Speech representation based on tensor factor analysis and its application to speaker recognition and language identification

Daisuke Saito, So Suzuki, and Nobuaki Minematsu
Graduate School of Engineering, The University of Tokyo, Tokyo, Japan
E-mail: {dsk_saito, ssuzuki, mine}@gavo.t.u-tokyo.ac.jp

***Abstract*—This paper proposes a novel approach to speech representation for both speaker recognition and language identification by characterizing the entire feature space by a tensor. In conventional studies of both tasks, i-vector is commonly used as the state-of-the-art representation. Here, i-vector extraction can be regarded as projection of utterance-based GMM supervector onto a low-dimensional space. In this paper, for the aim of explicit modeling of the correlation among mean vectors of a GMM, an utterance is not modeled as its GMM-based supervector but as its matrix and the entire set of utterances is modeled as its tensor. By applying tensor factor analysis, we obtain a new representation for an input utterance. Experimental evaluations for speaker recognition and language identification show that our proposed approach has effectiveness especially for the speaker recognition task.**

I. Introduction

Language identification (LID) and speaker recognition (SR) is a technique to identify language and speaker information from an input utterance, respectively. Speech varies due to conditions such as the speaker and microphone, so the variation of these irrelevant factors should be dealt with properly to diminish degradation of the performance of both tasks. In conventional studies of both tasks, i-vector is commonly used as the state-of-the-art representation [1], [2]. I-vector is derived by factor analysis of Gaussian Mixture Model Supervector (GMM-SV) which is a stacked vector of all the means in the GMM [3]. I-vector extraction can be regarded as Principal Component Analysis (PCA) to GMM-SV. At this point of view, representation of speaker characteristics based on Eigenvoice Conversion (EVC), which was proposed in the field of voice conversion, is almost the same as i-vector [4]. In these two methods, the means of utterance-based GMM are represented by a high-dimensional vector (GMM-SV), so the correlation among acoustic factors that GMM captures is difficult to be treated. To solve this problem, speaker representation based on tensor factor analysis was proposed [5], [6]. In this representation, for the aim of explicit modeling of the correlation among mean vectors of a GMM, an utterance is not modeled as its GMM-SV but as its matrix and the entire set of utterances of various speakers is modeled as its tensor. By introducing tensor factor analysis to the tensor of the training data set, speaker characteristics of an input utterance are represented. In this paper, we propose a new method of speech representation that has connection to [5], and also apply the both methods to the language identification task. As only preliminary experimental evaluation was done in [5], we evaluate the effectiveness of both methods in both tasks respectively.

II. Conventional language/speaker representation

A. GMM-SV

GMM-SV is a high-dimensional stacked vector of all the means in the utterance-based GMM [3]. Since each GMM component is expected to capture an acoustic factor such as a phoneme and phone, GMM-SV can be regarded as a feature which represents each acoustic factor in an input utterance.

Before extracting GMM-SV, utterance-based GMM needs to be estimated. It is estimated by doing Maximum a posteriori (MAP) adaptation of an input utterance from Universal Background Model (UBM). UBM is a language/speaker-independent GMM that is constructed from utterances of various languages and speakers.

B. i-vector

Let $\boldsymbol{M}$ be a GMM-SV. $\boldsymbol{M}$ extracted from an input utterance is decomposed based on factor analysis by using the total variability matrix $\boldsymbol{T}$, which captures the variation of text, languages, speakers or microphones as follows:

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w}. \tag{1}$$

In Equation 1, $\boldsymbol{m}$ denotes the supervector of UBM and $\boldsymbol{w}$ is called i-vector [2]. GMM-SV includes irrelevant factors (such as text, languages, or microphones in the case of speaker recognition). In the case of i-vector, on the other hand, projection of GMM-SV by the total variability matrix can help removing these factors. This projection can be regarded as PCA of GMM-SV, so i-vector is almost the same as speech representation based on EVC.

C. Speaker representation based on EVC

EVC is one of the GMM-based voice conversion methods [4]. In EVC, Eigenvoice, which is a method of adaptation of acoustic models [7], is introduced to speaker adaptation for voice conversion. In GMM-based voice conversion, a conversion model is constructed based on a joint GMM of the feature vector of input speaker $\boldsymbol{X}_t$ and that of output

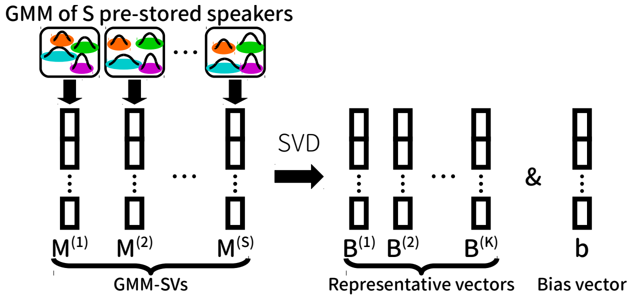

Fig. 1. Construction of speaker space based on Eigenvoice.

speaker $\boldsymbol{Y}_t$. In the case of EVC, the feature vector of output speaker $\boldsymbol{Y}_t$ is represented by using the features of S pre-stored speakers.

First, S GMM-SVs of pre-stored speakers are extracted. The number of dimensions of GMM-SV is DM (D and M denote the number of dimensions of the mean vector and the number of components respectively). Second, a speaker space is constructed by Singular Value Decomposition (SVD) of S GMM-SVs. The speaker space is represented by a bias vector and $K(\leq S)$ basis vectors (See Figure 1). Therefore, GMM-SV of the output speaker $\boldsymbol{M}^{(tar)}$ is represented by linear combination of basis vectors $\boldsymbol{B} = [\boldsymbol{B}_1^\top, \boldsymbol{B}_2^\top, \cdots, \boldsymbol{B}_M^\top]^\top \in \mathcal{R}^{DM \times K}$ and a bias vector $\boldsymbol{b} = [\boldsymbol{b}_1^{(0)^\top}, \cdots, \boldsymbol{b}_M^{(0)^\top}]^\top \in \mathcal{R}^{DM\times 1}$ as follows:

$$\boldsymbol{M}^{(tar)} = \boldsymbol{B}\boldsymbol{w} + \boldsymbol{b} \tag{2}$$

As GMM-SV of the output speaker is manipulated by the K-dimensional weight vector $\boldsymbol{w}$, $\boldsymbol{w}$ can be regarded as the representation of the output speaker. By using the features of S pre-stored speakers, the output speaker is effectively represented even in the case of a small amount of training utterances.

The weight vector can be estimated based on a maximum likelihood (ML) or MAP criterion using utterances of the output speaker. Let $\boldsymbol{Y}_t^{(tar)}$ be the feature vector of the output speaker at t-th frame, and $\boldsymbol{Y}^{(tar)}$ be the sequence of $\boldsymbol{Y}_t^{(tar)}$. $\boldsymbol{w}$ is estimated based on a ML criterion as follows:

$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \tag{3}$$

The following updating equation is derived by introducing the auxiliary function:

$$\hat{\boldsymbol{w}} = \left\{ \sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \boldsymbol{B}_m \right\}^{-1} \sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \overline{\boldsymbol{Y}}_m^{(tar)} \tag{4}$$

$$\overline{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t}, \ \overline{\boldsymbol{Y}}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} (\boldsymbol{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \tag{5}$$

$$\gamma_{m,t} = p(m | \boldsymbol{Y}_t^{(tar)}, \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) \tag{6}$$

On the other hand, $\boldsymbol{w}$ is estimated based on a MAP criterion as follows:

$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{w}) p(\boldsymbol{w}) \tag{7}$$

If $p(\boldsymbol{w})$ is assumed $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ as in the case of i-vector, the following updating equation is derived:

$$\hat{\boldsymbol{w}} = \left\{ \boldsymbol{I} + \sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \boldsymbol{B}_m \right\}^{-1} \sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{(YY)^{-1}} \overline{\boldsymbol{Y}}_m^{(tar)} \tag{8}$$

D. Speaker representation based on SAT for EVC

Speaker Adaptive Training (SAT) was introduced to EVC [8]. SAT was proposed as a method of construction of initial model for speaker adaptation, and better conversion performance was achieved by introducing SAT to EVC. SAT is done for each speaker in [8], but can also be done for each utterance. In this section, utterance-level SAT is explained. In the case of utterance-level SAT, basis vectors, a bias vector, weight vectors of the training utterances are estimated to maximize the likelihood of all the training utterances as follows:

$$\hat{\boldsymbol{\Lambda}}(\hat{\boldsymbol{w}}_1^N) = \operatorname*{argmax}_{\boldsymbol{\lambda}, \boldsymbol{w}_1^N} \prod_{n=1}^{N} \prod_{t_n=1}^{Tn} p(\boldsymbol{Y}_{t_n}^{(n)} | \boldsymbol{\lambda}(\boldsymbol{w}_n)) \tag{9}$$

In Equation 9, $\boldsymbol{Y}_{t_n}^{(n)}$ denotes the feature vector of n-th training utterance at t_n-th frame. $\boldsymbol{\lambda}(\boldsymbol{w}_n)$ denotes parameters of the GMM adapted to n-th training utterance represented by the weight vector $\boldsymbol{w}_n$ and $\boldsymbol{\Lambda}(\boldsymbol{w}_1^N)$ denotes the set of $\boldsymbol{\lambda}(\boldsymbol{w}_n)$. $\boldsymbol{w}_1^N$ denotes the set of the weight vectors of all the training utterances $(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_N)$.

The parameters are updated according to the following procedure:

1) Update the weight vectors $\hat{\boldsymbol{w}}_n$ of all the training utterances using the current basis vectors and bias vector.
2) Update the basis vectors $\hat{\boldsymbol{B}}_1, \ldots, \hat{\boldsymbol{B}}_K$ and the bias vector $\hat{\boldsymbol{b}}$ using $\hat{\boldsymbol{w}}_n$.
3) Iterate updating these parameters several times.

The updating equation of the weight vector $\hat{\boldsymbol{w}}_n$ is as follows:

$$\hat{\boldsymbol{w}}_n = \left\{ \sum_{m=1}^{M} \overline{\gamma}_m^{(n)} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{-1} \boldsymbol{B}_m \right\}^{-1} \sum_{m=1}^{M} \boldsymbol{B}_m^\top \boldsymbol{\Sigma}_m^{-1} (\overline{\boldsymbol{Y}}_m^{(n)} - \overline{\gamma}_m^{(n)} \boldsymbol{b}_m) \tag{10}$$

$$\overline{\boldsymbol{Y}}_m^{(n)} = \sum_{t_n=1}^{T_n} \gamma_{m,t_n}^{(n)} \boldsymbol{Y}_{t_n}^{(n)} \tag{11}$$

$$\gamma_{m,t_n}^{(n)} = p\Big(m | \boldsymbol{Y}_{t_n}^{(n)}, \boldsymbol{\lambda}(\boldsymbol{w}_n)\Big), \ \overline{\gamma}_m^{(n)} = \sum_{t_n=1}^{T_n} \gamma_{m,t_n}^{(n)} \tag{12}$$

The updating equations of the basis vectors $\hat{\boldsymbol{B}}_1, \cdots, \hat{\boldsymbol{B}}_K$ and the bias vector $\hat{\boldsymbol{b}}$ are as follows:

$$\hat{\boldsymbol{v}}_m = \left\{ \sum_{n=1}^{N} \overline{\gamma}_m^{(n)} \boldsymbol{W}_n^\top \boldsymbol{\Sigma}_m^{-1} \boldsymbol{W}_n \right\}^{-1} \sum_{n=1}^{N} \boldsymbol{W}_n^\top \boldsymbol{\Sigma}_m^{-1} \overline{\boldsymbol{Y}}_m^{(n)} \tag{13}$$

$$\hat{\boldsymbol{v}}_m = \begin{bmatrix} \hat{\boldsymbol{b}}_m^\top & \hat{\boldsymbol{B}}_m^{(1)} & \cdots & \hat{\boldsymbol{B}}_m^{(K)} \end{bmatrix}^\top \in \mathcal{R}^{(K+1)D \times 1} \tag{14}$$

$$\hat{\boldsymbol{W}}_n = \begin{bmatrix} 1 & \hat{\boldsymbol{w}}_n^\top \end{bmatrix} \otimes \boldsymbol{I} \ \in \mathcal{R}^{D \times (K+1)D} \tag{15}$$

E. Relations of speaker representation based on EVC, SAT for EVC, and i-vector

In II-B, II-C, we explained that speaker representation based on EVC and i-vector can be derived from PCA of GMM-SV. However, both methods are different in terms of implementation of PCA. Speaker representation based on EVC is derived from deterministic PCA (DPCA), which is done to maximize variance of training data. On the other hand, i-vector extraction can be regarded as Probabilistic PCA (PPCA), which is done to maximize likeihood of training data. In the case of DPCA, orthogonal basis vectors are calculated and the number of dimensions can be reduced by truncating basis vectors. In the case of PPCA, basis vectors are not generally orthogonal and the number of dimensions is set beforehand. Utterance-level SAT for EVC can be regarded as PPCA as in the case of i-vector.

III. Language/speaker representation based on tensor factor analysis

A. Overviews

In the case of EVC and i-vector, a feature space is constructed based on GMM-SV. However, since GMM-SV is simply a stacked vector of the means of GMM, it does not have the axis of the GMM components explicitly. In [5], [6], the mean vectors of GMM are represented as a matrix, whose row and column respectively correspond to the GMM component and the dimension of the mean vector, and tensor factor analysis are applied to the tensor of the entire set of training utterances to deal with correlation among the GMM components explicitly. In this paper, we apply this method to language/speaker representation.

B. Language/speaker representation based on Tucker decomposition

SVD can be rewritten as decomposition of a second-order tensor as follows:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \boldsymbol{S} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \tag{16}$$

Tucker decomposition is an extension of SVD to a third-order tensor as follows [9]:

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \tag{17}$$

The mean vectors of utterance-based GMM are expressed by a $M \times D$ matrix for each training utterance. M and D respectively denote the number of GMM components and that of the mean vector. The bias matrix $\boldsymbol{b}$ is calculated as the mean of these $M \times D$ matrices, and then it is subtracted from each matrix. Let N be the number of training utterances. These $M \times D$ matrices is expressed by a third-order tensor $\mathcal{M} \in \mathcal{R}^{M\times D\times N}$. $\mathcal{M}$ is decomposed by appling Tucker decomposition as follows:

$$\mathcal{M} = \mathcal{G}^{M\times D\times N} \times_1 \boldsymbol{U}^{(M)} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)} \tag{18}$$

In Equation 18, $\boldsymbol{U}^{(M)} \in \mathcal{R}^{M\times M}, \boldsymbol{U}^{(D)} \in \mathcal{R}^{D\times D}$, and $\boldsymbol{U}^{(N)} \in \mathcal{R}^{N\times N}$ capture the effects of the GMM component, the dimension of the mean vector, and the utterance index respectively. Here, the following matrix of n-th utterance is derived by fixing the third mode of $\mathcal{M}$:

$$\hat{\boldsymbol{\mu}}^{(n)} = \mathcal{G} \times_1 \boldsymbol{U}^{(M)} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)}(n,:) \tag{19}$$

In Equation 19, if $\boldsymbol{U}^{(N)}(n,:) \in \mathcal{R}^{1\times N}$ and the other part are regarded as the weight vector and the basis respectively, Equation 19 is the same as SVD. In this paper, Equation 19 is grouped to capture correlations among GMM components as follows:

$$\hat{\boldsymbol{\mu}}^{(n)} = \boldsymbol{U}^{(M)} \left\{ \mathcal{G} \times_2 \boldsymbol{U}^{(D)} \times_3 \boldsymbol{U}^{(N)}(n,:) \right\} = \boldsymbol{U}^{(M)} \boldsymbol{W}_n^\top \tag{20}$$

In Equation 20, $\boldsymbol{U}^{(M)}$ and $\boldsymbol{W}_n \in \mathcal{R}^{D\times M}$ are regarded as the basis matrix and the weight matrix. By truncating the basis matrix, an input utterance is decomposed as follows:

$$\boldsymbol{\mu}^{(new)} = \boldsymbol{U}^{(M)} \boldsymbol{W}_{(new)}^\top + \boldsymbol{b} \tag{21}$$

In Equation 21, $\boldsymbol{U}^{(M)} \in \mathcal{R}^{M\times K_M} (K_M \leq N)$ is the basis matrix, $\boldsymbol{W}_{(new)} \in \mathcal{R}^{D\times K_M}$ is the weight matrix. A $D \times K_M$ matrix represents language or speaker. Estimation of the weight matrix $\boldsymbol{W}$ can be regarded as effective estimation of GMM.

In [5], the weight matrix $\boldsymbol{W}$ is calculated based on the minimizing mean square errors (MMSE) criterion for Equation 21. Furthermore, $\boldsymbol{W}$ can also be estimated based on a ML or MAP criterion as in the case of the weight vector based on EVC. $\boldsymbol{W}$ is estimated based on a ML criterion as follows [6]:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}) \tag{22}$$

The following updating equation is derived by introducing the auxiliary function:

$$\mathrm{vec}(\hat{\boldsymbol{W}}) = \left[\sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{\Sigma}_m^{(YY)^{-1}} \right]^{-1} \mathrm{vec}(\boldsymbol{C}) \tag{23}$$

$$\boldsymbol{C} = \sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{(YY)^{-1}} (\overline{\boldsymbol{Y}}_m^{(tar)} - \overline{\gamma}_m^{(tar)} \boldsymbol{b}_m^{(0)}) \boldsymbol{U}_m \tag{24}$$

$$\boldsymbol{U}_m = \boldsymbol{U}^{(M)}(m,:) \in \mathcal{R}^{1\times K_M},\ \boldsymbol{b}_m^{(0)} = \boldsymbol{b}(m,:)^\top \in \mathcal{R}^{D\times 1} \tag{25}$$

In Equation 23, vec() is the vec operator that stacks the columns of a matrix into a vector.

On the other hand, $\boldsymbol{W}$ is estimated based on a MAP criterion as follows:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, p(\boldsymbol{Y}^{(tar)} | \boldsymbol{\lambda}^{(EV)}, \boldsymbol{W}) p(\boldsymbol{W}) \tag{26}$$

If $p(\mathrm{vec}(\boldsymbol{W}_n))$ is assumed $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the following updating equation of the weight matrix $\hat{\boldsymbol{W}}_n$ is derived:

$$\mathrm{vec}(\hat{\boldsymbol{W}}) = \left[\boldsymbol{I} + \sum_{m=1}^{M} \overline{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{\Sigma}_m^{(YY)^{-1}} \right]^{-1} \mathrm{vec}(\boldsymbol{C}) \tag{27}$$

$$\boldsymbol{C} = \sum_{m=1}^{M} \boldsymbol{\Sigma}_m^{(YY)^{-1}} \left[\overline{\boldsymbol{Y}}_t^{(tar)} - (\boldsymbol{W}_n \boldsymbol{U}_m^\top + \boldsymbol{b}_m) \right] \tag{28}$$

IV. Bilinear basis for tensor-based language/speaker representation

In Section III, only $\boldsymbol{U}^{(M)}$ is used as the basis matrix. However, the dimensions of the mean vectors of GMM can also have some correlation, so we use $\boldsymbol{U}^{(M)}$ and $\boldsymbol{U}^{(D)}$ as the bilinear basis matrices, In the case of bilinear basis, Equation 19 is grouped as follows:

$$\hat{\boldsymbol{\mu}}^{(n)} = \boldsymbol{U}^{(M)} \left\{ \mathcal{G} \times_3 \boldsymbol{U}^{(N)}(n,:) \right\} \boldsymbol{U}^{(D)^\top} \tag{29}$$

$$= \boldsymbol{U}^{(M)} \boldsymbol{W}'^{\top}_{n} \boldsymbol{U}^{(D)^\top} \tag{30}$$

By truncating the basis matrices, an input utterance is decomposed as follows:

$$\boldsymbol{\mu}^{(new)} = \boldsymbol{U}^{(M)} \boldsymbol{W}'^{\top}_{(new)} \boldsymbol{U}^{(D)^\top} + \boldsymbol{b} \tag{31}$$

$\boldsymbol{U}^{(M)} \in \mathcal{R}^{M \times K_M}$ and $\boldsymbol{U}^{(D)} \in \mathcal{R}^{D \times K_D}$ are the bilinear basis matrices, $\boldsymbol{W}'_{(new)} \in \mathcal{R}^{K_D \times K_M}$ is the weight matrix $(K_M \leq M, K_D \leq D)$. By using bilinear basis, dimensionality reduction along the axis of the dimension of the mean vectors is possible. Language/speaker representation based on bilinear basis can be regarded as a constraint on i-vector and EVC.

$\boldsymbol{W}'$ can also be estimated based on a ML criterion and the updating equation is as follows:

$$\mathrm{vec}(\hat{\boldsymbol{W}}') = \left[\sum_{m=1}^{M} \bar{\gamma}_m^{(tar)} \boldsymbol{U}_m^\top \boldsymbol{U}_m \otimes \boldsymbol{U}^{(D)^\top} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{U}^{(D)} \right]^{-1} \mathrm{vec}(C) \tag{32}$$

$$\boldsymbol{C} = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{m,t} \boldsymbol{U}^{(D)^\top} \boldsymbol{\Sigma}_m^{-1} (\boldsymbol{Y}_t^{(tar)} - \boldsymbol{b}_m^{(0)}) \boldsymbol{U}_m \tag{33}$$

V. Experimental evaluation for LID

A. Experimental conditions

To evaluate the proposed representation, LID experiments were carried out. We used The National Institute of Standards and Technology Language Recognition Evaluation (NIST LRE) 2003/2005/2007 as the speech corpora. These contain telephone conversation that have three types of duration of 3 sec, 10 sec and 30 sec. The number of target languages is 14: Arabic, Bengali, Farsi, German, Japanese, Korean, Russian, Spanish, Tamil, Thai, Vietnamese, Chinese, English, and Hindustani.

We used 56-dimensional MFCC (7MFCC+49SDC) as raw features, to which Cepstrum Mean Normalization (CMN) and Vocal Tract Length Normalization (VTLN) were applied. A diagonal-covariance UBM with 2048 mixtures was constructed from 24,577 utterances. To construct bases parameters, 23,665 out of 24,577 utterances were utilized and then language representation for each utterance were extracted. Table I shows the number of dimension for each feature. Probabilistic Linear Discriminant Analysis (PLDA) was used for classification; LDA and whitening was applied for each representation as preprocessing. For training of PLDA, 23,665 utterances were used. 6,474 utterances were used as test data. Equal Error Rate (EER) was used as evaluation criterion.

TABLE I
The number of dimensions of each feature.

	Language	Speaker
i-vector / EV-based	600	400
Tensor-based	1792 (= 56 × 32)	1920 (= 60 × 32)
Tensor-based bilinear	1568 (= 49 × 32)	1440 (= 45 × 32)

TABLE II
Results of LID task (EER [%]).

	3s	10s	30s
i-vector	24.14	16.03	11.68
EV-based (MMSE)	27.25	18.12	12.56
EV-based (ML)	24.51	15.71	10.91
EV-based (MAP)	25.63	15.57	**9.22**
EV-based SAT (ML)	**21.96**	14.29	10.16
EV-based SAT (MAP)	22.39	**13.52**	11.05
Tensor-based (MMSE)	27.53	17.56	10.31
Tensor-based (ML)	28.38	18.59	13.86
Tensor-based (MAP)	30.58	19.79	13.28
Tensor-based bilinear (MMSE)	28.41	18.35	11.47
Tensor-based bilinear (ML)	28.63	19.22	14.03

B. Results

Table II shows the results of experiment. 3s, 10s, 30s mean the duration of the duration. MMSE, ML, MAP mean the criteria for representation. From Table II, SAT works well in 3s and 10s cases for EV-based approaches. This suggest that SAT works effectively in the case that the duration of the session is short. Compared with all the representation, the order from better to worse becomes EV-based, i-vector, and tensor-based ones.

VI. Experimental evaluation for SR

A. Experimental conditions

As evaluation for SR, speaker recognition experiment using JNAS database [13] was carried out. In this corpus, Utterances recorded from 306 speakers (153 males and 153 females) by headset (H) and desktop (D) microphones are included. In the experiment, 120 speakers (60 males and 60 females) were used.

20-dimensional MFCCs with their delta and acceleration to which CMN was applied, were used. 2048 mixture UBM was constructed by 37,200 utterances from 138 speakers in JNAS. Table I shows the dimension of features. PLDA was used as classifier. This experiment was done under matched-microphone conditions (H-H, D-D) and mismatched-microphone conditions (H-D, D-H). 30 utterance per speaker was used for training, and 14,894 utterances in total were used for test. EER was used as evaluation criterion.

B. Results

Table III shows the result. Different from the case of LID, tensor-based bilinear representation based on MMSE criterion achieved the best performance. This result suggest that representation derived from tensor factor analysis captures the essential information of speaker effectively.

TABLE III
RESULTS OF SR TASK (EER [%]).

	H-H	H-D	D-H	D-D
i-vector	0.38	3.18	4.46	0.43
EV-based (MMSE)	0.35	2.67	**3.18**	0.37
EV-based (ML)	0.84	3.58	4.86	0.61
EV-based (MAP)	0.35	4.16	5.07	0.35
EV-based SAT (ML)	0.61	4.39	7.02	0.58
EV-based SAT (MAP)	0.56	4.29	6.19	0.56
Tensor-based (MMSE)	0.43	3.21	4.04	0.35
Tensor-based (ML)	2.72	5.72	7.17	2.04
Tensor-based (MAP)	0.40	4.99	5.97	0.39
Tensor-based bilinear (MMSE)	**0.33**	**2.66**	3.27	**0.30**
Tensor-based bilinear (ML)	2.21	5.07	6.32	1.63

VII. CONCLUSION

This paper has proposed a novel approach to speech representation for both speaker recognition and language identification by characterizing the entire feature space by a tensor. An utterance is not modeled as its GMM-based supervector but as its matrix and the entire set of utterances is modeled as its tensor. By applying tensor factor analysis, we obtain a new representation for an input utterance. Experimental evaluations for speaker recognition and language identification show that our proposed approach has effectiveness especially for the speaker recognition task.

REFERENCES

[1] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via Ivectors and Dimensionality Reduction," Proc. INTERSPEECH, pp. 857–860, 2011.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2011.

[3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support Vector Machines using GMM Supervectors for Speaker Verification," IEEE Signal Processing Letters, vol. 13, pp. 308–311, 2006.

[4] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice Conversion Based on Gaussian Mixture Model," Proc. INTERSPEECH, pp. 2446–2449, 2006.

[5] Trinh Tuan Tu, Daisuke Saito, Nobuaki Minematsu, and Kekichi Hirose, "Speaker identification using tensor-based representation of speakers," The proceeding of Spring meeting of Acoustic Society of Japan, pp. 217–220, 2015 (in Japanese).

[6] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-Many Voice Conversion Based on Tensor Representation of Speaker Space," Proc. INTERSPEECH, pp. 653–656, 2011.

[7] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, "Rapid Speaker Adaptation in Eigenvoice Space," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 6, pp. 695–707, 2000.

[8] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Speaker Adaptive Training for One-to-Many Eigenvoice Conversion Based on Gaussian Mixture Model," Proc. INTERSPEECH, pp. 1981–1984, 2007.

[9] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," Psychometria, vol. 31, no. 3, pp. 279–311, 1966.

[10] D. Saito, N. Minematsu, and K. Hirose, "Effects of Speaker Adaptive Training on Tensor-based Arbitrary Speaker Conversion," Proc. INTERSPEECH, pp. 98–101, 2012

[11] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling," Proc. ICASSP, vol. 1, pp. 305–308, 1994.

[12] M. H. Bahari, N. Dehak, and H. Van hamme, "Gaussian mixture model weight supervector decomposition and adaptation," Tech. Rep., 2013.

[13] "Jnas: Japanese newspaper article sentences,"
http://www.milab.is.tsukuba.ac.jp/jnas/instruct.html