Available online at www.sciencedirect.com
SCIENCE DIRECT®
Speech Communication 45 (2005) 435–454
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Generative factor analyzed HMM for automatic speech recognition

Kaisheng Yao [a,*], Kuldip K. Paliwal [b], Te-Won Lee [a]

[a] *Institute for Neural Computation, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0523, USA*
[b] *School of Microelectronic Engineering, Griffith University, Brisbane, Queensland 4111, Australia*

Received 10 November 2003; received in revised form 6 November 2004; accepted 4 January 2005

Abstract

We present a generative factor analyzed hidden Markov model (GFA-HMM) for automatic speech recognition. In a standard HMM, observation vectors are represented by mixture of Gaussians (MoG) that are dependent on discrete-valued hidden state sequence. The GFA-HMM introduces a hierarchy of continuous-valued latent representation of observation vectors, where latent vectors in one level are acoustic-unit dependent and latent vectors in a higher level are acoustic-unit independent. An expectation maximization (EM) algorithm is derived for maximum likelihood estimation of the model.

We show through a set of experiments to verify the potential of the GFA-HMM as an alternative acoustic modeling technique. In one experiment, by varying the latent dimension and the number of mixture components in the latent spaces, the GFA-HMM attained more compact representation than the standard HMM. In other experiments with varies noise types and speaking styles, the GFA-HMM was able to have (statistically significant) improvement with respect to the standard HMM.

Keywords: Hidden Markov models; Factor analysis; Mixture of Gaussian; Speech recognition; Expectation maximization algorithm

1. Introduction

In the automatic speech recognition (ASR) problem, one is presented with multi-dimensional data with D^y dimension where it is assumed that the data is generated from acoustic sources that are modeled as discrete state q in a hidden Markov model (HMM) (Rabiner and Juang, 1993). The transition of the states is assumed to encode the transition of the speech unit and the content of the uttered speech can be inferred by the well-known Viterbi algorithm (Viterbi, 1967). The task in speech modeling for ASR within the

* Corresponding author. Tel.: +1 858 822 2720; fax: +1 858 565 7440.
E-mail addresses: kyao@ti.com (K. Yao), k.paliwal@me.gu.edu.au (K.K. Paliwal), tewon@ucsd.edu (T.-W. Lee).

0167-6393/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2005.01.002

HMM framework is to obtain a compact and accurate model of the observations. However, this is a hard problem, since the observation vector is high dimensional and the elements in the observation vector contain second as well as higher order statistical information. Traditional approaches in modeling speech observations in an HMM make use of mixture of Gaussians (MoG) with usually a diagonal covariance matrix in each state, which implicitly models the intra-frame correlations.

Despite its pattern recognition appearance, the speech model in an HMM can be viewed in statistics as a latent representation. In particular, the discrete state q is the discrete latent representation of the speech unit and the discrete Gaussian index m in the MoG is the discrete latent representation of the density in that state. In this context, it is therefore natural to describe the D^y dimensional observation vector $\boldsymbol{y}(t)$ at time t as correlated in terms of a smaller set of D^x dimensional continuous-valued latent vector $\boldsymbol{x}(t)$. In this case, the most straightforward description of the continuous-valued latent representation of $\boldsymbol{y}(t)$ is given by the following linear model

$$y_n(t) = \sum_{l=1}^{D^x} \Lambda_{nl} x_l(t) + v_n(t), \quad n = 1, \ldots, D^y, \qquad (1)$$

where $y_n(t)$ denotes the nth element in vector $y(t)$ at time t. The $y_n(t)$ depends on linear combination of elements in $\boldsymbol{x}(t)$ with matrix $\boldsymbol{\Lambda} = [\Lambda_{nl}]_{D^y \times D^x}$. The density of $\boldsymbol{y}(t)$ is also related to the D^y-dimensional noise $\boldsymbol{v}(t)$ with element $v_n(t)$. Note that the problem in Eq. (1) is general, since without certain constraints imposed on the model, the solution is non-trivial.

In the context of the continuous-valued latent representation, Eq. (1) presents solutions with different physical meanings depending on the different constraints on the model (Roweis and Ghahramani, 1999; Frey, 1999). In independent component analysis (ICA) (Comon, 1994; Bell and Sejnowski, 1995) the constraints are as follows: (1) $v_n(t) = 0$; i.e., no distortions in observation (no additive noise) $\boldsymbol{y}(t)$, (2) element $x_l(t)$ in $\boldsymbol{x}(t)$ is independent from each other, and (3) At most one element $x_l(t)$ is Gaussian distributed or $x_l(t)$ has usually a non-Gaussian density. Maximum likelihood estimation of $\boldsymbol{\Lambda}$ leads to the ICA solution (Pearlmutter and Parra, 1997). Another interesting related model is independent factor analysis (IFA) (Attias, 1998). IFA is obtained by the following constraints: (1) element of $\boldsymbol{x}(t)$, $x_l(t)$, is independent and distributed as non-Gaussian density, (2) $\boldsymbol{v}(t)$ is distributed as diagonal Gaussian density.

Though advanced algorithms (Bell and Sejnowski, 1995; Attias, 1998; Amari et al., 2000; Cardoso, 1997; Hyvarinen et al., 2001) have been derived for signal processing within the framework of continuous-valued latent representation, there are few works applied to ASR. One reason is that the MoG can approximate any observation vector distribution given a sufficient number of model parameters and enough training data. Thus, by increasing the amount of training data and/or increasing number of model parameters, speech models by MoGs in HMMs can reach high recognition accuracy for input speech. Due to this claim, one might expect that the above continuous-valued latent representation may not be useful in ASR. However, there are several important differences in speech recognition research. Firstly, the number of parameters in the model and the amount of training data increase monotonically in order to achieve an improved performance. Secondly, the larger the number of parameters in a model, the larger the amount of training data is needed in order to have accurate estimation of the parameters. Thirdly, given the amount of training data, even when the number of parameters is increased, the performance of the model can easily reach a saturation point. These observations could give rise to problems in spontaneous speech recognition since the amount of training data is not sufficient for a reliable estimation of all acoustic units. Some heuristically justified approaches have been applied to address the above problems, for example, the method of parameter-tying (Bellegarda and Nahamoo, 1990). But parameter-tying has its own drawbacks since it considerably requires some artistry to design the way to share parameters.

Continuous-valued latent representation can be useful for modeling speech in a compact way. Note

that, with the linear model in Eq. (1), density of $\boldsymbol{y}(t)$ can be modeled by density of $\boldsymbol{x}(t)$, density of $\boldsymbol{v}(t)$, and parameter $\boldsymbol{\Lambda}$ (Rubin and Thayer, 1982). This motivates the recent application of factor analysis for speech recognition (Saul and Rahim, 2000). In this case, $\boldsymbol{x}(t)$ is distributed as $\mathcal{N}(\boldsymbol{x}(t);\boldsymbol{0},\boldsymbol{I})$. However, factor analysis can only explicitly model the correlations of elements in observation vectors. To model higher order statistics of the observation vectors explicitly, the latent vector $\boldsymbol{x}(t)$ has to be non-Gaussian as suggested by ICA (Comon, 1994) and IFA (Attias, 1998).

In this paper, we present a novel speech model for automatic speech recognition. The key to our approach lies in the introduction of a hierarchical continuous-valued latent representation. Observation vector $\boldsymbol{y}(t)$ is correlated with a *model dependent* continuous-valued latent vector $\boldsymbol{x}(t)$. Since elements in vector $\boldsymbol{x}(t)$ depend on the same acoustic unit, the elements have correlations. The correlations can be compactly represented by another continuous-valued latent representation $\boldsymbol{z}(t)$, which is independent of the acoustic model. In this paper, the $\boldsymbol{z}(t)$ is distributed as a standard Gaussian $\mathcal{N}(\boldsymbol{z}(t);\boldsymbol{0},\boldsymbol{I})$. Noise in the observation vector $\boldsymbol{y}(t)$ is modeled as a MoG and it depends on the state q_t at time t. This model is called a generative factor analyzed HMM (GFA-HMM), described schematically in Fig. 1. The model gives a more compact representation of intra-frame statistics than the standard HMM, and it shows improved performances over the standard HMM given the same amount of training data.

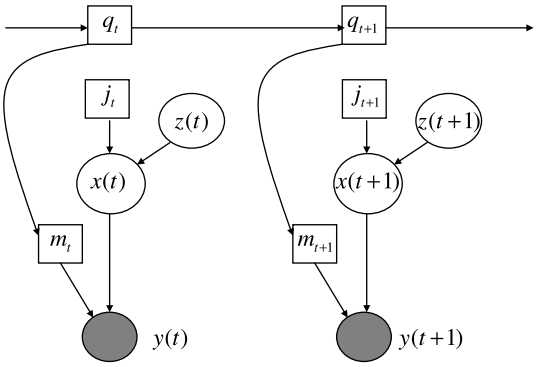

Fig. 1. Graphical model representation of FA-HMM.

The originality of this new model can be viewed from the following points. First, compared to the standard HMM, which represents observation vectors as dependent solely on discrete states/mixtures, the GFA-HMM has another continuous-valued latent representation of the observation vectors. The latent vector in the continuous-valued latent representation can be simply modeled by a standard diagonal Gaussian $\mathcal{N}(\boldsymbol{x}(t);\boldsymbol{0},\boldsymbol{I})$, which reduces the model to HMMs with factorized covariance matrix (Saul and Rahim, 2000). When the continuous-valued latent vector $\boldsymbol{x}(t)$ are distributed as MoG, the model reduces to factor-analyzed HMM (FA-HMM) proposed in (Rosti and Gales, 2002). Second, compared to ICA (Comon, 1994; Bell and Sejnowski, 1995) and IFA (Attias, 1998), the continuous-valued latent vectors in the GFA-HMM is dynamic; i.e., the latent representation is dependent on state q_t at time t. This construction of the model makes much sense in speech recognition, since speech signals are non-stationary and the latent representation should be acoustic-unit dependent.

1.1. Notation

Vectors are denoted by bold-faced lower-case letters and matrices are denoted by bold-faced upper-case letters. Elements of vectors and matrices are not bold-faced. Time index is in the parenthesis of vectors, matrices, or elements. Superscript T denotes transpose. For example, element $x_i(t)$ is the ith element in vector $\boldsymbol{x}(t)$ at time t. Λ_{nl} is the element at cross of nth row and lth column in matrix $\boldsymbol{\Lambda}$.

Sequence is denoted by (,). Set is denoted as {,}. The multi-variable Gaussian distribution for a vector $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted as

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{|\det(2\pi)^{D^x}\boldsymbol{\Sigma})|^{\frac{1}{2}}} \times \exp[-(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})/2]. \quad (2)$$

Denote distribution of a vector $\boldsymbol{x}$ as $p(\boldsymbol{x})$. Supposing the distribution is parametric and is with parameter Θ, the expectation operation with the distribution $p(\boldsymbol{x})$ is denoted as $E_{\Theta}[\cdot]$. Thus, the

mean of the above function can be represented as $\boldsymbol{\mu} = E_{\{\boldsymbol{\mu},\boldsymbol{\Sigma}\}}(\boldsymbol{x})$ and the covariance $\boldsymbol{\Sigma}$ is $E_{\{\boldsymbol{\mu},\boldsymbol{\Sigma}\}}[\delta\boldsymbol{x}\delta\boldsymbol{x}^{\mathrm{T}}]$ ($\delta\boldsymbol{x}$ denotes $\boldsymbol{x} - \boldsymbol{\mu}$).

2. Generative factor analyzed HMM

GFA-HMM is a generative model (Everitt, 1984) for modeling speech by word or phoneme model. Fig. 1 shows graphical model of the GFA-HMM. Round circle and rectangular square each denotes continuous- and discrete-valued node. Shaded nodes denote observations. $q_t = \{1,\ldots,S\}$ denotes discrete state at time t. $Q(T) = (q_1,\ldots,q_t, q_{t+1},\ldots,q_{\mathrm{T}})$ is the discrete state sequence with first-order state transition probability a_{pq} from state p to state q, which accounts for transition of speech units. Continuous-observation node $\boldsymbol{y}(t)$ is dependent on mixture index m_t, which is state-dependent. Continuous variable node $\boldsymbol{x}(t)$ is dependent on phoneme (word) index.[1] Continuous variable node $\boldsymbol{z}(t)$ is independent of the discrete state sequence and model index.

The continuous-valued nodes, $\boldsymbol{y}(t)$, $\boldsymbol{x}(t)$, and $\boldsymbol{z}(t)$ are hierarchical. In the language of generative model, vector $\boldsymbol{x}(t)$ is generated from $\boldsymbol{z}(t)$ through factor analysis (FA) by a state-dependent loading matrix $\mathrm{C} \in R^{D^x \times D^z}$; i.e.,

$$\mathbf{z}(t) \sim p(\mathbf{z}(t)) = \mathcal{N}(\mathbf{z}(t); \mathbf{0}, \mathbf{I}), \tag{3}$$

$$\mathbf{x}(t) = \mathbf{C}\mathbf{z}(t) + \boldsymbol{\zeta}(t), \tag{4}$$

where vector $\boldsymbol{\zeta}(t)$ denotes noise term in vector $\boldsymbol{x}(t)$. The noise is modeled by mixture of Gaussians $\{\mathcal{N}(\boldsymbol{\zeta}(t); \boldsymbol{\xi}_j^x, \mathbf{V}_j^x)\}_{j=1,\ldots,M^x}$, with component weight c_j. $\mathbf{V}_j^x$ is diagonal. M^x denotes number of the mixture components for $\boldsymbol{\zeta}(t)$. We denote the space spanned by $\mathbf{z}(t) \in R^{D^z}$ as *normalized space*, and the space spanned by $\mathbf{x}(t) \in R^{D^x}$ as *factor space*.

The marginal density of $\boldsymbol{x}(t)$ given mixture component j is mixture of Gaussians; i.e.,

$$\mathbf{x}(t) \sim p(\mathbf{x}(t)) = \sum_{j=1}^{M^x} c_j \mathcal{N}(\mathbf{x}(t); \boldsymbol{\xi}_j^x, \mathbf{V}_j^x + \mathbf{C}\mathbf{C}^{\mathrm{T}}). \tag{5}$$

[1] The graphical model in Fig. 1 depicts a *state-level* description of the GFA-HMM. We thus do not explicitly show in the figure the dependence of $\boldsymbol{x}(t)$ on the *model index*.

The covariance matrix of the marginal density of $\boldsymbol{x}(t)$ in each component j is factored as a diagonal matrix $\mathbf{V}_j^x$ and a full matrix $\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}$. Thus, Eqs. (3) and (4) represent phoneme(word)-dependent factor analysis of $\boldsymbol{x}(t)$, which models the covariance matrix of $\boldsymbol{x}(t)$ in an explicit and compact way. In fact, $\boldsymbol{x}(t)$ can be considered as generated from a MoG with intra-frame correlations explicitly modeled by $\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}$.

The latent representation for observation $\boldsymbol{y}(t)$ given $\boldsymbol{x}(t)$ and $q_t = q$ is thus

$$\mathbf{x}(t) \sim \mathrm{Model}^{\mathrm{MoG}}, \tag{6}$$

$$\mathbf{y}(t) = \boldsymbol{\Lambda}\mathbf{x}(t) + \mathbf{v}_{q_t}, \tag{7}$$

where the observation noise $\mathbf{v}_{q_t}$ is distributed according to MoG $\{\mathcal{N}(\mathbf{v}_{q_t}; \boldsymbol{\mu}_{qm}^y, \boldsymbol{\Sigma}_{qm}^y)\}_{m=1,\ldots,M_q^y}$ with mixture weight π_{qm}. M_q^y is the number of mixture components in state q for conditional density of $\boldsymbol{y}(t)$ given $\boldsymbol{x}(t)$. $\boldsymbol{\Sigma}_{qm}^y$ is diagonal with σ_{qmn}^2 for element (n,n). $\boldsymbol{\Lambda}$ is state-dependent loading matrix with dimension of $D^y \times D^x$. We denote the space spanned by $y(t) \in R^{D^y}$ as *observation space*.

By Eqs. (6) and (7), the conditional density of $\boldsymbol{y}(t)$ given $\boldsymbol{x}(t)$ in state q and observation component m is Gaussian; i.e.,

$$p(\mathbf{y}(t)|\mathbf{x}(t), q, m) = \mathcal{N}(\mathbf{y}(t); \boldsymbol{\mu}_{qm}^y + \boldsymbol{\Lambda}\mathbf{x}(t), \boldsymbol{\Sigma}_{qm}^y). \tag{8}$$

Marginalizing the conditional density with respect to $\boldsymbol{x}(t)$ can obtain the density of $\boldsymbol{y}(t)$ at state q in component m; i.e.,

$$p(\mathbf{y}(t)|q,m) = \sum_j c_j \int p(\mathbf{y}(t)|\mathbf{x}(t), q, m) p(\mathbf{x}(t)|j)\, \mathrm{d}\mathbf{x}(t). \tag{9}$$

The above equation can be computed as follows

$$p(\mathbf{y}(t)|q,m) = \sum_{j=1}^{M^x} c_j \mathcal{N}(\mathbf{y}(t); \boldsymbol{\mu}_{qm}^y + \boldsymbol{\Lambda}\boldsymbol{\xi}_j^x, \boldsymbol{\Sigma}_{qm}^y + \boldsymbol{\Lambda}(\mathbf{V}_j^x + \mathbf{C}\mathbf{C}^{\mathrm{T}})\boldsymbol{\Lambda}^{\mathrm{T}}). \tag{10}$$

With $p(\boldsymbol{y}(t)|q,m)$ by Eq. (10), $p(\boldsymbol{y}(t)|q)$ can be obtained as

$$p(\mathbf{y}(t)|q) = \sum_{m=1}^{M_q^y} \pi_{qm} p(\mathbf{y}(t)|q,m). \tag{11}$$

With Eq. (11) at hand, the likelihood of GFA-HMM for input sequence $\boldsymbol{Y}(T) = (\boldsymbol{y}(1), \ldots, \boldsymbol{y}(T))$ can be calculated given a state sequence $Q(T) = (q_1, \ldots, q_t, \ldots, q_T)$; i.e.,

$$p(\mathbf{Y}(T)|Q(T)) = \prod_{t=1}^{T} p(\mathbf{y}(t)|q_t). \tag{12}$$

The Viterbi process (Viterbi, 1967) is normally applied to obtain the state sequence having the largest posterior probability

$$\widehat{Q}(T) = \arg\max_{Q(T)} p(\mathbf{Y}(T)|Q(T))p(Q(T)), \tag{13}$$

where $p(Q(T)) = \prod_{t=1}^{T} a_{q_{t-1}q_t}$.

Since the observation vector $\boldsymbol{y}(t)$ is generated from $\boldsymbol{x}(t)$, which has correlations between elements in $\boldsymbol{x}(t)$ modeled by state dependent factor analysis, we denote the model as whole; i.e., Eqs. (3)–(7), as generative factor-analyzed HMM (GFA-HMM).

The model is a generalization of several generative models. When the dimension of $\boldsymbol{z}(t)$ is zero, the model reduces to FA-HMM (RG) (Rosti and Gales, 2002). Further constraints on $\boldsymbol{x}(t)$ to be distributed as $\mathcal{N}(\mathbf{x}(t); \mathbf{0}, \mathbf{I})$ results in the HMM with factorized covariance matrix (FA-HMM (SR)) (Saul and Rahim, 2000). Standard HMM can be obtained by setting dimension of $\boldsymbol{x}(t)$ to be zero.

The model captures dynamics of speech features through transition of states. The model is also non-linear, because, as shown in Eq. (4), the densities of noise $\boldsymbol{\zeta}(t)$ in the continuous-valued latent vector $\boldsymbol{x}(t)$ are mixture of Gaussians with diagonal covariance matrix $\mathbf{V}_j^x$ that might be different for each j.

2.1. Number of free parameters

Referring to Fig. 1, the number of free parameters (NoFP) in GFA-HMM can be calculated separately for each of the latent representations. In particular, for Eq. (4), NoFP is $D^z \times D^x + 2 \times M^x \times D^x$. For Eq. (7), NoFP is $D^y \times D^x + 2 \times S \times M^y \times D^y$, where $M^y = \max\{M_q^y : q = 1, \ldots, S\}$. As a whole, the NoFP for GFA-HMM is given as[2]

$$\begin{aligned}\text{NoFP}_{\text{GFA-HMM}} &= (D^z + D^y) \times D^x \\ &\quad + 2 \times (M^x \times D^x + S \times M^y \times D^y).\end{aligned} \tag{14}$$

Note that the standard HMM has NoFP as $2 \times S \times M^y \times D^y$.

The seemingly more complex formula of NoFP for GFA-HMM does not mean that the GFA-HMM requires more free parameters to achieve a performance comparable to standard HMM. On the contrary, a fair comparison should be based on the number of free parameters given a performance.

The GFA-HMM imposes explicit continuous-valued latent representation of observation vectors, which lack in standard HMMs. This structural information can make GFA-HMM be more compact over standard HMM in the sense that the GFA-HMM can achieve better performance than standard HMM with the number of free parameters that is less than that in the standard HMM. We will verify this statement through experiments in Section 5.

3. Maximum likelihood estimation of the GFA-HMM

For clarity, we plot the graphical model of the GFA-HMM together with its parameters to be estimated in Fig. 2. Note that the only observable variable is $\boldsymbol{y}(t)$. Other variables are hidden. Denote the sequence of continuous-valued latent vector $\boldsymbol{x}(t)$ as $X(T) := (\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(T))$ and the sequence of the higher-layer continuous-valued latent vector $\boldsymbol{z}(t)$ as $Z(T) := (\boldsymbol{z}(1), \boldsymbol{z}(2), \ldots, \boldsymbol{z}(T))$. Further denote the discrete-valued latent sequences, $Q(T) := (q_1, q_2, \ldots, q_T)$ for hidden state q, $M(T) := (m_1, m_2, \ldots, m_T)$ for hidden Gaussian mixture index m, and $J(T) := (j_1, j_2, \ldots, j_T)$ for hidden Gaussian mixture index j. These latent sequences are 'missing' to the observations for the model. The 'complete' data to the model is composed of both the 'missing' sequences and the observation sequence $Y(T)$.

As shown in Fig. 2, the model parameters in GFA-HMM include state transition probability a_{qp} for transition from state q to p, Gaussian

[2] NoFP of the transition probabilities is not considered in the current work.

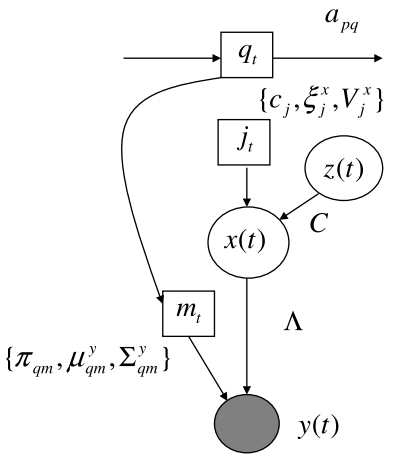

Fig. 2. Estimation parameters for the generative factor analyzed HMM (GFA-HMM). a_{qp} is the discrete state transition probability. $\{c_j, \boldsymbol{\xi}_j^x, \mathbf{V}_j^x\}$ denotes parameter for node j_t in modeling density of the noise vector $\boldsymbol{\zeta}(t)$ in latent vector $\boldsymbol{x}(t)$, where c_j, $\boldsymbol{\xi}_j^x$, and $\mathbf{V}_j^x$ each denotes the mixture component weight, mean vector, and covariance matrix. $\boldsymbol{C}$ is the model-dependent loading matrix mapping $\boldsymbol{z}(t)$ and $\boldsymbol{x}(t)$. $\{\pi_{qm}, \boldsymbol{\mu}_{qm}^y, \boldsymbol{\Sigma}_{qm}^y\}$ is the parameter for node m_t in modeling density of the observation noise $\mathbf{v}_{q_t}$ in observation vector $\boldsymbol{y}(t)$, where π_{qm}, $\boldsymbol{\mu}_{qm}^y$, and $\boldsymbol{\Sigma}_{qm}^y$ each denotes the mixture weight, mean vector, and covariance matrix.

density parameters in component m and j, and loading matrices $\boldsymbol{C}$ and $\boldsymbol{\Lambda}$. We denote the GFA-HMM parameters collectively by

$$\Theta = (a_{qp}, \mathbf{C}, c_j, \boldsymbol{\xi}_j^x, \mathbf{V}_j^x, \boldsymbol{\Lambda}, \pi_{qm}, \boldsymbol{\mu}_{qm}^y, \boldsymbol{\Sigma}_{qm}^y). \tag{15}$$

With the parameters, the model likelihood given the 'complete' data is by

$$\begin{aligned}&\log \prod_{t=1}^{T} p(\mathbf{y}(t), \mathbf{x}(t), \mathbf{z}(t), q, m, j|\Theta)\\ &= \log \prod_{t=1}^{T} \{p(\mathbf{z}(t)) a_{q_{t-1}q_t} c_{j_t} p(\mathbf{x}(t)|\mathbf{z}(t), j_t)\\ &\quad \times \pi_{q_t m_t} p(\mathbf{y}(t)|\mathbf{x}(t), q_t, m_t)\}^{\delta_{qmj}(t)},\end{aligned} \tag{16}$$

where $\delta_{qmj}(t) = 1$, if $q_t = q$, $m_t = m$ and $j_t = j$. Otherwise, $\delta_{qmj}(t) = 0$.

Since the above likelihood has 'missing' data, we cannot perform maximum likelihood estimation on the model parameter by direct maximization of the above likelihood w.r.t. model parameters. Expectation Maximization (EM) algorithm (Dempster et al., 1977) is one method that is commonly used for parameter estimation of the model with 'missing' data. In the following, we apply EM algorithm for maximum likelihood estimation of the model parameter Θ.

Note that, since the sequences $Q(T)$, $X(T)$, $Z(T)$, $M(T)$, and $J(T)$ are hidden, instead of the likelihood in Eq. (16), the EM algorithm maximizes the following auxiliary function, which is defined as the average of the joint log-likelihood in Eq. (16) calculated on new model parameter $\widetilde{\Theta}$ over posterior probabilities of the hidden sequences calculated from the previous model parameter Θ, i.e.,

$$\begin{aligned}\widehat{\Theta} &= \arg\max_{\widetilde{\Theta}} Q(\Theta, \widetilde{\Theta})\\ &= \arg\max_{\widetilde{\Theta}} E_{\Theta}\left[\log \prod_{t=1}^{T} p(\mathbf{y}(t), \mathbf{x}(t), \mathbf{z}(t), q, m, j|\widetilde{\Theta})\right].\end{aligned} \tag{17}$$

Referring to Fig. 2, the auxiliary function can be written as

$$\begin{aligned}Q(\Theta, \widetilde{\Theta}) &= E_{\Theta}\left[\log \prod_{t=1}^{T} p(\mathbf{y}(t), \mathbf{x}(t), \mathbf{z}(t), q, m, j|\widetilde{\Theta})\right]\\ &= E_{\Theta}\left[\log \prod_{t=1}^{T} \{\tilde{a}_{q_{t-1}q_t} \tilde{\pi}_{q_t m_t}\}^{\delta_{qm}(t)}\right]\\ &\quad + E_{\Theta}\left[\log \prod_{t=1}^{T} \{p(\mathbf{z}(t)) \tilde{c}_{j_t} p(\mathbf{x}(t)|\mathbf{z}(t), j_t, \widetilde{\Theta})\}^{\delta_j(t)}\right]\\ &\quad + E_{\Theta}\left[\log \prod_{t=1}^{T} \{p(\mathbf{y}(t)|\mathbf{x}(t), q_t, m_t, \widetilde{\Theta})\}^{\delta_{qm}(t)}\right],\end{aligned} \tag{18}$$

where $\delta_{qm}(t)$, and $\delta_j(t)$ are calculated given previous model parameter Θ. Since components in RHS of (18) are functions of $\{\tilde{a}_{qp}, \tilde{\pi}_{qm}\}$, $\{\widetilde{\mathbf{C}}, \tilde{c}_j, \widetilde{\boldsymbol{\xi}}_j^x, \widetilde{V}_j^x\}$ and $\{\widetilde{\boldsymbol{\Lambda}}, \widetilde{\boldsymbol{\mu}}_{qm}^y, \widetilde{\boldsymbol{\Sigma}}_{qm}^y\}$, respectively, parameter estimation can be carried out on them separately. EM algorithm for parameter estimation requires posterior statistics of the discrete- and continuous-valued hidden sequences.

3.1. Posterior statistics

Traditional HMM has a discrete-valued latent representation of observations; i.e., state q and

Gaussian mixture index m. The GFA-HMM also has a hierarchy of continuous-valued latent representation by $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$. Accordingly, posterior statistics are calculated on both discrete and continuous latent variables. In the following, we derive posterior statistics of the discrete-valued latent sequences in Section 3.1.1, which follows closely to the standard HMMs (Rabiner and Juang, 1993), and posterior statistics of continuous-valued latent variables in Section 3.1.2.

3.1.1. Posterior statistics of discrete-valued variables

We first derive the formulae for posterior statistics of discrete sequences $Q(T)$, $M(T)$ and $J(T)$. Denote the posterior probability of being in state q at time t given observation sequence $\boldsymbol{Y}(T)$ and model parameter Θ, $p(q|\boldsymbol{Y}(T),\Theta)$, as $\gamma_q(t)$. With the likelihood in Eq. (11), it can be obtained by the forward-backward algorithm as in standard HMM (Rabiner and Juang, 1993); i.e.,

$$\gamma_q(t) = \frac{\alpha_q(t)\beta_q(t)}{\sum_{i=1}^{S}\alpha_i(t)\beta_i(t)}, \tag{19}$$

where $\alpha_q(t) = p((\boldsymbol{y}(1),\ldots,\boldsymbol{y}(t)), q_t = q|\Theta)$ accounts for the probability of the partial observation sequence $(\boldsymbol{y}(1),\ldots,\boldsymbol{y}(t))$ and state q at time t given model parameter Θ, while $\beta_i(t) = p((\boldsymbol{y}(t+1),\ldots,\boldsymbol{y}(T))|q_t = i,\Theta)$ is the probability of the partial observation sequence $(\boldsymbol{y}(t+1),\ldots,\boldsymbol{y}(T))$ given state i at time t and model parameter Θ.

These partial probabilities can be inducted as

$$\alpha_q(t+1) = \left[\sum_{i=1}^{S}\alpha_i(t)a_{iq}\right]p(\mathbf{y}(t+1)|q), \tag{20}$$

$$\beta_q(t) = \sum_{i=1}^{S} a_{qi}p(\mathbf{y}(t+1)|i)\beta_i(t+1). \tag{21}$$

Denote the posterior probability at mixture component m and j in state q given observation sequence $\boldsymbol{Y}(T)$ and model parameter Θ as $\gamma_{qmj}(t)$, which is written as

$$\begin{aligned}\gamma_{qmj}(t) &= p(q_t = q, m_t = m, j_t = j|\mathbf{Y}(T),\Theta)\\ &= \frac{\pi_{qm}c_j p(\mathbf{y}(t)|q_t = q, m_t = m, j_t = j,\Theta)}{\sum_m\sum_j \pi_{qm}c_j p(\mathbf{y}(t)|q_t = q, m_t = m, j_t = j,\Theta)}\gamma_q(t).\end{aligned} \tag{22}$$

The posterior probability at mixture component m in state q, $\gamma_{qm}(t)$, is given as $\sum_j \gamma_{qmj}(t)$. Similarly for $\gamma_q(t) = \sum_{m,j}\gamma_{qmj}(t)$ and $\gamma_j(t) = \sum_{q,m}\gamma_{qmj}(t)$. These posterior probabilities, $\gamma_q(t)$, $\gamma_{qm}(t)$ and $\gamma_j(t)$, provide soft-version of $\delta_q(t)$, $\delta_{qm}(t)$ and $\delta_j(t)$, respectively, in (18).

3.1.2. Posterior statistics of the continuous-valued latent vectors

In this section, we assume that density of $\boldsymbol{x}(t)$ can be approximated well by $\mathcal{N}(\cdot;\boldsymbol{\xi}^x,\mathbf{V}^x)$. These mean $\boldsymbol{\xi}^x$ and covariance V^x may be obtained from the MoG in Eq. (5), or, alternatively as applied in this paper, they are updated by Eqs. (32) and (33) in a data-driven way.

We can view the dependence among observation $\boldsymbol{y}(t)$, node m_t and $x(t)$ locally as shown in Fig. 3. Inference of posterior statistics at node $x(t)$ consists of obtaining the posterior mean and variance of $\boldsymbol{x}(t)$ given $\boldsymbol{y}(t)$ and Θ. Regarding the posterior distribution, by Bayes rule, it is given as

$$p(\mathbf{x}(t)|\mathbf{y}(t),q,m,\Theta) = \frac{p(\mathbf{y}(t)|\mathbf{x}(t),q,m,\Theta)p(\mathbf{x}(t)|\Theta)}{p(\mathbf{y}(t)|q,m,\Theta)}. \tag{23}$$

It is easy to verify that the posterior distribution is Gaussian. Briefly derived in Appendix A, the above equation can arrive at $\mathcal{N}(\mathbf{x}(t);\boldsymbol{\phi}^x_{qm}(t),\boldsymbol{\Psi}^x_{qm}(t))$, where the posterior mean and posterior covariance are respectively given as

$$\begin{aligned}\boldsymbol{\phi}^x_{qm}(t) &= E_\Theta[\mathbf{x}(t)|\mathbf{y}(t),q,m]\\ &= \boldsymbol{\Psi}^x_{qm}(t)[(\mathbf{V}^x)^{-1}\boldsymbol{\xi}^x + \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Sigma}^{y-1}_{qm}(\mathbf{y}(t) - \boldsymbol{\mu}^y_{qm})],\end{aligned} \tag{24}$$

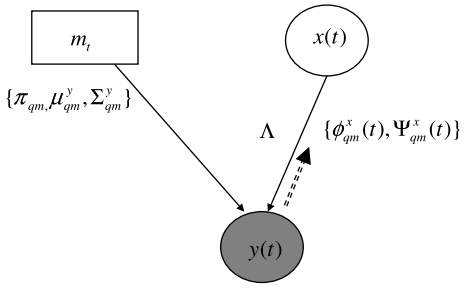

Fig. 3. Graphical model associated to $\boldsymbol{x}(t)$, $\boldsymbol{y}(t)$, and m_t.

$$\begin{aligned}\boldsymbol{\Psi}^x_{qm}(t) &= E_\Theta[(\mathbf{x}(t)-\boldsymbol{\phi}^x_{qm})(\mathbf{x}(t)-\boldsymbol{\phi}^x_{qm})^{\mathrm{T}}|\mathbf{y}(t),q,m] \\ &= [(\mathbf{V}^x)^{-1}+\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Sigma}^{y-1}_{qm}\boldsymbol{\Lambda}]^{-1}. \end{aligned} \tag{25}$$

Combining π_{qm} with likelihood $p(\boldsymbol{y}(t)|q,m)$ given in Eq. (10), the 'observation' at node $x(t)$ is the posterior mean which has the largest posterior probability at q_t and m_t, i.e.,

$$\boldsymbol{\phi}^x(t) = \boldsymbol{\phi}^x_{q^\star m^\star}(t), \tag{26}$$

$$\boldsymbol{\Psi}^x(t) = \boldsymbol{\Psi}^x_{q^\star m^\star}(t), \tag{27}$$

where $q^\star m^\star = \arg\max_{q,m}\gamma_{qm}(t)$.

Taking $\phi^x(t)$ as observation at node $x(t)$, we can apply in a similarly way as the above derivation to nodes $z(t)$, $x(t)$, and j_t. The dependence of these nodes can be seen locally in Fig. 4. The posterior distribution of $\boldsymbol{z}(t)$ given $\boldsymbol{x}(t)$, $j_t = j$ and model parameter Θ is Gaussian. Given $\boldsymbol{\phi}^x(t)$ as the 'observation' at node $x(t)$, the same way for derivation of Eq. (24) and Eq. (25) can be followed to the posterior statistics of $\boldsymbol{z}(t)$. Now, with Eq. (3), the posterior statistics of $\boldsymbol{z}(t)$ can be written as $\mathcal{N}(\mathbf{z}(t);\boldsymbol{\phi}^z_j(t),\boldsymbol{\Psi}^z_j(t))$, where the posterior mean and covariance are

$$\begin{aligned}\boldsymbol{\phi}^z_j(t) &= E_\Theta[\mathbf{z}(t)|\mathbf{y}(t),j] \\ &= \boldsymbol{\Psi}^z_j(t)\mathbf{C}^T\mathbf{V}^{x-1}_j(\boldsymbol{\phi}^x(t)-\boldsymbol{\xi}^x_j), \end{aligned} \tag{28}$$

$$\boldsymbol{\Psi}^z_j(t) = (\mathbf{I}+\mathbf{C}^{\mathrm{T}}(\mathbf{V}^x_j)^{-1}\mathbf{C})^{-1}. \tag{29}$$

The posterior 'observation' to node $\boldsymbol{z}(t)$ is approximated as

$$\boldsymbol{\phi}^z(t) = \boldsymbol{\phi}^z_{j^\star}(t), \tag{30}$$

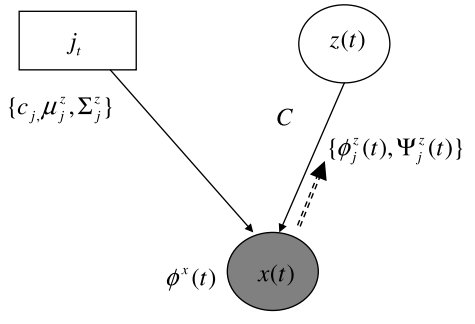

Fig. 4. Graphical model associated to $\boldsymbol{z}(t)$, $\boldsymbol{x}(t)$, and j_t.

$$\boldsymbol{\Psi}^z(t) = \boldsymbol{\Psi}^z_{j^\star}(t), \tag{31}$$

where $j^\star = \arg\max_j \gamma_j(t)$.

3.2. Parameter estimation

3.2.1. Updating in observation space

The first parameters to be updated are at node $x(t)$. Given the 'observation' sequence ($\boldsymbol{\phi}^x(1)$,..., $\boldsymbol{\phi}^x(t)$,..., $\boldsymbol{\phi}^x(T)$), the mean and covariance at node $x(t)$ are given as

$$\widehat{\boldsymbol{\xi}}^x = \frac{\sum_{t=1}^T \boldsymbol{\phi}^x(t)}{T}, \tag{32}$$

$$\widehat{\mathbf{V}}^x = \frac{\sum_{t=1}^T (\boldsymbol{\phi}^x(t)-\widehat{\boldsymbol{\xi}}^x)(\boldsymbol{\phi}^x(t)-\widehat{\boldsymbol{\xi}}^x)^{\mathrm{T}}}{T}. \tag{33}$$

Setting derivatives in Eq. (17) w.r.t. loading matrix $\widehat{\boldsymbol{\Lambda}}$ to zero gives

$$\begin{aligned}&\sum_t \gamma_{qm}(t)\boldsymbol{\Sigma}^{y-1}_{qm}\widehat{\boldsymbol{\Lambda}}(\boldsymbol{\phi}^x(t)\boldsymbol{\phi}^x(t)^{\mathrm{T}}+\boldsymbol{\Psi}^x(t)) \\ &\quad = \sum_t \gamma_{qm}(t)\boldsymbol{\Sigma}^{y-1}_{qm}(\mathbf{y}(t)-\boldsymbol{\mu}^y_{qm})\boldsymbol{\phi}^x(t)^{\mathrm{T}}, \end{aligned} \tag{34}$$

through which the new loading matrix $\widehat{\boldsymbol{\Lambda}}$ can be obtained. (Detailed formulae are shown in Appendix B.)

Mean and covariance matrix at node m_t are then updated respectively by

$$\widehat{\boldsymbol{\mu}}^y_{qm} = \frac{1}{\sum_t \gamma_{qm}(t)}\sum_t \gamma_{qm}(t)[\mathbf{y}(t)-\widehat{\boldsymbol{\Lambda}}\boldsymbol{\phi}^x(t)], \tag{35}$$

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}^y_{qm} &= \mathrm{diag}\frac{1}{\sum_t \gamma_{qm}(t)}\sum_t \gamma_{qm}(t) \\ &\quad \times [(\mathbf{y}(t)-\widehat{\boldsymbol{\mu}}^y_{qm}-\widehat{\boldsymbol{\Lambda}}\boldsymbol{\phi}^x(t))(\mathbf{y}(t) \\ &\quad -\widehat{\boldsymbol{\mu}}^y_{qm}-\widehat{\boldsymbol{\Lambda}}\boldsymbol{\phi}^x(t))^{\mathrm{T}}+\widehat{\boldsymbol{\Lambda}}\boldsymbol{\Psi}^x(t)\widehat{\boldsymbol{\Lambda}}^{\mathrm{T}}] \\ &\approx \mathrm{diag}\frac{1}{\sum_t \gamma_{qm}(t)}\sum_t \gamma_{qm}(t) \\ &\quad \times [(\mathbf{y}(t)-\widehat{\boldsymbol{\mu}}^y_{qm}-\widehat{\boldsymbol{\Lambda}}\boldsymbol{\phi}^x(t))(\mathbf{y}(t) \\ &\quad -\widehat{\boldsymbol{\mu}}^y_{qm}-\widehat{\boldsymbol{\Lambda}}\boldsymbol{\phi}^x(t))^{\mathrm{T}}], \end{aligned} \tag{36}$$

where the last equation in updating covariance matrix is obtained by assuming that the posterior covariance $\boldsymbol{\Psi}^x(t)$ is small.

The weight of mixture component m at state q can be updated by

$$\hat{\pi}_{qm} = \frac{\sum_{t=1}^{T} \gamma_{qm}(t)}{\sum_{t=1}^{T} \sum_{m=1}^{M_q^y} \gamma_{qm}(t)}. \quad (37)$$

Transition probability a_{qp} is updated by

$$\hat{a}_{qp} = \frac{\sum_t \gamma_{qp}(t)}{\sum_t \sum_p \gamma_{qp}(t)}. \quad (38)$$

3.2.2. Updating in Factor Space

Since Eq. (3) and Eq. (4) are factor analysis on 'observation' $\boldsymbol{\phi}^x(t)$, in a similar way shown in Eq. (34), the loading matrix $\widehat{\mathbf{C}}$ can be obtained from

$$\sum_t \sum_j \gamma_j(t) (\mathbf{V}_j^x)^{-1} \widehat{\mathbf{C}} (\boldsymbol{\phi}^z(t) \boldsymbol{\phi}^z(t)^{\mathrm{T}} + \boldsymbol{\Psi}^z(t)) = \sum_t \sum_j \gamma_j(t) (\mathbf{V}_j^x)^{-1} (\boldsymbol{\phi}^x(t) - \boldsymbol{\xi}_j^x) \boldsymbol{\phi}^z(t)^{\mathrm{T}}. \quad (39)$$

Then, updated mean vector and diagonal covariance matrix in j_t are respectively obtained by

$$\widehat{\boldsymbol{\xi}}_j^x = \frac{1}{\sum_t \gamma_j(t)} \sum_t \gamma_j(t) (\boldsymbol{\phi}^x(t) - \widehat{\mathbf{C}} \boldsymbol{\phi}^z(t)), \quad (40)$$

$$\widehat{V}_j^x = \mathrm{diag} \frac{1}{\sum_t \gamma_j(t)} \sum_t \gamma_j(t) \times \{ [\boldsymbol{\phi}^x(t) - \widehat{\boldsymbol{\xi}}_j^x - \widehat{\mathbf{C}} \boldsymbol{\phi}^z(t)] \times [\boldsymbol{\phi}^x(t) - \widehat{\boldsymbol{\xi}}_j^x - \widehat{\mathbf{C}} \boldsymbol{\phi}^z(t)]^{\mathrm{T}} + \widehat{\mathbf{C}} \boldsymbol{\Psi}^z(t) \widehat{\mathbf{C}}^{\mathrm{T}} \} \quad (41)$$

The weights of the components j in factor space can be found by

$$\hat{c}_j = \frac{\sum_{t=1}^{T} \gamma_j(t)}{\sum_{t=1}^{T} \sum_{j=1}^{M^x} \gamma_j(t)}. \quad (42)$$

3.3. Summary of the training process

GFA-HMM may be initialized with parameters obtained from standard HMMs. The initialization is conducted by retaining $\{a_{pq}, \pi_{qm}, \boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm}\}$ trained based on standard HMMs. Variance term $\mathbf{V}_j^x$ is set to diagonal unit matrix. Mean term $\boldsymbol{\xi}_j^x$ is initialized by a random vector distributed in $\mathcal{N}(\cdot; \mathbf{0}, \mathbf{I})$. Loading matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{C}$ are initialized to matrices with each element distributed in $\mathcal{N}(\cdot; 1, 1)$.

After initialization, EM-process may be carried out by the following iterative process:

(1) E-step: Obtain posterior statistics via Eqs. (19)–(22), and (24)–(31).
(2) M-step: Update model parameters via Eqs. (32)–(42).

In practice, to reduce computational cost in E-step, we do force alignment using well-trained standard HMMs. E-step is then performed on these segments. Because the alignment is given by standard HMMs, this practice implicitly assumes that the state transition probabilities $\{a_{pq}\}$ in GFA-HMM are the same as those in standard HMMs. Therefore, $\{a_{pq}\}$ are usually left unchanged in M-step.

4. Relationship to mixtures of PCA, semi-tied covariance modeling, and maximum likelihood linear transformation

Because the proposed GFA-HMM is a method of modeling intra-frame statistics, it may relate to other techniques for the same purpose. In fact, the form of estimation routine described for GFA-HMM may be applied to other estimation problems in speech recognition. This section briefly discusses schemes to optimize mixtures of PCA transform, semi-tied covariance modeling and maximum likelihood linear transformation.

Probabilistic principle component analysis (PPCA) is related to factor analysis (FA). This relationship is revealed by Tipping and Bishop (1997). It is shown in (Tipping and Bishop, 1997) that PPCA is a particular case of FA with observation noise term to be isotropic; i.e., $\boldsymbol{\Sigma}_{qm}^y = \sigma_{qm}^2 \mathbf{I}$ in Eq. (8) and σ_{qm}^2 is a scalar variable. Therefore, for estimating mixtures of PPCA model in (Tipping and Bishop, 1997), the previously shown EM steps can be directly applied by further setting $\boldsymbol{\zeta}(t) \sim \mathcal{N}(\cdot; \mathbf{0}, \mathbf{I})$ and $\boldsymbol{C} = \boldsymbol{0}$ in Eq. (4). Usually, for maximum likelihood of mixtures of PPCA (MPCA) model, a flooring scheme on σ_{qm}^2 is employed in order to avoid singular covariance matrix by $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$ (note that $rank(\boldsymbol{\Lambda}) < D.$) in Eq. (10).

It is shown that traditional PCA is a limiting case of PPCA to $\sigma_{qm}^2 \to 0$.

Notice that GFA-HMM and MPCA both assume that $D^x < D^y$. There are other ways in modeling intra-frame statistics. In semi-tied covariance (SMC) modeling (Gales, 1999), covariance matrix at each mixture in modeling density of $\boldsymbol{y}(t)$ is given by

$$\boldsymbol{\Sigma}_{qm}^{y} = \boldsymbol{\Lambda}^{(r)}\mathrm{diag}(\boldsymbol{\Sigma}_{qm})\boldsymbol{\Lambda}^{(r)T}, \tag{43}$$

where $\boldsymbol{\Lambda}^{(r)}$ may be shared among many mixture components. Diagonal matrix $\mathrm{diag}(\boldsymbol{\Sigma}_{qm})$ may differs at each mixture components. This model may be related to GFA-HMM by setting (1) $\boldsymbol{\Sigma}_{qm}^{y} = \mathbf{0}$, (2) $\boldsymbol{\Lambda}$ is tied among many mixtures, (3) $\boldsymbol{\xi}_j^x = \mathbf{0}$, and $D^x = D^y$ in Eq. (8). Heteroscedastic linear discriminant analysis (HLDA) (Kumar, 1997) may be considered as a special case of SMC. HLDA can be obtained from SMC by splitting $\boldsymbol{\Lambda}^{(r)}$ into two submatrix, where one is mixture-dependent and the other is tied among all mixture components.

The maximum likelihood linear transformation (MLLT) (Gopinath, 1998) and its extensions, e.g., (Olsen and Gopinath, 2002), are closely related to SMC modeling in terms of having the same equation as Eq. (43) in modeling covariance matrices. However, MLLT transforms speech feature vectors. The above E-M processes may still be applicable to these modeling techniques by setting GFA-HMM to be (1) $\boldsymbol{\Sigma}_{qm}^{y} = \mathbf{0}$, (2) $\boldsymbol{\Lambda}$ is tied among many mixtures, (3) $\boldsymbol{\xi}_j^x = \mathbf{0}$, and $D^x \geqslant D^y$ in Eq. (8).

5. Experimental results

In this section, we conducted a series of experiment to make comparison of GFA-HMMs with other alternative acoustic models. Features for recognition were 39-dimensional MFCC plus C0 and its first- and second-order coefficients. That is $D^y = 39$.

Relative performance improvement in speech recognition are measured by word error rate reduction (ERR):

$$ERR = (WA2 - WA1)/(100 - WA1),$$

where WA1 and WA2 measure word accuracies by system 1 and system 2, respectively.

We first conducted experiments in Section 5.1 to show the convergence property of GFA-HMM. Then, in Section 5.2, we compared GFA-HMM with some alternative acoustic models in terms of their change of log-likelihood during training process. We then conducted a preliminary experiment in Section 5.3 to show that GFA-HMM may have fewer number of parameters as compared to standard HMM. Such property may result in an improved performance with respect to standard HMM for small amount of training data. To further study effects of latent dimension D^x on recognition performances, we conducted experiments in Section 5.4.1 in multiple speaking styles. Experiments were also conducted in Section 5.4.2 to study performances against number of Gaussian mixtures M^y in observation space. These experiments revealed a strong dependence of performances on D^x and M^y. Based on the empirical selection from these experiments, we conducted noisy digits recognition in Section 5.5 to show performance improvement by GFA-HMM w.r.t. standard HMM in a varying amount of training data.

5.1. Number of iterations required

For the GFA-HMM, the parameter estimation process is iterative. Fig. 5 shows a typical change in log-likelihood function value against iteration numbers, with the change of log-likelihood of a standard HMM shown for comparison.[3] It can be seen that the majority of the gain in log-likelihood of both GFA-HMM and standard HMM occurs during the first several model update iterations. After the fourth iteration, the log-likelihood of GFA-HMM increases from -3.93×10^{-3} to above -2.00×10^{-3}, and the log-likelihood of standard HMM increases from -3.92×10^{-3} to above -1.60×10^{-3}. Whereas standard HMM does not vary log-likelihood afterwards, the log-likelihood of GFA-HMM increases above that by standard HMM at fifth iteration and continuous to increase by the tenth iteration.

[3] The GFA-HMM and standard HMM were initialized with the same parameters in $\{\pi_{qm}, \boldsymbol{\mu}_{qm}^{y}, \boldsymbol{\Sigma}_{qm}^{y}\}$.

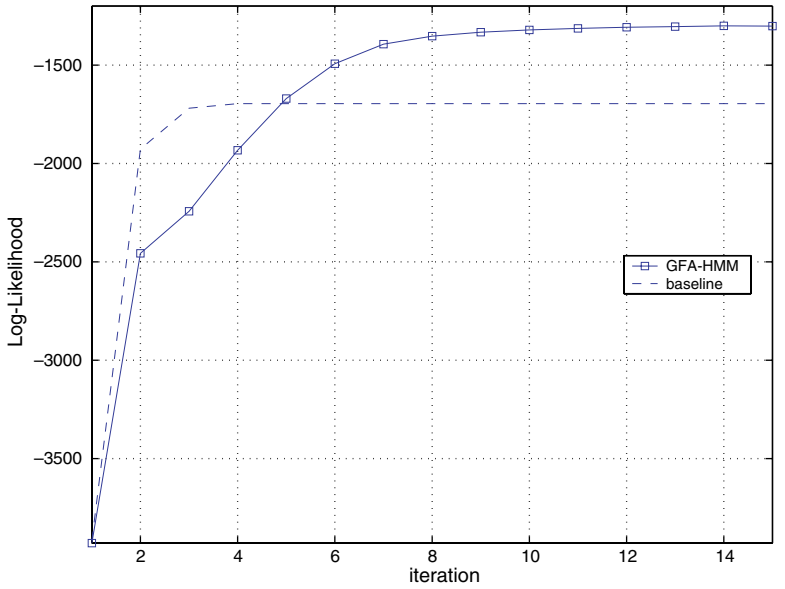

Fig. 5. Increase in log-likelihood against iteration number when optimizing a GFA-HMM. The log-likelihood obtained with a standard diagonal covariance HMM is shown as the baseline. GFA-HMM is with 5-state and 3 Gaussian mixtures per state.

Because the majority of the training time is spent obtaining the sufficient (posterior) statistics from the training data, rather than estimating the model parameters given those statistics. Instead of the actual parameter estimation, the important ability to guarantee stability and convergence are crucial for, particularly, complex systems involving discrete and continuous hidden variables. Therefore, we usually initialize GFA-HMM with parameters by well-trained standard HMMs, and iterate estimation process for three to five iterations.

5.2. Comparison with some alternative models

Notice that both GFA-HMM and FA-HMM (RG) (Rosti and Gales, 2002) were proposed for explicitly modeling intra-frame covariance and implicitly modeling of higher-order intra-frame statistics. It is thus interesting to make comparison between those models, together with FA-HMM (SR) (Saul and Rahim, 2000). One reasonable measure for comparison is the likelihood of those models. Given a training set, the difference in likelihood among the models may suggest differences among these models.

To make a fair comparison, the number of Gaussian mixtures at node $x(t)$ were set, say two, the same for GFA-HMM and FA-HMM (RG).[4] Dimension at $\boldsymbol{x}(t)$ was set to 3. Dimension at $\boldsymbol{z}(t)$ for GFA-HMM was one. We plot in Fig. 6 a typical change of log-likelihood obtained by those acoustic models, trained from a short segment of training data. It is seen that FA-HMM (RG) and FA-HMM (SR) had the same likelihood trend. In fact, for FA-HMM (RG), we increased the number of Gaussian mixtures M^x in node $x(t)$ up to 8, but log-likelihood of the FA-HMM (RG) did not vary much accordingly. The observation may suggest that adding MoG in node $x(t)$ is weak to differentiate FA-HMM (RG) from FA-HMM (SR).

In contrast, the log-likelihood by GFA-HMM was lower than that by FA-HMM (RG) after the fourth iteration. This observation clearly shows that adding another layer $\boldsymbol{z}(t)$ can effectively make GFA-HMM 'depart' from FA-HMM (SR) and FA-HMM (RG). The layer $\boldsymbol{z}(t)$ keeps much shared variance in $\boldsymbol{x}(t)$ by the common loading matrix $\boldsymbol{C}$,

[4] FA-HMM (SR) may be considered as FA-HMM (RG) with only one mixture M^x.

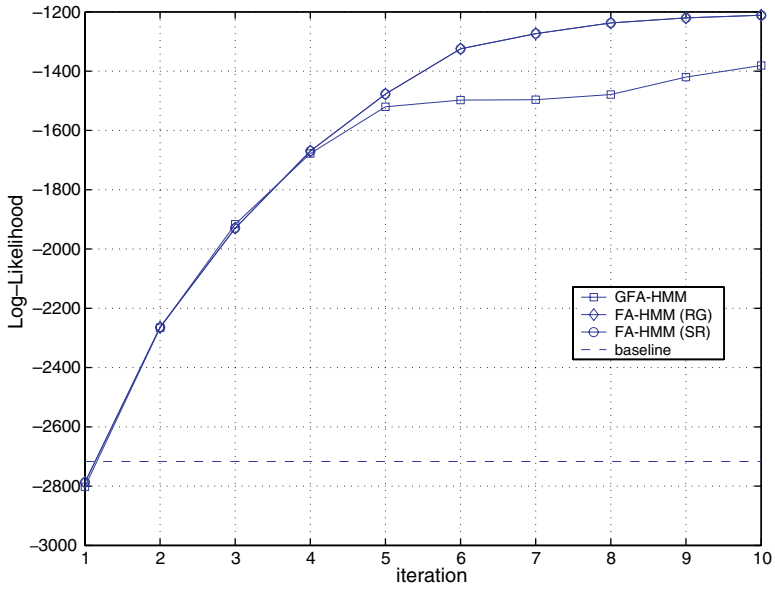

Fig. 6. A typical change of log-likelihood versus iteration number. Compared to FA-HMM (RG) and FA-HMM (SR), GFA-HMM usually has lower likelihood due to larger variance by introducing another layer $\boldsymbol{z}(t)$ to model covariance of $\boldsymbol{x}(t)$. These models were initialized with parameters from a well-trained standard HMM with its log-likelihood shown in dashed line.

whereas FA-HMM (SR) and FA-HMM (RG) do not have such mechanism.

5.3. Small training set experiments

We used one thousand utterances[5] from the clean training set of the Aurora 2 database (Pearce, 2000). Testing was conducted with 1000 clean utterances from the testing set of the database. Acoustic models were trained by EM algorithm with six iterations. Number of states, S, was eight for digit models and one for silence model.

Given the fixed number of state, standard HMM could only adjust the number of mixture components M_q^y. Accordingly, the number of free parameters (NoFP) for a model was $S \times (2D^y) \times M_q^y$. The structure of GFA-HMM is flexible. In this work, we set $M_q^y = 1$. This reduced GFA-HMM to a model with only one observation mixture. Therefore, if any performance improvement over standard HMM can be observed, much of the gain should be attributed to the introduction of the continuous-valued latent representation. For the configuration of the latent representation, the dimension of $\boldsymbol{z}(t)$, D^z, was set to one. We varied the number of mixture components for j_t, M^x, and the dimension of vector $\boldsymbol{x}(t)$, D^x. For this configuration, the GFA-HMM has NoFP as $S \times (2D^y) + (D^y + 1) \times D^x + (2 \times D^x) \times M^x$.

Mixture components were incrementally obtained by mixture splitting, which repeatedly splits the mixture with the largest weight until the desired number of components is obtained. In the training stage, variance of elements in noise vectors $\boldsymbol{\zeta}(t)$ and $\mathbf{v}_{q_t}$ were floored to 1.0 and 0.001, respectively.

5.3.1. Results

Performances by standard HMM and the GFA-HMM are shown in Table 1. Varying number of mixture components M_q^y can change recognition word accuracy (WA). In particular, the highest WA for standard HMM was attained to 88.96% with $M_q^y = 4$.

[5] We limited number of training utterances to 1000, which was less than the standard number, 8440, of training utterances in Aurora 2 database, in order to show the efficacy of the method. Please see Section 5.4 and Section 5.5 for results with larger training sets.

Table 1
For the given test set, we compare the performance between standard HMM and the GFA-HMM in terms of the number of free parameters (NoFP) for one digit model and word accuracy (WA in %)

	Dimension D^x	M_q^y	1	2	3	4
Traditional HMM	0	NoFP	624	1248	1872	2496
		WA	88.48	88.64	88.96	88.96
		M_q^x	1	2	3	4
GFA-HMM ($M_q^y = 1$, $D^z = 1$)	1	NoFP	666	668	670	672
		WA	88.80	89.73	90.30	90.93
	2	NoFP	708	712	716	720
		WA	86.44	89.09	89.73	89.66

The GFA-HMM can achieve higher recognition accuracy over standard HMM with the same amount of training data. For example, word accuracy increased consistently by increasing mixture component M_q^x while keeping $D^x = 1$. The highest WA was 90.93% by setting $D^x = 1$ and $M_q^x = 4$. This is compared to the highest WA achieved by standard HMM, leading to a relative error rate reduction of 18%. Moreover, the NoFPs could be much lower than those by standard HMMs. For example, in the situation where the highest word accuracies were achieved by standard HMM and GFA-HMM, the NoFP for GFA-HMM was 672, whereas the NoFP for standard HMM was 2496.

It is known that the larger the number of free parameters, the more the data needed for reliable estimation of the parameters. By varying dimension D^x and number of mixture M^x, GFA-HMM may be more compact than standard HMM. Therefore, it is observed through experiments that GFA-HMM could achieve higher performance than standard HMM even it had the same amount of training data.

As shown in Table 1, increasing the dimension of latent vector $\boldsymbol{x}(t)$, D^x, did not result in improvement of recognition accuracy. This comes from the flooring scheme used in the experiments. The variances of elements in noise vector $\boldsymbol{\zeta}(t)$ in the latent vector $\boldsymbol{x}(t)$ were floored to 1.0. Increasing dimension D^x of the latent vector may result in a more noisy model if the underling generative process of the observations are different from the structure of GFA-HMM specified for experiments. However, the GFA-HMM still outperformed standard HMM in the situation of $D^x = 2$. For example, the GFA-HMM achieved 89.66% WA when $M_q^x = 4$, a relative error rate reduction of 6% over the highest WA achieved by standard HMM.

5.4. Speech recognition in different speaking styles

Acoustic models in the previous section were word-based. It may raise the following question. Notice that, whereas the fixed state number may be adequate for some digits, for instance 'one', the state number may not be sufficient for other digits such as 'seven'. Therefore, in this section, we investigate properties of GFA-HMM in phoneme level. Experiments were conducted on SUSAS (Speech Under Simulated and Actual Stress) database (Hansen et al., 1998), which contains stressed speech from fairground rides and helicopters. A common highly confusable vocabulary set of 35 aircraft communication words make up the database. It includes male and female voice signals in multi-style (neutral, angry, fast, slow, question, and *et al*) situations. Due to the stressful conditions, speakers may utter voice signals with characteristics, such as pitch, energy, voice duration, and formants that are much different from neutral speech signals.

5.4.1. Performances as a function of D^x

This section presents recognition accuracies by GFA-HMM as a function of dimension D^x in continuous latent variable $\boldsymbol{x}(t)$.

The training set includes all training utterances in neutral speaking style, which has 3780 utterances from male and female speakers with different

Table 2
Word accuracy (in %) achieved on SUSAS database as a function of D^x

	Neutral	Angry	Fast	Slow	Question
Baseline	94.12	63.80	81.48	77.92	86.83
GFA-HMM ($M^x = 2$, $D^x = 3$)	94.71	59.51	80.86	74.03	88.82
GFA-HMM ($M^x = 2$, $D^x = 5$)	94.71	63.19	79.63	77.92	88.82
GFA-HMM ($M^x = 2$, $D^x = 8$)	92.35	66.26	82.72	75.32	87.62
GFA-HMM ($M^x = 2$, $D^x = 10$)	95.88	64.42	83.95	75.32	89.11
GFA-HMM ($M^x = 2$, $D^x = 13$)	90.59	63.19	79.01	76.13	89.11
FA-HMM (RG) ($M^x = 2$, $D^x = 13$)	82.94	64.42	72.84	79.08	88.61

GFA-HMMs were with $M^y = 3$ and $M^x = 2$. Standard HMMs were with $M^y = 3$.

American English accents. Testing set includes all the testing utterances in five speaking styles. Baseline system was based on standard HMMs in a phoneme level. There were 35 phonemes and each phoneme was a 5-state HMM with three diagonal Gaussian mixtures per state. The state output distributions of the baseline models by standard HMM were mixed up to three components running four iterations prior to mixture splitting as described in (Young et al., 1997). Baseline performances (denoted as Baseline) are shown in Table 2.

GFA-HMM was initialized from standard HMMs and trained with EM algorithms for four iterations. The number of MoGs in node $x(t)$, M^x, was fixed to 2. We varied dimension of $\boldsymbol{x}(t)$ in GFA-HMM, whereas the dimension of $\boldsymbol{z}(t)$ was fixed to 2. Performances by GFA-HMM[6] are denoted according to D^x in Table 2.[7,8]

Recognition results show that performances by GFA-HMM are dependent on dimension of $\boldsymbol{x}(t)$. For example, in neutral speech, recognition accuracy with $D^x = 3$ was increased to 94.71% from the baseline by 94.12%, corresponding to 10.0% ERR. The highest ERR in neutral speech was achieved at $D^x = 10$, reaching 29.9%. Performances at $D^x = 10$ in other speaking styles except 'Slow' were also higher than baseline. In average, GFA-HMM achieved 10.1% ERR at $D^x = 10$ as compared to baseline.

We noticed that, when D^x was further increased to 13, performances by GFA-HMM might be dropped. For example, performances in all speaking styles except 'Question' had lower recognition accuracies compared to baseline at $D^x = 13$.

We also tested performances by FA-HMM (RG) (Rosti and Gales, 2002). D^x in FA-HMM (RG) was chosen to be 13 by referring an experimental setup in (Rosti and Gales, 2002). Its performances in the set of speaking styles were shown in the last row. In fact, GFA-HMM achieved better performances than FA-HMM (RG); in the same dimension $D^x = 13$, GFA-HMM obtained an averaged 10.9% ERR with respect to FA-HMM (RG).[9]

5.4.2. Effects of increasing observation mixture components M^y

This section conducted experiments to test GFA-HMM performances w.r.t. M^y, the number of Gaussian mixtures in observation node $y(t)$. GFA-HMM configuration was fixed to $D^z = 2$, $D^x = 10$, and $M^x = 2$. Results are summarized in Table 3. It is seen that, given the configuration, there is an optimal M^y in terms of recognition

[6] Performances were obtained by re-scoring lattice generated from standard HMMs in order to reduce computational costs. GFA-HMM rescored likelihood on the lattices, which had reduced number of state sequences compared to that in standard HMM. Without this scheme, GFA-HMM may have D^y times of computational cost compared to a standard HMM with the same M^y and D^y.

[7] Because D^z was set to two, D^x had to be larger than D^z in order to avoid non-singularity in loading matrix $\boldsymbol{C}$. Therefore, experiments were conducted for GFA-HMM with $D^x > 2$ only.

[8] In Table 2, GFA-HMM had NoFP from 837 for $D^x = 3$ to 1287 for $D^x = 13$. Baseline had 702 NoFP. FA-HMM (RG) had 1209 NoFP.

[9] We did not conduct experiments in other dimension D^x to compare GFA-HMM with FA-HMM (RG) due to excessive computational requirements. However, we believe that the results could be representative.

Table 3
Word accuracy (in %) achieved on SUSAS database as a function of M^y

	Neutral	Angry	Fast	Slow	Question
GFA-HMM ($M^y = 2$)	92.94	61.35	80.63	73.91	87.13
GFA-HMM ($M^y = 3$)	95.88	64.42	83.95	75.32	89.11
GFA-HMM ($M^y = 4$)	95.29	63.91	83.33	79.71	89.11
GFA-HMM ($M^y = 10$)	90.59	59.51	79.63	69.48	83.17

GFA-HMMs were with $D^z = 2$, $D^x = 10$, and $M^x = 2$. Performances of GFA-HMM are measured against M^y, the number of Gaussian mixtures in observation node $y(t)$.

accuracy. With $M^y = 3$ and $M^y = 4$ respectively, GFA-HMM achieved 95.88% and 95.29% word accuracy in neutral speaking style, relatively higher than other settings of M^y. This observation is consistent in other speaking styles.

5.5. Performances as a function of number of training utterances

This section conducts experiments to reveal the trade-off of the standard HMM versus GFA-HMM as a function of the amount of training data. The first set of experiments conducted in Section 5.5.1 shows performances by GFA-HMM in multi-conditional training data, where noisy speech utterances are included to train a system. In Section 5.5.2, GFA-HMMs were trained with clean speech utterances.

5.5.1. Performances on multi-conditional training data

To test performances as a function of training utterances, we split the whole Aurora 2 multi-conditional training set (Pearce, 2000) into four subsets, where each of them consists of specific types of noise shown in Table 4. These training subsets are denoted by their number of utterances. For example, training subset 5229 consists of Babble, Car, and Exhibition noise, whereas subset 2154 only has Exhibition noise. Testing was done on the Aurora 2 testing set, which includes utterances in the four noise types ranging from −5 dB to clean in a 5 dB step. We measured the relative error rate reduction with respect to standard HMMs in each type of noise for a given training subset. A hypothesis testing was conducted to measure statistical significance of the relative error rate reduction.

Table 4
Training subsets and their included noise types

	Subway	Babble	Car	Exhibition
8440	Y	Y	Y	Y
5229	N	Y	Y	Y
3841	N	N	Y	Y
2154	N	N	N	Y

Signal-to-noise ratio ranges from clean, 20 dB to 5 dB in a −5 dB step.

The null hypothesis H_0 is that, there is no improvement by GFA-HMM relative to standard HMM. This hypothesis H_0 will be rejected if (1) there is statistically significant error rate reduction in those types of noise that does not appear in training subset, and (2) there is no statistically significant decrease of word accuracy in those types of noise that appears in training subset. The confidence level we set is 95%.

GFA-HMM had the following setup. Dimension of $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ were respectively 3 and 1. After initialization with parameters from traditional 18-state and 3 mixtures/state HMMs,[10] 5 iterations of training process in Section 3 were applied.

Tables 5–8 show word accuracies by GFA-HMM in each type of noise and SNR conditions, together with their ERRs in percentage and t-statistics of the ERRs in each type of noise.[11] For 95% confidence level and 7 types of SNR conditions, a threshold of 2.37 was set for statistical significance. For those t-statistics with absolute value smaller than the threshold, the ERRs are

[10] HMMs were trained with scripts from Aurora 2 database.

[11] t-Statistic is calculated by $\sqrt{n}\frac{mean(\mathbf{x})}{std(\mathbf{x})}$, where n is the number of samples in $\boldsymbol{x}$. Here $\boldsymbol{x}$ denotes ERRs in each SNR conditions.

Table 5
Word accuracy (in %) achieved by GFA-HMM on 8440 subset of Aurora 2

	Subway	Babble	Car	Exhibition
Clean	98.43	98.58	98.33	98.80
20 dB	98.04	97.52	97.82	97.25
15 dB	96.93	96.98	97.44	96.82
10 dB	94.41	95.34	95.47	94.01
5 dB	86.03	87.39	81.75	86.30
0 dB	54.50	62.91	40.32	58.99
−5 dB	20.76	28.30	17.83	20.46
ERR (in %)	−2.59	0.00	−2.66	−1.49
(*t*-statistic)	(−0.65)	(0.84)	(−1.89)	(−1.35)

GFA-HMMs were with $M^y = 3$, $M^x = 2$, $D^x = 3$, and $D^z = 1$. Standard HMMs were with $M^y = 3$. Both types of HMMs have 18 states.

Table 6
Word accuracy (in %) achieved by GFA-HMM on 5229 subset of Aurora 2

	Subway	Babble	Car	Exhibition
Clean	94.60	94.71	94.30	94.29
20 dB	94.14	96.22	96.18	95.62
15 dB	91.28	95.56	95.65	94.57
10 dB	86.61	93.20	92.72	90.68
5 dB	77.10	84.73	76.35	79.94
0 dB	53.52	59.43	35.28	44.25
−5 dB	20.79	28.04	16.55	16.41
ERR (in %)	6.82	0.42	−1.26	−1.45
(*t*-Statistic)	(3.35)	(1.15)	(0.33)	(−0.74)

GFA-HMMs were with $M^y = 3$, $M^x = 2$, $D^x = 3$, and $D^z = 1$. Standard HMMs were with $M^y = 3$. Both types of HMMs have 18 states.

Table 7
Word accuracy (in %) achieved by GFA-HMM on 3841 subset of Aurora 2

	Subway	Babble	Car	Exhibition
Clean	95.06	94.68	94.72	94.63
20 dB	95.09	93.17	96.03	95.74
15 dB	92.57	87.82	95.23	94.38
10 dB	88.49	79.41	92.69	90.81
5 dB	78.85	65.96	80.47	81.83
0 dB	56.25	38.60	40.05	52.27
−5 dB	22.54	12.62	16.52	17.80
ERR (in %)	1.81	2.36	−1.26	−40.65
(*t*-Statistic)	(3.06)	(4.31)	(0.49)	(−1.16)

GFA-HMMs were with $M^y = 3$, $M^x = 2$, $D^x = 3$, and $D^z = 1$. Standard HMMs were with $M^y = 3$. Both types of HMMs have 18 states.

Table 8
Word accuracy (in %) achieved by GFA-HMM on 2154 subset of Aurora 2

	Subway	Babble	Car	Exhibition
Clean	96.68	96.07	96.00	96.33
20 dB	95.58	80.65	93.02	96.20
15 dB	94.69	68.14	88.99	94.79
10 dB	91.34	53.05	78.62	91.73
5 dB	81.98	36.61	59.59	83.83
0 dB	56.19	19.56	29.14	50.69
−5 dB	20.14	8.10	12.23	17.59
ERR (in %)	0.93	1.15	1.97	−1.72
(*t*-statistic)	(1.69)	(2.84)	(3.11)	(−1.21)

GFA-HMMs were with $M^y = 3$, $M^x = 2$, $D^x = 3$, and $D^z = 1$. Standard HMMs were with $M^y = 3$. Both types of HMMs have 18 states.

considered to be non-significance. Otherwise, the ERRs are significant.

Observations of the results shown in these tables are as follows. First, Table 5 reveals that there is no statistical difference in performance between GFA-HMMs and standard HMMs for training with all types of noise. Through other tables with less amount of training data and fewer types of training noise, we found that GFA-HMM achieved statistically significant improvement with respect to standard HMM. For example, for training subset 5229 in Table 6, GFA-HMM had 6.82% ERR with t-statistic of 3.35 in Subway noise, which is not included in the training set. In other types of noise that occurs in training set, GFA-HMM does not decrease performances as compared to standard HMMs. In Table 8 for training subset 2154, we observed statistically significant improvement in Babble and Car noise that are not appeared in the training subset. The table also shows that there is no significant difference between GFA-HMMs and standard HMMs in Exhibition noise that appears in the training set.

5.5.2. Performances for training with 8440 clean utterances

GFA-HMMs were trained with 8440 clean speech utterances in Aurora 2 database. Testing set was the same in the previous section. Recognition accuracies by GFA-HMMs are shown in Table 9, together with the ERRs in each SNR

Table 9
Word accuracy (in %) achieved by GFA-HMM on 8440 clean training utterances on Aurora 2

	Subway	Babble	Car	Exhibition	ERRs (in %)
Clean	98.22	98.46	97.71	98.33	−0.55
20 dB	77.86	83.01	87.32	78.96	0.57
15 dB	65.43	65.72	69.73	67.63	0.11
10 dB	51.09	45.04	48.34	54.15	0.58
5 dB	30.70	26.18	24.43	33.14	0.50
0 dB	12.62	14.24	9.31	12.43	0.09
−5 dB	7.92	7.80	7.01	7.37	0.06
Average ERR (in %) from 20 dB to 0 dB					0.33
(*t*-statistic)					(3.33)

GFA-HMMs were with $M^y = 3$, $M^x = 2$, $D^x = 3$, and $D^z = 1$. Standard HMMs were with $M^y = 3$. Both types of HMMs have 18 states.

conditions shown in the last column. The ERRs achieved in 20–0 dB are averaged and its value is shown in the right corner of the table. It is seen that GFA-HMM attained 0.33% of Averaged ERR relative to HMM in noisy conditions. Although the absolute value of improvement was not large, the improvement was consistent in SNRs from 20 dB to 0 dB. The t-statistic of the ERRs was 3.33, showing that the improvement was also statistically significant in 95% confidence level. Performance in clean conditions did not degrade significantly, because the t-statistic in clean condition was a small value of −0.15.[12]

6. Discussion

A generative model using continuous-valued latent representation for speech recognition was proposed by Saul and Rahim (2000). In their model, the covariance of the observation vectors is modeled by factor analysis. This corresponds to introducing a state-dependent factor analysis into the standard HMMs. The model can be directly obtained from GFA-HMM by imposing the following two constraints: (1) $\boldsymbol{x}(t)$ is distributed as a standard Gaussian $\mathcal{N}(\mathbf{x}(t); \mathbf{0}, \mathbf{I})$, (2) There is no noise $\boldsymbol{\zeta}(t)$ in $\boldsymbol{x}(t)$. EM algorithm derived for GFA-HMM can also be directly applied to the model by Saul and Rahim (2000). Reducing GFA-HMM to one state and zero dimension of $\boldsymbol{z}(t)$ may obtain factor analyzed Gaussian mixture models (FA-GMMs), which are applied for speaker identification (Ding et al., 2002; Yamamoto et al., 2004).

A recent model proposed by Rosti and Gales (2002) uses mixture of Gaussians to model the densities of $\boldsymbol{x}(t)$, which is the continuous-valued latent vectors to generate observation vectors. This model can be obtained by setting dimension of $\boldsymbol{z}(t)$ to be zero in GFA-HMM.

With the introduction of the hierarchical continuous-valued latent representation, GFA-HMM is more flexible than the standard HMM. By tuning latent dimensions and number of mixtures in the latent spaces, compact representation of observations can be achieved via the GFA-HMM. This gives strength to the model for acoustic modeling with fewer parameters but may achieve higher performances comparable to standard HMM.

In the view of computational complexity, the GFA-HMM does need more computational resources than the standard HMM, because of this hierarchical continuous-valued latent representation. However, we may pre-calculate the inversed matrices in order to speed up the program, because most of the computations are due to calculations of the matrix operations, e.g., matrix inverse, By such scheme, we did not see dramatic increase of computation costs. To train GFA-HMMs, a Pentium 4 computer may take 1–2 h for one iteration of 8440 digits utterances, which is comparable to the time to train a standard HMM.

An important issue that deserves further work is the determination of the number D^x, the dimension of the latent vector $\boldsymbol{x}(t)$. D^x is not a simple

[12] The value is not shown in the table.

parameter since increasing the number increases dimension of the driving noises $\boldsymbol{\zeta}(t)$. As a result, the model might not fit to real data, if the true dimension D^x is not that large. For this reason, a model selection scheme is suggested as a further work.

So far, our empirical results (in Sections 5.3 and 5.4) show that, performances of GFA-HMM depend on the underlying dimension D^x and the number of mixture components M^y, especially for multiple speaking styles. For example, whereas $D^x = 5$ is optimal for slow speech, it may be optimal at 10 for fast speech. Due to excessive computational costs, we have not yet made many experiments to 'select' the best model for each speaking styles. However, based on our experimental results so far, we may suggest that $D^x = 10$ is probably good for many speaking styles. Regarding M^y, our observation of an optimal M^y in Section 5.4.2 is consistent to the results on FA-HMM (Rosti and Gales, 2002) reported in (Rosti and Gales, 2001). Performances of their proposed factor analyzed HMMs (Rosti and Gales, 2002) also depend on M^y. Increasing M^y does not have a non-decreasing performance improvement. On the contrary, large M^y may hurt performances. Based on our results, we suggest $M^y = 3$ for GFA-HMMs.

7. Conclusions

We presented a model for representing observations using a generative factor-analyzed HMM (GFA-HMM) by introducing a hierarchical continuous-valued latent representation, which also depends on the discrete hidden state sequence of the HMM. The observation vectors are interpreted as arising from state-dependent hierarchical continuous-valued vectors, where the covariance matrix is modeled by factor analysis. Noise in the observation is modeled by state-dependent mixture of Gaussians. The model reduces to standard HMM when the observation vectors are dependent on the state-dependent mixture of Gaussians solely. EM algorithm is derived in this paper for parameter estimation of the hierarchical latent representation. The EM algorithm reduces to the well-known EM algorithm for training standard HMM when the observation vectors are not dependent on the hierarchical continuous-valued latent representation. The model may achieve a more compact representation of the observations when compared to standard HMM. In a set of experiments, we verified that, with properly selected latent dimensions and mixture components in the latent spaces, the proposed acoustic model achieves consistent and statistically significant improvement relative to standard HMMs, especially in conditions with unseen noise and speaking styles. This may be advantageous for speech recognition with limited amount of data (e.g., spontaneous speech and fewer types of training noise).

Acknowledgement

The authors thank two anonymous reviewers for their comments that improved presentation of the paper. The first author thanks Dr. Jianwu Dang and Dr. Satoshi Nakamura at ATR SLT for helpful discussions. Part of the work was done when the first author was with ATR SLT.

Appendix A. Derivation of posterior mean and covariance of continuous latent variable

Posterior mean and covariance of $\boldsymbol{x}(t)$ given observed vector $\boldsymbol{y}(t)$ and the associated state, q, and mixture index, m can be derived in the following way. First, assume that $\mathbf{x}(t) \sim \mathcal{N}(\mathbf{x}(t); \boldsymbol{\xi}^x, \mathbf{V}^x)$. This provides the prior distribution of $\boldsymbol{x}(t)$. Then, notice that, given q, m and $\boldsymbol{x}(t)$, observed vector $\boldsymbol{y}(t)$ is distributed in $\mathcal{N}(\mathbf{y}(t); \boldsymbol{\Lambda}\mathbf{x}(t) + \boldsymbol{\mu}^y_{qm}, \boldsymbol{\Sigma}^y_{qm})$. Write $\boldsymbol{x}(t)$ as a function of $\boldsymbol{y}(t)$, which gives $\boldsymbol{x}(t) = (\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{-1}\ \boldsymbol{\Lambda}^T(\boldsymbol{y} - \boldsymbol{v}_{qt})$ according to Eq. (7). This term provides a distribution of $\boldsymbol{x}(t)$ without *a priori* as

$$\mathbf{x}(t) \sim \mathcal{N}(\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^{\mathrm{T}}(\mathbf{y} - \boldsymbol{\mu}^y_{qm}), \quad (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Sigma}^y_{qm}((\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^{\mathrm{T}})^{\mathrm{T}}. \tag{A.1}$$

Finally, product the above distribution with the prior distribution of $\boldsymbol{x}(t)$, and use the rule of product of Gaussian density (e.g., in Frey, 1999). Eqs. (24) and (25) are arrived as a result.

Appendix B. EM algorithm

We provide a more detailed derivation for re-estimation formulae in Section 3.2. We begin by evaluating auxiliary function (18) with posterior statistics obtained in Section 3.1. For example, up to a constant D, referring to Eq. (8), the last component in the right of Eq. (18) can be expressed as

$$\begin{aligned} & E_{\Theta}\left[\log \prod_{t=1}^{T}\{p(\mathbf{y}(t)|\mathbf{x}(t), q_t, m(t), \widetilde{\Theta})\}^{\delta_{qm}(t)}\right] \\ & = D - \frac{1}{2}\sum\nolimits_t \gamma_{qm}(t)\log|\widetilde{\boldsymbol{\Sigma}}^{y}_{qm}| - \frac{1}{2}\sum\nolimits_t \gamma_{qm}(t) \\ & \quad \times [(\mathbf{y}(t) - \widetilde{\boldsymbol{\mu}}^{y}_{qm} - \widetilde{\boldsymbol{\Lambda}}\boldsymbol{\phi}^{x}(t))^{\mathrm{T}}(\widetilde{\boldsymbol{\Sigma}}^{y}_{qm})^{-1}(\mathbf{y}(t) - \widetilde{\boldsymbol{\mu}}^{y}_{qm} \\ & \quad - \widetilde{\boldsymbol{\Lambda}}\boldsymbol{\phi}^{x}(t)) + \mathrm{tr}(\widetilde{\boldsymbol{\Lambda}}\boldsymbol{\Psi}^{x}(t)\widetilde{\boldsymbol{\Lambda}}^{\mathrm{T}}(\widetilde{\boldsymbol{\Sigma}}^{y}_{qm})^{-1})], \end{aligned} \tag{B.1}$$

where $tr(\cdot)$ is the trace operation. By setting the first order derivative of the above function w.r.t. $\{\widetilde{\boldsymbol{\Lambda}}, \widetilde{\boldsymbol{\mu}}^{y}_{qm}, \widetilde{\boldsymbol{\Sigma}}^{y}_{qm}\}$ to zero, updating formulae for the parameters can be obtained as shown in Eq. (34)–(36).

Note that the loading matrix $\widehat{\boldsymbol{\Lambda}}$ in Eq. (34) has to be estimated row by row (Woodland et al., 1996). The nth row vector $\widehat{\lambda}_n$ of the new loading matrix $\widehat{\boldsymbol{\Lambda}}$ can be written as

$$\widehat{\lambda}_n = \mathbf{k}_n^{\mathrm{T}}\mathbf{G}_n^{-1}, \tag{B.2}$$

where the D^x by D^x matrices $\boldsymbol{G}_n$ and D^x dimensional vector are given as

$$\mathbf{G}_n = \sum_t \sum_q \sum_{m=1}^{M^y} \gamma_{qm}(t)\frac{1}{\sigma^2_{qmn}}(\boldsymbol{\phi}^{x}(t)\boldsymbol{\phi}^{x}(t)^{\mathrm{T}} + \boldsymbol{\Psi}^{x}(t)) \tag{B.3}$$

$$\mathbf{k}_n = \sum_t \sum_q \sum_{m=1}^{M^y} \gamma_{qm}(t)\frac{1}{\sigma^2_{qmn}}(y_n(t) - \boldsymbol{\mu}^{y}_{qmn})\boldsymbol{\phi}^{x}(t), \tag{B.4}$$

where $y_n(t)$ and $\boldsymbol{\mu}^{y}_{qmn}$ are, respectively, the nth element of observation vector $\boldsymbol{y}(t)$ and the nth element of the observation noise mean vector $\boldsymbol{\mu}^{y}_{qm}$.

Note that the loading matrix $\widehat{\mathbf{C}}$ in Eq. (39) should be calculated row-by-row similarly as that for $\widehat{\boldsymbol{\Lambda}}$; i.e.,

$$\widehat{C}_n = \mathbf{l}_n^{\mathrm{T}}\mathbf{H}_n^{-1}, \tag{B.5}$$

where the D^z by D^z matrices $\boldsymbol{H}_n$ and D^z dimensional vector are given as

$$\mathbf{H}_n = \sum_t \sum_{j=1}^{M^x} \gamma_j(t)\frac{1}{(v^{x}_{jn})^2}(\boldsymbol{\phi}^{z}(t)\boldsymbol{\phi}^{z}(t)^{\mathrm{T}} + \boldsymbol{\Psi}^{z}(t)), \tag{B.6}$$

$$\mathbf{l}_n = \sum_t \sum_{j=1}^{M^x} \gamma_j(t)\frac{1}{(v^{x}_{jn})^2}(\boldsymbol{\phi}^{x}(t) - \boldsymbol{\xi}^{x}_{jn})\boldsymbol{\phi}^{z}(t), \tag{B.7}$$

where v^{x}_{jn} is the nth element in variance $\mathbf{V}^{x}_{j}$.

References

Amari, S.-I., Cichocki, A., Yang, H.H., 2000. Blind Signal Separation and Extraction: Neural and Information-theoretic Approaches. Wiley, pp. 63–138.

Attias, H., 1998. Independent factor analysis. Neural Comput. 11, 803–851.

Bell, A.J., Sejnowski, T.J., 1995. A Non-linear Information Maximization Algorithm That Performs Blind Separation. Advances in Neural Information Processing Systems. MIT Press, pp. 467–474.

Bellegarda, J., Nahamoo, D., 1990. Tied mixture continuous parameter modeling for speech recognition. IEEE Trans. Acoust. Speech, Signal Process. 38, 2033–2045.

Cardoso, J.-F., 1997. Infomax and maximum likelihood for source separation. IEEE Lett. Signal Process. 4, 112–114.

Comon, P., 1994. Independent component analysis: a new concept? Signal Process. 36, 287–314.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. 3g, 1–38.

Ding, P., Liu, Y., Xu, B., 2002. Factor analyzed Gaussian mixture models for speaker identification. In: ICSLP. pp. 1341–1344.

Everitt, B.S., 1984. An Introduction to Latent Variable Models. Chapman and Hall.

Frey, B., 1999. Factor analysis using batch and online EM. Internal UW/CS Adaptive Computation TR-99-2. University of Waterloo.

Gales, M.J.F., 1999. Semi-tied covariance matrices for hidden Markov models. IEEE Trans. Speech and Audio Process. 7 (3), 272–281.

Gopinath, R., 1998. Maximum likelihood linear modeling with Gaussian distributions for classification. In: ICASSP. pp. 12–15.

Hansen, J.H.L. et al., 1998. Getting started with the SUSAS: Speech under simulated and actual stress database. Technical report: RSPL - 98 - 10. Robust speech processing laboratory, Colorado University.

Hyvarinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. John Wiley & Sons, Inc.

Kumar, N., 1997. Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition. Ph.D. thesis. Johns Hopkins University.

Olsen, P., Gopinath, R., 2002. Modeling inverse covariance matrices by basis expansion. In: ICASSP. pp. 13–17.

Pearce, D., 2000. Aurora project: Experimental framework for the performance evaluation of distributed speech recognition front-ends. In: ISCA ITRW ASR2000.

Pearlmutter, B., Parra, L., 1997. Maximum likelihood blind source separation: a context-sensitive generalization of ICAAdvances in Neural Information Processing Systems, Vol. 9. MIT Press, pp. 613–619.

Rabiner, L., Juang, B.-H., 1993. Fundamentals of Speech Recognition. Prentice Hall PTR.

Rosti, A-V.I., Gales, M.J.F., 2001. Linear Gaussian models for speech recognition. CUED/F-INFENG/TR 420. Cambridge University.

Rosti, A.-V.I., Gales, M.J.F., 2002. Factor analysed hidden Markov models. In: ICASSP. pp. 949–952.

Roweis, S., Ghahramani, Z., 1999. A unifying review of linear Guassian models. Neural Comput. 11 (2), 305–345.

Rubin, D., Thayer, D., 1982. EM algorithms for ML factor analysis. Psychometrika 47 (1), 69–76.

Saul, L.K., Rahim, M.G., 2000. Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. IEEE Trans. Speech and Audio Process. 8 (2), 115–125.

Tipping, M.E., Bishop, C.M., 1997. Mixtures of probabilistic principal component analysis. Technical report, NCRG-97-003. Aston University.

Yamamoto, H. et al., 2004. Parameter sharing and minimum classification error training of mixture of factor analyzers for speaker identification. In: ICASSP. pp. 29–32.

Viterbi, A.J., 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans. Information Theory 13, 260–269.

Woodland, P.C., Gales, M.J.F., Pye, D., 1996. Improving environmental robustness in large vocabulary speech recognition. In ICASSP. pp. 65–68.

Young, S. et al., 1997. The HTK Book. version 2.1.