Odyssey 2016
21-24 Jun 2016, Bilbao, Spain

Rapid Computation of I-vector

Longting Xu[1,2], *Kong Aik Lee*[2], *Haizhou Li*[2], *Zhen Yang*[1]

[1]Broadband Wireless Communication and Sensor Network Technology Key Lab.
Nanjing University of Posts and Telecommunications, China
[2]Institute for Infocomm Research
Agency for Science, Technology and Research (A*STAR), Singapore
1011010417@njupt.edu.cn, kalee@i2r.a-star.edu.sg

Abstract

I-vector has been one of the state-of-the-art techniques in speaker recognition. The main computational load of the standard i-vector extraction is to evaluate the posterior covariance matrix, which is required in estimating the i-vector. This limits the potential use of i-vector on handheld devices and for large-scale cloud-based applications. Previous fast approaches focus on simplifying the posterior covariance computation. In this paper, we propose a method for rapid computation of i-vector which bypasses the need to evaluate a full posterior covariance thereby speeds up the extraction process with minor impact on the recognition accuracy. This is achieved by the use of subspace-orthonormalizing prior and the uniform-occupancy assumption that we introduce in this paper. From the experiments conducted on the extended core task of NIST SRE'10, we obtained significant speed-up with modest degradation in performance over the standard i-vector.

1. Introduction

Aside from the session variability, one major difficulty in speaker recognition is to deal with the continuous variable-length nature of speech utterances. It was first shown in [1], the variable-length nature of speech utterance could be accounted for with Gaussian mixture model (GMM). Later in [2, 3], it was shown that the speaker, channel, and other forms of session variability could be substantially captured with low-rank subspaces in the GMM parameter space. In [4], the authors took one step further by modeling the various subspaces with a single total variability matrix. The end result is a compression process that maps variable-length speech utterances into fixed-length low dimensional vectors referred to as the i-vectors. Due to its low dimensionality, simple and yet effective techniques, such as linear discriminant analysis (LDA) [5] and probabilistic LDA (PLDA) [6, 7, 8] could be conveniently applied for channel compensation and as the backend classifier.

An i-vector is the posterior mean of a latent variable in the total variability space [2, 4]. The computation of i-vector involves the estimation and inversion of the posterior precision matrix, which has shown to be the major bottleneck for implementation on handheld devices [9] or cloud-based systems that process a large amount of online requests [10]. A smaller computational and memory footprint would definitely benefit these applications. Many fast methods have been proposed to extract i-vectors without seriously compromising the recognition accuracy. In [11], the authors proposed to diagonalize the posterior precision matrix using eigenvalue decomposition or heteroscedastic LDA (HLDA) and thereby simplifying the matrix inversion. The approach in [12] employs a fixed occupancy counts thereby approximating the posterior covariance with the same estimate for all utterances. In [13] matrix factorization and iterative conjugate gradient method were used to speed up the matrix inversion. In our previous work [14], methods based on sparse coding were proposed. All the above approaches try to solve the problem by simplifying the estimation of posterior covariance. In this paper, we attempt to solve the problem by bypassing the need to evaluate the full posterior covariance as part of i-vector computation. This is achieved by asserting an informative prior as opposed to a standard Gaussian prior assumed in the standard i-vector extractor.

The standard i-vector extractor assumes a standard Gaussian prior with zero mean and unit variance on the i-vector latent variable [4]. In general factor analysis framework, informative prior is seldom used as the prior mean and covariance could be absorbed into the global mean and loading matrix [5]. In this paper, we set the prior covariance to take a specific form such that when it is absorbed as part of the total variability matrix it will set the columns of the matrix to become orthogonal and unit norm (i.e., orthonormal). We refer to this prior as the *subspace-orthonormalizing prior* and we show that such prior could be formed easily using the total variability matrix. The orthonormal property imposed on the total variability space sets the stage for subsequent computational speed-up and approximation that leads to the fast method for rapid computation of i-vector proposed in this paper. It is worth mentioning that the use of informative prior for i-vector extraction was reported earlier in [15, 16]. In particular, source-specific priors were shown to be effective in dealing with source variation encountered in heterogeneous datasets. Notice that we use informative prior for a different purpose in this work.

The paper is organized as follows. Section 2 reviews the i-vector paradigm. Section 3 formulates the subspace-orthonormalizing prior and its application for i-vector extraction. Section 4 presents a detailed derivation of the proposed fast method for i-vector extraction. Section 5 shows the use of subspace-orthonormalizing prior in the EM training of the total variability matrix. This is followed by experiment results in Section 6. Section 7 concludes the paper.

2. I-vector extraction

An i-vector extractor aims to find a compressed representation of speech utterance in the parameter space of the GMM supervector [2] with the use of factor analysis [4, 5]. Dimension reduction is achieved by controlling the size of the latent variable

doi: 10.21437/Odyssey.2016-7

$\mathbf{x}$ in the model:

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

where $\mathbf{m}$ and $\mathcal{M}$ are the GMM and UBM supervectors, respectively. The low-rank matrix $\mathbf{T}$ is the so-called total variability matrix since its column space captures the speaker, channel, phonetic and other sources of variability.

Let $\mathcal{O} = \{o_1, o_2, \ldots, o_T\}$ represents the feature vector sequence of a given utterance. Given $\mathcal{O}$ and the current estimate $\mathbf{T}$, an i-vector is given by the maximum *a posterior* (MAP) estimate, as follows:

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

where $\mathcal{M}_c$ and $\mathbf{T}_c$ indicate the c-th component in the supervector $\mathcal{M}$ and matrix $\mathbf{T}$. Let C be the number of Gaussian components in the UBM and F be the dimension of the acoustic feature vector. The $CF \times M$ matrix $\mathbf{T} = \left[\mathbf{T}_1^{\mathsf{T}}, \mathbf{T}_2^{\mathsf{T}}, \cdots, \mathbf{T}_C^{\mathsf{T}}\right]^{\mathsf{T}}$ comprises of C component matrices $\mathbf{T}_c$ from all the mixtures stacked up column wise.

Assuming a standard Gaussian prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I})$, the i-vector in (2) is given by the posterior mean of $\mathbf{x}$, as follow

$$\phi = \mathbf{L}^{-1} \cdot \mathbf{T}^{\mathsf{T}}\mathbf{\Sigma}^{-1}\mathbf{F} \tag{3}$$

where

$$\mathbf{L}^{-1} = \left(\mathbf{I} + \mathbf{T}^{\mathsf{T}}\mathbf{\Sigma}^{-1}\mathbf{N}\mathbf{T}\right)^{-1} \tag{4}$$

Here, $\mathbf{N}$ and $\mathbf{F}$ are the utterance-dependent Baum Welch statistics computed based on the UBM. The matrix $\mathbf{\Sigma}$ is constructed by having its diagonal blocks made up by the covariance matrices $\mathbf{\Sigma}_c$ of the UBM[1]. The $CF \times 1$ vector $\mathbf{F}$ is obtained by concatenating the first-order statistics

$$\mathbf{F}_c = \sum\nolimits_t \gamma_c(t)\left(o_t - \mathcal{M}_c\right) \tag{5}$$

centred to the mean vector $\mathcal{M}_c$ of the UBM. In a similar manner, $\mathbf{N}$ is a $CF \times CF$ diagonal matrix, whose diagonal blocks are $N_c\mathbf{I}$, where N_c is the zero-order statistics computed for the c-th Gaussian by summing the frame occupancy $\gamma_c(t)$ over the entire sequence:

$$N_c = \sum\nolimits_t \gamma_c(t) \tag{6}$$

To speed up the computation, it is customary to pre-whiten the first-order statistics [18], as follow

$$\widetilde{\mathbf{F}}_c = \mathbf{\Sigma}_c^{-1/2}\mathbf{F}_c = \mathbf{\Sigma}_c^{-1/2}\left[\sum\nolimits_t \gamma_c(t)\left(o_t - \mu_c\right)\right] \tag{7}$$

The i-vector is now given by

$$\phi = \mathbf{L}^{-1} \cdot \mathbf{T}^{\mathsf{T}}\widetilde{\mathbf{F}} \tag{8}$$

where

$$\mathbf{L}^{-1} = \left(\mathbf{I} + \mathbf{T}^{\mathsf{T}}\mathbf{N}\mathbf{T}\right)^{-1} \tag{9}$$

We shall use the form in (7)-(9) in the subsequent parts of this paper. To further speed up the computation, the terms $\mathbf{T}_c^{\mathsf{T}}\mathbf{T}_c$ in (9) are usually pre-computed and stored. The memory demand for storing $\mathbf{T}_c^{\mathsf{T}}\mathbf{T}_c$ for all the C mixtures is $O\left(CM^2\right)$ in addition to $O\left(CFM\right)$ required to store the total variability matrix $\mathbf{T}$. The computational complexity is $O\left(CFM + M^2\right)$ for (8) and $O\left(CM^2 + M^3\right)$ for (9).

[1]The UBM could be a GMM trained in an unsupervised manner [1] or a DNN trained to model se es via supervised training [17].

3. I-vector extraction with subspace orthonormalization

We first show the use of informative prior of the form $\mathbf{x} \sim \mathcal{N}\left(\mu_{\mathbf{p}}, \mathbf{\Sigma}_{\mathbf{p}}\right)$, where $\mu_{\mathbf{p}} \neq \mathbf{0}$ and $\mathbf{\Sigma}_{\mathbf{p}} \neq \mathbf{I}$, for i-vector extraction. We then introduce the so-called subspace-orthonomalizing prior and its application for i-vector extraction.

3.1. Posterior inference with informative prior

In its conventional form, a standard Gaussian prior is typically used for i-vector extraction. That is, we assume that the latent variable $\mathbf{x}$ in (1) follows a standard normal distribution, such that $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I})$. Consider a more general case, where the prior $p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mu_{\mathbf{p}}, \mathbf{\Sigma}_{\mathbf{p}}\right)$ has mean $\mu_{\mathbf{p}}$ and covariance $\mathbf{\Sigma}_{\mathbf{p}}$, the i-vector becomes [15]

$$\phi = \mathbf{L}^{-1} \cdot \left(\mathbf{T}^{\mathsf{T}}\widetilde{\mathbf{F}} + \mathbf{\Sigma}_{\mathbf{p}}^{-1}\mu_{\mathbf{p}}\right) \tag{10}$$

and the posterior covariance is now given by

$$\mathbf{L}^{-1} = \left(\mathbf{\Sigma}_{\mathbf{p}}^{-1} + \mathbf{T}^{\mathsf{T}}\mathbf{N}\mathbf{T}\right)^{-1} \tag{11}$$

In [15] and [16], it was shown that the use of informative prior is beneficial to model and compensate for the source variability when a heterogeneous dataset is concerned.

3.2. Subspace-orthonormalizing prior

In this paper, we consider a more specific form of prior with zero mean vector $\mu_{\mathbf{p}} = \mathbf{0}$ and covariance matrix

$$\mathbf{\Sigma}_p = \left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1} \tag{12}$$

made dependent on the loading matrix $\mathbf{T}$. As we shall show next, this has the effect of orthnormalizing the loading matrix in the i-vector extraction process. Using the aforementioned prior in (10), the i-vector is given by

$$\phi = \mathbf{L}^{-1} \cdot \mathbf{T}^{\mathsf{T}}\widetilde{\mathbf{F}} \tag{13}$$

where

$$\mathbf{L}^{-1} = \left(\mathbf{T}^{\mathsf{T}}\mathbf{T} + \mathbf{T}^{\mathsf{T}}\mathbf{N}\mathbf{T}\right)^{-1} \tag{14}$$

Notice that (13) is the same as the conventional form in (8) since the prior has zero mean. The difference happens in the posterior covariance in (14) with the term $\mathbf{T}^{\mathsf{T}}\mathbf{T}$ replacing the identity matrix $\mathbf{I}$ in (9). Substituting (14) into (13), and after some algebraic manipulation, we arrive at

$$\phi = \left[\mathbf{I} + \left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}\mathbf{N}\mathbf{T}\right]^{-1}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}\widetilde{\mathbf{F}} \tag{15}$$

Note that we have assumed the matrix $\mathbf{T}^{\mathsf{T}}\mathbf{T}$ is invertible to arrive at the equation above. We recall the following matrix inversion identity [19]

$$(\mathbf{I} + \mathbf{P}\mathbf{Q})^{-1}\mathbf{P} = \mathbf{P}(\mathbf{I} + \mathbf{Q}\mathbf{P})^{-1} \tag{16}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are two matrices of arbitrary size with the constraint that their product forms a square matrix. We apply the identity on (15) by letting $\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}} = \mathbf{P}$ and $\mathbf{N}\mathbf{T} = \mathbf{Q}$ which leads to

$$\phi = \left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}\left[\mathbf{I} + \mathbf{N} \cdot \mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}\right]^{-1}\widetilde{\mathbf{F}} \tag{17}$$

Taking a closer look at (17), we note that the term $\mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}$ forms an orthogonal projection onto the subspace spanned by $\mathbf{T}$ [19]. The matrix $\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}$ acts as a normalizing factor that recovers the norm and orthonormal basis of the subspace. Since this normalization matrix is introduced through the prior $\mathbf{\Sigma}_p$, we refer to (12) as the subspace-orthonormalizing prior.

4. Computation speed-up

The major computational load of i-vector extraction is to compute the matrix inversion in (9) or (14) for the standard and subspace-orthonormalized models, respectively. The expression in (17) show analytically the effects of subspace-orthnormalization when the prior Σ_p conforms to (12). Using (17) directly for i-vector extraction would incur more computation than that in (8) and (13) as it involves the inversion of a larger $CF \times CF$ matrix. Having say so, (17) serves as the starting point to derive a very fast method to extract i-vector proposed in this paper.

4.1. Rapid computation of i-vector

We start with a singular value decomposition (SVD) on $\mathbf{T}$ of the form $\mathbf{USV}^{\mathsf{T}}$. The columns of $\mathbf{U}$ are referred to as the left-singular vector of $\mathbf{T}$ in SVD terminology [19]. We partition the $CF \times CF$ orthogonal matrix $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2]$ to consist of two rectangular matrices such that $\mathbf{T}$ spans the same subspace as $\mathbf{U}_1$ while orthogonal to $\mathbf{U}_2$. With this, $\mathbf{U}_1$ has the same size as $\mathbf{T}$ while $\mathbf{U}_2$ fills up the remaining of the vector space. Since $\mathbf{UU}^{\mathsf{T}} = \mathbf{I}$ and $\mathbf{U}_1 \perp \mathbf{U}_2$, it follows that

$$\left[\mathbf{I} + \mathbf{N}\cdot\mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}\right]^{-1} = \left(\mathbf{I} + \mathbf{N} - \mathbf{N}\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\right)^{-1} \tag{18}$$

Let $\mathbf{A} = \mathbf{I} + \mathbf{N}$, we expand the right-hand-side of (18) using matrix inversion lemma [19], as follows

$$\mathbf{K} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{N}\left(\mathbf{I} - \mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{N}\right)^{-1}\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\mathbf{A}^{-1}$$

Applying the matrix inversion identity (16), we arrive at

$$\mathbf{K} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{N}\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\left(\mathbf{I} - \mathbf{A}^{-1}\mathbf{N}\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\right)^{-1}\mathbf{A}^{-1} \tag{19}$$

Until now the solution in (19) is exact. The matrix

$$\mathbf{A}^{-1}\mathbf{N} = (\mathbf{I} + \mathbf{N})^{-1}\mathbf{N} \tag{20}$$

is diagonal with element

$$\frac{N_c}{1 + N_c} \simeq \alpha \quad \forall c \tag{21}$$

where $0 \leq \alpha < 1$. We adopt the approximation $\mathbf{A}^{-1}\mathbf{N} = \alpha\mathbf{I}$ such that

$$\mathbf{K} \approx \mathbf{A}^{-1} + \alpha\cdot\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\left(\mathbf{I} - \alpha\cdot\mathbf{U}_2\mathbf{U}_2^{\mathsf{T}}\right)^{-1}\mathbf{A}^{-1} \tag{22}$$

Notice that the assumption in (21) allows us to decompose the matrix inversion in (22) into two components that are orthogonal to each others. We do not need to know the exact value for α, as the second term comprising of $\mathbf{U}_2$ will be discarded.

Using (22) in (17), and recognizing the fact that $\mathbf{T} \perp \mathbf{U}_2$, we have the following approximation for i-vector extraction

$$\hat{\phi} = \left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}(\mathbf{I} + \mathbf{N})^{-1}\widetilde{\mathbf{F}} \tag{23}$$

The complexity of computing i-vector using (23) is $O(CFM)$. The matrix $\widetilde{\mathbf{T}} = \mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}$ could be pre-computed and stored, where the memory demand is $O(CFM)$. Compared to the standard form in (8) and (9), our proposal in (23) is 12 time faster and with much smaller memory requirement (see details in Section 6). It is worth mentioning that there are two modifications that leads to the fast i-vector extraction above, namely, the introduction of subspace-orthonormalizing prior of (12), and the uniform-occupancy assumption in (21).

Notice that we do not need to estimate the posterior covariance (14) first to compute (23). It is worth mentioning that, posterior covariance is determined by the zero-order statistics and the loading matrix $\mathbf{T}$. Both of these are used in (23). Following the same steps as above, full posterior covariance could be computed easily, as follows

$$\begin{aligned}\hat{\mathbf{L}}^{-1} &= \left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}\mathbf{T}^{\mathsf{T}}(\mathbf{I} + \mathbf{N})^{-1}\mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1} \\ &= \widetilde{\mathbf{T}}^{\mathsf{T}}(\mathbf{I} + \mathbf{N})^{-1}\widetilde{\mathbf{T}}\end{aligned} \tag{24}$$

It has been shown in [20] and [21] the posterior covariance could be propagated to the downstream as part of uncertainty modeling. Taking into account (24), the computational complexity is $O(CM^2)$, when pre-computing $\widetilde{\mathbf{T}}_c^{\mathsf{T}}\widetilde{\mathbf{T}}_c$ and storing them. Nevertheless, it should be noted that uncertainty propagation is usually not necessary for long segment.

4.2. Relation to PCA

Equation (23) shares a number of similarities with dimensionality reduction mapping. Let $\mathbf{f}$ be defined as $\mathbf{f} \equiv (\mathbf{I} + \mathbf{N})^{-1}\widetilde{\mathbf{F}}$ and $\widetilde{\mathbf{T}} \equiv \mathbf{T}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-1}$. The supervector-size vector $\mathbf{f}$ gives a fixed length representation of an utterance which is then mapped to a lower dimensional space via projection onto the columns of $\widetilde{\mathbf{T}}$:

$$\hat{\phi} = \widetilde{\mathbf{T}}^{\mathsf{T}}\mathbf{f} \tag{25}$$

The matrix $\widetilde{\mathbf{T}}$ is made up of the total variability matrix $\mathbf{T}$ trained using a maximum likelihood cost function [4]. Taking $\mathbf{f}$ as input data, we could train the transformation matrix $\widetilde{\mathbf{T}}$ using any dimension reduction technique, for instance, principle component analysis (PCA). Suppose we have n training utterances, we form the matrix $\mathbf{M}$ with each column being the vector $\mathbf{f}$ for a training utterance. Since $CF \gg n$, we perform eigenvalue decomposition on the matrix $\mathbf{M}^{\mathsf{T}}\mathbf{M}/n$ to obtain the transformation matrix as $\widetilde{\mathbf{T}} = \mathbf{MQ}$, where $\mathbf{Q}$ is an $n \times m$ matrix consisting of m leading eigenvectors. We compare the performance of probabilistic and deterministic ways of training the matrix $\widetilde{\mathbf{T}}$ in the next section.

5. EM steps for hyperparameter training

The subspace-orthonormalizing prior was introduced in Section 3 for i-vector extraction. We further show in Section 4, the subspace-orthonormalizing prior, together with the uniform-occupancy assumption, constitutes the necessary conditions for rapid computation of i-vector proposed in this paper. The subspace-orthonormalizing prior could be utilized in the EM training of the total variability matrix, by so doing, we insert some form of the self-orthonormalizing feature in the EM update of the $\mathbf{T}$ matrix.

Table 1: *Performance comparison under nine common conditions (CCs) of NIST SRE'10 extended-core task. Each entry shows the EER (%) and min DCF10 at the top and bottom rows, respectively. Entries in **bold** beat the others under the same CC.*

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
baseline	**2.1986**	**3.8313**	4.3591	2.8421	3.3182	5.3826	6.3412	2.6505	2.1288
	0.3657	**0.6088**	0.6553	0.4593	**0.5179**	0.8506	0.7698	**0.5199**	0.2393
proposed (exact)	2.3014	3.8716	4.3417	**2.7644**	**3.2932**	**5.3464**	**6.3280**	**2.6176**	1.9896
	0.3680	0.6091	**0.6468**	**0.4581**	0.5192	0.8550	0.7565	0.5259	**0.2212**
proposed (fast)	2.4699	4.4485	**4.1780**	2.8356	3.6514	5.9624	7.2228	2.9741	1.8361
	0.3681	0.6507	0.7142	0.4963	0.5414	**0.8349**	**0.7331**	0.5786	0.2881
$(\mathbf{I}+\mathbf{N})^{-1}\,\widetilde{\mathbf{F}}$	2.4144	4.5298	4.9959	3.2292	3.8124	6.2883	6.8657	2.8703	**1.7414**
	0.4247	0.7267	0.7856	0.5281	0.6106	0.9183	0.8013	0.5915	0.2994
$(\mathbf{I}+\mathbf{N})^{-1}\,\mathbf{F}$	3.5880	6.7725	6.2079	4.4367	4.6302	7.3215	9.7463	3.8837	2.2043
	0.5502	0.8591	0.8916	0.6586	0.6537	0.9457	0.8589	0.6860	0.3936

Table 2: *Performance comparison of different approaches using the same $\mathbf{T}$ matrix trained using the subspace-orthonormalizing priors detailed in algorithm 1. Each entry shows the EER (%) and min DCF10 at the top and bottom rows, respectively. Entries in **bold** beat the others under the same CC.*

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
baseline	**2.2123**	**3.8453**	**4.2795**	2.8363	**3.3065**	5.3342	6.6660	2.6034	2.1221
	0.3568	**0.6072**	0.6593	0.4623	**0.5162**	0.8539	0.7609	**0.5248**	0.2377
proposed (exact)	2.3570	3.9401	4.2906	**2.8029**	3.3238	**5.3129**	**6.2746**	**2.5969**	1.8906
	0.3659	0.6171	**0.6518**	**0.4622**	0.5183	0.8548	0.7669	0.5292	**0.2375**
proposed (fast)	2.4472	4.4163	4.1910	2.8324	3.6438	5.8565	7.2230	2.9446	**1.8506**
	0.3683	0.6516	0.7171	0.4953	0.5409	**0.8346**	**0.7333**	0.5712	0.2916

In the E-step, using the current estimate of $\mathbf{T}$, we compute the posterior mean using (13) and covariance using (14). In the M-step, we accumulate statistics across all utterances, and perform a single update on $\mathbf{T}$. The EM steps are repeated until convergence. Algorithm 1 lists the details of the EM algorithm. Note that the first-order statistics $\widetilde{\mathbf{F}}$ have been whitened and centred according to (7). It is worth mentioning that, the fast method given by (23) and (24) could be used for posterior estimation in the E-step. However, this is generally not required as the EM training of $\mathbf{T}$ matrix is usually done once in an off-line manner.

6. Experiments

The acoustic features used in the experiments consists of 19-dimensional mel frequency cepstral coefficients (MFCC). Delta and delta-delta features were appended giving rise to 57-dimensional feature vector. We used gender-dependent UBM consisting of 512 mixtures with full covariance matrices. The total variability matrix $\mathbf{T}$ was trained using Swichboard, NIST SRE'04, 05, and 06 data. The rank of the matrix $\mathbf{T}$ is set to M = 400. The dimensionality of the i-vectors was first reduced to 300 with LDA and length normalization [22] was applied. The PLDA model was trained to have 200 speaker factors with a full residual covariance for channel modeling. Experiments were performed on the extended core task of NIST SRE'10 [23] consisting of nine common conditions (CCs) with different recording conditions for the training and test segments.

We evaluate the performance of the proposed method, both exact and fast versions, taking the standard i-vector given by (8) and (9) as the baseline. Table 1 shows the performance comparison in terms of *equal error rate* (EER) and *minimum DCF* (min DCF) for the female partition of the core-extended task. The *proposed* (*exact*) method refers to i-vector extraction using the subspace-orthonormalizing prior, which is given by (13)

Algorithm 1: EM steps for each iterative update of the $\mathbf{T}$ matrix given n training utterances with the use of subspace-orthnormalizing prior

input : $\mathbf{T}, \mathbf{N}, \widetilde{\mathbf{F}}$
output: $\mathbf{T}$
begin
  // Reset accumulators
  $\mathbf{A} = \mathbf{0}, \mathbf{C} = \mathbf{0}$;
  // Expectation Step
  for $s = 1$ **to** n **do**
    $\mathbf{L}_s = \left(\mathbf{T}^{\mathsf{T}}\mathbf{T} + \mathbf{T}^{\mathsf{T}}\mathbf{N}_s\mathbf{T}\right)$;
    $\phi_s = \mathbf{L}_s^{-1} \cdot \mathbf{T}^{\mathsf{T}}\widetilde{\mathbf{F}}_s$;
    $E[\mathbf{x_s}\mathbf{x_s}^{\mathsf{T}}] = \phi_s\phi_s^{\mathsf{T}} + \mathbf{L}_s^{-1}$;
    // Accumulate Statistics
    for $c = 1$ **to** C **do**
      $\mathbf{A}_c = \mathbf{A}_c + \mathbf{N}_c(s) \cdot E[\mathbf{x_s}\mathbf{x_s}^{\mathsf{T}}]$;
    end
    $\mathbf{C} = \mathbf{C} + \widetilde{\mathbf{F}}_s \cdot \phi_s^{\mathsf{T}}$;
  end
  // Maximization Step
  for $c = 1$ **to** C **do**
    $\mathbf{T}_c = \mathbf{C}_c \cdot \mathbf{A}_c^{-1}$ // Solves for $\mathbf{T}$
  end
end

Table 3: *Computational Complexity and Memory Cost on different approaches.*

	Complexity	Memory Cost	Time ratio
slow baseline	$O\left(CFM^2 + M^3\right)$	$O\left(CFM\right)$	106.44
fast baseline	$O\left(M^3 + CFM + CM^2\right)$	$O\left(CFM + CM^2\right)$	11.99
proposed (exact)	$O\left(M^3 + CFM + CM^2\right)$	$O\left(CFM + CM^2\right)$	12.65
proposed (fast)	$O\left(CFM\right)$	$O\left(CFM\right)$	1

and (14). Compare to the standard i-vector baseline, the performance is almost the same in terms of EER and min DCF across all nine common conditions. This is important as it assures that introducing a new prior during the i-vector extraction stage does not degrade the performance (We shall show later that using the new prior in the EM update does improve slightly the performance). Notice that the *proposed* (*exact*) method incurs the same computational complexity as the standard i-vector, i.e., the *fast baseline* as shown in Table 3. As mentioned briefly in the last paragraph of Section 2, we pre-computed the matrix $\mathbf{T}_c^{\mathsf{T}}\mathbf{T}_c$ for faster computation in the fast baseline. Looking back at Table I and II, it is worth mentioning that the performance of the fast and slow baseline are exactly the same, so we do not specially refer to the baseline as fast or slow.

Table 3 shows a comparison of computational complexity and the memory demand required by each method. The time ratio was computed with respect to the *proposed* (*fast*) method based on time required to process one thousand utterances with the same setup and on the same machine. In Table 3, we show that the *proposed* (*fast*) method in (23) is 10 times and 100 times faster than the *fast* and *slow* variants of the standard i-vector baseline with a small memory demand. This comes at the cost of a small degradation in recognition accuracy. For all the 9 CCs in Table 1, the relative degradation ranges from 10.04% to 16.11% in EER and 0.67% to 20.40% in min DCF. In particular, for the tel-tel common condition 5, the relative degradation is 10.04% in EER and 4.54% in min DCF.

Also shown in Table 1, in the last two rows, are the results for the deterministic approach using PCA as described in Section 4.2. In particular, we trained the transformation matrix $\widetilde{\mathbf{T}}$ using PCA by taking either $(\mathbf{I}+\mathbf{N})^{-1}\widetilde{\mathbf{F}}$ or $(\mathbf{I}+\mathbf{N})^{-1}\mathbf{F}$ as inputs. Comparing these results, it is evident that the whiten $\widetilde{\mathbf{F}}$ gives a better performance than the non-whiten counterpart $\mathbf{F}$. Comparing the results in the last three rows of Table 1, it is obvious that the transformation matrix constructed from the total variability matrix gives a better performance compared to that obtained using PCA. Figure 1 shows the DET plot for tel-tel common condition (CC5).

Table 2 shows the performance of the standard i-vector and the proposed (both exact and fast versions) methods with the same set-up as in Table 1 except for the $\mathbf{T}$ matrix used. Different from that in Table 1, we trained the $\mathbf{T}$ matrix using subspace-orthonormalizing prior as detailed in Algorithm 1. Comparing the EER and minDCF in the two tables, we observe a better performance could be obtained for most of the common conditions, though the improvement is marginal.

7. Conclusions

We have introduced the use of subspace-orthonormalizing prior for i-vector extraction. The orthonormal property imposed on the total variability space allows us to reformulate the equation to compute i-vector without having to evaluate the full posterior covariance beforehand. In conjunction with the uniform-occupancy assumption, the use of the new prior leads to a very fast method for rapid computation of i-vector. Compared to the standard forms with and without matrix pre-computation (i.e., the slow and fast i-vector baselines), the proposed method manages to speed up the i-vector extraction process by a factor of 12 and 106, respectively. Experiments conducted on SRE'10 extended-core task show that the relative degradation in EER is 10.04% on the tel-tel task.

With the use of DNN senone posterior becomes more popular, the computational demand is expected to increase tremendously for large senone set. We believe that the results presented in this paper would path the way for further research to cope with the computational requirement in this direction.

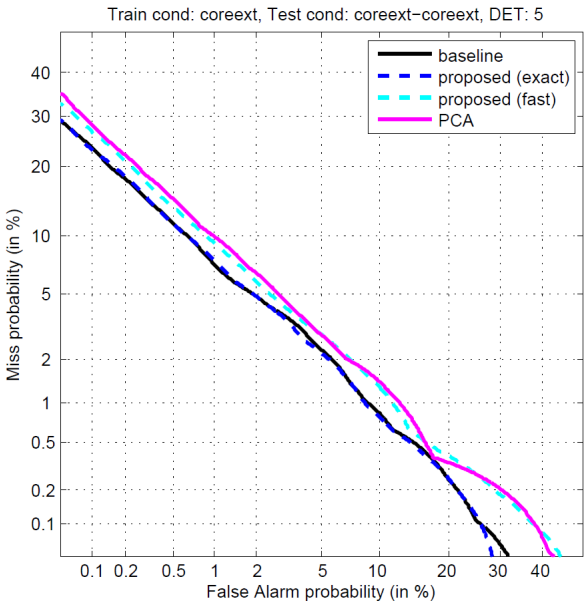

Figure 1: DET curves on the tel-tel common condition (CC) 5 of NIST SRE'10 core-extended task using different methods.

8. Acknowledgment

This work of Longting Xu is supported by the National Natural Science Foundation of China (No.60971129, 61271335), the Scientific Innovation Research Programs of College Graduate in Jiangsu Province (No.CXZZ13_0488), Key Laboratory of the Ministry of Public Security Smart Speech Technology (No.2014ISTKFK T02), the Natural Science Foundation of Jiangsu Province (No. BK20140891), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No.13KJB510020), the National Natural Science Foundation of China (Grant No.61501251), the NUPTSF (Grant No. NY214191), and China Scholarship Council.

9. References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, pp. 19–41, 2000.

[2] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

[3] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 980–988, 2008.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE Computer Vision*, 2007, pp. 1–8.

[7] P. Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Proc. Odyssey*, 2010, p. 14.

[8] K. A. Lee, A. Larcher, C. H. You, B. Ma, and H. Li, "Multi-session PLDA scoring of i-vector for partially open-set speaker detection," in *Proc. INTERSPEECH*, 2013, pp. 3651–3655.

[9] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *IEEE Signal Processing Society Speech and language Technical Committee Newsletter*, 2013.

[10] Nuance vocalpassword, "http://www.nuance.com/for-business/customer-service-solutions/voice-biometrics/vocalpassword/index.htm," 2015.

[11] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, "Simplification and optimization of i-vector extraction," in *Proc. IEEE ICASSP*, 2011, pp. 4516–4519.

[12] H. Aronowitz and O. Barkan, "Efficient approximated i-vector extraction," in *Proc. IEEE ICASSP*, 2012, pp. 4789–4792.

[13] S. Cumani and P. Laface, "Factorized sub-space estimation for fast and memory effective i-vector extraction," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 1, pp. 248–259, 2014.

[14] L. Xu, K. A. Lee, H. Li, and Z. Yang, "Sparse coding of total variability matrix," in *Proc. INTERSPEECH*, 2015, pp. 1022–1026.

[15] S. E. Shepstone, K. A. Lee, Z-H. Tan H. Li, and S. H. Jensen, "Source-specific informative prior for i-vector extraction," in *Proc. IEEE ICASSP*, April 2015, pp. 4185–4189.

[16] S. E. Shepstone, K. A. Lee, Z-H. Tan H. Li, and S. H. Jensen, "Total variability modeling using source-specific priors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[17] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE ICASSP*, 2014, pp. 1695–1699.

[18] P. Kenny, "A small footprint i-vector extractor," in *Proc. Odyssey*, 2012, vol. 2012.

[19] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*, Technical University of Denmark, 2008.

[20] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. IEEE ICASSP*, 2013, pp. 7649–7653.

[21] S. Cumani, O. Plchot, and R. Fér, "Exploiting i-vector posterior covariances for short-duration language recognition," in *Pro. INTERSPEECH*, 2015, pp. 1002–1006.

[22] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Proc. Interspeech*, 2011, pp. 249–252.

[23] 2010 NIST Speaker Recognition Evaluation, "http://www.itl.nist.gov/iad/mig//tests/sre/2010/," 2010.