EURASIP Journal on
Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Speaker adaptation in the maximum *a posteriori* framework based on the probabilistic 2-mode analysis of training models

Yongwon Jeong

Abstract

In this article, we describe a speaker adaptation method based on the probabilistic 2-mode analysis of training models. Probabilistic 2-mode analysis is a probabilistic extension of multilinear analysis. We apply probabilistic 2-mode analysis to speaker adaptation by representing each of the hidden Markov model mean vectors of training speakers as a matrix, and derive the speaker adaptation equation in the maximum *a posteriori* (MAP) framework. The adaptation equation becomes similar to the speaker adaptation equation using the MAP linear regression adaptation. In the experiments, the adapted models based on probabilistic 2-mode analysis showed performance improvement over the adapted models based on Tucker decomposition, which is a representative multilinear decomposition technique, for small amounts of adaptation data while maintaining good performance for large amounts of adaptation data.

Keywords: Speech recognition, Speaker adaptation, Probabilistic tensor analysis, Tucker decomposition

1 Introduction

In automatic speech recognition (ASR) systems using hidden Markov models (HMMs) [1], mismatches between the training and testing conditions lead to performance degradation. One of such mismatches results from speaker variation. Thus, speaker adaptation techniques [2] are employed to transform a well-trained canonical model (e.g., speaker-independent (SI) HMM) to the target speaker. Speaker adaptation requires fewer adaptation data than needed to build a speaker-dependent (SD) model. Among speaker adaptation techniques, eigenvoice (EV) [3] expresses the model of a new speaker as a linear combination of basis vectors, which are built from the principal component analysis (PCA) of the HMM mean vectors of training speakers.

In a similar approach, speaker adaptation based on tensor analysis using Tucker decomposition [4] was investigated in [5], where bases were constructed from the multilinear decomposition of a tensor that consisted of the HMM mean vectors of training speakers. In the approach, all the training models were collectively arranged in a third-order tensor (3-D array):

$$\mathcal{M}^{R\times D\times S} \tag{1}$$

where the first, second, and third modes (dimensions) were for the mixture component, dimension of the mean vector, and training speaker. In [5], Tucker decomposition was used to build bases and in the experiments, speaker adaptation using Tucker decomposition showed better performance than eigenvoice and maximum likelihood linear regression (MLLR) adaptation [6]. The improvement seemed to be attributable to the increased number of adaptation parameters and compact bases. Also noticed in [5] was that the increased number of adaptation parameters did not guarantee a good performance when the amount of adaptation data was small (the determination of the proper number of adaptation parameters for given adaptation data is a model-order selection problem). Extending the tensor-based approach, in [7], the fourth mode for noise was added (so, $\mathcal{M}$ became a 4-D array) so that the training models of various speakers and noise conditions were decomposed.

In this article, we describe a speaker adaptation method using probabilistic 2-mode analysis, which is an application of probabilistic tensor analysis (PTA) [8] to the second-order tensor (i.e., matrix); PTA is an application

Correspondence: jeongy@pusan.ac.kr
School of Electrical Engineering, Pusan National University, Busan 609–735, Republic of Korea

Springer

© 2013 Jeong; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of probabilistic PCA (PPCA) [9] to tensor objects. Using probabilistic 2-mode analysis, we derive bases from training models in a probabilistic framework, and formulate the speaker adaptation equation in the maximum *a posteriori* (MAP) framework [10]. The speaker adaptation equation based on the probabilistic approach becomes similar to MAP linear regression (MAPLR) adaptation [11] as shown below. The experiments showed that the proposed method further improved the performance of the speaker adaptation based on Tucker decomposition for small amounts of adaptation data.

The rest of this article is organized as follows. Section 2.1 explains some tensor algebra and tensor decomposition. Section 2.3 explains the probabilistic 2-mode analysis of a set of mean vectors of training HMMs. In Section 2.5, the estimation of the prior distribution of the adaptation parameter is described. Section 2.6 describes the speaker adaptation in the MAP framework using the bases and the prior. Section 2.2 describes the speaker adaptation using Tucker decomposition, which is compared with the probabilistic 2-mode analysis-based method. We explain the experiments in Section 3 and conclude the article in Section 4. Some of the notations used in this article are summarized in Table 1.

2 Methods

2.1 Multilinear decomposition

Following the convention of multilinear algebra, we denote vectors, matrices, and tensors by lowercase boldface letters (e.g., $\mathbf{m}$), uppercase boldface letters (e.g., $\mathbf{M}$), and calligraphic letters (e.g., $\mathcal{M}$), respectively, in this article.

A tensor is a multidimensional array, and an N-dimensional array is called the Nth-order tensor (or N-way array). The order of a tensor is the number of indices for addressing the tensor; so the order of $\mathcal{M}^{I_1 \times I_2 \times \cdots \times I_N}$ is N. Scalar, vector, and matrix are zeroth-, first-, and second-order tensors, respectively. There are three indices for addressing the array in a third-order tensor as depicted in Figure 1.

Tensor algebra is performed in terms of matrix and vector representations of tensors; the mode-n flattening (matricization) of tensor $\mathcal{M}$, which is denoted as $\mathbf{M}_{(n)}$, is obtained by reordering the elements as follows:

$$\mathbf{M}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{(n-1)} \times I_{(n+1)} \times \cdots \times I_N)}. \tag{2}$$

That is, all the column vectors along the mode n are arranged into a matrix. For example, a third-order tensor $\mathcal{M}^{I \times J \times K}$ can be flattened into an $I \times (JK)$, $J \times (KI)$, or $K \times (IJ)$ matrix as depicted in Figure 2; for a $\mathcal{M}^{2 \times 2 \times 2}$ tensor:

$$\mathcal{M} = \{\mathbf{M}(:,:,1)\ \mathbf{M}(:,:,2)\} \tag{3}$$

$$\mathbf{M}(:,:,1) = \begin{bmatrix} m_{111} & m_{121} \\ m_{211} & m_{221} \end{bmatrix}, \quad \mathbf{M}(:,:,2) = \begin{bmatrix} m_{112} & m_{122} \\ m_{212} & m_{222} \end{bmatrix},$$

the mode-n flattening is given as:

$$\begin{aligned} \mathcal{M} &\rightarrow \left[\begin{array}{c|c} m_{111} & m_{121} \\ m_{211} & m_{221} \end{array}\right] \Bigg| \left[\begin{array}{c|c} m_{112} & m_{122} \\ m_{212} & m_{222} \end{array}\right] \\ &\rightarrow \mathbf{M}_{(1)} = \begin{bmatrix} m_{111} & m_{121} & m_{112} & m_{122} \\ m_{211} & m_{221} & m_{212} & m_{222} \end{bmatrix} \\ \mathcal{M} &\rightarrow \left[\begin{array}{cc} m_{111} & m_{121} \\ \hline m_{211} & m_{221} \end{array}\right] \Bigg| \left[\begin{array}{cc} m_{112} & m_{122} \\ \hline m_{212} & m_{222} \end{array}\right] \\ &\rightarrow \mathbf{M}_{(2)} = \begin{bmatrix} m_{111} & m_{112} & m_{211} & m_{212} \\ m_{121} & m_{122} & m_{221} & m_{222} \end{bmatrix} \\ \mathcal{M} &\rightarrow \left[\begin{array}{c|c} m_{111} & m_{121} \\ \hline m_{211} & m_{221} \end{array}\right] \left[\begin{array}{c|c} m_{112} & m_{122} \\ \hline m_{212} & m_{222} \end{array}\right] \\ &\rightarrow \mathbf{M}_{(3)} = \begin{bmatrix} m_{111} & m_{211} & m_{121} & m_{221} \\ m_{112} & m_{212} & m_{122} & m_{222} \end{bmatrix}. \end{aligned} \tag{4}$$

The operation of the mode-n flattening will be denoted as $\mathrm{mat}_n(\cdot)$, i.e., $\mathrm{mat}_n(\mathcal{M}) = \mathbf{M}_{(n)}$.

Table 1 Notations used in the article

Notation	Meaning
r	Index for the mixture component $(1, \ldots, R)$
s	Index for the training speaker $(1, \ldots, S)$
D	Dimension of the acoustic feature vector
$\boldsymbol{\mu}$	HMM mean vector
$\mathbf{M}$	Matrix representation of HMM mean vector
$\boldsymbol{\mathcal{M}}$	Tensor representation of training models
$\boldsymbol{\mathcal{G}}$	Core tensor
$\mathbf{U}$	Mode matrix, factor loading matrix
$\mathbf{w}$	Weight vector
$\mathbf{W}$	Weight matrix, latent matrix
$\mathbf{C}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Sigma}$	covariance matrix
$\mathbf{E}$	Error matrix

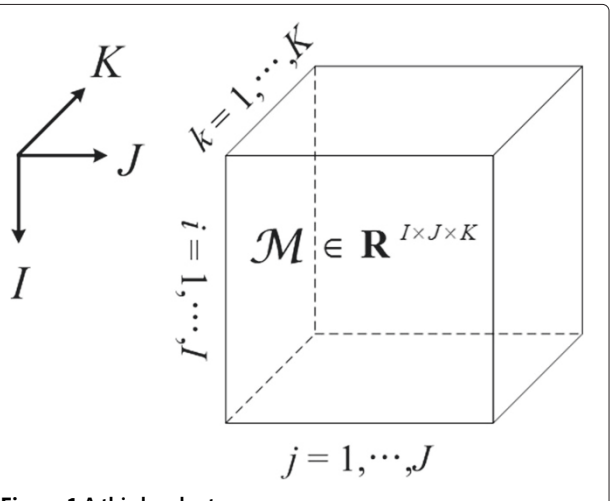

Figure 1 A third-order tensor.

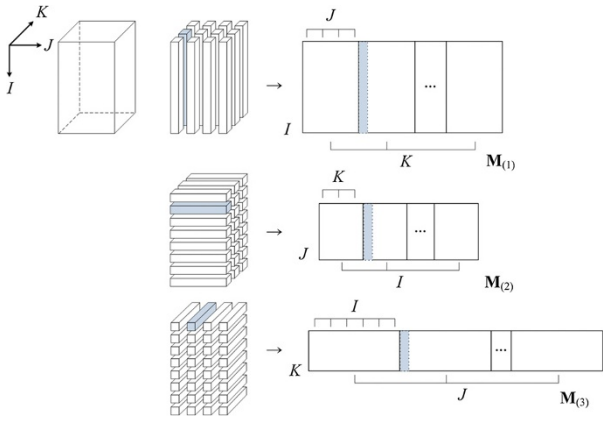

Figure 2 Mode-*n* flattening of a third-order tensor.

Multiplication of a tensor and a matrix is performed by *n*-mode product; the *n*-mode product of a tensor $\mathcal{W}$ with a matrix $\mathbf{U}$ is denoted as

$$\mathcal{M} = \mathcal{W} \times_n \mathbf{U} \tag{5}$$

and is carried out by matrix multiplication in terms of flattened matrices:

$$\mathbf{M}_{(n)} = \mathbf{U}\,\mathbf{W}_{(n)} \tag{6}$$

or elementwise

$$\left(\mathcal{W} \times_n \mathbf{U}\right)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} w_{i_1 i_2 \ldots i_N} u_{j i_n} \tag{7}$$

where w and u denote the elements of $\mathcal{W}$ and $\mathbf{U}$, respectively. If $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathbf{U}^T \in \mathbb{R}^{K_n \times I_n}$, then the dimension of $\mathcal{W} \times_n \mathbf{U}^T$ becomes $I_1 \times I_2 \times \cdots \times I_{n-1} \times K_n \times I_{n+1} \times \cdots \times I_N$.

As an extension of singular value decomposition (SVD) to tensor objects, Tucker decomposition decomposes a tensor as follows [4]:

$$\mathcal{M}^{I_1 \times I_2 \times \cdots \times I_N} \simeq \mathcal{W}^{K_1 \times K_2 \times \cdots \times K_N} \prod_{n=1}^{N} \times_n \mathbf{U}_n \tag{8}$$

where $\mathbf{U}_n \in \mathbb{R}^{I_n \times K_n}$, $K_n \leq I_n$ $(n = 1, \ldots, N)$. The core tensor $\mathcal{W}$ and mode matrices $\mathbf{U}_n$'s correspond to the matrices of singular values and orthonormal basis vectors in matrix SVD, respectively. An example of Tucker decomposition of a third-order tensor is illustrated in Figure 3.

The core tensor $\mathcal{W}$ and mode matrices $\mathbf{U}_n$'s in Tucker decomposition can be computed such that they minimize

$$\text{Error} = \left\| \mathcal{M} - \mathcal{W} \prod_{n=1}^{N} \times_n \mathbf{U}_n \right\|^2 \tag{9}$$

where the norm of a tensor is defined as $\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N}^2}$. A representative technique for Tucker decomposition is the alternating least squares (ALS) [12]; the basic idea is to compute each mode matrix $\mathbf{U}_n$ alternatingly with other mode matrices fixed. For more details on Tucker decomposition, refer to [4]. In the following section, we explain probabilistic 2-mode analysis in the context of speaker adaptation.

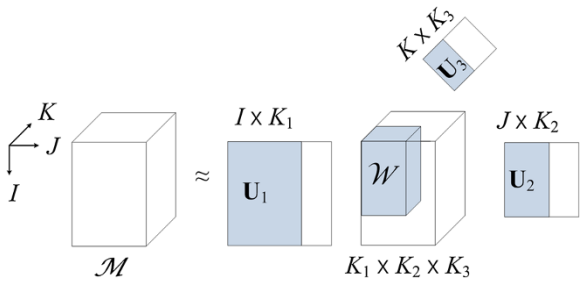

Figure 3 Tucker decomposition of a third-order tensor.

2.2 Speaker adaptation using Tucker decomposition

The probabilistic 2-mode analysis based method is a probabilistic extension of the Tucker decomposition based method. Thus, we compare the probabilistic approach with the Tucker decomposition based method in the experiments. In this section, we explain the speaker adaptation based on the Tucker decomposition of training models in [5]. In this article, speaker adaptation is performed by updating the mean vectors of the output distribution of an HMM. The HMM mean vectors of each training speaker are arranged in an $R \times D$ matrix:

$$\mathbf{M}_s = \left[\boldsymbol{\mu}_{s;1} \ldots \boldsymbol{\mu}_{s;r} \ldots \boldsymbol{\mu}_{s;R}\right]^T, \quad s = 1, \ldots, S. \tag{10}$$

Here, $\boldsymbol{\mu}_{s;r}$ denotes the mean vector corresponding to mixture r of the sth training speaker model.

All the centered HMM mean vectors of training speakers, $\left\{\mathbf{M}_s - \overline{\mathbf{M}}\right\}_{s=1}^{S}$ where $\overline{\mathbf{M}} = 1/S \sum_s \mathbf{M}_s$, are collectively expressed as a third-order tensor $\widetilde{\mathcal{M}}$, and we decompose the training tensor by Tucker decomposition as follows:

$$\begin{aligned} \widetilde{\mathcal{M}}^{R \times D \times S} &\simeq \mathcal{G}^{K_R \times K_D \times K_S} \times_1 \mathbf{U}_{\text{mixture}} \times_2 \mathbf{U}_{\text{dim}} \times_3 \mathbf{U}_{\text{speaker}} \\ &= \left(\mathcal{G}^{K_R \times K_D \times K_S} \times_3 \mathbf{U}_{\text{speaker}}\right) \times_1 \mathbf{U}_{\text{mixture}} \times_2 \mathbf{U}_{\text{dim}}. \end{aligned} \tag{11}$$

In the above equation, $\mathbf{U}_{\text{mixture}} \in \mathbb{R}^{R \times K_R}$, $\mathbf{U}_{\text{dim}} \in \mathbb{R}^{D \times K_D}$, and $\mathbf{U}_{\text{speaker}} \in \mathbb{R}^{S \times K_S}$ are basis matrices for the mixture component, dimension of the mean vector, and training speaker, respectively ($K_R \leq R - 1$, $K_D \leq D - 1$, and $K_S \leq S - 1$); the core tensor $\mathcal{G}$ is common across the mixture component, dimension of the mean vector, and training speaker. In Equation (11), the sth row vector of $\mathbf{U}_{\text{speaker}}$, which is denoted as $\mathbf{u}_{\text{speaker};s}$, corresponds to the speaker weight of the sth speaker, thus the low-rank approximation of the sth speaker model is given by

$$\mathbf{M}_s \simeq \left(\mathcal{G}^{K_R \times K_D \times K_S} \times_3 \mathbf{u}_{\text{speaker};s}\right) \times_1 \mathbf{U}_{\text{mixture}} \times_2 \mathbf{U}_{\text{dim}} + \overline{\mathbf{M}}. \tag{12}$$

If we define the augmented speaker weight $\mathbf{W}_s^{K_R \times K_D} \equiv \mathcal{G}^{K_R \times K_D \times K_S} \times_3 \mathbf{u}_{\text{speaker};s}$, Equation (12) becomes

$$\begin{aligned}\mathbf{M}_s &\simeq \mathbf{W}_s \times_1 \mathbf{U}_{\text{mixture}} \times_2 \mathbf{U}_{\text{dim}} + \overline{\mathbf{M}} \\ &= \mathbf{U}_{\text{mixture}} \mathbf{W}_s \mathbf{U}_{\text{dim}}^T + \overline{\mathbf{M}}.\end{aligned} \tag{13}$$

Thus, we express the model of a new speaker as

$$\mathbf{M}_{\text{new}} = \mathbf{U}_{\text{mixture}} \mathbf{W}_{\text{new}} \mathbf{U}_{\text{dim}}^T + \overline{\mathbf{M}}. \tag{14}$$

For the given adaptation data $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, we derive the equation for finding the speaker weight in a maximum likelihood (ML) criterion:

$$\begin{aligned}&\sum_t \sum_r \gamma_r(t)\, \mathbf{C}_r^{-1} \underbrace{\mathbf{U}_{\text{dim}} \mathbf{W}_{\text{new}}^T}_{\equiv \mathbf{W}_{\text{new,aug}}^T} \mathbf{u}_{\text{mixture};r}^T \mathbf{u}_{\text{mixture};r} \\ &= \sum_t \sum_r \gamma_r(t)\, \mathbf{C}_r^{-1} \left(\mathbf{o}_t - \overline{\mathbf{m}}_r^T\right) \mathbf{u}_{\text{mixture};r}\end{aligned} \tag{15}$$

where $\gamma_r(t)$ denotes the occupation probability of being at mixture r at t given $\mathbf{O}$, $\mathbf{C}_r$ the covariance matrix of the rth Gaussian component of an SI HMM (in this article, a diagonal covariance matrix is used); $\mathbf{u}_{\text{mixture};r}$ and $\overline{\mathbf{m}}_r$ denote the rth row vectors of $\mathbf{U}_{\text{mixture}}$ and $\overline{\mathbf{M}}$, respectively. In the above equation, $\mathbf{W}_{\text{new, aug}}$ can be computed using a technique similar to MLLR adaptation and the weight of the new speaker is obtained by

$$\hat{\mathbf{W}}_{\text{new}} = \mathbf{W}_{\text{new, aug}} \mathbf{U}_{\text{dim}} \tag{16}$$

which is plugged into Equation (14) to produce the model updated for the new speaker.

2.3 Probabilistic 2-mode analysis

The advantage of probabilistic 2-mode analysis over Tucker decomposition is similar to that of PPCA over standard PCA; probabilistic 2-mode analysis can deal with missing entries in the data tensor (although this is not the case in our experiments). In the modeling perspective, probabilistic 2-mode analysis assumes a distribution of latent variables, thus it is suitable for a MAP framework.

In this section, the ensemble of training models is expressed as

$$M = \{\mathbf{M}_s\}_{s=1}^S. \tag{17}$$

Assuming the HMM mean vectors of training speakers are drawn from the matrix-variate normal distribution [13], we derive the adaptation equation based on the probabilistic 2-mode analysis of training models. We use probabilistic 2-mode analysis, the second-order case of PTA [8], to decompose the training models expressed in matrix form. The latent tensor model is expressed as

$$\mathcal{M} = \mathcal{W} \prod_{n=1}^N \times_n \mathbf{U}_n + \mathcal{M}_{\text{mean}} + \mathcal{E} \tag{18}$$

where $\mathcal{W}$ denotes the latent tensor, $\mathbf{U}_n$'s the factor loading matrices, $\mathcal{M}_{\text{mean}}$ the mean, and $\mathcal{E}$ is the error/noise process. The 2-mode case of the latent tensor model is given by

$$\mathbf{M} = \mathbf{W} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 + \mathbf{M}_{\text{mean}} + \mathbf{E} \tag{19}$$

which becomes, for the training models $\{\mathbf{M}_1, \ldots, \mathbf{M}_S\}$,

$$\begin{aligned}\mathbf{M}_s &= \mathbf{W}_s \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 + \mathbf{M}_{\text{mean}} + \mathbf{E}_s \\ &= \mathbf{U}_{\text{mixture}}\, \mathbf{W}_s\, \mathbf{U}_{\text{dim}}^T + \mathbf{M}_{\text{mean}} + \mathbf{E}_s\end{aligned} \tag{20}$$

where $\mathbf{W}_s \in \mathbb{R}^{K_R \times K_D}$ denotes the latent matrix, $\mathbf{U}_{\text{mixture}} \in \mathbb{R}^{R \times K_R}$ and $\mathbf{U}_{\text{dim}} \in \mathbb{R}^{D \times K_D}$ the factor loading matrices ($K_R \leq R - 1$ and $K_D \leq D - 1$), $\mathbf{M}_{\text{mean}}$ the mean, and $\mathbf{E}_s$ the error/noise process. (Mode matrices and dimensions are defined as follows: $\mathbf{U}_1 = \mathbf{U}_{\text{mixture}}$, $\mathbf{U}_2 = \mathbf{U}_{\text{dim}}$, $I_1 = R$, $I_2 = D$, $K_1 = K_R$, and $K_2 = K_D$.) The distribution of $\mathbf{W}_s$ is assumed to be a matrix-variate normal, i.e., $\mathbf{W}_s \sim \mathcal{N}(\mathbf{0}_{K_R \times K_D}, \mathbf{I}_{K_R} \otimes \mathbf{I}_{K_D})$ where $\otimes$ denotes the Kronecker product, and independent of $\mathbf{E}_s$ whose elements follow $\mathcal{N}(0, \sigma^2)$. Figure 4 shows the graphical model representing the probabilistic 2-mode model.

In Equation (20), it is computationally intractable to calculate $\mathbf{U}_n$'s simultaneously. So, the following decoupled predictive density is defined:

$$p\left(\mathbf{M} | \mathbf{M}_{\text{mean}}, \{\mathbf{U}_n\}_{n=1}^N, \sigma^2\right) \propto \prod_{n=1}^N p\left(\mathbf{M} \bar{\times}_n \mathbf{U}_n^T | \bar{\mathbf{t}}_n, \sigma_n^2\right) \tag{21}$$

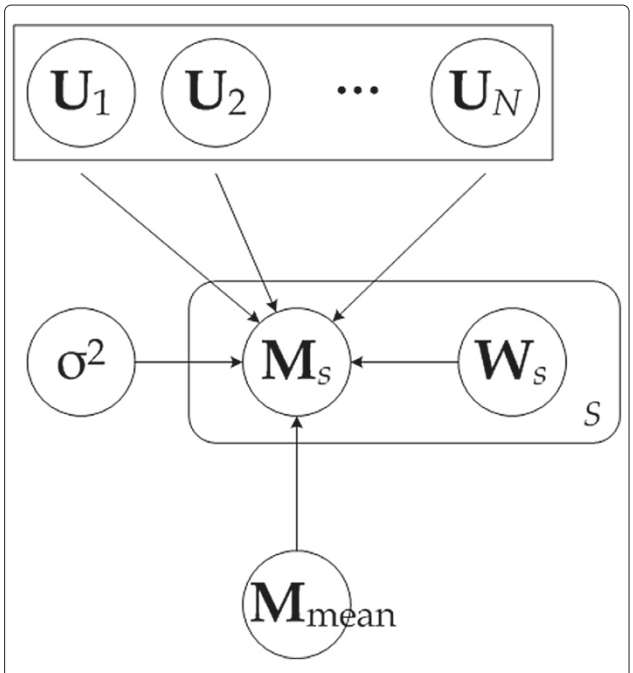

Figure 4 Graphical model representation of the probabilistic 2-mode model.

where $\bar{\mathbf{t}}_n \in \mathbb{R}^{I_n \times 1}$ and σ_n^2 denote the mean vector and noise variance, respectively, for mode n; $\mathbf{M}\bar{\times}_n\mathbf{U}_n^T \equiv \mathbf{M} \times_1 \mathbf{U}_1^T \ldots \times_{n-1} \mathbf{U}_{n-1}^T \times_{n+1} \mathbf{U}_{n+1}^T \ldots \times_N \mathbf{U}_N^T$, i.e., the product of $\mathbf{M}$ with all the mode matrices except mode n, which is called the contracted n-mode product [14]. That is, the nth probabilistic function is defined as the projected $\mathbf{M}$ by all $\mathbf{U}_j$'s expect $\mathbf{U}_n$. Given observed data M, the decoupled posterior probabilistic function is defined as

$$p\big(\mathbf{M}_{\text{mean}}, \{\mathbf{U}_n\}_{n=1}^N, \sigma^2|M\big) \propto \prod_{n=1}^{N} p\big(\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2, M, \{\mathbf{U}_j\}_{j=1,j\neq n}^N\big). \tag{22}$$

By Bayes' theorem, the nth posterior distribution can be expressed in terms of the decoupled likelihood function and the decoupled prior distribution:

$$\begin{aligned} &p\big(\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2, M, \{\mathbf{U}_j\}_{j=1,j\neq n}^N\big) \propto \\ &\quad p\big(M, \{\mathbf{U}_j\}_{j=1,j\neq n}^N|\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2\big)\, p\big(\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2\big). \end{aligned} \tag{23}$$

Therefore, the decoupled predictive density is given by

$$\begin{aligned} p\,(\mathbf{M}|M) \propto& p\left(\mathbf{M}|\mathbf{M}_{\text{new}}, \{\mathbf{U}_n\}_{n=1}^N, \sigma^2\right) \\ &\times p\big(\mathbf{M}_{\text{mean}}, \{\mathbf{U}_n\}_{n=1}^N, \sigma^2|M\big) \\ =& \prod_{n=1}^{N} p\left(\mathbf{M}\bar{\times}_n\mathbf{U}_n^T|\bar{\mathbf{t}}_n, \sigma_n^2\right) \\ &\times p\left(M, \{\mathbf{U}_j\}_{j=1,j\neq n}^N|\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2\right). \end{aligned} \tag{24}$$

($p\big(\bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2\big)$ is dropped out for a fixed $\mathbf{U}_n$). This is the 2-mode case of the PTA in [8]. In our case, Equation (24) is given by

$$\begin{aligned} p\,(\mathbf{M}|M) \propto& p\left(\mathbf{U}_{\text{dim}}^T\mathbf{M}^T|\bar{\mathbf{t}}_{\text{mixture}}, \sigma_{\text{mixture}}^2\right) \\ &\times p\big(M, \mathbf{U}_{\text{dim}}|\bar{\mathbf{t}}_{\text{mixture}}, \mathbf{U}_{\text{mixture}}, \sigma_{\text{mixture}}^2\big) \\ &\times p\big(\mathbf{U}_{\text{mixture}}^T\mathbf{M}|\bar{\mathbf{t}}_{\text{dim}}, \sigma_{\text{dim}}^2\big) \\ &\times p\big(M, \mathbf{U}_{\text{mixture}}|\bar{\mathbf{t}}_{\text{dim}}, \mathbf{U}_{\text{dim}}, \sigma_{\text{dim}}^2\big). \end{aligned} \tag{25}$$

Now, $\mathbf{U}_n$'s are obtained by maximizing the following posterior distribution:

$$p\big(\{\mathbf{U}_n\}_{n=1}^N|M\big) \approx \prod_{n=1}^{N} p\big(\mathbf{U}_n|M\bar{\times}_n\mathbf{U}_n^T\big) \tag{26}$$

where $p\big(\mathbf{U}_n|M\bar{\times}_n\mathbf{U}_n^T\big) \equiv \prod_{s=1}^{S} \mathbf{M}_s\bar{\times}_n\mathbf{U}_n^T$. The expectation-maximization (EM) algorithm [15] is applied to compute $\mathbf{U}_n$'s. The application of the EM algorithm to construct probabilistic 2-mode model is explained in the next section.

2.4 Construction of probabilistic 2-mode model for speaker adaptation

In Equation (20), for the given training models, the maximum likelihood (ML) estimate of $\mathbf{M}_{\text{mean}}$ is given as $\overline{\mathbf{M}} = (1/S)\sum_s \mathbf{M}_s$ and $\{\mathbf{U}_n, \sigma_n^2\}$ can be estimated as follows. First, let us define the followings: Let $\mathbf{t}_{n;j} \in \mathbb{R}^{I_n \times 1}$ be the jth column vector of

$$\mathbf{T}_{(n)} = \text{mat}_n\big(\mathbf{M}\bar{\times}_n\mathbf{U}_n^T\big) \tag{27}$$

for $1 \le j \le \bar{I}_n S$ ($\bar{I}_n = \prod_{j=1,j\neq n}^{N} I_j$) and $\mathbf{x}_{n;j} \in \mathbb{R}^{K_n \times 1}$ be the jth column vector of

$$\mathbf{X}_{(n)} = \text{mat}_n\big(\mathbf{M}\prod_{n=1}^{N} \times_n\mathbf{U}_n^T\big). \tag{28}$$

Let us suppose $\mathbf{t}_n|\mathbf{x}_n \sim \mathcal{N}(\mathbf{U}_n\mathbf{x}_n + \bar{\mathbf{t}}_n, \sigma_n^2\mathbf{I}_{I_n})$ and $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}_{K_n\times 1}, \mathbf{I}_{K_n})$. Then, by integrating out $\mathbf{x}_n$, $\mathbf{t}_n \sim \mathcal{N}(\bar{\mathbf{t}}_n, \mathbf{G}_n)$ where $\bar{\mathbf{t}}_n = 1/(\bar{I}_n S)\sum_{j=1}^{\bar{I}_n S} \mathbf{t}_{n;j}$ and $\mathbf{G}_n = \mathbf{U}_n\mathbf{U}_n^T + \sigma_n^2\mathbf{I}_{I_n}$. Consequently,

$$\mathbf{x}_n|\mathbf{t}_n \sim \mathcal{N}\big(\mathbf{H}_n^{-1}\mathbf{U}_n^T(\mathbf{t}_n - \bar{\mathbf{t}}_n), \sigma_n^2\mathbf{H}_n^{-1}\big) \tag{29}$$

where $\mathbf{H}_n = \mathbf{U}_n^T\mathbf{U}_n + \sigma_n^2\mathbf{I}_{K_n}$. The right-hand side of Equation (26) becomes

$$\log p\big(\mathbf{U}_n|M\bar{\times}_n\mathbf{U}_n^T\big) \propto -\frac{\bar{I}_n S}{2}\Big(\log|\mathbf{G}_n| + \text{tr}\big[\mathbf{G}_n^{-1}\mathbf{S}_n\big]\Big) \tag{30}$$

where $\mathbf{S}_n = 1/(\bar{I}_n S - 1)\sum_{j=1}^{\bar{I}_n S}\big(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\big)\big(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\big)^T$ and $\text{tr}[\,\cdot\,]$ denotes the trace of a matrix. Summing up for all the modes, we obtain the following log-likelihood function of the posterior distribution:

$$\begin{aligned} L =& \sum_n \log p\big(\mathbf{U}_n|M\bar{\times}_n\mathbf{U}_n\big) \propto \\ &-\sum_n\left\{\frac{\bar{I}_n S}{2}\Big(\log|\mathbf{G}_n| + \text{tr}\big[\mathbf{G}_n^{-1}\mathbf{S}_n\big]\Big)\right\}. \end{aligned} \tag{31}$$

The graphical model representation of the decoupled probabilistic model is shown in Figure 5.

We seek to find $\mathbf{U}_n$'s that maximize the log-likelihood function alternatingly. Mode matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ are initialized with the results from the Tucker decomposition which minimizes the reconstruction error:

$$\text{Error} = \sum_s \left\|\mathbf{M}_s - \big(\mathbf{W}_s \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 + \overline{\mathbf{M}}\big)\right\|^2. \tag{32}$$

With the initial $\mathbf{U}_1$ and $\mathbf{U}_2$, the following procedure is performed for each mode ($n = 1, 2$).

Each training model is projected into mode matrices except mode n and expressed in a mode-n matrix:

$$\mathbf{T}_{s,(n)} = \text{mat}_n\big(\mathbf{M}_s\bar{\times}_n\mathbf{U}_n^T\big). \tag{33}$$

All the column vectors of $\big\{\mathbf{T}_{s,(n)}\big\}_{s=1}^S$ constitute the training data set:

$$\{\mathbf{t}_{n;j}\}, \quad 1 \le j \le \bar{I}_n S. \tag{34}$$

Then, with an initial estimate of σ_n^2 (e.g., 0.005 was used in the experiments), the EM algorithm is iterated as follows until $\mathbf{U}_n$ and σ_n^2 converge.

E-step: From Equation (31), the expectation of the log-likelihood function of complete data w.r.t. $p\big(\mathbf{x}_{n;j}|\mathbf{t}_{n;j}, \bar{\mathbf{t}}_n, \mathbf{U}_n, \sigma_n^2\big)$ is given as

$$\begin{aligned}\langle L_{\mathrm{c}}\rangle &= \sum_n \sum_s E\big[\log p(\mathbf{M}_s, \mathbf{W}_s|\{\mathbf{U}\}_{j=1,j\neq n}^N)\big] \\ &= \sum_n \sum_{j=1}^{I_n S} E\big[\log p(\mathbf{t}_{n;j}, \mathbf{x}_{n;j}|\{\mathbf{U}\}_{j=1,j\neq n}^N)\big]\end{aligned} \tag{35}$$

where

$$\log p(\mathbf{t}_{n;j}, \mathbf{x}_{n;j}|\{\mathbf{U}\}_{j=1,j\neq n}^N) = \log p(\mathbf{x}_{n;j}) + \log p(\mathbf{t}_{n;j}|\mathbf{x}_{n;j}) \tag{36}$$

$$\propto -\|\mathbf{x}_{n;j}\|^2 - \frac{I_n}{2}\log(\sigma_n^2) - \frac{1}{\sigma_n^2}\|\mathbf{t}_{n;j} - \mathbf{U}_n\mathbf{x}_{n;j} - \bar{\mathbf{t}}_n\|^2.$$

So,

$$\begin{aligned}\langle L_{\mathrm{c}}\rangle \propto -\sum_n \sum_{j=1}^{\bar{I}_n S} \Big\{ &\mathrm{tr}\big[\langle \mathbf{x}_{n;j}\mathbf{x}_{n;j}^T\rangle\big] \\ &+ \frac{1}{\sigma_n^2}(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n)^T(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n) \\ &+ \frac{I_n}{2}\log(\sigma_n^2) + \frac{1}{\sigma_n^2}\mathrm{tr}\big[\mathbf{U}_n^T\mathbf{U}_n\langle \mathbf{x}_{n;j}\mathbf{x}_{n;j}^T\rangle\big] \\ &- \frac{2}{\sigma_n^2}\langle \mathbf{x}_{n;j}\rangle^T\mathbf{U}_n^T(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n)\Big\}\end{aligned} \tag{37}$$

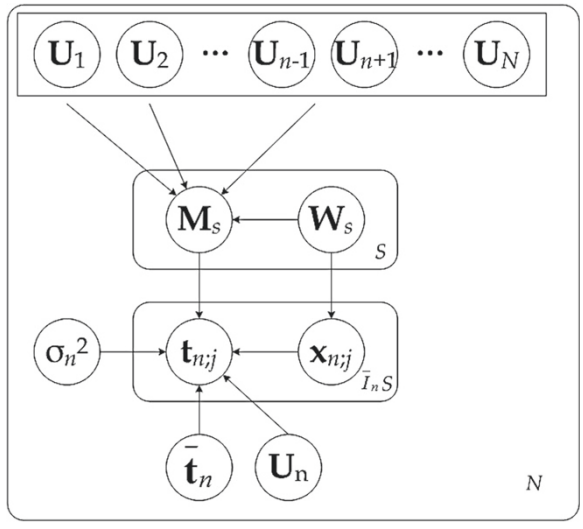

Figure 5 Graphical model representation of the decoupled probabilistic model.

with the sufficient statistics are given as follows from Equation (29):

$$\begin{aligned}\langle \mathbf{x}_{n;j}\rangle &= \mathbf{H}_n^{-1}\mathbf{U}_n^T\big(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\big) \\ \langle \mathbf{x}_{n;j}\mathbf{x}_{n;j}^T\rangle &= \sigma_n^2\,\mathbf{H}_n^{-1} + \langle \mathbf{x}_{n;j}\rangle\langle \mathbf{x}_{n;j}\rangle^T.\end{aligned} \tag{38}$$

M-step: Model parameters are updated by maximizing $\langle L_{\mathrm{c}}\rangle$ w.r.t. $\mathbf{U}_n$ and σ_n^2. Setting $\partial_{\mathbf{U}_n}\langle L_{\mathrm{c}}\rangle = 0$ produces

$$\mathbf{U}_n = \Big[\sum_{j=1}^{\bar{I}_n S}\big(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\big)\langle \mathbf{x}_{n;j}\rangle^T\Big]\Big[\sum_{j=1}^{\bar{I}_n S}\langle \mathbf{x}_{n;j}\mathbf{x}_{n;j}^T\rangle\Big]^{-1}. \tag{39}$$

Next, setting $\partial_{\sigma_n^2}\langle L_{\mathrm{c}}\rangle = 0$ produces

$$\sigma_n^2 = \frac{1}{I_n\bar{I}_n S}\sum_{j=1}^{\bar{I}_n S}\Big\{\|\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\|^2 - 2\langle \mathbf{x}_{n;j}\rangle^T\mathbf{U}_n^T\big(\mathbf{t}_{n;j} - \bar{\mathbf{t}}_n\big) + \mathrm{tr}\big[\langle \mathbf{x}_{n;j}\mathbf{x}_{n;j}^T\rangle\mathbf{U}_n^T\mathbf{U}_n\big]\Big\}. \tag{40}$$

Essentially, the procedure applies PPCA to the data set $\{\mathbf{t}_{n;j}\}$ for each mode.

2.5 Estimation of prior distribution

Given model parameters $\{\overline{\mathbf{M}}, \mathbf{U}_n, \sigma_n^2\}$, the weight matrix for the training speaker model $\mathbf{M}_s$ is obtained by

$$\begin{aligned}\mathbf{W}_s &= \big(\mathbf{M}_s - \overline{\mathbf{M}}\big)\prod_{n=1}^{2}\times_n\big(\mathbf{H}_n^{-1}\,\mathbf{U}_n^T\big) \\ &= \big(\mathbf{H}_1^{-1}\,\mathbf{U}_{\mathrm{mixture}}^T\big)\big(\mathbf{M}_s - \overline{\mathbf{M}}\big)\big(\mathbf{H}_2^{-1}\,\mathbf{U}_{\mathrm{dim}}^T\big)^T.\end{aligned} \tag{41}$$

From the set of weight matrices $\{\mathbf{W}_s\}_{s=1}^S$, the distribution of the weight is estimated. In deriving the adaptation equation in the MAP framework, the parameters for the prior distribution can be obtained in closed-form solutions if $p(\mathbf{W})$ follows a conjugate distribution. Hence, we assume the prior distribution of the weight to be a matrix-variate normal:

$$p(\mathbf{W}) \propto \frac{1}{|\Sigma|^{K_D/2}|\Psi|^{K_R/2}}\exp\Big\{-\frac{1}{2}\mathrm{tr}\big[(\mathbf{W}-\mathbf{W}_{\mathrm{mean}})^T\Sigma^{-1}(\mathbf{W}-\mathbf{W}_{\mathrm{mean}})\Psi^{-1}\big]\Big\}. \tag{42}$$

Furthermore, the hyperparameters of $p(\mathbf{W})$ can easily be estimated in an ML criterion if Ψ is known [16]. So,

$\boldsymbol{\Psi}$ is assumed to be the identity matrix [17], and the hyperparameters are estimated as:

$$\widehat{\mathbf{W}}_{\text{mean}} = \frac{1}{S}\sum_{s}\mathbf{W}_s = \mathbf{0}_{K_R\times K_D} \tag{43}$$
$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{S-1}\sum_{s}\mathbf{W}_s\mathbf{W}_s^T.$$

2.6 Speaker adaptation in the MAP framework

Based on Equation (20), we express the model of a new speaker as

$$\mathbf{M}_{\text{new}} = \mathbf{U}_{\text{mixture}}\mathbf{W}_{\text{new}}\mathbf{U}_{\text{dim}}^T + \overline{\mathbf{M}}. \tag{44}$$

For the given adaptation data $\mathbf{O} = \{\mathbf{o}_1,\ldots,\mathbf{o}_T\}$, we estimate the adaptation parameter in a MAP criterion:

$$\begin{aligned}\boldsymbol{\Lambda}_{\text{MAP}} &= \arg\max_{\boldsymbol{\Lambda}} p(\boldsymbol{\Lambda}|\mathbf{O}) \\ &\propto \arg\max_{\boldsymbol{\Lambda}} p(\mathbf{O}|\boldsymbol{\Lambda})\,p(\boldsymbol{\Lambda}) \\ &\propto \arg\max_{\boldsymbol{\Lambda}} \log p(\mathbf{O}|\boldsymbol{\Lambda}) + \log p(\boldsymbol{\Lambda})\end{aligned} \tag{45}$$

where $\boldsymbol{\Lambda} = \{\mathbf{W}_{\text{new}}\}$ denotes the model parameter.

Using the EM algorithm, we obtain the following auxiliary Q-function to be optimized (discarding the terms that are independent of the model parameter):

$$\begin{aligned}Q(\boldsymbol{\Lambda},\hat{\boldsymbol{\Lambda}}) = &-\frac{1}{2}\sum_t\sum_r \gamma_r(t)\,\text{tr}\big[(\mathbf{o}_t-\mathbf{m}_{\text{new};r}^T)^T\mathbf{C}_r^{-1}(\mathbf{o}_t-\mathbf{m}_{\text{new};r}^T)\big] \\ &-\frac{1}{2}\,\text{tr}\big[(\mathbf{W}_{\text{new}}\mathbf{U}_{\text{dim}})^T\widehat{\boldsymbol{\Sigma}}^{-1}(\mathbf{W}_{\text{new}}\mathbf{U}_{\text{dim}})\big]\end{aligned} \tag{46}$$

where $\boldsymbol{\Lambda}$ and $\hat{\boldsymbol{\Lambda}}$ denote the current and updated model parameters, respectively, and $\mathbf{m}_{\text{new};r} = \mathbf{u}_{\text{mixture};r}\mathbf{W}_{\text{new}}\mathbf{U}_{\text{dim}}^T + \overline{\mathbf{m}}_r$. In finding the speaker weight, we compute $\mathbf{W}_{\text{new,aug}} \equiv \mathbf{W}_{\text{new}}\mathbf{U}_{\text{dim}}$, from which $\mathbf{W}_{\text{new}}$ is obtained. Solving in this way, we can use the row-by-row technique in MLLR adaptation [6]. Setting $\partial_{\mathbf{W}_{\text{new}}}Q(\boldsymbol{\Lambda},\hat{\boldsymbol{\Lambda}}) = 0$ yields the following equation:

$$\begin{aligned}&\sum_t\sum_r \gamma_r(t)\mathbf{C}_r^{-1}\underbrace{\mathbf{U}_{\text{dim}}\mathbf{W}_{\text{new}}^T}_{\equiv\mathbf{W}_{\text{new, aug}}^T}\mathbf{u}_{\text{mixture};r}^T\mathbf{u}_{\text{mixture};r} + \underbrace{\mathbf{U}_{\text{dim}}\mathbf{W}_{\text{new}}^T}_{\equiv\mathbf{W}_{\text{new, aug}}^T}\widehat{\boldsymbol{\Sigma}}^{-1} \\ &= \sum_t\sum_r \gamma_r(t)\,\mathbf{C}_r^{-1}\big(\mathbf{o}_t-\overline{\mathbf{m}}_r^T\big)\mathbf{u}_{\text{mixture};r}.\end{aligned} \tag{47}$$

The above equation can be solved for $\mathbf{W}_{\text{new, aug}}$ in a similar way to MLLR adaptation in [6]: we define the followings:

$$\begin{aligned}\mathbf{V}_r &= \sum_t \gamma_r(t)\,\mathbf{C}_r^{-1} \\ \mathbf{D}_r &= \mathbf{u}_{\text{mixture};r}^T\mathbf{u}_{\text{mixture};r} \\ \mathbf{G}_{(i)} &= \sum_r \nu_r(i,i)\,\mathbf{D}_r \\ \mathbf{Z} &= \sum_t\sum_r \gamma_r(t)\,\mathbf{C}_r^{-1}\big(\mathbf{o}_t-\overline{\mathbf{m}}_r^T\big)\mathbf{u}_{\text{mixture};r} \\ \boldsymbol{\Sigma}_{(i)} &= \frac{1}{S-1}\sum_s \mathbf{w}_{s;i}\mathbf{w}_{s;i}^T\end{aligned} \tag{48}$$

where $\nu_r(i,i)$ denotes the (i,i) element of $\mathbf{V}_r$ and $\mathbf{w}_{s;i}$ the ith column vector of $\mathbf{W}_{s,\text{aug}} \equiv \mathbf{W}_s\mathbf{U}_{\text{dim}}$. Then, the speaker weight can be computed:

$$\mathbf{w}_{\text{new, aug},(i)}^T = \Big[\mathbf{G}_{(i)} + \boldsymbol{\Sigma}_{(i)}^{-1}\Big]^{-1}\mathbf{z}_{(i)}^T, \quad i = 1,\ldots,D \tag{49}$$

where $\mathbf{w}_{\text{new, aug},(i)}$ denotes the ith row of $\mathbf{W}_{\text{new,aug}}$ and $\mathbf{z}_{(i)}$ the ith row vector of $\mathbf{Z}$. The method becomes similar to MAPLR adaptation in [11]. Finally, the speaker weight is obtained as

$$\hat{\mathbf{W}}_{\text{new}} = \mathbf{W}_{\text{new, aug}}\mathbf{U}_{\text{dim}}^{+} \tag{50}$$

where $[\,\cdot\,]^+$ denotes the pseudoinverse of a matrix. The weight is plugged into Equation (44) to produce the model adapted to the new speaker.

2.7 Speaker adaptation techniques compared in the experiments

In this section, we briefly review the speaker adaptation techniques compared with the probabilistic 2-mode analysis based method: eigenvoice adaptation [3], MLLR adaptation [6], and MAPLR adaptation [11].

In eigenvoice adaptation, the collection of HMM mean vectors of speaker s is arranged in an $(RD)\times 1$ vector:

$$\boldsymbol{\mu}_s = \begin{bmatrix}\boldsymbol{\mu}_{s;1}\\ \boldsymbol{\mu}_{s;2}\\ \vdots\\ \boldsymbol{\mu}_{s;R}\end{bmatrix}. \tag{51}$$

Then, the set of S supervectors, $\{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_S\}$, is decomposed by PCA to produce the adaptation model

$$\boldsymbol{\mu}_{\text{new}} = \boldsymbol{\Phi}\,\mathbf{w}_{\text{new}} + \bar{\boldsymbol{\mu}} \tag{52}$$

where $\boldsymbol{\Phi} = \big[\boldsymbol{\phi}_1\ldots\boldsymbol{\phi}_K\big]$, the basis matrix consisting of the K dominant eigenvectors from PCA, and $\bar{\boldsymbol{\mu}} = 1/S\sum_s\boldsymbol{\mu}_s$.

Table 2 Word recognition accuracy (%) of the Tucker 3-mode and probabilistic 2-mode based methods

Method	(K_R, K_D)	Number of free parameters	Number of adaptation sentences				
			1	2	3	4	5
Tucker 3-mode	(20, 35)	700	91.84	92.98	93.07	92.99	93.11
	(20, 38)	760	91.82	92.83	93.11	93.01	93.01
	(30, 35)	1050	90.77	92.99	93.18	93.09	92.94
	(30, 38)	1140	90.77	92.86	93.18	93.01	92.86
	(40, 35)	1400	89.39	92.85	93.11	93.24	93.03
	(40, 38)	1520	89.16	92.77	93.20	93.14	92.98
	(50, 35)	1750	87.95	92.34	93.24	93.26	93.13
	(50, 38)	1900	87.75	92.47	93.27	93.31	93.16
Probabilistic 2-mode	(20, 35)	700	93.07	93.18	93.26	93.27	93.16
	(20, 38)	760	92.96	93.07	93.03	93.24	93.13
	(30, 35)	1050	92.98	93.20	93.24	93.24	93.31
	(30, 38)	1140	92.94	93.33	93.33	93.27	93.31
	(40, 35)	1400	93.13	93.20	93.39	93.24	93.01
	(40, 38)	1520	93.14	93.22	93.33	93.20	93.24
	(50, 35)	1750	93.26	93.35	93.37	93.29	93.29
	(50, 38)	1900	93.37	93.44	93.42	93.31	93.39

The number of mixture components $R = 3472 \cdot 8$ and the dimension of acoustic feature vector $D = 39$. The number of free parameters is $K_R \times K_D$.

The $K \times 1$ weight vector can be obtained by maximizing the likelihood of the adaptation data, which is given by

$$\hat{\mathbf{w}}_{\text{new}} = \left[\sum_t \sum_r \gamma_r(t)\, \boldsymbol{\Phi}_r^T \mathbf{C}_r^{-1} \boldsymbol{\Phi}_r \right]^{-1} \times \left[\sum_t \sum_r \gamma_r(t)\, \boldsymbol{\Phi}_r^T \mathbf{C}_r^{-1} \left(\mathbf{o}_t - \bar{\boldsymbol{\mu}}_r \right) \right] \tag{53}$$

where $\boldsymbol{\Phi}_r$ and $\bar{\boldsymbol{\mu}}_r$ denote the $D \times K$ submatrix and $D \times 1$ subvector corresponding to the rth mixture of $\boldsymbol{\Phi}$ and $\bar{\boldsymbol{\mu}}$, respectively.

In MLLR adaptation, the updated model for a new speaker is obtained by linearly transforming the SI model (assuming a global regression matrix):

$$\boldsymbol{\mu}_{\text{new},\, r} = \mathbf{W}_{\text{new}}\, \boldsymbol{\xi}_r, \quad \boldsymbol{\xi}_r = \begin{bmatrix} \omega \\ \boldsymbol{\mu}_{\text{SI},\, r} \end{bmatrix} \tag{54}$$

where $\boldsymbol{\mu}_{\text{SI},\, r}$ denotes the mean vector of the SI HMM corresponding to mixture r and ω is the bias offset term: $\omega = 1$ to include the term and $\omega = 0$ otherwise ($\omega = 1$ in our experiments). The $D \times (D+1)$ transformation matrix can be obtained in an ML criterion, which yields the following equation:

$$\sum_t \sum_r \gamma_r(t)\, \mathbf{C}_r^{-1} \mathbf{o}_t\, \boldsymbol{\xi}_r^T = \sum_t \sum_r \gamma_r(t)\, \mathbf{C}_r^{-1} \mathbf{W}_{\text{new}}\, \boldsymbol{\xi}_r\, \boldsymbol{\xi}_r^T. \tag{55}$$

The above equation can be solved for $\mathbf{W}_{\text{new}}$:

$$\hat{\mathbf{w}}_{\text{new},\,(i)}^T = \mathbf{G}_{(i)}^{-1}\, \mathbf{z}_{(i)}^T, \quad i = 1, \ldots, D \tag{56}$$

where $\hat{\mathbf{w}}_{\text{new},\,(i)}$ and $\mathbf{z}_{(i)}$ denote the ith row vectors of $\hat{\mathbf{W}}_{\text{new}}$ and $\mathbf{Z}$, respectively; $\mathbf{G}_{(i)}$ and $\mathbf{Z}$ are defined as:

$$\begin{aligned} \mathbf{V}_r &= \sum_t \gamma_r(t)\, \mathbf{C}_r^{-1} \\ \mathbf{D}_r &= \boldsymbol{\xi}_r\, \boldsymbol{\xi}_r^T \\ \mathbf{G}_{(i)} &= \sum_r v_r(i,i)\, \mathbf{D}_r \\ \mathbf{Z} &= \sum_t \sum_r \gamma_r(t)\, \mathbf{C}_r^{-1} \mathbf{o}_t\, \boldsymbol{\xi}_r^T \end{aligned} \tag{57}$$

where $v_r(i,i)$ denotes the (i,i) element of $\mathbf{V}_r$.

In MAPLR adaptation, the prior for the transformation matrix is used in the MLLR framework. The parameters for the prior are obtained from the MLLR transformation matrices of training speakers $\{\mathbf{W}_1, \ldots, \mathbf{W}_S\}$:

$$\begin{aligned} \bar{\mathbf{w}}_{(i)} &= \frac{1}{S} \sum_s \mathbf{w}_{s,\,(i)} \\ \boldsymbol{\Sigma}_{(i)} &= \frac{1}{S-1} \sum_s \left(\mathbf{w}_{s,\,(i)} - \bar{\mathbf{w}}_{(i)} \right)^T \left(\mathbf{w}_{s,\,(i)} - \bar{\mathbf{w}}_{(i)} \right) \end{aligned} \tag{58}$$

where $\mathbf{w}_{s,\,(i)}$ denotes the ith row vector of $\mathbf{W}_s$. Then, the transformation matrix for a new speaker is obtained in a

Table 3 p-values from the matched-pair t-test

Methods	Number of adaptation sentences				
	1	2	3	4	5
Prob. 2-mode and Tucker 3-mode	< 0.01	0.22	0.08	0.03	0.34
Prob. 2-mode and MAPLR	0.10	< 0.01	< 0.01	0.01	0.02
Prob. 2-mode and MLLR, block-diagonal	< 0.01	0.01	0.04	< 0.01	0.05
Prob. 2-mode and EV	< 0.01				
Tucker 3-mode and MLLR, block-diagonal	0.43	0.18	0.63	0.17	0.22
Tucker 3-mode and EV	0.94	< 0.01	< 0.01	< 0.01	< 0.01

For the probabilistic 2-mode and Tucker 3-mode based models, $K_R = 20$ and $K_D = 35$.

MAP criterion. Deriving the equation in the same way as above, we can obtain the following:

$$\hat{\mathbf{w}}_{\text{new},(i)}^{T} = \left[\mathbf{G}_{(i)} + \boldsymbol{\Sigma}_{(i)}^{-1}\right]^{-1}\left[\mathbf{z}_{(i)} + \bar{\mathbf{w}}_{(i)}\boldsymbol{\Sigma}_{(i)}^{-1}\right]^{T}. \quad (59)$$

3 Experiments

We carried out the large-vocabulary continuous-speech recognition (LVCSR) experiments using the Wall Street Journal corpus WSJ0 [18]. In building the SI model, we used 12754 utterances of 101 speakers from the corpus. As the acoustic feature vector, we used the 39-dimensional vector consisting of 13-dimensional mel-frequency cepstral coefficients (MFCCs) including the 0th cepstral coefficient, their derivative coefficients, and their acceleration coefficients. The feature vector was extracted with the 20-ms Hamming window with the frame sliding of 10 ms. Using the HMM toolkit (HTK) [19], we built a tied-state triphone model (word-internal triphones) with 3472 tied states and 8-mixture Gaussian.

To build training models for constructing bases, we transformed the SI model by MLLR adaptation [6] using 32 regression classes followed by maximum *a posteriori* (MAP) adaptation [10]. We used the 101 adapted models to build the Tucker decomposition and probabilistic tensor based models as well as eigenvoice.

For adaptation and recognition test, we used Nov'92 5K non-verbalized adaptation and test sets. The number of testing speakers was 8; adaptation set was used for adaptation and testing set of 330 sentences was used for recognition test (the number of testing utterances per speaker was about 40). The length of an adaptation sentence was about 6 s and the adaptation was performed in supervised mode. In recognition test, we used WSJ 5K non-verbalized 5k closed-vocabulary set and WSJ standard 5K non-verbalized closed bigram.

The word recognition accuracy of the SI model is 91.54%. Table 2 shows the results of the Tucker decomposition and probabilistic 2-mode based methods ($K_S = 100$ in the Tucker decomposition based model). In the table, the probabilistic 2-mode based method shows improved performance over the Tucker decomposition based method for small amounts of adaptation data, which can be evidently seen in Figure 6 for the Tucker decomposition and probabilistic 2-mode based models with ($K_R = 20, K_D = 35$). The results of MAPLR [11] are also shown in the figure. The use of MAP framework contributes to improved performance for small amounts of adaptation data. The number of free parameters of each method is given as follows: $20 \cdot 35$ for the Tucker 3-mode and probabilistic 2-mode based models, and $39 \cdot 40$ for MAPLR adaptation. In Figure 7, the Tucker decomposition based method is compared with MLLR and eigenvoice adaptation techniques. The figure shows that the Tucker decomposition based method outperforms MLLR and eigenvoice adaptation techniques for adaptation sentences > 1. It can be inferred from the figure that eigenvoice adaptation will outperform the Tucker decomposition based method or MLLR for sparse adaptation data. The p-values from the matched-pair t-test are shown in Table 3; although the values are not always small, the performance improvement of the probabilistic 2-mode based method seems meaningful. Additionally, Figure 8

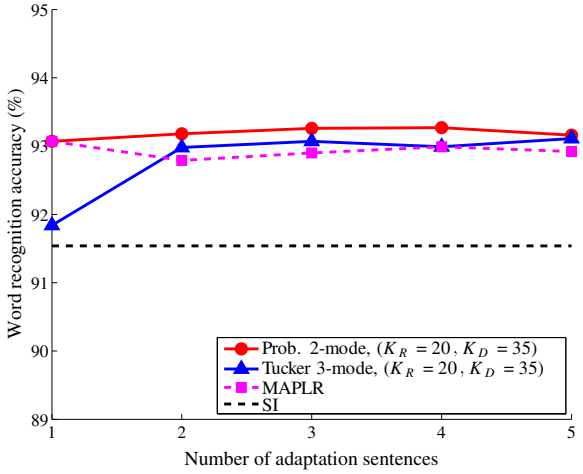

Figure 6 Word recognition accuracy of the probabilistic 2-mode based model, Tucker 3-mode based model, and MAPLR adaptation.

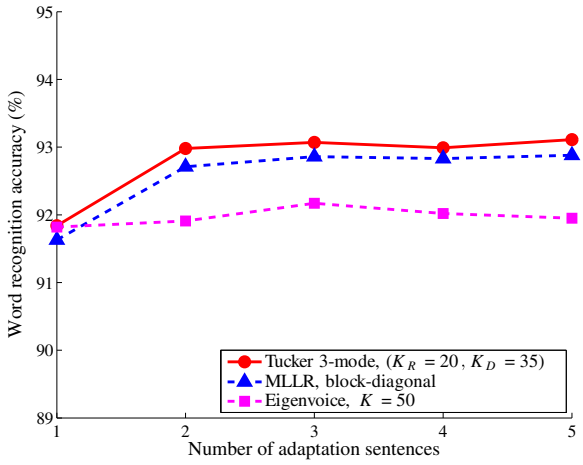

Figure 7 Word recognition accuracy of the Tucker 3-mode based model, MLLR, and eigenvoice adaptation.

shows the performance of the probabilistic 2-mode based model with ($K_R = 20, K_D = 35$), MLLR adaptation with a full regression matrix, and MAPLR adaptation for adaptation data of about 6–240 s; for adaptation sentences ≥ 10 (about 60 s), the probabilistic 2-mode based model shows the comparable performance with MLLR adaptation and MAPLR adaptation. In Figure 8, the p-values are given as: $p < 0.01$ for 1–5 adaptation sentences between the probabilistic 2-mode based model and MLLR adaptation, $p < 0.05$ for 2–5 adaptation sentences between the probabilistic 2-mode based model and MAPLR adaptation. The number of free parameters of each method is summarized in Table 4.

We think that the performance improvement of the proposed method over MLLR or MAPLR adaptation comes

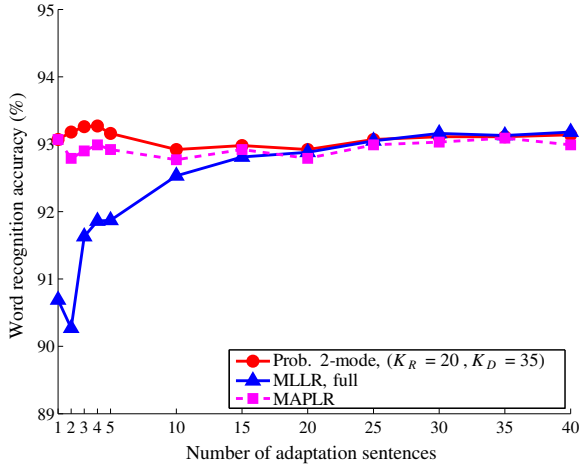

Figure 8 Word recognition accuracy of the probabilistic 2-mode based model, MLLR, and MAPLR adaptation.

Table 4 Number of free parameters of adaptation techniques

Method	Number of free parameters
Probabilistic 2-mode based model	20 · 35 ($K_R \cdot K_D$)
Tucker 3-mode based model	20 · 35 ($K_R \cdot K_D$)
MLLR, 3-block-diagonal regression matrix	13 · 40
MLLR, full regression matrix	39 · 40
MAPLR adaptation	39 · 40
Eigenvoice	50

from the use of basis vectors and speaker weight of large dimension. Additionally, we think that the performance improvement of the probabilistic 2-mode based method in the MAP framework over the Tucker decomposition based method in the ML framework for small amounts of adaptation data (e.g., 1 adaptation sentence) is due to its constraint on the weight. If the amount of adaptation data is small (e.g., 1 adaptation sentence), the weight cannot be reliably estimated in the ML framework where the weight is estimated using only adaptation data without constraint, as done in the Tucker decomposition based method. The results confirm that constraint on the weight in the MAP framework can produce better model when the amount of adaptation data is small.

The selection of appropriate dimensions of model parameters (e.g., K_R and K_D) in the probabilistic 2-mode analysis depends on the training models and also available adaptation data. The selection of model parameters affects the performance of the system, but how to choose the optimum model parameters is not obvious, which needs a further study.

4 Conclusions

In this article, we applied probabilistic tensor analysis to the adaptation of HMM mean vectors to a new speaker. The training models consisted of the mean vectors of HMMs expressed in matrix form and the training set was decomposed by probabilistic 2-mode analysis. The prior distribution of the adaptation parameter was estimated from the training models. Then, the speaker adaptation equation was derived in the MAP framework. Compared with the speaker adaptation method based on Tucker 3-mode decomposition in the ML framework, the proposed method further improved the performance for small amounts of adaptation data.

Abbreviations

ALS: Alternating Least Squares; ASR: Automatic Speech Recognition; EM: Expectation-Maximization; HMM: Hidden Markov Model; HTK: HMM Toolkit; LVCSR: Large-Vocabulary Continuous-Speech Recognition; MAP: Maximum *A Posteriori*; MAPLR: Maximum *A Posteriori* Linear Regression; MFCC: Mel-Frequency Cepstral Coefficient; ML: Maximum Likelihood; MLLR: Maximum Likelihood Linear Regression; PCA: Principal Component Analysis; PPCA: Probabilistic Principal Component Analysis; PTA: Probabilistic Tensor

Analysis; SD: Speaker-Dependent; SI: Speaker-Independent; SVD: Singular Value Decomposition; WSJ: Wall Street Journal.

Competing interests
The author declares that they have no competing interests.

Received: 29 May 2012 Accepted: 13 March 2013
Published: 11 April 2013

References

1. LR Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE. **77**(2), 257–286 (1989)
2. M Gales, S Young, The application of hidden Markov models in speech recognition. Found. Trends Signal, Process. **1**(3), 195–304 (2008)
3. R Kuhn, J-C Junqua, P Nguyen, N Niedzielski, Rapid speaker adaptation in eigenvoice space. IEEE Trans. Speech Audio Process. **8**(6), 695–707 (2000)
4. TG Kolda, BW Bader, Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
5. Y Jeong, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. Speaker adaptation based on the multilinear decomposition of training speaker models (Dallas, TX, 2010), pp. 4870–4873
6. CJ Leggetter, PC Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Comput. Speech Lang. **9**(2), 171–185 (1995)
7. Y Jeong, Acoustic model adaptation based on tensor analysis of training models. IEEE Signal Process. Lett. **18**(6), 347–350 (2011)
8. D Tao, M Song, X Li, J Shen, J Sun, X Wu, C Faloutsos, SJ Maybank, Bayesian tensor approach for 3-D face modeling. IEEE Trans. Circ. Syst. Video Technol. **18**(10), 1397–1410 (2008)
9. ME Tipping, CM Bishop, Probabilistic principal component analysis. J. R. Stat. Soc. Ser. B-Stat. Methodol. **61**(3), 611–622 (1999)
10. J-L Gauvain, C-H Lee, Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. Speech Audio Process. **2**(2), 291–298 (1994)
11. C Chesta, O Siohan, C-H Lee, in *Proceedings of EUROSPEECH*, vol. 1. Maximum *a posteriori* linear regression for hidden Markov model adaptation (Budapest, Hungary, 1999), pp. 211–214
12. JD Carroll, JJ Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika. **35**(3), 283–319 (1970)
13. AK Gupta, DK Nagar, *Matrix Variate Distributions*. (Chapman and Hall/CRC, Boca Raton, FL, 1999)
14. BW Bader, TG Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Trans. Math. Softw. **32**(4), 635–653 (2006)
15. AP Dempster, NM Laird, DB Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B-Stat. Methodol. **39**(1), 1–38 (1977)
16. AK Gupta, T Varga, *Elliptically Contoured Models in Statistics*. (Kluwer, Norwell, MA, 1993)
17. O Siohan, C Chesta, C-H Lee, Joint maximum a posteriori adaptation of transformation and HMM parameters. IEEE Trans. Speech Audio Process. **9**(14), 417–428 (2001)
18. DB Paul, JM Baker, in *Proceedings of DARPA Speech and Natural Language Workshop*. The design for the Wall Street Journal-based CSR corpus (Austin, TX, 1992), pp. 357–362
19. S Young, G Evermann, D Kershaw, G Moore, J Odell, D Ollason, D Povey, V Valtchev, P Woodland, *The HTK Book, Version 3.2*. (Cambridge University Engineering Department, England, 2002)

doi:10.1186/1687-4722-2013-7
Cite this article as: Jeong: **Speaker adaptation in the maximum *a posteriori* framework based on the probabilistic 2-mode analysis of training models.** *EURASIP Journal on Audio, Speech, and Music Processing* 2013 **2013**:7.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Immediate publication on acceptance
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com