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

RESEARCH Open Access

PLDA in the i-supervector space for text-independent speaker verification

Ye Jiang[1], Kong Aik Lee[2] and Longbiao Wang[1*]

Abstract

In this paper, we advocate the use of the uncompressed form of i-vector and depend on subspace modeling using *probabilistic linear discriminant analysis* (PLDA) in handling the speaker and session (or channel) variability. An i-vector is a low-dimensional vector containing both speaker and channel information acquired from a speech segment. When PLDA is used on an i-vector, dimension reduction is performed twice: first in the i-vector extraction process and second in the PLDA model. Keeping the full dimensionality of the i-vector in the i-supervector space for PLDA modeling and scoring would avoid unnecessary loss of information. We refer to the uncompressed i-vector as the i-supervector. The drawback in using the i-supervector with PLDA is the inversion of large matrices in the estimation of the full posterior distribution, which we show can be solved rather efficiently by portioning large matrices into smaller blocks. We also introduce the Gaussianized rank-norm, as an alternative to whitening, for feature normalization prior to PLDA modeling. We found that the i-supervector performs better during normalization. A better performance is obtained by combining the i-supervector and i-vector at the score level. Furthermore, we also analyze the computational complexity of the i-supervector system, compared with that of the i-vector, at four different stages of loading matrix estimation, posterior extraction, PLDA modeling, and PLDA scoring.

Keywords: Speaker verification; Probabilistic linear discriminant analysis; I-supervector; I-vector

1 Introduction

Recent research in text-independent speaker verification has been focusing on the problem of compensating the mismatch between training and test speech segments. Such mismatch in most part is due to the variations induced by the transmission channel. There are two fundamental approaches to tackling this problem. The first approach operates at the front-end via the exploration of discriminative information in speech in the form of features (e.g., voice source, spectro-temporal, prosodic, high-level) [1-6]. The second approach relies on the effective modeling of speaker characteristic in the classifier design (e.g., GMM-UBM, GMM-SVM, JFA, i-vector, PLDA) [4,7-15]. In this paper, we focus on the speaker modeling.

Over the past few years, many approaches based on the use of Gaussian mixture models (GMM) in a GMM universal background model (GMM-UBM) framework [7] have been proposed to improve the performance of speaker verification system. The GMM-UBM is a generative model in which a speaker model is trained only on data from the same speaker. New criteria have then been developed that allow discriminative learning of generative models. Support vector machine (SVM) is acknowledged as one of the pre-eminent discriminative approaches [16-18], and it has been successfully combined with GMM, such as the GMM-SVM [8,9,19-21]. Nevertheless, approaches based on GMM-SVM are unable to cope well with the channel effects [22,23]. To compensate for the channel effects, it was shown using the joint factor analysis (JFA) technique that the speaker and channel variability can be confined as two disjoint subspaces in the parameter spaces of GMM [12,24]. The word 'joint' refers to the fact that not only the speaker, but also the channel variability is treated in a single JFA model. However, it was been reported that the channel space obtained by the JFA does contain some residual speaker information [25].

Inspired by the JFA approach, it was shown in [13] that speaker and session variability can be represented by a single subspace referred to as the total variability space. The major motivation for defining such a subspace is to

* Correspondence: wang@vos.nagaokaut.ac.jp
[1]Top Runner Incubation Center for Academia-Industry Fusion, Nagaoka University of Technology, Nagaoka 940-2188, Japan
Full list of author information is available at the end of the article

extract a low-dimensional identity vector (i.e., the so-called i-vector) from the feature sequence of a speech segment. The advantage of i-vector is that it represents a speech segment as a fixed-length vector instead of a variable-length sequence of acoustic features. This greatly simplifies the modeling and scoring processes in speaker verification. For instance, we can assume that the i-vector is generated from a Gaussian density [13] instead of the mixture of Gaussian densities as usual in the case of acoustic features [7]. In this regard, linear discriminant analysis (LDA) [13,26,27], nuisance attribute projection (NAP) [8,13,28], within-class covariance normalization (WCCN) [13,29,30], probabilistic LDA (PLDA) [10,31], and the heavy-tailed PLDA [32] have shown to be effective for such fixed-length data. In this paper, we focus on PLDA with Gaussian prior instead of heavy-tailed prior. It was recently shown in [33] that the advantage of the heavy-tailed assumption diminishes with a simple length normalization on the i-vector before PLDA modeling.

Because the total variability matrix is always a low-rank rectangular matrix, a dimension reduction process is also imposed by the i-vector extractor [12]. In this study, we advocate the use of the uncompressed form of the i-vector. Similar to that in [13], our extractor converts speech sequence into a fixed-length vector but retains its dimensionality in the full supervector space. Modeling of speaker and session variability is then carried out using PLDA, which has shown to be effective in handling high-dimensional data. By doing so, we avoid reducing the dimensionality of the i-vector twice: first in the extraction process and second in the PLDA model. Any dimension reduction procedure will unavoidably discard information. Our intention is therefore to keep the full dimensionality until the scoring stage with PLDA and to investigate the performance of PLDA in the i-supervector space. We refer to the uncompressed form of i-vector as the *i-supervector*, or the *identity* supervector, following the nomenclature in [13,29]. Similar to that in the i-vector extraction, the i-supervector is computed as the posterior mean of a latent variable, but with a much higher dimensionality.

The downside of using i-supervector with PLDA is that we have to deal with the inversion of large matrices. The size of the matrices becomes enormous when more sessions are available for each speaker in the development data[a]. One option is to estimate the subspaces in a decoupled manner, which might lead to suboptimal solution [12,24]. In [34], we showed that the joint estimation of subspaces can be accomplished by partitioning large matrices into smaller blocks, thereby making the inversion and the joint estimation feasible. In this study, we present the same approach with more detail and further refinement. We also look into various normalization methods and introduce the use of the Gaussianized rank-norm for the PLDA. In the experiments, we compare the performance of both i-vector and i-supervector under no normalization and various normalization conditions. Meanwhile, a fusion system that combines the i-vector and i-supervector is presented as well. In addition, we provide an analysis of the computational complexity associated with the i-vector and i-supervector at four different stages: loading matrix estimation, i-vector and i-supervector extraction, PLDA model training, and verification score calculation.

The paper is organized as follows. In Section 2, we introduce the i-vector paradigm, which includes the formulation of the i-vector and i-supervector and its relationship to the classical maximum *a posteriori* (MAP). Section 3 introduces the probabilistic LDA, where we show that the inversion of a large matrix in PLDA can be solved by exploiting some inherent structure of the precision matrix. Section 4 deals with PLDA scoring and introduces the Gaussianized rank-norm. We present some experimental results in Section 5 and conclude the paper in Section 6.

2 I-vector paradigm

2.1 I-vector extraction

The purpose of i-vector extraction is to represent variable-length utterances with fixed-length low-dimensional vectors. The fundamental assumption is that the feature vector sequence, $\mathcal{O}$, was generated from a session-specific GMM. Furthermore, the mean supervector obtained by stacking the means from all mixtures, $\mathbf{m}$, is constrained to lie in a low-dimensional subspace with origin $\mathscr{M}$ as follows:

$$\mathbf{m} = \mathscr{M} + \mathbf{T}\mathbf{x}, \tag{1}$$

where $\mathbf{m}$ and $\mathscr{M}$ are the mean supervectors of the speaker (and session)-dependent GMM and the UBM, respectively. The subspace spanned by the columns of the matrix $\mathbf{T}$ captures the speaker and session variability, and hence the name *total variability* [13]. The weighted combination of the columns of $\mathbf{T}$, as determined by the latent variable $\mathbf{x}$, gives rise to the mean supervector $\mathbf{m}$ with $\mathscr{M}$ as an additive factor.

The i-vector extraction process was formulated as the MAP estimation problem [13,35]. Notice that in (1), the equation is concerned with the construction of the mean supervector $\mathbf{m}$ from the parameters $\{\mathscr{M}, \mathbf{T}\}$ and the latent variable $\mathbf{x}$. The variable $\mathbf{x}$ is unobserved (or latent) as is the supervector. The optimal value of $\mathbf{x}$ is determined by the observed sequence $\mathcal{O}$ and is given by the mode (equivalent to the mean in the current case) of the posterior distribution of the latent variable $\mathbf{x}$:

$$\phi = \arg\max_{\mathbf{x}} \left[\prod_{c=1}^{C} \prod_{t=1}^{N_c} \mathcal{N}(o_t | \mathcal{M}_c + \mathbf{T}_c \mathbf{x}, \Phi_c) \right] \mathcal{N}(\mathbf{x} | 0, \ \mathbf{I}). \tag{2}$$

The first point to note is that the latent variable $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$ is assumed to follow a standard normal prior. The parameters $\mathcal{M}_c$ and Φ_c denote the mean vector and covariance matrix of the c-th mixture of the UBM, while N_c indicates the number of frames o_t aligned to each of the C mixtures. Also, we decompose the total variability matrix $\mathbf{T} = \left[\mathbf{T}_1^{\mathrm{T}}, \ \mathbf{T}_2^{\mathrm{T}}, \ \ldots, \ \mathbf{T}_C^{\mathrm{T}} \right]^{\mathrm{T}}$ to its component matrices, one associated with each Gaussian. Given an observation sequence $\mathcal{O}$, its i-vector representation is given by (2), the solution [13] of which is given by

$$\phi_x = \mathbf{L}_x^{-1} \left(\sum_{c=1}^{C} \mathbf{T}_c^{\mathrm{T}} \Phi_c^{-1} \mathbf{f}_c \right), \tag{3}$$

where

$$\mathbf{L}_x^{-1} = \left(\mathbf{I} + \sum_{c=1}^{C} N_c \mathbf{T}_c^{\mathrm{T}} \Phi_c^{-1} \mathbf{T}_c \right)^{-1} \tag{4}$$

is the posterior covariance, $\mathbf{f}_c = \sum_t \gamma_{c,t} o_t - N_c \mathcal{M}_c$ is the centralized first-order statistics [35] for the c-th Gaussian, and $\gamma_{c,t}$ denotes the occupancy of vector o_t to the c-th Gaussian. Since $\mathbf{T}$ is always a low-rank rectangular matrix, the dimension D of the i-vector is much smaller compared to that of the supervector, i.e., $D \ll C{\cdot}F$, where F is the dimensionality of the acoustic feature.

2.2 I-supervector extraction

Consider the case where the latent variable is allowed to grow into the full supervector space, for which $D = C{\cdot}F$. One straightforward approach to achieving this is by using a *CF*-by-*CF* full matrix for $\mathbf{T}$ in (1). However, the number of parameters would be enormous, causing difficulty in the training. Another option is to impose a diagonal constraint on the loading matrix as follows:

$$\mathbf{m} = \mathcal{M} + \mathbf{D}\mathbf{z}, \tag{5}$$

where $\mathbf{D}$ is now a *CF*-by-*CF* diagonal matrix and the latent variable $\mathbf{z}$ has the same dimensionality as the mean supervector $\mathbf{m}$. Similar to the variable $\mathbf{x}$ in (1), the variable $\mathbf{z}$ is unobserved. Given an observed sequence $\mathcal{O}$, we estimate the mode of the posterior distribution as follows:

$$\phi = \underset{\mathbf{z}}{\operatorname{argmax}} \left[\prod_{c=1}^{C} \prod_{t=1}^{N_c} \mathcal{N}(o_t | \mathcal{M}_c + \mathbf{D}_c \mathbf{z}_c, \Phi_c) \mathcal{N}(\mathbf{z}_c | 0, \mathbf{I}) \right]. \tag{6}$$

Here, $\mathbf{z}_c$ is the sub-vector of $\mathbf{z}$ and $\mathbf{D}_c$ is the F-by-F sub-matrix corresponding to the mixture c. Notice that such notations are necessary as the likelihood is computed over the acoustic vector o_t. Following the procedure as in [35], it can be shown that solution to (6) is a *CF*-by-1 supervector:

$$\phi_z = \mathbf{L}_z^{-1} \left(\mathbf{D}^{\mathrm{T}} \boldsymbol{\Phi}^{-1} \mathbf{f} \right), \tag{7}$$

where $\mathbf{L}_z^{-1}$ is a *CF*-by-*CF* diagonal matrix given by

$$\mathbf{L}_z^{-1} = \left(\mathbf{I} + \mathbf{D}^{\mathrm{T}} \mathbf{N} \boldsymbol{\Phi}^{-1} \mathbf{D} \right)^{-1}. \tag{8}$$

In (7), $\mathbf{f}$ is the *CF*-by-1 supervector obtained by concatenating the $\mathbf{f}_c$ from all mixtures (see Figure 1). In

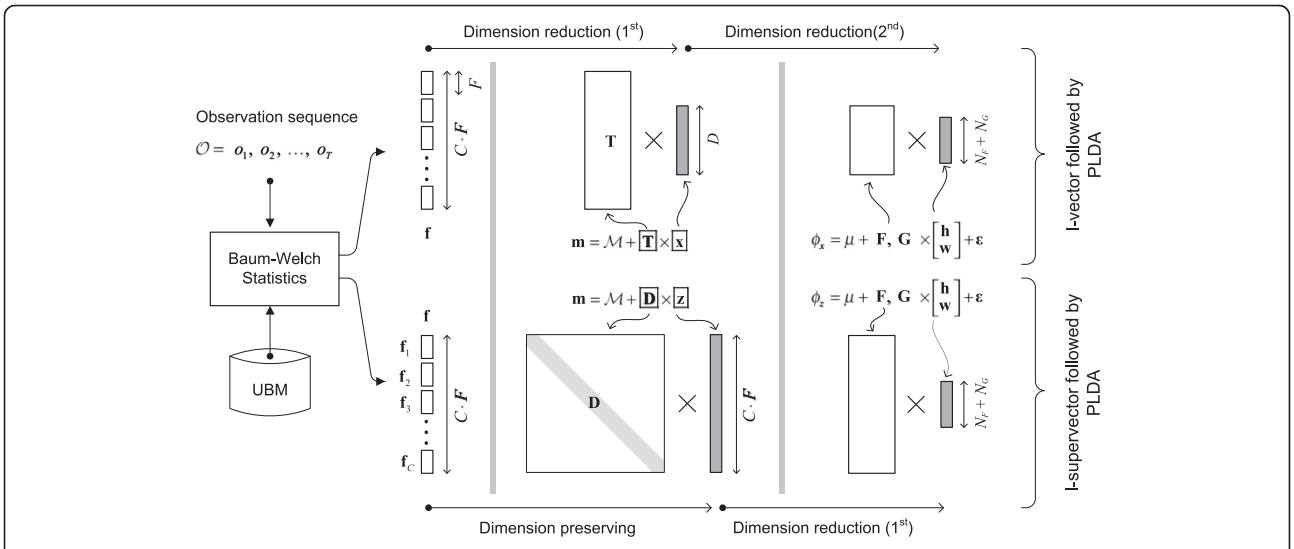

Figure 1 Comparison between i-vector (upper panel) and i-supervector (lower panel) extraction followed by PLDA modeling.

(8), $\mathbf{N}$ is the *CF*-by-*CF* diagonal matrix whose diagonal blocks are $N_c\mathbf{I}$, and $\mathbf{\Phi}$ is a block diagonal matrix with Φ_c at its diagonal. Recall that N_c and $\mathbf{f}_c$ are the occupancy count and centralized first-order statistics extracted using the UBM.

We refer to ϕ_z as the *i-supervector* analogous to the *i-vector* since ϕ_z is computed as the posterior mean of a latent variable similar to that in the i-vector extraction, but with a much higher dimensionality. It is worth to note that there exist some subtle differences between the i-supervector extraction and the classical MAP estimation of GMM [36]. In particular, the so-called *relevance* MAP widely used in the GMM-UBM [7] could be formulated in similar notations. In particular, the mean supervector of the adapted GMM is given by

$$\phi_{\text{rel}} = \mathscr{M} + (\tau\mathbf{I} + \mathbf{N})^{-1}\mathbf{f}. \tag{9}$$

One could deduce (9) from (7) and (8) by setting $\mathbf{D}^{\mathrm{T}}\mathbf{D} = \tau^{-1}\mathbf{\Sigma}$ and using the results in (5). The parameter τ is referred to as the relevance factor, which is set empirically in the range between 8 and 16 [7]. This is different from that in (7), where the matrix $\mathbf{D}$ is trained from a dataset using the EM algorithm in a manner similar to the matrix $\mathbf{T}$ for the i-vector. Secondly, the i-supervector is taken as the posterior of the latent variable $\mathbf{z}$ which is absent in the relevance MAP formulation.

The i-supervector extractor can be implemented by adopting the diagonal modeling part of the JFA [12,24] with a slight modification: the diagonal model $\mathbf{D}$ is trained per utterance instead of per speaker basis in order to capture both speaker and session variability. Figure 2 summarizes the EM steps. The diag(.) operator sets the off-diagonal elements to zeros, only the diagonal elements are computed in our implementation. Notice that the sufficient statistics $\{\mathbf{f}, \mathbf{N}\}$ are session-dependent. We omitted the session index for simplicity.

2.3 From i-vector to i-supervector

The i-vector extraction is formulated in probabilistic terms based on a latent variable model as in (2), similarly for the case of i-supervector in (6). One obvious benefit is that in addition to obtain the i-vector as the posterior mean ϕ_x of the latent variable $\mathbf{x}$, we could also compute the posterior covariance (4) which quantifies the uncertainty of the estimate and fold in the information in subsequent modeling [37]. Nevertheless, any form of dimension reduction would unavoidably discard information. Following the same latent variable modeling paradigm, we proposed the i-supervector as an uncompressed form of i-vector representation.

Figure 1 compares the i-vector and i-supervector approaches from the extraction process to the subsequent PLDA modeling (recall that the parameter C denotes the number of mixtures in the UBM. F is the size of the acoustic feature vectors. D is the length of the i-vector while the i-supervector has a much higher dimensionality of $C{\cdot}F$.). The biggest difference is that there are two rounds of

Maximum likelihood estimation of loading matrix $\mathbf{D}$

Input : $\mathbf{\Phi}$ and $\{\mathbf{f}, \mathbf{N}\}$ of all utterances
Output : $\mathbf{D}$
begin

Initialize $\mathbf{D}$ randomly and set $\mathbf{\Omega}_b$ and $\mathbf{\Omega}_b$ to zeros

Repeat 1 to `max_iter`

// Expectation step, loop for each session

For each session

$$\mathbf{L}_z^{-1} = \left(\mathbf{I} + \mathbf{D}^{\mathrm{T}}\mathbf{N}\mathbf{\Phi}^{-1}\mathbf{D}\right)^{-1}$$

$$\phi_z = \mathbf{L}_z^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{\Phi}^{-1}\mathbf{f}$$

$$E\left\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\right\} = \mathbf{L}_z^{-1} + diag\left(\phi_z\phi_z^{\mathrm{T}}\right)$$

$$\mathbf{\Omega}_a = \mathbf{\Omega}_a + diag\left(\mathbf{N}\cdot E\left\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\right\}\right)$$

$$\mathbf{\Omega}_b = \mathbf{\Omega}_b + diag\left(\mathbf{f}\phi_z^{\mathrm{T}}\right)$$

end

// Maximization step

$\mathbf{D} = \mathbf{\Omega}_b\mathbf{\Omega}_a^{-1}$ // Update the diagonal model

until

end

Figure 2 The EM steps for estimating the loading matrix D for the i-supervector extractor.

dimension reduction which occurred in the i-vector PLDA system, whereas there is only one time reduction for the case of i-supervector PLDA. In this paper, our motivation is to keep the full dimensionality of the supervector as the input to the PLDA model which has shown to be an efficient model for high-dimensional data [10]. We envisage that more information would be preserved via the use of i-supervector, which could be exploited with the use of PLDA.

3 PLDA modeling in i-supervector space

3.1 Probabilistic LDA

The i-vector and the i-supervector represent a speech segment as a fixed-length vector instead of a variable-length sequence of vectors. Taking the fixed-length vector ϕ_{ij} as input, PLDA assumes that it is generated from a Gaussian density as follows:

$$p\left(\phi_{ij}\right) = \mathcal{N}\left(\phi_{ij}|\boldsymbol{\mu}, \mathbf{F}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma}\right), \tag{10}$$

where $\boldsymbol{\mu}$ denotes the global mean and $\boldsymbol{\Gamma} = \mathbf{F}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma}$ is the covariance matrix. Here, ϕ_{ij} is i-supervector (or i-vector) representing the j-th session of the i-th speaker. We use ϕ referring either to the i-vector ϕ_x or i-supervector ϕ_z in the subsequent discussion.

The strength of PLDA lies at the modeling of the covariance $\boldsymbol{\Gamma}$ in structural form as $\mathbf{F}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma}$. To see this, we rewrite (10) as marginal density:

$$p(\phi_{ij}) = \iint p(\phi_{ij}|\mathbf{h}_i, \mathbf{w}_{ij})\mathcal{N}(\mathbf{h}_i|0, \mathbf{I})\,\mathcal{N}(\mathbf{w}_{ij}|0, \mathbf{I})\,d\mathbf{h}_i\,d\mathbf{w}_{ij}, \tag{11}$$

where the conditional density is given by

$$p(\phi_{ij}|\mathbf{h}_i, \mathbf{w}_{ij}) = \mathcal{N}(\phi|\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij}, \boldsymbol{\Sigma}). \tag{12}$$

In the above equations, $\mathbf{h}_i$ is the speaker-specific latent variable pertaining to the i-th speaker, while $\mathbf{w}_{ij}$ is the session-specific latent variable corresponding to the j-th session of the i-th speaker. Both latent variables are assumed to follow a standard Gaussian prior. The low-rank matrices $\mathbf{F}$ and $\mathbf{G}$ model the subspaces corresponding to speaker and session variability (we denote their rank as N_F and N_G, respectively), while the diagonal matrix $\boldsymbol{\Sigma}$ covers the remaining variation. From (12), the mean vector of the conditional distribution is given by

$$\boldsymbol{\mu}_{ij} = \boldsymbol{\mu} + [\,\mathbf{F}, \mathbf{G}\,]\begin{bmatrix}\mathbf{h}_i \\ \mathbf{w}_{ij}\end{bmatrix}. \tag{13}$$

Comparing (1) and (13), we see that both i-vector extraction process and the PLDA model involve dimension reduction via a similar form of subspace modeling. This observation motivates us to explore the use of PLDA on i-supervector. The extraction process serves as the front-end which converts a variable-length sequence $\mathcal{O}$ to a fixed-length vector without reducing the dimension. Speaker modeling and channel compensation are then carried out in the original supervector space.

The downside of using i-supervector with PLDA is that we have to deal with large matrices as illustrated in the lower panel of Figure 1. The size of the matrices becomes enormous when more sessions are available for each speaker in the development data. This is typically the case for speaker recognition where the number of utterances per speaker is usually in the range from ten to over a hundred [38,39]. In the following, we estimate the parameters $\{\,\boldsymbol{\mu},\ \mathbf{F},\ \mathbf{G},\ \boldsymbol{\Sigma}\,\}$ of the PLDA model using the expectation maximization (EM) algorithm. We show how large matrices could be partitioned into sub-matrices, thereby making the matrix inversion and EM steps feasible.

3.2 E-step: joint estimation of posterior means

We assume that our development set consists of speech samples from N speakers each having J sessions, though the number of sessions J could be different for each speaker. All the J observations from the i-th speaker are collated to form the compound system [10]:

$$\underbrace{\begin{bmatrix}\phi_{i1}\\ \phi_{i2}\\ \vdots\\ \phi_{iJ}\end{bmatrix}}_{\tilde{\phi}_i} = \underbrace{\begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{\mu}\\ \vdots\\ \boldsymbol{\mu}\end{bmatrix}}_{\tilde{\boldsymbol{\mu}}} + \underbrace{\begin{bmatrix}\mathbf{F} & \mathbf{G} & 0 & \cdots & 0\\ \mathbf{F} & 0 & \mathbf{G} & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{F} & 0 & 0 & \cdots & \mathbf{G}\end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix}\mathbf{h}_i\\ \mathbf{w}_{i1}\\ \mathbf{w}_{i2}\\ \vdots\\ \mathbf{w}_{iJ}\end{bmatrix}}_{\mathbf{y}_i} + \underbrace{\begin{bmatrix}\boldsymbol{\varepsilon}_{i1}\\ \boldsymbol{\varepsilon}_{i2}\\ \vdots\\ \boldsymbol{\varepsilon}_{iJ}\end{bmatrix}}_{\boldsymbol{\varepsilon}_i}. \tag{14}$$

Each row in (14) says that each observation $\phi_{ij} = \boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\varepsilon}_{ij}$ consists of a speaker-dependent component $\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i$ and session-dependent component $\mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\varepsilon}_{ij}$, where $\boldsymbol{\varepsilon}_{ij} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ is responsible for the residual variation in (10). In the E-step, we infer the posterior mean of the compound latent variable $\mathbf{y}_i = \left[\,\mathbf{h}_i^{\mathrm{T}},\ \ \mathbf{w}_{i1}^{\mathrm{T}},\ \ \ldots,\ \ \mathbf{w}_{iJ}^{\mathrm{T}}\,\right]^{\mathrm{T}}$ as follows:

$$E\{\mathbf{y}_i\} = \mathbf{L}^{-1} \cdot \tilde{\mathbf{A}}^{\mathrm{T}} \tilde{\boldsymbol{\Sigma}}^{-1} \left(\tilde{\phi_i} - \boldsymbol{\mu}\right), \tag{15}$$

where $\tilde{\boldsymbol{\Sigma}}$ is a block diagonal matrix whose diagonal blocks are $\boldsymbol{\Sigma}$, and $\mathbf{L}^{-1}$ is the posterior covariance given by

$$\mathbf{L}^{-1} = \left[\tilde{\mathbf{A}}^{\mathrm{T}} \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\mathbf{A}} + \mathbf{I}\right]^{-1}. \tag{16}$$

The posterior inference involves the inversion of the matrix $= \tilde{\mathbf{A}}^{\mathrm{T}} \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\mathbf{A}} + \mathbf{I}$. Following the notations in (14), we could express the matrix inversion as

$$\mathbf{L}^{-1}=\begin{bmatrix} J\mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{F}+\mathbf{I} & \mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G} & \mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G} & \cdots & \mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G} \\ \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{F} & \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G}+\mathbf{I} & 0 & \cdots & 0 \\ \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{F} & 0 & \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G}+\mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{F} & 0 & 0 & \cdots & \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G}+\mathbf{I} \end{bmatrix}^{-1}. \tag{17}$$

The matrix is large as we consider the joint inference of latent variables $\{\mathbf{h}_i,\ \mathbf{w}_{i1},\ \ldots,\ \mathbf{w}_{iJ}\}$ representing a speaker and all sessions from the same speaker. The size of the matrix increases with the number of sessions J, while more sessions are always desirable for more robust parameter estimation. Direct inversion of the matrix becomes intractable.

The precision matrix $\mathbf{L}$ possesses a unique structure since all sessions from the same speakers are tied to one speaker-specific latent variable. As depicted in (17) and (18), the matrix $\mathbf{L}$ can be partitioned into four sub-matrices: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}^{\mathrm{T}}$, and $\mathbf{C}$. Using the *partitioned inverse formula* [40], the inverse of the matrix $\mathbf{L}$ could be obtained as

$$\mathbf{L}^{-1}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^{\mathrm{T}} & \mathbf{C}\end{bmatrix}^{-1}=\begin{bmatrix}\mathbf{M} & -\mathbf{M}\mathbf{B}\mathbf{C}^{-1}\\ -\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{M} & \mathbf{C}^{-1}+\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{M}\mathbf{B}\mathbf{C}^{-1}\end{bmatrix}, \tag{18}$$

Where

$$\mathbf{M}=\left(\mathbf{A}-\mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}\right)^{-1}. \tag{19}$$

The matrix $\mathbf{M}^{-1}$ is known as the Schur complement of $\mathbf{L}$ with respect to $\mathbf{C}$ [18]. Using these formulae, there are still two matrices to be inverted. The first is $\mathbf{C}^{-1}$ in the left-hand side of (18) and the second is the $\mathbf{M}$ in (19). The inversion $\mathbf{C}^{-1}$ is simple as $\mathbf{C}$ is block diagonal, where the inversion $Q = (\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G} + \mathbf{I})^{-1}$ can be computed directly from the N_G-by-N_G matrix. Using the notations in (14) and (18), $\mathbf{M}$ is given by

$$\mathbf{M}=\left[J\,\mathbf{F}^{\mathrm{T}}\mathbf{J}\mathbf{F}+\mathbf{I}\right]^{-1}, \tag{20}$$

where $\mathbf{J}$ is obtained via the matrix inversion lemma:

$$\mathbf{J}=\left[\mathbf{G}\mathbf{G}^{\mathrm{T}}+\boldsymbol{\Sigma}\right]^{-1}=\boldsymbol{\Sigma}^{-1}-\boldsymbol{\Sigma}^{-1}\mathbf{G}\left(\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G}+\mathbf{I}\right)^{-1}\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}. \tag{21}$$

Using (18) in (15), it can be shown that the posterior mean of the speaker-specific latent variable $\mathbf{h}_i$ is given by

$$E\{\mathbf{h}_i\}=\mathbf{M}\left[\mathbf{F}^{\mathrm{T}}\mathbf{J}\cdot\sum_{j=1}^{J}\left(\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}\right)\right], \tag{22}$$

while the session-specific posterior mean of $\mathbf{w}_{ij}$ could be inferred as

$$E\{\mathbf{w}_{ij}\}=\underbrace{\left(\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G}+\mathbf{I}\right)^{-1}}_{Q}\ \mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left[\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}-\mathbf{F}\cdot E\{\mathbf{h}_i\}\right]. \tag{23}$$

One interesting point to note from (23) is that the i-supervector $\boldsymbol{\phi}_{ij}$ is first centralized to the global mean $\boldsymbol{\mu}$ and the speaker mean $\mathbf{F}\cdot E\{\mathbf{h}_i\}$ before projection to the session variability space.

From computation perspective, the matrices $Q = (\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{G} + \mathbf{I})^{-1}$, $\boldsymbol{\Lambda} = Q\cdot\mathbf{G}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{F}$, and $\mathbf{J}$ could be pre-computed and used for all sessions and speakers in the E-step. The matrix $\mathbf{M}$ depends on the number of sessions J per speaker. In the event where J is different for each speaker (which is usually the case), we compute

$$\mathbf{M}_J\equiv\left[J\cdot\mathbf{F}^{\mathrm{T}}\mathbf{J}\mathbf{F}+\mathbf{I}\right]^{-1}=\mathbf{V}[J\cdot\mathbf{E}+\mathbf{I}]^{-1}\mathbf{V}^{\mathrm{T}}, \tag{24}$$

where $\mathbf{F}^{\mathrm{T}}\mathbf{J}\mathbf{F} = \mathbf{V}\mathbf{E}\mathbf{V}^{\mathrm{T}}$ is obtained via eigenvalue decomposition in which $\mathbf{V}$ is the square matrix of eigenvectors and is $\mathbf{E}$ the diagonal matrix of eigenvalues.

3.3 M-step: model estimation

The M-step can also be formulated in terms of sub-matrices. Let $\tilde{\mathbf{w}}_{ij}=\begin{bmatrix}\mathbf{h}_i^{\mathrm{T}}, & \mathbf{w}_{ij}^{\mathrm{T}}\end{bmatrix}^{T}$ be a compound vector by appending $\mathbf{h}_i$ to each session $\mathbf{w}_{ij}$ belonging to the same speaker. We update the loading matrices $\mathbf{F}$ and $\mathbf{G}$ jointly as follows:

$$[\hat{\mathbf{F}},\hat{\mathbf{G}}]=\left\{\sum_{i=1}^{N}\sum_{j=1}^{J}\left(\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}\right)E\left\{\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\}\right\}\left\{\sum_{i=1}^{N}\sum_{j=1}^{J}E\left\{\tilde{\mathbf{w}}_{ij}\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\}\right\}^{-1}, \tag{25}$$

where $E\left\{\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\}$ could be obtained by concatenating the results from (22) and (23). The second moment $E\left\{\tilde{\mathbf{w}}_{ij}\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\}$ is computed for each individual session and speaker as follows:

$$E\left\{\tilde{\mathbf{w}}_{ij}\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\}=\begin{bmatrix}\mathbf{M} & -\mathbf{M}\boldsymbol{\Lambda}^{\mathrm{T}}\\ -\boldsymbol{\Lambda}\mathbf{M} & Q+\boldsymbol{\Lambda}\mathbf{M}\boldsymbol{\Lambda}^{\mathrm{T}}\end{bmatrix}+E\left\{\tilde{\mathbf{w}}_{ij}\right\}\left\{\tilde{\mathbf{w}}_{ij}^{\mathrm{T}}\right\} \tag{26}$$

The covariance matrix of the PLDA model could then be updated as

$$\hat{\boldsymbol{\Sigma}}=\frac{1}{N\cdot J}\sum_{i=1}^{N}\sum_{j=1}^{J}\mathrm{diag}\left[\left(\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}\right)\left(\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}\right)^{\mathrm{T}}-[\hat{\mathbf{F}},\hat{\mathbf{G}}]E\left\{\tilde{\mathbf{w}}_{ij}\right\}\left(\boldsymbol{\phi}_{ij}-\boldsymbol{\mu}\right)^{\mathrm{T}}\right], \tag{27}$$

where the operator diag(·) diagonalizes a matrix by setting the off-diagonal elements to zeros.

4 Likelihood ratio computation

4.1 Model comparison

Speaker verification is a binary classification problem, where a decision has to be made between two hypotheses with respect to a decision threshold. The null hypothesis H_0 says that the test segment is from the target speaker, while the alternative H_1 hypothesizes the opposite. Using the latent variable modeling approach with PLDA, H_0 and H_1 correspond to the models as shown in Figure 3. In the model H_0, $\{\phi_1, \phi_2\}$ belong to the same speaker and hence share the same speaker-specific latent variable $\mathbf{h}_{1,2}$. On the other hand, $\{\phi_1, \phi_2\}$ belong to different speakers and hence have separate latent variables, $\mathbf{h}_1$ and $\mathbf{h}_2$, in the model H_1. The verification score is calculated as the log-likelihood ratio between two models:

$$s(\phi_1, \phi_2) = \log p(\phi_1, \phi_2|H_0) - \log p(\phi_1, \phi_2|H_1), \quad (28)$$

where the likelihood terms are evaluated using (10) (we shall give more details in the next section). One key feature of the PLDA scoring function in (28) is that no speaker model is built or trained. The verification scores are computed by comparing the likelihood of two different models which describe the relationship between the training and test i-supervectors (or i-vector) through the use of PLDA model.

4.2 PLDA verification score

To solve for (28), we first recognize from Figure 1 that the generative equation for the model H_0 is given by

$$\underbrace{\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}}_{\tilde{\phi}} = \underbrace{\begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{\mu}\end{bmatrix}}_{\tilde{\boldsymbol{\mu}}} + \underbrace{\begin{bmatrix}\mathbf{F} & \mathbf{G} & 0\\ \mathbf{F} & 0 & \mathbf{G}\end{bmatrix}}_{\tilde{\mathbf{A}}} \begin{bmatrix}\mathbf{h}_{12}\\ \mathbf{w}_1\\ \mathbf{w}_2\end{bmatrix} + \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\end{bmatrix}. \quad (29)$$

Using the compound form of (29) in (10), we compute the log-likelihood of the model H_0 by

$$\begin{aligned} \log p(\phi_1, \phi_2|H_0) &= \log N(\tilde{\phi}\,|\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\Sigma}) \\ &= -\frac{1}{2}(\tilde{\phi} - \tilde{\boldsymbol{\mu}})^{\mathrm{T}}(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\Sigma})^{-1}(\tilde{\phi} - \tilde{\boldsymbol{\mu}}) \\ &\quad -\frac{1}{2}\log|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\Sigma}| - \alpha\log(2\pi), \end{aligned} \quad (30)$$

where $\alpha = C \cdot F$ for the case of i-supervector while $\alpha = D$ for the case of i-vector. To evaluate the log-likelihood function, we have to solve for the inversion and log-determinant of the following covariance matrix:

$$(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\boldsymbol{\Sigma}}) = \begin{bmatrix}\mathbf{F}\\ \mathbf{F}\end{bmatrix}\begin{bmatrix}\mathbf{F}^{\mathrm{T}} & \mathbf{F}^{\mathrm{T}}\end{bmatrix} + \begin{bmatrix}\mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma} & 0\\ 0 & \mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma}\end{bmatrix}. \quad (31)$$

The inversion of the above matrix could be obtained by applying twice the matrix inversion lemma. In particular, we first compute $\mathbf{J} = (\mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma})^{-1}$, the result of which is given by (21), and apply again the matrix inversion lemma on the right-hand side of (31), which leads to

$$(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\boldsymbol{\Sigma}})^{-1} = \begin{bmatrix}\mathbf{J} & 0\\ 0 & \mathbf{J}\end{bmatrix} - \begin{bmatrix}\mathbf{J}\mathbf{F}\\ \mathbf{J}\mathbf{F}\end{bmatrix} \times \mathbf{M}_2 \times \begin{bmatrix}\mathbf{F}^{\mathrm{T}}\mathbf{J}, & \mathbf{F}^{\mathrm{T}}\mathbf{J}\end{bmatrix}, \quad (32)$$

where $\mathbf{M}_2$ is computed using the solution in (24) by setting $J = 2$. Now, to solve for the log-determinant of the same matrix in (31), we apply twice the matrix determinant lemma in a much similar way as the matrix inversion. Taking the log of the result leads to

$$\log|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}} + \tilde{\boldsymbol{\Sigma}}| = -2\log|\mathbf{J}| - \log|\mathbf{M}_2|. \quad (33)$$

Using (32) and (33) in (30), we arrive at

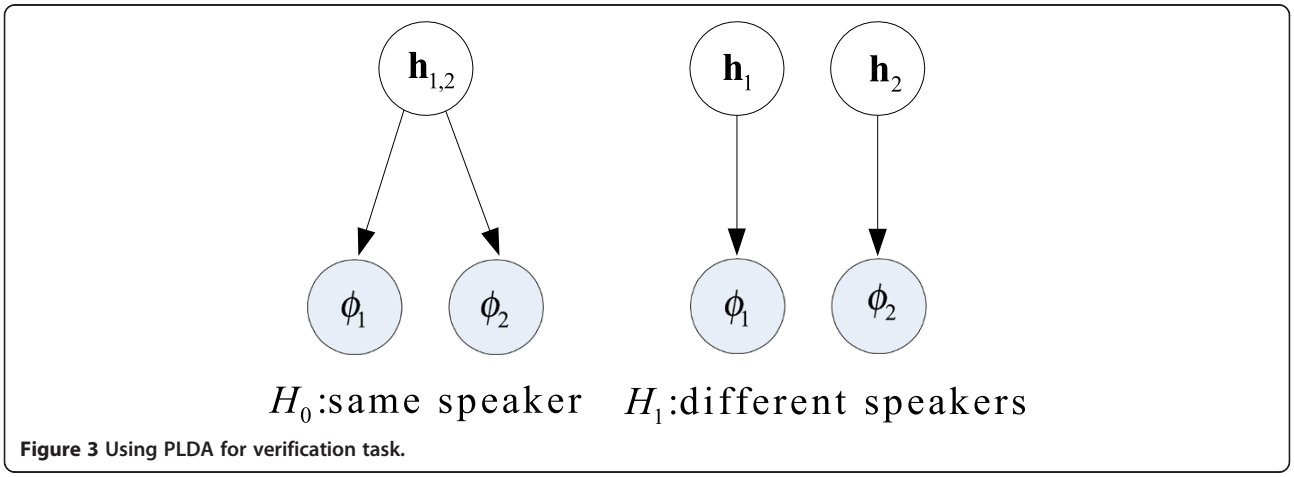

Figure 3 Using PLDA for verification task.

$$\log p(\phi_1, \phi_2|H_0) = \frac{1}{2}\left[\sum_{l=1}^{2}\mathbf{F}^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu})\right]^{\mathrm{T}}\mathbf{M}_2\left[\sum_{l=1}^{2}\mathbf{F}^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu})\right] - \frac{1}{2}\sum_{l=1}^{2}(\phi_l-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu}) + \frac{1}{2}\log|\mathbf{M}_2| + \log|\mathbf{J}| - \alpha\log(2\pi). \tag{34}$$

For the alternative hypothesis H_1, we form the following compound equation:

$$\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu}\\ \boldsymbol{\mu}\end{bmatrix} + \begin{bmatrix}\mathbf{F} & \mathbf{G} & 0 & 0\\ 0 & 0 & \mathbf{F} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{h}_1\\ \mathbf{w}_1\\ \mathbf{h}_2\\ \mathbf{w}_2\end{bmatrix} + \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\end{bmatrix}. \tag{35}$$

The first thing to note is that the first and second rows of the system are decoupled and therefore could be treated separately. The log-likelihood of the alternative hypothesis H_1 is therefore given by the following sum of log-likelihoods:

$$log p(\phi_1,\ \phi_2\big|H_1) = \sum_{l=1}^{2} log\mathcal{N}(\phi_l|\boldsymbol{\mu}, \mathbf{F}\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{G}^{\mathrm{T}} + \boldsymbol{\Sigma}). \tag{36}$$

Using a similar approach as for the case of the null hypothesis, it can be shown that the solution to (36) is given by

$$\log p(\phi_1, \phi_2|H_1) = \frac{1}{2}\sum_{l=1}^{2}\left[\mathbf{F}^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu})\right]^{\mathrm{T}}\mathbf{M}_1\left[\mathbf{F}^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu})\right] - \frac{1}{2}\sum_{l=1}^{2}(\phi_l-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu}) + \log|\mathbf{M}_1| + \log|\mathbf{J}| - \alpha\log(2\pi) \tag{37}$$

Using (34) and (37) in (28), canceling out common terms, we arrive at the following log-likelihood ratio score for the verification task:

$$s(\phi_1, \phi_2) = \frac{1}{2}\left[\sum_{l=1}^{2}\varphi_l^{\mathrm{T}}\right]\mathbf{M}_2\left[\sum_{l=1}^{2}\varphi_l\right] - \frac{1}{2}\sum_{l=1}^{2}\varphi_l^{\mathrm{T}}\mathbf{M}_1\varphi_l + K. \tag{38}$$

For brevity of notations, we have let

$$\varphi_l = \mathbf{F}^{\mathrm{T}}\mathbf{J}(\phi_l-\boldsymbol{\mu}). \tag{39}$$

One way to look at (39) is that it centralizes the vector ϕ_l and projects it onto the subspace $\mathbf{F}$ where speaker information co-vary the most (i.e., dimension reduction) while de-emphasizing the subspace pertaining to channel variability. In (38), $K = \log|\mathbf{M}_2|/2 - \log|\mathbf{M}_1|$ is constant for the given set of parameters $\{\mathbf{F},\ \mathbf{G},\ \boldsymbol{\Sigma}\}$. Though K diminishes when score normalization is applied, we could calculate the two log-determinant terms easily by using the property of eigenvalue decomposition. In particular, we compute $\log|\mathbf{M}_2|$ as $-\sum_{n=1}^{N_F} log(2\lambda_n + 1)$ and $\log|\mathbf{M}_1|$ as $-\sum_{n=1}^{N_F} log(\lambda_n + 1)$, where $\{\lambda_n: n = 1, 2,..., N_F\}$ are the eigenvalues of the matrix $\mathbf{F}^{\mathrm{T}}\mathbf{J}\mathbf{F}$ (*c.f.* (24)).

4.3 I-supervector pre-conditioning

Another prerequisite for good performance with PLDA is that the i-supervectors have to follow a normal distribution, as in (10). It has been shown in [33], for the case of i-vector, that whitening followed by length normalization helps toward this goal. However, whitening can never be possible for i-supervector due to data scarcity. To this end, we advocate the use of a Gaussianized version of rank norm [34,41]. The i-supervector is processed element-wise with warping functions mapping each dimension to a standard Gaussian distribution (instead of uniform distribution as in rank norm). To put it mathematically, let $\phi_l(m)$, for $m = 1, 2,..., CF$, denote the elements of the i-supervector ϕ_l. We first get the normalized rank of $\phi_l(m)$ with respect to a background set B_m as follows:

$$r_m = \frac{|\{b\in B_m : b < \phi_l(m)\}|}{|B_m|}, \tag{40}$$

where $|\cdot|$ denotes the cardinality of a set. The Gaussianized value is then obtained by using the inverse cumulative density function (CDF) of a standard Gaussian distribution (i.e., the probit function) as follows:

$$\phi_l(m) \leftarrow \sqrt{2}\,\mathrm{erf}^{-1}(2r_m - 1), \tag{41}$$

where $\mathrm{erf}^{-1}(\cdot)$ denotes the inverse error function. This can then be followed by length normalization prior to PLDA modeling.

5 Experiment

5.1 Experimental setup

Experiments were carried out on the core task (short2-short3) of NIST SRE08 [42]. We use two well-known metrics in evaluating the performance, namely, *equal error rate* (EER) and *minimum detection cost* (MinDCF). Two gender-dependent UBMs consisting of 512 Gaussians were trained using data drawn from the SRE04. Speech parameters were represented by a 54-dimensional vector of *mel frequency cepstral coefficients* (MFCC) with first and second derivatives appended.

The loading matrices $\mathbf{T}$ in (1) and $\mathbf{D}$ in (5) were both trained with similar sets of data drawn from Switchboard, SRE04, and SRE05. We use 500 factors for $\mathbf{T}$, while $\mathbf{D}$ was a diagonal matrix by definition. The dimensionality of i-vector was therefore 500, while i-supervector is of $CF = 27{,}648$ in dimensionality. The rank of the matrices $\mathbf{F}$ and $\mathbf{G}$ in the PLDA model was set to 300 and 200,

respectively, for the case of i-supervector. For i-vector, best result was found with the rank of **F** set to 300 and using a full matrix for **Σ**, for which **G** was no longer required. This observation was consistent with that reported in [32]. Table 1 summarizes all the corpora used to train the UBM, loading matrices **T** and **D**, PLDA model, whitening transformation matrix, Gaussianized rank-norm, and the cohort data for s-norm [32].

5.2 Feature and score normalization

Experiments were performed on the so-called det1 (int-int), det4 (int-tel), det5 (tel-mic), and det6 (tel-tel) common conditions as defined in NIST SRE08 short2-short3 core task. The term int refers to interview style recorder over microphone channel. For the det1 common condition, the training and test utterances were both int style of speech. Similar definition applied for other common conditions. The first set of experiments aimed at verifying the effectiveness of PLDA model in the i-supervector space without normalization (raw). Table 2 shows the results. It is evident that the i-supervector system performed much better than i-vector in all the four common conditions for both male and female trials. For the particular case of female trials, the EER for the i-supervector system was lower by 10.27%, 15.46%, 28.42%, and 16.58% in det1, det4, det5, and det6 compared to that of the i-vector system. One possible reason may be that the Gaussian assumption in (10) can be better fulfilled in the i-supervector space with higher dimensionality compared to that of the i-vector.

The second set of experiments aimed at investigating the effectiveness of different normalization methods on i-supervector prior to PLDA modeling (i.e., length normalization, whitening, and Gaussianized rank-norm) and also the effects of score normalization (we used s-norm as reported in [32]). For simplicity, we used only telephone data and report the results on det6 (i.e., tel-tel common condition) in Table 3. We observed similar performance for other common conditions. From Table 3, it is clear that length normalization (len) always outperforms raw for both i-vector and i-supervector. Notice that i-vector gains huge improvement from length normalization. For the MALE subset, we observed 20.0% and 6.5% of relative improvement in EER when length normalization was applied on i-vector and i-supervector, respectively. Whitening followed by length normalization (white + len) further improves the performance for i-vector. Similarly, in the case of i-supervector, we used Gaussianized rank-norm followed by length normalization (grank + len) to cope with the high dimensionality. Finally, we also noticed that s-norm gives consistent improvement for both i-vector and i-supervector.

Table 1 Corpora used for training various components of the system

	Switchboard	NIST SRE04	NIST SRE05		NIST SRE06	
	Tel	Tel	Tel	Mic	Tel	Mic
UBM		X				
T	X	X	X			
D	X	X	X			
PLDA model	X	X	X		X	
Whitening	X	X	X	X	X	X
G-rank-norm	X	X	X	X	X	X
s-norm			X	X	X	X

The terms 'Tel' and 'Mic' refer to telephone (landline or mobile) and microphone channel recordings.

Table 2 Performance comparison of i-vector and i-supervector on NIST SRE08 core task with no normalization applied

	Male		Female	
	EER	MinDCF	EER	MinDCF
i-vector				
Det1 (raw)	9.6696	4.3332	13.9834	5.4534
Det4 (raw)	5.9883	2.7667	14.5646	5.8298
Det5 (raw)	5.9601	2.4829	11.4183	3.9617
Det6 (raw)	6.1785	3.1206	8.1486	3.7028
i-supervector				
Det1 (raw)	8.7329	3.9681	12.5471	5.3874
Det4 (raw)	4.7950	2.4082	12.3123	5.7750
Det5 (raw)	5.6250	2.3139	8.1731	3.8216
Det6 (raw)	5.2632	2.6605	6.7976	3.3368

5.3 Channel factors in i-supervector space

The low-rank matrices **G** model the subspace corresponding to channel variability as described in Section 3.1. We evaluated the performance of the i-supervector system at different numbers of channel factors, N_G. Table 4 shows the results for the det6 common condition. We can see that when $N_G = 0$, which corresponds to a fully diagonal PLDA model, the EER and MinDCF for both of male and female were very poor. A slight increment in the number

Table 3 Performance comparison of normalization methods on i-vector and i-supervector

	Male		Female	
	EER	MinDCF	EER	MinDCF
i-vector				
raw	6.1785	3.1206	8.1486	3.7028
len	4.9411	2.6286	6.4409	3.0581
white + len	4.5458	2.4546	6.3193	3.0065
white + len + snorm	4.3478	2.2155	6.1530	3.0034
i-supervector				
raw	5.2632	2.6605	6.7976	3.3368
len	4.9199	2.6271	6.3667	3.3624
grank + len	4.8982	2.6676	6.0976	3.2588
grank + len + snorm	4.5888	2.3737	6.2639	3.1132

Table 4 Performance of i-supervector PLDA system with different numbers of channel factors, N_G

Number of channel factors, N_G	Male		Female	
	EER	MinDCF	EER	MinDCF
0	15.9148	5.6324	19.1796	6.8120
10	8.3524	3.9297	10.2550	4.5087
20	6.1785	3.2124	8.9246	3.9208
30	5.6114	2.8483	7.6497	3.5515
40	5.1487	2.7037	7.6497	3.4161
50	4.8552	2.5573	6.9290	3.3179
100	4.6911	2.4875	6.5196	3.2814
150	4.5974	2.3983	6.4856	3.1239
200	4.5888	2.3737	6.2639	3.1132
250	5.0099	2.5092	6.4302	3.1420
300	4.7693	2.5843	6.2084	3.1114

of channel factors N_G to 10 reduces the EER by 47% and 46% for male and female sets, respectively. Further increment in N_G reduces the EER gradually until it levels off at N_G = 200 after which no further improvement could be attained. We set N_G = 200 for the i-supervector PLDA system in subsequent experiments.

5.3.1 Performance comparison

In this section, we compared the performance of i-supervector and i-vector under different train-test channel conditions. The PLDA models used for i-vector and i-supervector were the same as described in Section 5.2. In addition, we included microphone data (drawn from SRE05 and SRE06) for the whitening transform, Gaussianized rank-norm, and s-norm to better handle the interview (int) and microphone (mic) channel conditions.

Table 5 shows the results when full normalization (i.e., white + len + snorm for i-vector, grank + len + snorm for i-supervector) was applied. Here, we consider the EER and MinDCF by pooling together the male and female scores. The DET curves under the four common conditions are plotted in Figure 4. Similar to the observation in Section 5.1, i-vector gives better performance than i-supervector for the case with full normalization except in det5 where the i-supervector gives a much lower EER though the MinDCF is slightly worse. This again shows that current normalization strategy (Gaussianized rank-norm followed by length normalization), though effective, has to be further improved. Also shown in Table 5 and Figure 4 are the results by fusing the i-vector and i-supervector systems. The fusion of the two systems gives competitive performance with slightly lower EER and MinDCF across all four common conditions. The two systems were fused at the score level as follows:

$$s = \beta \times s_1 + (1-\beta) \times s_2 \tag{42}$$

where s_1 and s_2 are i-vector and i-supervector scores, respectively. The fusion weight β is set to 0.5, 0.5, 0.3, and 0.4, respectively, for det1, det4, det5, and det6.

Table 5 Performance comparison under various train-test channel conditions of NIST SRE08 short2-short3 core task

	Conditions			
	det1 (int-int)		det4 (int-tel)	
	EER	MinDCF	EER	MinDCF
i-vector	7.2964	3.5189	5.7919	2.7576
i-supervector	7.8769	3.6724	5.9421	3.0541
Fusion	7.0711	3.4100	5.2489	2.5892
	det5 (tel-mic)		det6 (tel-tel)	
	EER	MinDCF	EER	MinDCF
i-vector	6.0462	2.0975	5.5602	2.7556
i-supervector	4.7554	2.2475	5.7740	2.8949
Fusion	4.4158	2.0260	5.3398	2.7537

5.4 Computation complexity comparison

The experiments were carried out using the following hardware configuration: Centos 6.4 system, Intel Xeon processor E5-2687w (8-core, 3.1 GHz/core) with 128 GB memory. We compared the total time and the real-time factor of i-supervector and i-vector systems at four different stages, namely, loading matrix estimation, posterior extraction, PLDA modeling, and PLDA scoring. The total variability matrix **T** in (1) and **D** in (5) were both trained using a similar set of data drawn from Switchboard, SRE04, and SRE05. Table 6 lists the time of training **T** and **D** with ten EM iterations. We can see that it takes about 16.75 h for training **T** using 348 h of speech, which implies a real-time factor of 0.048. On the contrary, it took only 380 s for training **D**. Because **D** is a diagonal matrix, simple vector multiplication can be used instead of large matrix multiplication.

After training the total variability space, we extracted i-vector and i-supervector for all utterances. Table 6 shows the time required for extracting the i-vectors and i-supervectors from the entire SRE04 dataset. The result shows that i-vector extraction consumes much more time than i-supervector. PLDA models were then trained for i-vectors and i-supervectors drawn from Switchboard, SRE04, and SRE05. We can see that training a PLDA model on the i-vector takes much lesser time than for the i-supervector. Finally, we compared the computation requirement for PLDA scoring on the NIST SRE08 short2-short3 core task with 98,776 trials. It can be seen that i-supervector scoring took more time than i-vector mainly due to its comparatively high dimensionality. In summary, the i-supervector system requires less computation at the front-end while the i-vector system is faster at the back-end PLDA.

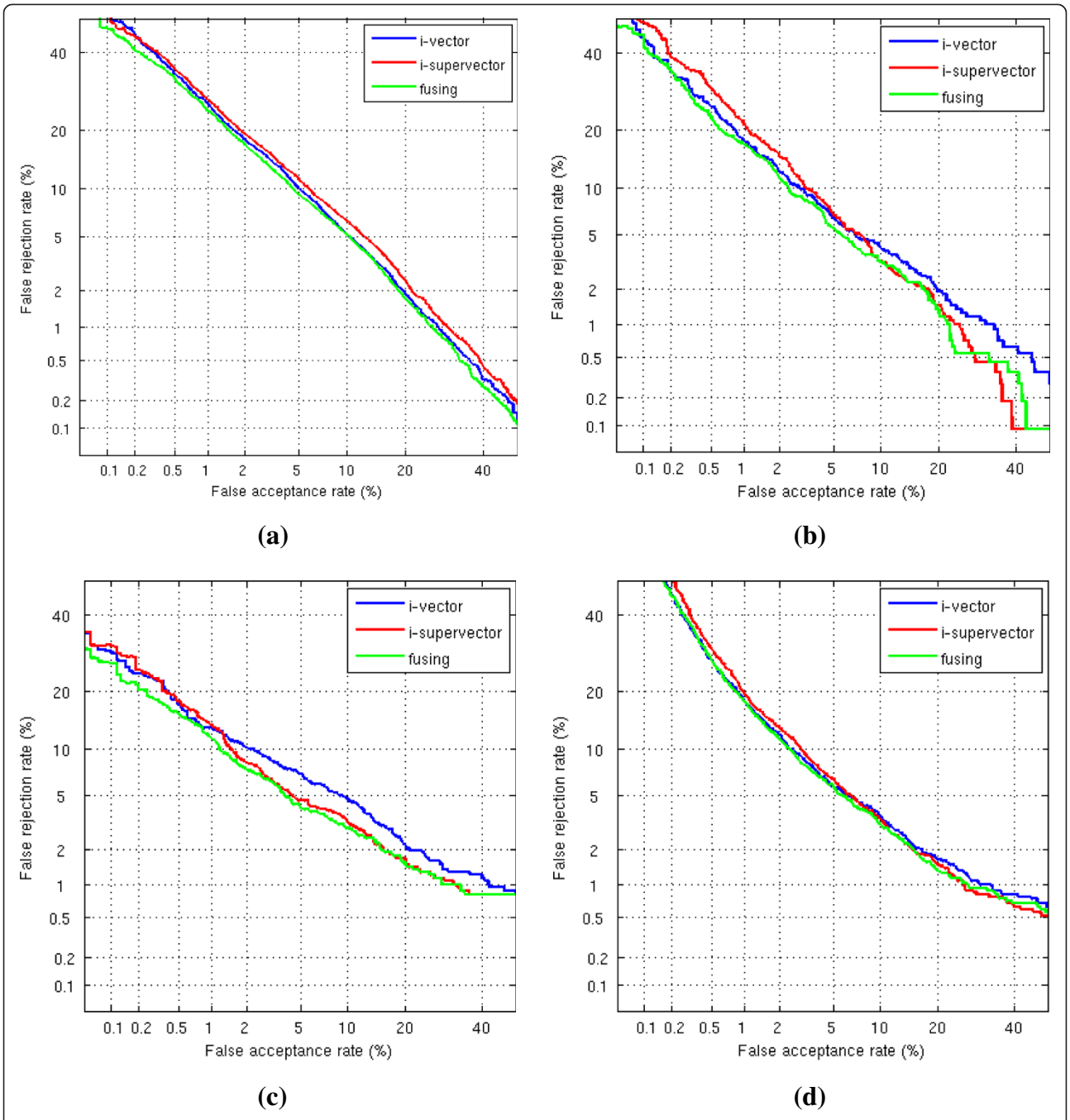

Figure 4 The DET curves of the i-vector and i-supervector evaluated on NIST SRE08 short2-short3 core task. (a) Common condition det1, **(b)** common condition det4, **(c)** common condition det5, and **(d)** common condition det6.

Table 6 Comparison of computational complexity (total time/real time factor) at various stages of implementation

	Loading matrix	Posterior extraction	PLDA modeling	PLDA scoring
i-vector	60,300 s/4.8e−2	470 s/2.83e−3	47 s/3.74e−7	142 s/4.30e−4
i-supervector	380 s/3.03e−6	24 s/1.45e−4	950 s/7.57e−6	425 s/1.29e−3

6 Conclusions

We have introduced the use of the uncompressed form of *i-vector* (i.e., the *i-supervector*) for PLDA-based speaker verification. Similar to i-vector, an i-supervector represents a variable-length speech utterance as a fixed-length vector. But different from i-vector, we keep the total variability space having the same dimensionality as the original supervector space. To this end, we showed how manipulation of high-dimensional matrices can be done efficiently in training and scoring with the PLDA model. We also introduced the use of Gaussianized rank-norm for feature normalization prior to PLDA modeling.

Compared to i-vector, we found that i-supervector performs better when no normalization (on both feature and score) was applied. This suggests that the Gaussian assumption imposed by PLDA becomes less stringent and easier to fulfill in the higher dimensional i-supervector space. However, the performance improvement given by the high dimensionality diminishes when full normalization is applied. As such, current normalization strategy, though effective, has to be improved for better performance. This is a point for future work. We also showed that fusion system can give competitive performance compared to either i-vector or i-supervector. Furthermore, we analyzed the computational complexity of the i-supervector system, compared to that of the i-vector, at four different stages, namely, loading matrix estimation, posterior extraction, PLDA modeling, and PLDA scoring. Actually, the results showed that the i-supervector system took much less time than the i-vector system in terms of loading matrix and posterior extraction.

Endnote

[a]The number of sessions is usually limited in face recognition for which PLDA was originally proposed in [10].

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was partially supported by a research grant from the Tateisi Science and Technology Foundation.

Author details
[1]Top Runner Incubation Center for Academia-Industry Fusion, Nagaoka University of Technology, Nagaoka 940-2188, Japan. [2]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore 138632, Singapore.

Received: 26 November 2013 Accepted: 21 June 2014
Published: 15 July 2014

References

1. G Doddington, Speaker recognition based on idiolectal differences between speakers, in *Proc. 7th European Conference on Speech Communication and Technology (Eurospeech)* (Scandinavia, 2001), pp. 2521–2524
2. CE Wilson, S Manocha, S Vishnubhotla, A new set of features for text-independent speaker identification, in *Proc. Interspeech* (Pittsburgh, PA, USA, 2006), pp. 1475–1478
3. T Kinnunen, KA Lee, H Li, Dimension reduction of the modulation spectrogram for speaker verification, in *The Speaker and Language Recognition Workshop* (Stellenbosch, South Africa, 2008)
4. T Kinnunen, HZ Li, An overview of text-independent speaker recognition: from features to supervectors. Speech Comm. **52**(1), 12–40 (2010)
5. L Wang, K Minami, K Yamamoto, S Nakagawa, Speaker identification by combining MFCC and phase information in noisy environments, in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Dallas, TX, USA, 2010), pp. 4502–4505
6. S Nakagawa, L Wang, S Ohtsuka, Speaker identification and verification by combining MFCC and phase information. IEEE Trans. Audio Speech Lang. Process. **20**(4), 1085–1095 (2012)
7. DA Reynolds, TF Quatieri, RB Dunn, Speaker verification using adapted Gaussian mixture models. Digital Signal Process. **10**(1), 19–41 (2000)
8. WM Campbell, DE Sturim, DA Reynolds, A Solomonoff, SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Philadelphia, USA, 2005), pp. 97–100
9. WM Campbell, DE Sturim, DA Reynolds, Support vector machines using GMM supervectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006)
10. SJD Prince, JH Elder, Probabilistic linear discriminant analysis for inferences about identity, in *Proc. International Conference on Computer Vision* (Rio De Janeiro, Brazil, 2007), pp. 1–8
11. L Wang, N Kitaoka, S Nakagawa, Robust distant speaker recognition based on position-dependent CMN by combining speaker-specific GMM with speaker-adapted HMM. Speech Comm. **9**(6), 501–513 (2007)
12. P Kenny, P Ouellet, N Dehak, V Gupta, P Dumouchel, A study of inter-speaker variability in speaker verification. IEEE Trans. Audio. Speech Lang. Process. **16**(5), 980–988 (2008)
13. N Dehak, P Kenny, R Dehak, P Dumouchel, P Ouellet, Front-end factor analysis for speaker verification. IEEE Trans. Audio. Speech Lang. Process. **19**(4), 788–798 (2011)
14. JMK Kua, J Epps, E Ambikairajah, i-Vector with sparse representation classification for speaker verification. Speech Comm. **55**(5), 707–720 (2013)
15. F Kelly, A Drygajlo, N Harte, Speaker verification in score-ageing-quality classification space. Comput. Speech Lang. **27**(5), 1068–1084 (2013)
16. V Wan, WM Campbell, Support vector machines for speaker verification and identification. IEEE Workshop Neural Netw. Signal Process. **2**, 77–784 (2000)
17. WM Campbell, JP Campbell, DA Reynolds, Support vector machines for speaker and language recognition. Comp. Speech Lang. **20**, 210–229 (2006)
18. C Bishop, *Pattern Recognition and Machine Learning* (Springer Science & Business Media, New York, 2006)
19. KA Lee, C You, H Li, T Kinnunen, A GMM-based probabilistic sequence kernel for speaker recognition, in *Proc. Interspeech* (Antwerp, Belgium, 2007), pp. 294–297
20. CH You, KA Lee, H Li, GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1300–1312 (2010)
21. X Dong, W Zhaohui, Speaker recognition using continuous density support vector machines. Electron. Lett. **37**(17), 1099–1101 (2001)
22. V Wan, S Renals, Speaker verification using sequence discriminant support vector machines. IEEE Trans. Speech Audio Process. **13**(2), 203–210 (2005)
23. N Dehak, G Chollet, Support vector GMMs for speaker verification, in *Proc. IEEE Odyssey: The Speaker and Language Recognition Workshop* (San Juan, Puerto Rico, 2006)
24. P Kenny, G Boulianne, P Ouellet, P Dumouchel, Speaker and session variability in GMM-Based speaker verification. IEEE Trans. Audio Speech Lang. Process. **15**(4), 1448–1460 (2007)
25. N Dehak, Discriminative and generative approaches for long- and short-term speaker characteristics modeling: application to speaker verification, in Ph.D. thesis (École de Technologie Supérieure, Université du Québec, 2009)
26. A Kanagasundaram, D Dean, R Vogt, M McLaren, S Sridharan, M Mason, Weighted LDA techniques for i-vector based speaker verification, in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (Kyoto, Japan, 2012), pp. 4781–4794
27. A Kanagasundaram, D Dean, S Sridharan, M McLaren, R Vogt, I-vector based speaker recognition using advanced channel compensation techniques. Comput. Speech Lang. **28**(1), 121–140 (2014)

28. BGB Fauve, D Matrouf, N Scheffer, J-F Bonastre, JSD Mason, State-of-the-art performance in text-independent speaker verification through open-source software, in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Honolulu, USA, 2007), pp. 1960–1968
29. M Senoussaoui, P Kenny, N Dehak, P Dumouchel, An i-vector extractor suitable for speaker recognition with both microphone and telephone speech, in *Proc.Odyssey: The Speaker and Language Recognition Workshop* (Brno, Czech, 2010)
30. A Kanagasundaram, R Vogt, D Dean, S Sridharan, M Mason, I-vector based speaker recognition on short utterances, in *Proc. Interspeech* (Florence, 2011), pp. 2341–2344
31. L Machlica, Z Zajic, An efficient implementation of probabilistic linear discriminant analysis, in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vancouver, Canada, 2013), pp. 7678–7682
32. P Kenny, Bayesian speaker verification with heavy-tailed priors, in *Proc. Odyssey: Speaker and Language Recognition Workshop* (Brno, Czech, 2010)
33. D Garcia-Romero, CY Espy-Wilson, Analysis of i-vector length normalization in speaker recognition systems, in *Proc. Interspeech* (Florence, Italy, 2011), pp. 249–252
34. Y Jiang, KA Lee, Z Tang, B Ma, A Larcher, H Li, PLDA modeling in i-vector and supervector space for speaker verification, in *Proc. Interspeech* (Portland, USA, 2012)
35. P Kenny, G Boulianne, P Dumouchel, Eigenvoice modeling with sparse training data. IEEE Trans. Speech Audio Process. **13**(3), 345–354 (2005)
36. J Gauvain, C-H Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chain. IEEE Trans. Speech Audio Process. **2**(2), 291–298 (1994)
37. P Kenny, T Stafylakis, P Ouellet, MJ Alam, P Dumouchel, PLDA for speaker verification with utterance of arbitrary duration, in *Proc. IEEE ICASSP* (Vancouver, Canada, 2013), pp. 7649–7653
38. H Li, B Ma, KA Lee, CH You, H Sun, A Larcher, IIR system description for the NIST 2012 speaker recognition evaluation, in *NIST SRE'12 Workshop* (Orlando, 2012)
39. R Saeidi, KA Lee, T Kinnunen, T Hasan, B Fauve, P-M Bousque, E Khoury, PL Sordo Martinez, JMK Kua, CH You, H Sun, A Larcher, P Rajan, V Hautamaki, C Hanilci, B Braithwaite, R Gonzales-Hautamki, SO Sadjadi, G Liu, H Boril, N Shokouhi, D Matrouf, L El Shafey, P Mowlaee, J Epps, T Thiruvaran, DA van Leeuwen, B Ma, H Li, JHL Hansen et al., I4U submission to NIST SRE2012: a large-scale collaborative effort for noise-robust speaker verification, in *Proc. Interspeech* (Lyon, France, 2013), pp. 1986–1990
40. KP Murphy, *Machine Learning-A Probabilistic Perspective* (MIT Press, Massachusetts, 2012), pp. 116–117
41. A Stolcke, S Kajarekar, L Ferrer, Nonparametric feature normalization for SVM-based speaker verification, in *Proc. ICASSP* (Ohio, USA, 2008), pp. 1577–1580
42. NIST, The NIST year 2008 speaker recognition evaluation plan, http://www.itl.nist.gov/iad/mig/tests/sre/2008/

doi:10.1186/s13636-014-0029-2
Cite this article as: Jiang *et al.*: PLDA in the i-supervector space for text-independent speaker verification. *EURASIP Journal on Audio, Speech, and Music Processing* 2014 **2014**:29.