grouped in the cluster g [49, p. 360]. The clustering yields a hierarchical binary tree with d layers and $2^d - 1$ nodes where each node corresponds to a cluster and therefore there are a total of $\mathcal{G} = 2^d - 1$ clusters. Each Gaussian mixture component corresponds to d nodes in the tree, one at each layer. The root node covers Gaussian mixture components of all HMMs and each of the leaf node covers one distinct Gaussian mixture component.

In the hierarchical tree, the model state and mixture indexes of each Gaussian mixture component are stored in the corresponding tree node. The transformation parameters of each tree node are estimated by using the proposed online Bayesian learning. In general, the parameters in higher layers serve as *coarse transformation* and those in lower layers serve as *fine transformation*. To retain details of acoustic models,

the transformation functions should be as fine as possible. In [7], a *bottom-up* strategy is proposed to automatically search for the transformation parameters of each Gaussian mixture component with the computational cost of $O(\mathcal{G} \log_2 d)$ for the search of the closest transformation node. We propose a more efficient strategy, referred to as the white-black tree-based *bottom-up top-down* strategy which has the computational cost of $O(\mathcal{G})$. A *bottom-up* procedure is first used to perform online Bayesian learning of the transformation parameters $\eta_g^{(k)}$ for the nodes containing adaptation data, and a *top-down* procedure is next used to perform transformations on all the Gaussian mixture components.

In the *bottom-up* procedure, we first mark all the nodes by white. We then find the leaf nodes of Gaussian mixture components with adaptation data and mark themselves and their parents by black. We then move up one layer and for each black node in this layer, we collect adaptation data and perform online Bayesian learning of transformation parameters $\eta_g^{(k)}$, and mark the parent of the node by black. When reaching the end of the current layer, we go up one layer. This procedure is repeated until reaching the root node.

In the *top-down* procedure, we start from the layer immediately below the root node, visit each node in the layer from left to right, perform transformations on Gaussian densities according to the coloring of each node, and iterate over layers until completion of the bottom layer of the tree:

> If the current node is white and its parent is black
> then transform the parameters of the Gaussian densities that are covered under this node by using the transformation parameters estimated at its parent node;
> else if the current node is black and the node is a leaf
> then transform the parameters of the Gaussian density of the node by using the transformation parameters estimated at this node;
> else move to next node.

Fig. 2 gives an illustration for this *bottom-up* estimation and *top-down* transformation procedure.

After this *bottom-up top-down* procedure, all the Gaussian mixture components are updated by the *finest* transformation parameters available.

V. BAYESIAN MODEL SELECTION

A transformation matrix can be chosen as full, diagonal, or block diagonal. The use of block diagonal matrix is based on the assumption that a separate transformation can be used for each type of speech features, including cepstral coefficients, energy, first-order time derivatives, resulting in a block diagonal structure with parameter correlation considered only within each type of features. The choice of the transformation matrix structure is in general a tradeoff between the number of parameters to be estimated and the amount of adaptation data required. The problem is referred to as *model parameterization complexity*. When building a hierarchical transformation tree, we can tie all Gaussian components together and apply a global transformation, which corresponds to a tree with a root node only, or we can grow a full tree until each leaf node contains one Gaussian component and apply specific transformation to
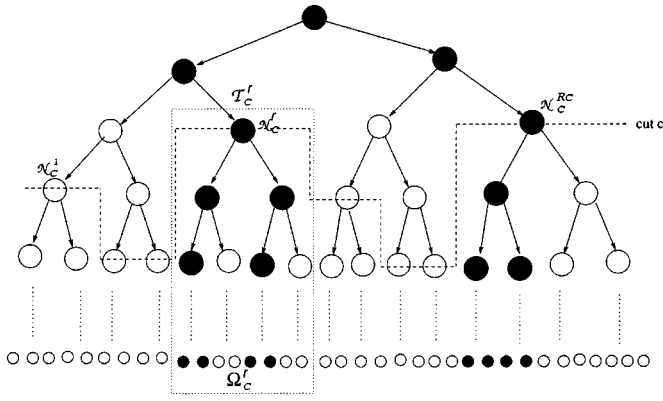
Fig. 2. White−black tree for bottom-up estimation and top-down transformation.

each Gaussian. In general, a shallow tree is simpler, but it produces a poor fit to the data; in contrast, a deep tree is more complex, but it provides a good fit to the data. Therefore, the choice of tree-structure is a tradeoff between model simplicity and goodness of fit to data. The problem is termed as *model structure complexity*. In order to balance the complexity of the tree-structured model with the goodness of fit to adaptation data, model-selection information criterion needs to be used. In this section, we show how to accommodate a Bayesian variant of Rissanen's MDL [30] into online Bayesian adaptation to control both model structural complexity and parameterization complexity to best fit the adaptation data, the goal being minimization of recognition error.

The model selection problem is coarsely prefigured by Occam's Razor [29]: given two hypotheses that fit the data equally well, prefer the simpler one. Rissanen distills such thinking in his *Principle of MDL: choose the model that gives the shortest description of data*. Originally Rissanen derived the description length by MLE. Here we apply Rissanen's method of parameter truncation to a Bayesian framework as proposed in [30]. Consider the multi-class hypothesis testing problem of determining which model $m \in \mathcal{M}$ generated a given data o. For each model m, the likelihood of observation and *a prior* density are denoted by $p(o|\eta, m)$ and $p(\eta|\phi, m)$. The Bayesian approach of Rissanen's MDL is to describe the observed data o with a two-stage code in which we first encode the MAP estimate $\tilde{\eta}$ and then encode o under the model determined by $\tilde{\eta}$. When both o and $\tilde{\eta}$ are discrete, the encoding length can be obtained according to Shannon's theory as

$$\mathcal{L}(o, \tilde{\eta}, m) = -[\log p(o|\tilde{\eta}, m) + \log p(\tilde{\eta}|\phi, m)].$$

When o and $\tilde{\eta}$ are both continuous variables, we need to truncate them each to a desired precision and measure the code length on the truncated values o_{tr} and $\tilde{\eta}_{tr}$. In practice, the effect of truncation o on the code length is insignificant whereas that of truncation $\tilde{\eta}$ on the code length is significant. Therefore we would like to focus on $\mathcal{L}(o, \tilde{\eta}_{tr}, m)$ and find the optimal precision for $\tilde{\eta}_{tr}$. Due to difficulties in a direct analysis

on $\mathcal{L}(o, \tilde{\eta}_{tr}, m)$, a second order Taylor series approximation is made on $\mathcal{L}(o, \tilde{\eta}_{tr}, m)$ around the MAP estimate $\tilde{\eta}$, which yields

$$
\begin{aligned}
&\mathcal{L}(o, \tilde{\eta}_{tr}, m) \\
&\approx -[\log p(o|\tilde{\eta}, m) + \log p(\tilde{\eta}|\phi, m)] \\
&\quad + \tfrac{1}{2}(\tilde{\eta} - \tilde{\eta}_{tr})^T [I_O(o|\tilde{\eta}, m) + I_p(\tilde{\eta}|\phi, m)](\tilde{\eta} - \tilde{\eta}_{tr}).
\end{aligned}
\tag{17}
$$

Adopting the worst-case minimax approach [30] which picks the truncated parameter $\tilde{\eta}_{tr}$ to minimize the maximum of (17) yields the total description lengths $\mathcal{L}$ as

$$
\begin{aligned}
\mathcal{L}(o, m) = &-[\log p(o|\tilde{\eta}, m) + \log p(\tilde{\eta}|\phi, m)] \\
&+ \tfrac{1}{2} \log \det[I_O(o|\tilde{\eta}, m) + I_p(\tilde{\eta}|\phi, m)] \\
&+ \frac{D}{2}(1 - \log 4)
\end{aligned}
\tag{18}
$$

where

$$I_O(o|\eta) = -\frac{\partial^2 \log p(o|\eta)}{\partial \eta \partial \eta^T}$$

is referred to as the observed incomplete-data information matrix, and D is the number of unknown parameters or the dimension of the statistical model. It can be shown that when the number of observation data k is large, the log determinant term can be approximated by $(D/2) \log k$, and it will dominate the last term. As a model becomes more complex, the sum of the first two terms decreases and that of the last two terms increases. The description length $\mathcal{L}(o, m)$ has its minimum at a model m with an appropriate complexity.

Define a cut in a tree be a set of nodes dividing the tree into an upper part and a lower part. One example of a cut is shown as the set of nodes on the dashed line in Fig. 2. Each node in a cut has a transformation matrix with one of three forms, namely diagonal, block diagonal, and full. A "model" corresponds to a cut where each node in the cut has a fixed form of transformation matrix, but the form may vary from node to node. The coarsest model consists of only the root node with a diagonal transformation matrix. As the cut goes downward in the tree, the number of nodes increases, and thus the model becomes finer. The finest model consists of all the leaf nodes with full transformation matrices.

Now we show how to calculate the description length defined in (18) for a tree cut. Denote a cut in a tree by c, and the rth node of this cut by $\mathcal{N}_c^r$. Assuming that there are R_c nodes in cut c, then there are a total of 3^{R_c} models in this cut. Let $\mathcal{T}_c^r$ be a subtree whose root node is $\mathcal{N}_c^r$, and Ω_c^r be the set of Gaussian components that lie in the leaf nodes of $\mathcal{T}_c^r$. For k sequences of feature observations $\underline{o}^k = \{\underline{o}_1, \ldots, \underline{o}_k\}$, with each $\underline{o}_i$ generated by a HMM, the likelihood of observation sequences is approximated by the joint likelihood of the dominant state and mixture index sequences and the observation sequences [38], that is $\log p(\underline{o}^k | \tilde{\eta}^{(k)}, \phi)$ $\approx \log p(\underline{o}^k, \underline{\tilde{s}}^k | \tilde{\eta}^{(k)}, \phi) = \log \max_{\underline{s}^k} p(\underline{o}^k, \underline{s}^k | \tilde{\eta}^{(k)}, \phi)$, where $\underline{\tilde{s}}^k = (\underline{\tilde{x}}^k, \underline{\tilde{z}}^k)$, and $\underline{\tilde{x}}^k, \underline{\tilde{z}}^k$ are the optimal state and mixture index sequences determined by the Viterbi algorithm. Each

feature vector $o_{j,t}$, $j = 1, \ldots, k$, $t = 1, \ldots, T_j$, is assigned to its corresponding Gaussian component indexed by $\tilde{x}_{j,t}$, $\tilde{z}_{j,t}$. Let $\mathcal{O}_c^r$ be the set of feature vectors in $\underline{o}^k$ assigned to the Gaussian components in Ω_c^r. Denote $\underline{A}_c^{(k)}$, $\underline{b}_c^{(k)}$ as the set of transformation parameters corresponding to the cut c, and define the indicator function $\mathcal{I}(A, B) = 1$ if A is true and B is true; and $\mathcal{I}(A, B) = 0$, otherwise. For LR, the description length $\mathcal{L}^c(\underline{o}^k, \underline{A}_c^{(k)}, \underline{b}_c^{(k)})$ for the cut c is approximated as follows.

$$\mathcal{L}^c\left(\underline{o}^k, \underline{A}_c^{(k)}, \underline{b}_c^{(k)}\right) \approx \sum_{r=1}^{R_c} \mathcal{L}\left(\mathcal{N}_c^r, \mathcal{O}_c^r, A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right) \quad (19)$$

with

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{N}_c^r, \mathcal{O}_c^r, A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right) \\
&= -\sum_{o_{j,t} \in \mathcal{O}_c^r} \sum_{(i,m) \in \Omega_c^r} \mathcal{I}(i = \tilde{x}_{j,t}, m = \tilde{z}_{j,t}) \\
&\quad \cdot \left[\left(-\tfrac{1}{2}\left(o_{j,t} - A_{\mathcal{N}_c^r}^{(k)}\mu_{i,m} - b_{\mathcal{N}_c^r}^{(k)}\right)^T \right. \right. \\
&\qquad \left. \cdot \Sigma_{i,m}^{-1}\left(o_{j,t} - A_{\mathcal{N}_c^r}^{(k)}\mu_{i,m} - b_{\mathcal{N}_c^r}^{(k)}\right) \right) \\
&\qquad \left. - \tfrac{1}{2}\log\det(\Sigma_{i,m}) + \log p\left(\left.\left(A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right)\right| \phi\right) \right] \\
&\quad + \tfrac{1}{2}\log\det\left[\sum_{o_{j,t} \in \mathcal{O}_c^r} \sum_{(i,m) \in \Omega_c^r} \mathcal{I}(i = \tilde{x}_{j,t}, m = \tilde{z}_{j,t}) \right. \\
&\qquad \left. \cdot I_O\left(o_{j,t} \left| \left(A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right), \lambda_{i,m}\right.\right) + I_P\left(\left.\left(A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right)\right| \phi\right) \right] \\
&\quad + \tfrac{1}{2}(1 - \log 4)D_{\mathcal{N}_c^r}
\end{aligned} \quad (20)$$

where the approximation is replacing the log determinant of the sum of information matrices as defined by (18) by the sum (over r) of the log determinants of the information matrices.

If the transformation matrix of $\mathcal{N}_c^r$ is full, then $D_{\mathcal{N}_c^r} = n^2 + n$; if the transformation matrix is B block diagonal, then $D_{\mathcal{N}_c^r} = (n^2/B) + n$; if the transformation matrix is diagonal, then $D_{\mathcal{N}_c^r} = 2n$. The entries of information matrix $I_O(o_{j,t}|A_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}, \lambda_{i,m})$ are given as

$$I_{o_{j,t}}(a_{\mathcal{N}_c^r, e, f}, a_{\mathcal{N}_c^r, p, q}, \lambda_{i,m}) = \mu_{i,m,f} r_{i,m,e,p} \mu_{i,m,q}$$

$$I_{o_{j,t}}(a_{\mathcal{N}_c^r, e, f}, b_{\mathcal{N}_c^r, p}, \lambda_{i,m}) = \mu_{i,m,f} r_{i,m,e,p}$$

$$I_{o_{j,t}}(b_{\mathcal{N}_c^r, p}, b_{\mathcal{N}_c^r, q}, \lambda_{i,m}) = r_{i,m,p,q}$$

where $e, f, p, q = 1, \ldots, n$.

For affine transformations, the description length $\mathcal{L}^c(\underline{o}^k, \underline{\hat{A}}_c^{(k)}, \underline{b}_c^{(k)})$ for a cut c can be similarly approximated as

$$\mathcal{L}^c\left(\underline{o}^k, \underline{\hat{A}}_c^{(k)}, \underline{b}_c^{(k)}\right) \approx \sum_{r=1}^{R_c} \mathcal{L}\left(\mathcal{N}_c^r, \mathcal{O}_c^r, \hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right)$$

with

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{N}_c^r, \mathcal{O}_c^r, \hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right) \\
&= -\sum_{o_{j,t} \in \mathcal{O}_c^r} \sum_{(i,m) \in \Omega_c^r} \mathcal{I}(i = \tilde{x}_{j,t}, m = \tilde{z}_{j,t}) \\
&\quad \cdot \left[\log\left(N\left(\hat{A}_{\mathcal{N}_c^r}^{(k)}(o_{j,t} - b_{\mathcal{N}_c^r}^{(k)}) \middle| \mu_{i,m}, \Sigma_{i,m}\right)\right) \right. \\
&\qquad \left. - \tfrac{1}{2}\log\left(\left|\hat{A}_{\mathcal{N}_c^r}^{(k)}\right|^2\right) + \log p\left(\left.\left(\hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right)\right| \phi\right) \right] \\
&\quad + \tfrac{1}{2}\log\det\left[\sum_{o_{j,t} \in \mathcal{O}_c^r} \sum_{(i,m) \in \Omega_c^r} \mathcal{I}(i = \tilde{x}_{j,t}, m = \tilde{z}_{j,t}) \right. \\
&\qquad \left. \cdot I_O\left(o_{j,t} \middle| \hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}, \lambda_{i,m}\right) + I_P\left(\left.\left(\hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}\right)\right| \phi\right) \right] \\
&\quad + \tfrac{1}{2}(1 - \log 4)D_{\mathcal{N}_c^r}
\end{aligned} \quad (21)$$

and the entries of $I_O(o_{j,t}|\hat{A}_{\mathcal{N}_c^r}^{(k)}, b_{\mathcal{N}_c^r}^{(k)}, \lambda_{i,m})$ are given as

$$\begin{aligned}
&I_{o_{j,t}}(\hat{a}_{\mathcal{N}_c^r, e, f}, \hat{a}_{\mathcal{N}_c^r, p, q}, \lambda_{i,m}) \\
&= -a_{\mathcal{N}_c^r, f, p} a_{\mathcal{N}_c^r, q, e} \\
&\quad + (o_{j,t,f} - b_{\mathcal{N}_c^r, f}) r_{i,m,e,p}(o_{j,t,q} - b_{\mathcal{N}_c^r, q})
\end{aligned}$$

$$I_{o_{j,t}}(\hat{a}_{\mathcal{N}_c^r, e, f}, b_{\mathcal{N}_c^r, p}, \lambda_{i,m})$$

$$= \begin{cases}
-\sum_{v=1}^{n} (o_{j,t,f} - b_{\mathcal{N}_c^r, f}) \\
\quad r_{i,m,j,v}\hat{a}_{\mathcal{N}_c^r, v, p} & \text{for } f \neq p \\
-\sum_{v=1}^{n} r_{i,m,e,v}\left(\sum_{w=1}^{n} \hat{a}_{\mathcal{N}_c^r, v, w}\right. \\
\quad ((o_{j,t,w} - b_{\mathcal{N}_c^r, w}) \\
\quad -\mu_{i,m,v} + \hat{a}_{\mathcal{N}_c^r, v, f} \\
\quad \left.(o_{j,t,f} - b_{\mathcal{N}_c^r, f})\right) & \text{for } f = p
\end{cases}$$

$$I_{o_{j,t}}(b_{\mathcal{N}_c^r, p}, b_{\mathcal{N}_c^r, q}, \lambda_{i,m})$$

$$= \sum_{w=1}^{n} \sum_{v=1}^{n} \hat{a}_{\mathcal{N}_c^r, v, p} r_{i,m,v,w} \hat{a}_{\mathcal{N}_c^r, w, q}$$

where $e, f, p, q = 1, \ldots, n$.

Note that (20) and (21) can be calculated in the online mode, since we can accumulate sufficient statistics recursively and plug-in the parameter estimates to obtain the description length values. The difference between the batch and the online schemes is that in batch algorithms, the current parameter estimates are used to decode all adaptation utterances, while in online algorithm, the current parameter estimates are used to decode the current adaptation utterance only, leaving the previous decoding results unchanged.

In general, learning a graphical model is NP-hard [11]. For a tree-structured model, we could in principle calculate model description length for every possible tree cut and take the model
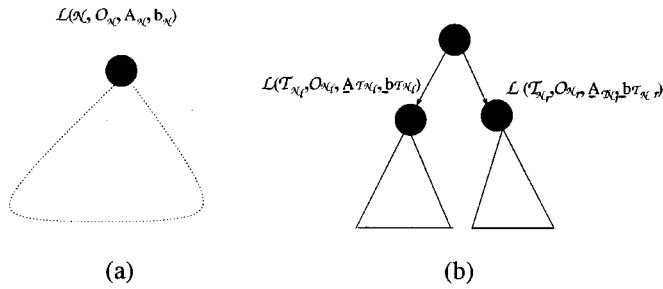
Fig. 3. Dynamic programming for optimal tree cut and model. The dashed curve means tying all Gaussian components together. The solid curve means taking subproblem solution.

with the MDL. However, since the number of cuts in a binary tree is exponential in the size of the tree, i.e., $O(2^{2^{d-1}})$, with d the number of layers, it is impractical to do so. An efficient dynamic programming algorithm is therefore proposed for determining the optimal size of hierarchical tree and the form of transformation matrices. Denote an internal node in the tree $\mathcal{T}$ as $\mathcal{N}$ and its left child node $\mathcal{N}_l$ and right child node $\mathcal{N}_r$. Denote the subtrees rooted at these nodes as $\mathcal{T}_\mathcal{N}$, $\mathcal{T}_{\mathcal{N}_l}$, and $\mathcal{T}_{\mathcal{N}_r}$, respectively, and denote the set of feature vectors clustered in the Gaussians of these nodes as $\mathcal{O}_\mathcal{N}$, $\mathcal{O}_{\mathcal{N}_l}$, and $\mathcal{O}_{\mathcal{N}_r}$, respectively. Finally, denote the optimal transformation matrices and biases for the subtrees rooted at these nodes as $\underline{A}_{\mathcal{T}_\mathcal{N}}$, $\underline{A}_{\mathcal{T}_{\mathcal{N}_l}}$, $\underline{A}_{\mathcal{T}_{\mathcal{N}_r}}$ and $\underline{b}_{\mathcal{T}_\mathcal{N}}$, $\underline{b}_{\mathcal{T}_{\mathcal{N}_l}}$, $\underline{b}_{\mathcal{T}_{\mathcal{N}_r}}$, respectively, with the corresponding MDLs as $\mathcal{L}(\mathcal{T}_\mathcal{N}, \mathcal{O}_\mathcal{N}, \underline{A}_{\mathcal{T}_\mathcal{N}}, \underline{b}_{\mathcal{T}_\mathcal{N}})$, $\mathcal{L}(\mathcal{T}_{\mathcal{N}_l}, \mathcal{O}_{\mathcal{N}_l}, \underline{A}_{\mathcal{T}_{\mathcal{N}_l}}, \underline{b}_{\mathcal{T}_{\mathcal{N}_l}})$, and $\mathcal{L}(\mathcal{T}_{\mathcal{N}_r}, \mathcal{O}_{\mathcal{N}_r}, \underline{A}_{\mathcal{T}_{\mathcal{N}_r}}, \underline{b}_{\mathcal{T}_{\mathcal{N}_r}})$. The recursive formula for the MDL is given as

$$
\mathcal{L}\left(\mathcal{T}_\mathcal{N}, \mathcal{O}_\mathcal{N}, \underline{A}_{\mathcal{T}_\mathcal{N}}, \underline{b}_{\mathcal{T}_\mathcal{N}}\right)
= \min \begin{cases} \mathcal{L}(\mathcal{N}, \mathcal{O}_\mathcal{N}, A_\mathcal{N}, b_\mathcal{N}), \\ \mathcal{L}\left(\mathcal{T}_{\mathcal{N}_l}, \mathcal{O}_{\mathcal{N}_l}, \underline{A}_{\mathcal{T}_{\mathcal{N}_l}}, \underline{b}_{\mathcal{T}_{\mathcal{N}_l}}\right) \\ \quad + \mathcal{L}\left(\mathcal{T}_{\mathcal{N}_r}, \mathcal{O}_{\mathcal{N}_r}, \underline{A}_{\mathcal{T}_{\mathcal{N}_r}}, \underline{b}_{\mathcal{T}_{\mathcal{N}_r}}\right) \end{cases} \tag{22}
$$

where each transformation matrix can be full, block diagonal, or diagonal. See Fig. 3 for an illustration of (22).

Either a *bottom-up* dynamic programming algorithm or a *top-down* recursive algorithm can be performed to obtain the MDL. In general, the *bottom-up* approach is more efficient than the *top-down* approach [12]. However, for this problem, the two approaches have the same computational complexity. Li *et al.* [35] used the *top-down* recursive algorithm to calculate the MDL in the MLE sense for generalizing case frames in natural language processing. Shinoda *et al.* [42] used the *top-down* recursive algorithm to calculate the MDL in the MLE sense for bias removal in batch-mode model selection. In this work we use the *bottom-up* approach for online mode model selection. The procedure is summarized as follows:

- Obtain diagonal, block diagonal, and full transformation matrix parameters and initialize the tree cut using block diagonal transformation matrices by the *bottom-up top-down* procedure described in Section IV.

- Iterate through the following steps until convergence.
  - Step 1) Transform HMM parameters.
  - Step 2) Decode current adaptation utterance by using the transformed HMM parameters. Assign each feature vector into a Gaussian component in a leaf node in the tree.
  - Step 3) Determine the optimal tree cut and transformation matrix form of each node in the cut by the following *bottom-up* dynamic programming algorithm:
    - i) Initialization: for each leaf node, if the node contains feature vector, then calculate its description lengths with full, block diagonal, and diagonal matrices by (20) or (21), and choose the minimum; otherwise, set its description length to be infinity.
    - ii) Recursion: in the *bottom-up* order, recursively compute the description length using (22). For each nonleaf node $\mathcal{N}$, if the MDL of the first line of (22) is smaller than or equal to the MDL of the second line, drop all the children of the node $\mathcal{N}$ and assign node $\mathcal{N}$ to the cut; else assign the children of the node $\mathcal{N}$ to the cut.
    - iii) Go up one level and repeat ii) until the root is reached.
  - Step 4) If the tree cut and the transformation matrix form of each node in the cut remain the same as in the previous iteration, then output the transformed HMM parameters and stop. Otherwise, return to Step 1.

VI. Experimental Results

The online Bayesian learning approach is applied to online speaker adaptation using a vocabulary of 26-letter English alphabet. Two severely mismatched speech databases, the OGI ISOLET and the TI46, were used for evaluating the adaptation algorithm. A full description of these two corpora can be found in [26]. For speaker independent training, the OGI ISOLET database was used. It consists of 150 speakers, each speaking a letter twice. For online Bayesian adaptation and testing, the English alphabet subset of the TI46 isolated word corpus was used. Among the 16 talkers, data from four males were incomplete. Therefore, only 12 speakers (four males and eight females) were used in this study. Each person uttered each of the letters 26 times, where ten were collected in the same session and the remaining 16 were collected in eight different sessions with two in each session. For each person and each letter, we divided equally those 16 tokens collected in eight different sessions into two parts, one for adaptive training, another for testing. Each letter was modeled by a single left-to-right five-state CDHMM. Each state had a Gaussian mixture density of four components with each component density having a diagonal covariance matrix. Speech features were extracted based
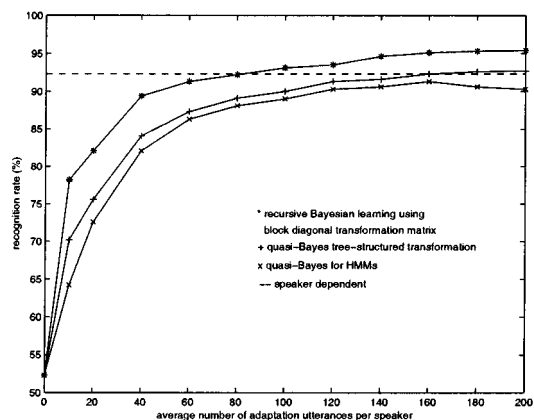
Fig. 4. Recognition performance by supervised online Bayesian learning.

Fig. 5. Recognition performance by supervised and unsupervised online Bayesian learning.

on a tenth-order LPC analysis, where the feature components were 12 cepstral coefficients, a normalized log energy and their first time derivatives.

A number of comparative experiments were conducted, including

1) the proposed online recursive Bayesian approach versus other online Bayesian approaches;
2) supervised versus unsupervised adaptation;
3) generalized Gaussian prior with different shape parameters γ;
4) diagonal, block diagonal (two blocks, one for instantaneous features and one for dynamic features), and full transformation matrices;
5) different depths of hierarchical tree;
6) LR versus affine transformation;
7) models selected by the Bayesian variant of MDL principle versus two models with fixed model complexity.

Recognition performances were evaluated by using parameter estimates of four iterations of (5) for each k. Hyperparameters were estimated by four iterations of the modified equation (2), where the step size was obtained by line search. By default, a six-layered hierarchical tree and a Gaussian prior were used in the experiments.

1) Comparison of Online Bayesian Adaptation Approaches: Starting with the SI models, we selected adaptation tokens for each letter randomly from the adaptation set and performed utterance-based supervised online adaptation. After each adaptation, we tested the recognizer on the test set to measure the performance. In Fig. 4, recognition results were averaged over 12 speakers as a function of total number of adaptation tokens per speaker. For recursive Bayesian estimation, we used affine transformation with block diagonal transformation matrices. We also included the results of quasi-Bayes online estimation for tree-structured transformation [7] and quasi-Bayes online estimation for HMMs [26], where in the latter case, full sets of HMM parameters were adapted and the hyperparameters were estimated by the method of moments. Our result is shown better than those of quasi-Bayes approaches, the main reason being that the recursive Bayesian learning technique accommodates easily affine transformations which when combined with the hierarchical tree is effective for both limited adaptation data and abundant adaptation data.
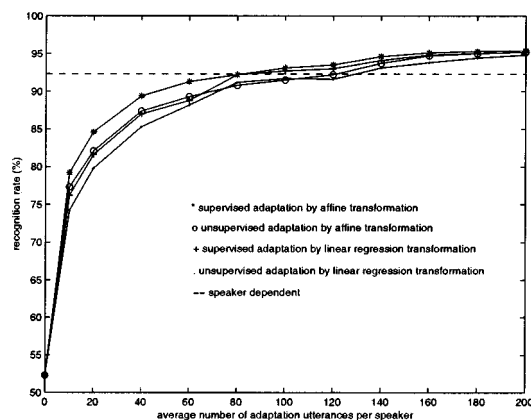
2) Comparison of Supervised and Unsupervised Adaptations: We evaluated the recognition performances when using supervised and unsupervised adaptation, respectively. Fig. 5 shows the results when block diagonal matrix of affine transformation and hierarchical tree with a depth of six were used. Supervised adaptation performed slightly better. As abundant adaptation data became available, the two results became very close.

3) Robustness of Priors: We investigated the robustness of generalized Gaussian prior models. The transformation matrices were block diagonal matrix and the depth of hierarchical tree was six. The recognition performances by using three values of the shape parameter $\gamma = 2, 1, 0.7$ are shown in Fig. 6. As the results indicate, recognition performance was improved when using *priors* with heavy tails than using standard Gaussian, since heavy-tailed *priors* are more robust to data and function mismatch. There was little difference between the results for $\gamma = 1$ and $\gamma = 0.7$.

4) Full, Block Diagonal, and Diagonal Transformation Matrices: Previously, Leggetter and Woodland [34] investigated the effect of full or diagonal transformation matrices on recognition performance in ML-based LR using a batch adaptation approach. Here, we investigate the effectiveness of online Bayesian learning of tree-structured affine and LR transformations. We evaluated the effect of model parameterization complexity on recognition performance by using full, block diagonal, as well as diagonal transformation matrices while model structural complexity was varied by using different depths of hierarchical tree. The amount of adaptation data was ten utterances per speaker in one case and 100 in another.

Fig. 7 illustrates the recognition performance when using the small amount of adaptation data. All transformations provided improvements over the initial model, but the effect of diagonal matrices was limited. The full matrices gave a substantial improvement when using one- or three-layered trees. However, as the depth of the tree was increased, the amount of data allocated to each leaf node became small and the matrices were poorly estimated, and thus the performance of full matrices dropped rapidly. With diagonal matrices, as deeper trees were used the performance gradually increased; however, this effect was very small and using 500 diagonal matrices was only 5.0% better than using one diagonal matrix. It is clear that the off-diagonal terms
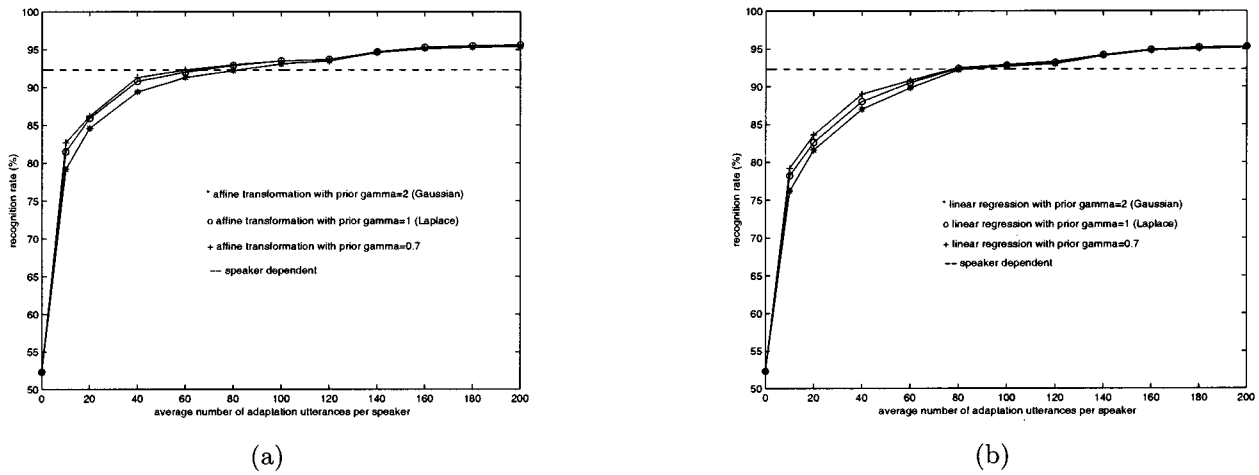
Fig. 6.   Recognition performance by GGD *priors*: (a) affine transformation and (b) LR transformation.
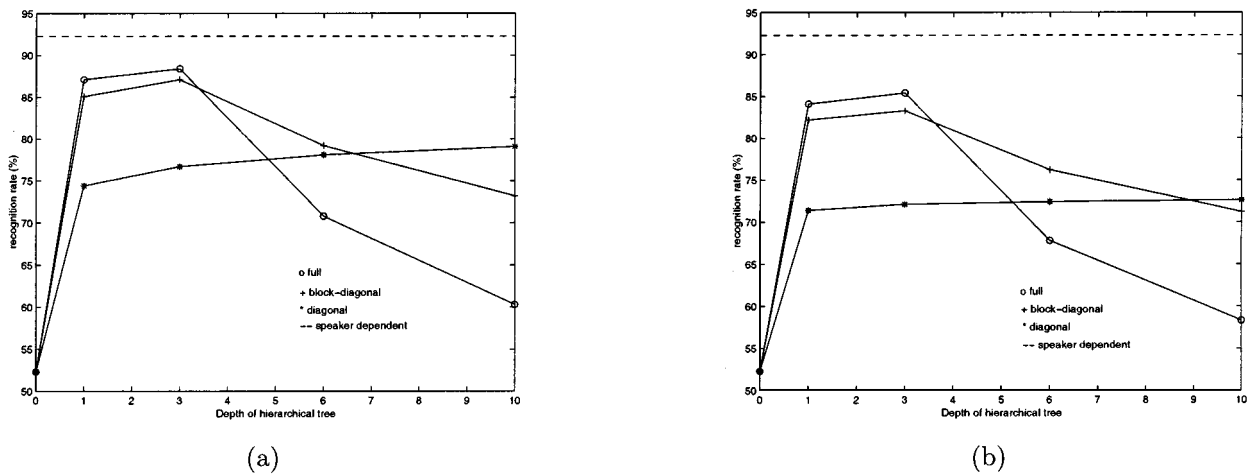

Fig. 7.   Full, block-diagonal, and diagonal matrix using ten adaptation utterances per speaker: (a) affine transformation and (b) LR transformation.
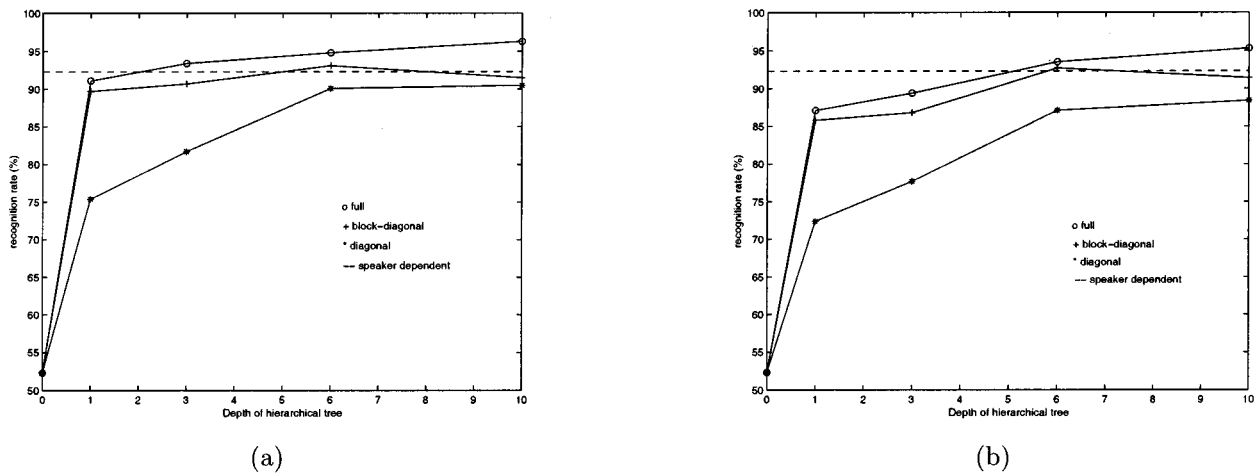

Fig. 8.   Full, block-diagonal, and diagonal matrix using 100 adaptation utterances per speaker: (a) affine transformation and (b) LR transformation.

that account for the interdependencies between feature components were important.

Fig. 8 gives the recognition performance when using the large amount of adaptation data. In this case, the amounts of data allocated to leaf nodes were abundant and the transformation matrices were well estimated. As expected, full transformation matrices gave substantially better performance than that of diagonal transformation matrices.

5) Depth of Hierarchical Tree: We investigated the effects of using hierarchical trees with different depths, i.e., the problem of *model structure complexity*. Previously, Shinoda and Watanabe [42] investigated this problem for bias trans-
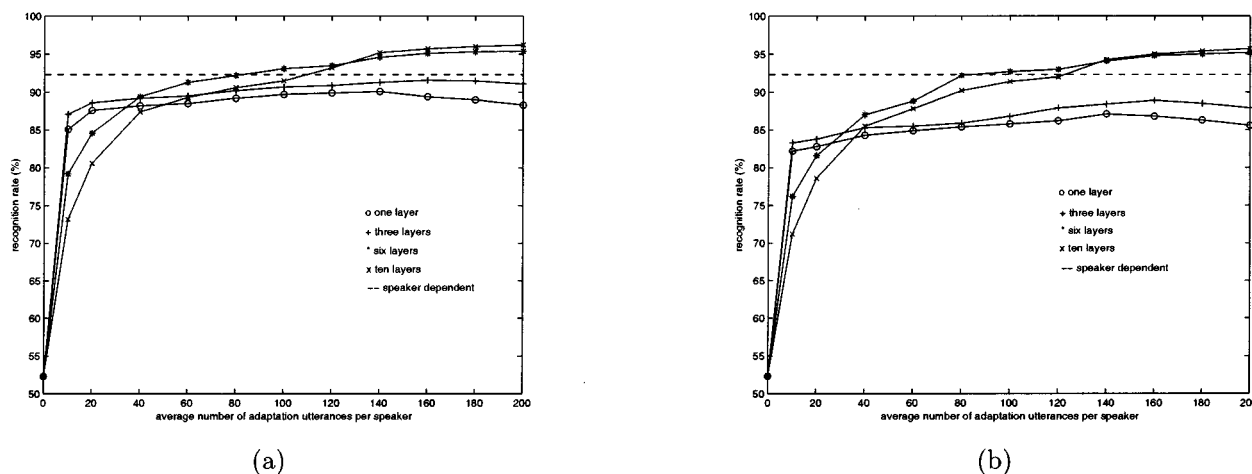
(a)   (b)

Fig. 9. Recognition performance by various depths: (a) affine transformation and (b) LR transformation.

formation by using batch approach. Here we investigate this problem for online Bayesian learning of tree-structured affine transformation parameters. Fig. 9 gives the recognition results of online learning of block-diagonal transformation matrices with various tree depths while varying the number of adaptation utterances. We observe that when the amount of adaptation data was small, better recognition performances were achieved by trees with shallower depths than trees with deeper depths, since a deeper tree-structure overfitted the limited training data. As more adaptation data was presented, the performance gradually improved for both shallow and deep trees, but the trees with shallow depths improved less, which indicates underfitting the adaptation data. As a sufficient amount of data was used, good recognition performance was achieved by trees with deeper depths. From the results, we see that a critical issue in the hierarchical tree-structured speaker adaptation is choosing a tree which neither underfits nor overfits the adaptation data.

6) Linear Regression and Affine Transformation: From the above results, it can be seen that both LR and affine transformation significantly improved recognition performance. On average, LR transformation gave a 80% recognition rate, and affine transformation gave a further 2–8% increase in recognition rate over LR. It is worth noting that the results by affine transformation were consistently better than LR at little additional cost in computation.

7) Bayesian Model Selection: We first examined the optimal model complexities for each given amount of adaptation data by using the MDL principle. Fig. 10 shows the number of tree nodes of the optimal cut for one speaker when the amount of adaptation data was 10, 60, and 180 utterances, respectively. As the amount of adaptation data increased, the optimal cut approached the leaf nodes and the number of nodes increased.

We then compared the recognition results obtained using the proposed MDL method with those obtained by fixed model complexity. The results are summarized in Fig. 11. Of the two fixed models, one is a ten-layer tree with block diagonal transformation matrices and the other is a three-layer tree with full transformation matrices. As shown in the figure, the proposed MDL method possesses the advantage of the small-sized tree when the amount of adaptation data is small, and it possesses the advantage of the large-sized tree when the amount of adaptation data is large. In addition, it outperforms both fixed
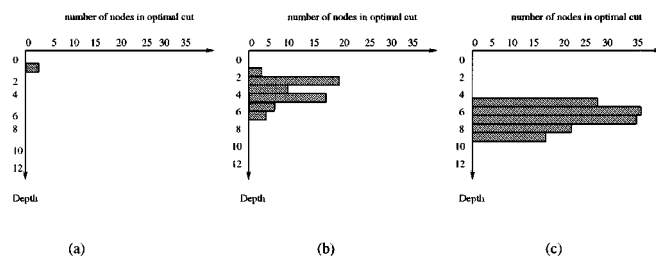

Fig. 10. Model complexity of the optimal model: (a) ten utterances, (b) 60 utterances, and (c) 100 utterances per speaker.

models in the intermediate range of adaptation data size, which is intuitively appealing since this is the condition that MDL provides solution to the uncertainty of model complexity. Therefore, MDL provides an optimal tradeoff between accuracy and complexity of model structure and parameterization over a full range of adaptation data size.

VII. DISCUSSION AND CONCLUSION

In this paper, we developed an online Bayesian learning technique for transformation of parameters of Gaussian densities of HMMs. A hierarchical tree of HMM Gaussian parameters is employed to dynamically control the transformation tying and the *a priori* knowledge of HMM Gaussian parameters is incorporated to estimate the transformation parameters at tree nodes. This technique allows updating parameter estimates per utterance and it can accommodate flexible forms of transformation functions and *a priori* probability densities of the transformation parameters. Speaker adaptation experiments showed consistently improved recognition accuracy for increasing amounts of adaptation data, and the performance was superior to existing online adaptation methods. We investigated the choice on *a priori* density functions and suggested using GGD as priors for online Bayesian learning of tree-structured transformation of Gaussian densities of HMMs. It was found that heavy tailed *a priori* density functions gave better recognition performance and thus are more robust to mismatch in prior estimation. Finally, the Bayesian variant of the MDL principle was incorporated into online Bayesian tree-structured transformation to obtain an optimal tradeoff between model structural and parame-
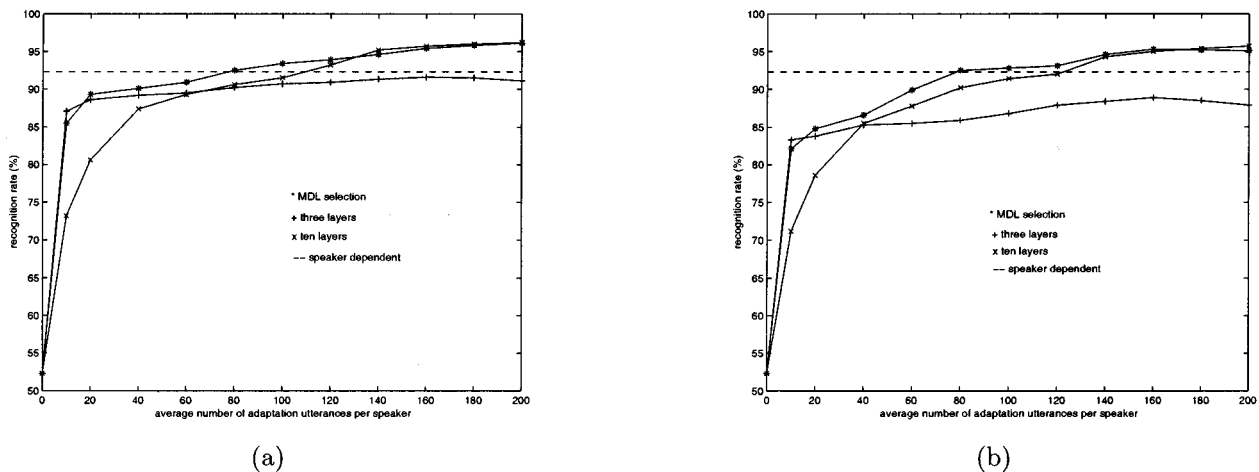
Fig. 11. Recognition performance by MDL principle: (a) affine transformation and (b) LR transformation.

terization complexities and goodness of fit to adaptation data. Experimental results show that the MDL method in general is capable of automatically selecting a set of model parameters that leads to best recognition performance for each given amount of adaptation data.

A disadvantage of the proposed method is its high computational complexity due to the computation of inverse matrix H in (5). For an n-dimensional mean vector, the full transformation A is an $n \times n$ matrix, and the shift vector b is an $n \times 1$ vector. Thus there are $n^2 + n$ parameters, and the information matrix will be $(n^2 + n) \times (n^2 + n)$. A fast algorithm for matrix inversion is $\mathcal{O}(D^3)$ for $D \times D$ matrix, so the proposed method needs $\mathcal{O}(n^6)$ operations for each k, while an incremental EM algorithm for MAPLR [5], [10] needs $\mathcal{O}(n^3)$ operations for each k.

One direction to further improve the performance of our adaptation technique is to incorporate *tree-structural learning* during the online learning process, that is, to update the clustering of Gaussian components in the hierarchical tree after each or several updates of parameter estimates. This problem has been investigated by Chien [6] in a batch algorithm. Another direction is using the *sequential hypothesis testing* technique [40] as a verification scheme for evaluating the transformation reliability, which was shown very useful for unsupervised speaker adaptation [9].

In designing a learning algorithm, it is often important to know the amount of samples needed for training a model, such that for new testing data, one has certain confidence that the probability of making an error is under certain level. This problem is known as *sample complexity* [14] and it is often formulated in a probably approximately correct (PAC) sense of learning [29]. Upper and lower bounds on sample complexity can be derived in the PAC framework for a learning algorithm, and they are often distribution free, i.e., valid regardless of the distribution from which the training data is drawn. Given independently observed feature sequences generated by HMMs, how to derive upper and lower bounds of sample complexity for the proposed online Bayesian learning for speaker adaptation is a challenging problem.

Batch algorithm which follows the same idea used in online algorithm can be similarly derived. An interesting and impor-

tant problem is the comparison of online and batch Bayesian learning algorithms. For a fixed parameter model, the recognition result obtained by the batch learning algorithm is often better than that by the online algorithm, since the online algorithm sees one example at a time and incurs a loss on each sample at its current model. If the loss is defined to be the recognition error or the normalized negative *a posteriori* likelihood function, it is then interesting to derive the upper and lower bounds for the relative loss, which relate the online loss to the best off-line loss.

REFERENCES

[1] S. Ahadi and P. 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. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 3, pp. 2033–2045, Dec. 1990.

[3] J. Bilmes, "Natural statistical models for automatic speech recognition," Ph.D. dissertation, Univ. California, Berkeley, 1999.

[4] W. Byrne and A. Gunawardana, "Comments on efficient training algorithms for HMMs using incremental estimation," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 751–754, Nov. 2000.

[5] C. Chesta, O. Siohan, and C. Lee, "Maximum *a posteriori* linear regression for hidden Markov model adaptation," in *Proc. EUROSPEECH*, Hungary, 1999, pp. 211–214.

[6] J. Chien, "Hybrid adaptation of tree structure and hidden Markov models for robust speech recognition," in *IEEE Workshop Robust Methods for Speech Recognition in Adverse Conditions*, Finland, 1999.

[7] ——, "Online hierarchical transformation of hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 656–668, Nov. 1999.

[8] J. Chien, C. Lee, and H. Wang, "A hybrid algorithm for speaker adaptation using MAP transformation and adaptation," *IEEE Signal Processing Lett.*, vol. 4, pp. 167–169, June 1997.

[9] J. Chien and J. Junqua, "Unsupervised hierarchical adaptation using reliable selection of cluster-dependent parameters," *Speech Commun.*, vol. 30, pp. 235–253, 2000.

[10] W. Chou, "Maximum *a posterior* linear regression with elliptically symmetric matrix variate *priors*," in *Proc. EUROSPEECH*, Hungary, 1999, pp. 1–4.

[11] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artif. Intell.*, vol. 42, pp. 393–405, 1990.

[12] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

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

[14] S. Dasgupta, "The sample complexity of learning fixed-structure Bayesian networks," *Mach. Learn.*, vol. 29, pp. 165–180, 1997.

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

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

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

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

[19] V. Digalakis, *et al.*, "Rapid speech recognizer adaptation to new speakers," in *Proc. ICASSP'99*, pp. 765–768.

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

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

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

[23] Y. Gotoh, M. Hochberg, and H. Silverman, "Efficient training algorithms for HMMs using incremental estimation," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 291–298, Nov. 1998.

[24] A. Gunawardana and W. Byrne, "Convergence of EM variants," Johns Hopkins Univ., Baltimore, MD, CLSP Res. Note, no. 32, 1999.

[25] X. Huang and M. Jack, "Semi-continuous hidden Markov models for speech signal," *Comput. Speech Lang.*, vol. 3, no. 3, pp. 239–251, 1989.

[26] Q. Huo and C. Lee, "Online 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.

[27] Q. Huo and B. Ma, "Online adaptive learning of CDHMM parameters based on multiple-stream *a priori* evolution and posterior pooling," in *Proc. EUROSPEECH*, Hungary, 1999, pp. 2721–2724.

[28] A. Kannan and Ostendorf, "Modeling parameter dependence in speaker adaptation using multiscale tree processes," IEEE Trans. Speech Audio Processing, vol. 6, Oct. 1998, submitted for publication.

[29] M. Kearns and U. Vazirani, *Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press, 1994.

[30] A. Lanterman, "Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model order estimation," Int. Stat. Rev., 2000, to be published.

[31] M. Lasry and R. Stern, "*A posteriori* estimation of correlated jointly Gaussian mean vectors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 530–535, July 1984.

[32] C. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Commun.*, vol. 25, pp. 29–47, 1998.

[33] C. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proc. IEEE*, vol. 88, pp. 1241–1269, Aug. 2000.

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

[35] H. Li and N. Abe, "Generalizing case frames using a thesaurus and the MDL principle," *Comput. Linguistics*, vol. 24, no. 2, pp. 217–244, 1998.

[36] D. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.

[37] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[38] N. Merhav and Y. Ephraim, "Hidden Markov modeling using a dominant state sequence with application to speech recognition," *Comput. Speech Lang.*, vol. 5, pp. 327–339, 1991.

[39] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed. Norwell, MA: Kluwer, 1998, pp. 355–368.

[40] V. Poor, *Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1994.

[41] B. Shahshahani, "A Markov random field approach to Bayesian speaker adaptation," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 183–191, Mar. 1997.

[42] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous model complexity control by MDL principle," in *Proc. ICASSP'96*, pp. 717–720.

[43] K. Shinoda and C. Lee, "Structural MAP speaker adaptation using hierarchical *priors*," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 1997, pp. 381–387.

[44] O. Siohan, C. Chesta, and C. Lee, "Joint maximum *a posterior* adaptation of transformation and hidden Markov model parameters," in *Proc. ICASSP'00*.

[45] A. Surendran, C. Lee, and M. Rahim, "Nonlinear compensation for stochastic matching," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 643–655, Nov. 1999.

[46] D. Titterington, "Recursive parameter estimation using incomplete data," *J. R. Statist. Soc. B*, vol. 46, pp. 257–267, 1984.

[47] S. Wang, "Statistical recursive estimation algorithms for speaker adaptation," Ph.D. dissertation, Univ. Illinois, Urbana-Champaign, 2000.

[48] G. Zavaliagkos, R. Schwartz, and J. Makhoul, "Batch, incremental and instantaneous adaptation techniques for speech recognition," in *Proc. ICASSP'95*, pp. 676–679.

[49] Y. Zhao, "A speaker-independent continuous speech recognition system using continuous mixture Gaussian density HMM of phoneme-sized units," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 345–361, July 1993.

[50] ——, "Self-learning speaker and channel adaptation based on spectral variation source decomposition," *Speech Commun.*, vol. 18, pp. 65–77, 1996.

[51] G. Zweig and S. Russell, "Probabilistic modeling with Bayesian networks for automatic speech recognition," *Aust. J. Intell. Inform. Process. Syst.*, vol. 5, no. 4, pp. 253–260, 1999.

Shaojun Wang received the B.E. and M.E. degrees in electric power and energy systems in electrical engineering from Tsinghua University, Beijing, China, in 1988 and 1992, respectively, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, in 1998 and 2000, respectively.

He is now a Postdoctoral Fellow in the Center for Automated Learning and Discovery, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. His research interests include speech signal processing and recognition, statistical language modeling and text processing, statistical learning algorithms, information retrieval, data mining, multimedia and human–machine communications, and network economics.

Yunxin Zhao (S'86–M'88–SM'94) received the Ph.D. degree from the University of Washington, Seattle, in 1988.

She was Senior Research Staff and Project Leader of the Speech Technology Laboratory, Panasonic Technologies, Inc., from 1988 to 1994. She was Assistant Professor of the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, from 1994 to 1998. She is currently Associate Professor of the Department of Computer Engineering and Computer Science, University of Missouri, Columbia. Her research interests are in spoken language processing, automatic speech recognition, multimedia interface, multimodal human–computer interaction, statistical pattern recognition, blind system identification and estimation, speech and signal processing, and biomedical applications.

Dr. Zhao was Associate Editor of IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING and is a Member of IEEE Speech Technical Committee. She received the 1995 NSF Career Award, and is listed in *American Men and Women of Science*, February 1998.