Fast and Memory Effective I-Vector Extraction using a Factorized Sub–space

Sandro Cumani[1,2], *Pietro Laface*[1]

[1]Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy
[2] Brno University of Technology, Brno, Czech Republic
Sandro.Cumani@polito.it, Pietro.Laface@polito.it

Abstract

Most of the state–of–the–art speaker recognition systems use a compact representation of spoken utterances referred to as i–vectors. Since the "standard" i–vector extraction procedure requires large memory structures and is relatively slow, new approaches have recently been proposed that are able to obtain either accurate solutions at the expense of an increase of the computational load, or fast approximate solutions, which are traded for lower memory costs. We propose a new approach particularly useful for applications that need to minimize their memory requirements. Our solution not only dramatically reduces the storage needs for i–vector extraction, but is also fast. Tested on the female part of the tel-tel extended NIST 2010 evaluation trials, our approach substantially improves the performance with respect to the fastest but inaccurate eigen-decomposition approach, using much less memory than any other known method.
Index Terms: Speaker recognition, i-vectors, i-vector extraction, Singular Value Decomposition.

1. Introduction

A simple and effective model for speaker recognition has been introduced in [1, 2]. In this approach, speaker and channel variabilities are modeled in a common constrained low dimensional space, and a speech segment is represented by a low-dimensional "identity vector" or i-vector. The low dimensionality of i–vectors makes them suitable for fast classification using either generative models based on Probabilistic Linear Discriminant Analysis (PLDA) [3, 4], or discriminative classifiers such as Support Vector Machines or Logistic Regression [5, 6].

Since the "standard" i–vector extraction procedure requires large memory structures and is relatively slow, new approaches have recently been proposed that are able to obtain either fast approximate solutions, [7, 8], possibly traded for lower memory costs, or accurate solutions at the expense of an increase of the computational load [9, 10, 11]. In [7] a simplification of the i–vector extraction is proposed based on an approximated simultaneous diagonalization of the terms composing the i–vector posterior covariance matrix. This "eigen–decomposition" approach is very fast and memory effective, and gives good performance, but cannot reach the accuracy of the standard one. In this paper we propose a new approach particularly useful for applications that need to optimize their memory requirements. The key idea in our solution is that it is possible to factorize the variability sub–space matrix $\mathbf{T}$ so that it is not necessary to store all its rows to perform i–vector extraction. These rows can be obtained as a linear combination and rotation of the atoms of a common dictionary.

The paper is organized as follows: Section 2 summarizes the i–vector model for speaker recognition. Section 3 recalls the eigen–decomposition i–vector estimation approach. Our novel factorized sub–space approach is illustrated in Section 4. Section 5 is devoted to the estimation of the dictionary and of the other matrices needed for approximating matrix $\mathbf{T}$. I–vector extraction with our approach is illustrated in Section 6. The experimental results are presented and commented in Section 7, and conclusions are drawn in Section 8.

2. I–vector model

The i–vector model [1, 2] constrains the GMM supervector $\mathbf{s}$, representing both speaker and channel characteristics of a given speech segment, to live in a single sub–space according to:

$$\mathbf{s} = \mathbf{m} + \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{T}\mathbf{w} \; , \tag{1}$$

where $\mathbf{m}$ is the UBM supervector, $\mathbf{T}$ is a low-rank rectangular matrix with $C \times F$ rows and M columns and C and F are the number of GMM components and feature dimension, respectively. $\mathbf{T}$ is normalized for convenience by $\mathbf{\Sigma}^{\frac{1}{2}}$, where $\mathbf{\Sigma}$ denotes the block–diagonal matrix whose diagonal contains the UBM covariance matrices $\mathbf{\Sigma}^{(c)}$. The M columns of $\mathbf{T}$ are vectors spanning the variability space, and $\mathbf{w}$ is a random vector of size M with a standard normal prior distribution. It is worth noting that the i–vector model (1) is equivalent to the classical i–vector model, but takes advantage of the UBM statistics whitening introduced in [7] to simplify the i–vector extraction. Following [12] and the notation in [7], given a sequence of feature vectors $\mathcal{X} = \mathbf{x}_1\mathbf{x}_2 \dots \mathbf{x}_t$ extracted for a speech segment, the corresponding i–vector $\mathbf{w}_\mathcal{X}$ is computed as the mean of the posterior distribution $\mathrm{p}(\mathbf{w}|\mathcal{X})$:

$$\mathbf{w}_\mathcal{X} = \mathbf{L}_\mathcal{X}^{-1}\mathbf{T}^*\mathbf{f}_\mathcal{X} \; , \tag{2}$$

where $\mathbf{L}$ is the precision matrix of the posterior distribution:

$$\mathbf{L}_\mathcal{X} = \mathbf{I} + \sum_c N_\mathcal{X}^{(c)}\mathbf{T}^{(c)^*}\mathbf{T}^{(c)} \; . \tag{3}$$

In these equations, $N_\mathcal{X}^{(c)} = \sum_t \gamma_t^{(c)}$ are the zero–order statistics estimated on the c-th Gaussian component of the UBM for the set of feature vectors in $\mathcal{X}$, $\mathbf{T}^{(c)}$ is the $F \times M$ sub-matrix of $\mathbf{T}$ corresponding to the c–th mixture component such that $\mathbf{T} = \left(\mathbf{T}^{(1)*}, \dots, \mathbf{T}^{(C)*}\right)^*$, and $\mathbf{f}_\mathcal{X}$ is the supervector stacking the covariance–normalized first–order statistics $\mathbf{f}_\mathcal{X}^{(c)}$, centered around the corresponding UBM means:

$$\mathbf{f}_\mathcal{X}^{(c)} = \mathbf{\Sigma}^{(c)^{-\frac{1}{2}}} \left[\sum_t \left(\gamma_t^{(c)}\mathbf{x}_t\right) - N_\mathcal{X}^{(c)}\mathbf{m}^{(c)}\right] \; , \tag{4}$$

Sandro Cumani is currently supported by Czech Ministry of Education project No. CZ.1.07/2.3.00/30.0005.

where $\mathbf{x}_t$ is the t–th feature vector in $\mathcal{X}$, $\gamma_t^{(c)}$ is its occupation probability and $\mathbf{\Sigma}^{(c)^{-1}}$ is the UBM c–th component precision matrix.

3. Approximate i–vector extraction

Since $\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}$ is a symmetric and semi–definite positive matrix, it can be eigen–decomposed as:

$$\mathbf{T}^{(c)^*}\mathbf{T}^{(c)} = \mathbf{G}^{(c)}\mathbf{D}^{(c)}\mathbf{G}^{(c)^*} , \tag{5}$$

where $\mathbf{G}^{(c)}$ is an orthogonal matrix, and matrix $\mathbf{D}^{(c)}$ is diagonal. $\mathbf{D}^{(c)}$ can be expressed, in terms of $\mathbf{G}^{(c)}$ and $\mathbf{T}^{(c)}$, as:

$$\mathbf{D}^{(c)} = \mathbf{G}^{(c)^*}\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}\mathbf{G}^{(c)} . \tag{6}$$

A simultaneous approximate diagonalization of the matrices $\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}$ has been introduced in [7] for fast computation of the i–vectors with low memory resources. In this approach, each $\mathbf{G}^{(c)}$ is replaced, for the sake of efficiency, by a single matrix $\mathbf{Q}$

$$\hat{\mathbf{D}}^{(c)} = \mathbf{Q}^*\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}\mathbf{Q} \tag{7}$$

and every $\hat{\mathbf{D}}^{(c)}$, which is thus no more diagonal, is forced to be diagonal by setting to zero its off-diagonal elements.

A suitable common orthogonalizing matrix $\mathbf{Q}$ has been proposed in [7], based on the eigen–decomposition $\mathbf{W} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ of the weighted average covariance matrix

$$\mathbf{W} = \sum_c \omega^{(c)}\mathbf{T}^{(c)^*}\mathbf{T}^{(c)} , \tag{8}$$

where $\omega^{(c)}$ is the weight of the UBM c–th component.

Substituting (7) in (3), one gets the approximated posterior distribution precision matrix

$$\tilde{\mathbf{L}}_{\mathcal{X}} = \mathbf{Q}\hat{\mathbf{L}}_{\mathcal{X}}\mathbf{Q}^* , \tag{9}$$

where $\hat{\mathbf{L}}_{\mathcal{X}} = \mathbf{I} + \sum_c N_{\mathcal{X}}^{(c)}\hat{\mathbf{D}}^{(c)}$. The approximated precision $\tilde{\mathbf{L}}_{\mathcal{X}}$ can then be used in (2) to compute the i–vector. Assuming that $\hat{\mathbf{D}}^{(c)}$ in (7) is diagonal has the remarkable advantage that $\hat{\mathbf{L}}_{\mathcal{X}}$ can be computed by C element–wise products of two vectors of dimension M, and its inversion cost in (2) becomes negligible.

This approach is very fast and memory effective, and its performance is good, but it does not reach the accuracy of the standard approach. Thus, alternative memory-aware accurate i–vector extraction methods have been recently introduced, based on a Variational Bayes formulation [9, 11], or on Conjugate Gradient (CG) [11], which compute i–vectors as accurate as the ones obtained by the standard technique, but require only a fraction of its memory. Since these approaches save memory, but are slower than the standard one, we introduce in the next section a new approach that substantially reduces the memory costs and gives higher performance compared to the eigen–decomposition technique of the previous section, using comparable processing resources.

4. Factorized sub–space estimation of matrix T

The eigen–decomposition approach [7] does not reach the accuracy of the standard one because useful information conveyed by the off–diagonal elements of $\hat{\mathbf{D}}^{(c)}$ is discarded. In order to overcome this weakness, we propose a new approach that is able to obtain more accurate estimates of the $\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}$ matrices.

Let's rewrite (6) as $(\mathbf{T}^{(c)}\mathbf{G}^{(c)})^*(\mathbf{T}^{(c)}\mathbf{G}^{(c)}) = \mathbf{D}^{(c)}$. Since $\mathbf{D}^{(c)}$ is diagonal, it can be proved that $\mathbf{T}^{(c)}\mathbf{G}^{(c)}$ can be decomposed as:

$$\mathbf{T}^{(c)}\mathbf{G}^{(c)} = \mathbf{O}^{(c)}\mathbf{\Pi}_M^{(c)} ,$$

where $\mathbf{O}^{(c)}$ is an orthonormal $F \times F$ matrix, and $\mathbf{\Pi}_M^{(c)}$ is an $F \times M$ matrix having at most one non–null element per row. Thus, $\mathbf{T}^{(c)}$ can be obtained as:

$$\mathbf{T}^{(c)} = \mathbf{O}^{(c)}\mathbf{\Pi}_M^{(c)}\mathbf{G}^{(c)^*} . \tag{10}$$

An accurate approximation $\hat{\mathbf{T}}^{(c)}$ of each matrix $\mathbf{T}^{(c)}$ can be obtained by replacing each $M \times M$ matrix $\mathbf{G}^{(c)}$ by a single, larger, $K \times M$ matrix $\mathbf{Q}$ as:

$$\hat{\mathbf{T}}^{(c)} \approx \mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q} . \tag{11}$$

where $\mathbf{\Pi}^{(c)}$ is, in this case, a sparse $F \times K$ matrix with at most one non–null element per row. Thus, each matrix $\hat{\mathbf{T}}^{(c)}$ is obtained by a linear combination and rotation of F vectors, selected from a set of K atoms collected in a shared dictionary represented by the $K \times M$ matrix $\mathbf{Q}$.

Compared to the eigen–decomposition approach, our proposal has substantial differences not only because the matrices $\mathbf{Q}$, $\mathbf{O}^{(c)}$ and $\mathbf{\Pi}^{(c)}$ are obtained by a completely different optimization process, described in Section 5, but also and above all because the size of the dictionary $\mathbf{Q}$ is not constrained to be $M \times M$. We have the freedom to select the first dimension of $\mathbf{Q}$. Thus, setting $K >> M$ allows us to estimate more accurate $\mathbf{T}^{(c)}$ matrices. Moreover, modelling directly the matrices $\mathbf{T}^{(c)}$ allows to avoid storing the full $\mathbf{T}$ matrix, needed in the eigen–decomposition approach to compute (2), thus keeping the storage cost small.

5. Matrix $\mathbf{T}^{(c)}$ approximation

The matrices $\mathbf{O}^{(c)}$, $\mathbf{\Pi}^{(c)}$, and $\mathbf{Q}$ are obtained by minimizing a weighted average square norm of the differences between matrices $\mathbf{T}^{(c)}$ and their approximations $\hat{\mathbf{T}}^{(c)}$:

$$\min_{\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}} \sum_c \omega^{(c)} ||\mathbf{T}^{(c)} - \mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}||^2 . \tag{12}$$

The optimization is performed by updating a matrix while keeping constant the others, according to the iterative sequence of optimizations illustrated in Table 1.

In order to solve our optimization problem we rewrite the objective function (12) as:

$$\min_{\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}} \sum_c \omega^{(c)} \Big[\mathrm{tr}\left(\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}\right) + \tag{13}$$

$$\mathrm{tr}\left(\mathbf{Q}^*\mathbf{D}^{(c)}\mathbf{Q}\right) - 2\,\mathrm{tr}\left(\mathbf{T}^{(c)^*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right)\Big] ,$$

where $\mathbf{D}^{(c)} = \mathbf{\Pi}^{(\mathbf{c})^*}\mathbf{\Pi}^{(c)}$ is a diagonal matrix.

5.1. Matrix Q optimization

We solve for $\mathbf{Q}$ by zeroing the gradient of (13) while keeping fixed $\mathbf{O}^{(c)}$ and $\mathbf{\Pi}^{(c)}$ obtaining:

$$\mathbf{Q} = \left(\sum_c \omega^{(c)}\mathbf{D}^{(c)}\right)^{-1} \left(\sum_c \omega^{(c)}\mathbf{\Pi}^{(c)*}\mathbf{O}^{(c)*}\mathbf{T}^{(c)}\right) . \tag{14}$$

Table 1: Iterative optimization sequence.

Update	Constant	Constant
$\mathbf{\Pi}^{(c)}$ eq. (21)	$\mathbf{Q}$	$\mathbf{O}^{(c)}$
$\mathbf{O}^{(c)}$ eq. (17)	$\mathbf{Q}$	$\mathbf{\Pi}^{(c)}$
$\mathbf{Q}$ eq. (14)	$\mathbf{O}^{(c)}$	$\mathbf{\Pi}^{(c)}$

5.2. Matrix $\mathbf{O}^{(c)}$ optimization

The optimization of each matrix $\mathbf{O}^{(c)}$ can be done independently from the others, by maximizing the third term in (13) keeping constants $\mathbf{\Pi}^{(c)}$ and $\mathbf{Q}$:

$$\max_{\mathbf{O}^{(c)}} \ \mathrm{tr}\left(\mathbf{T}^{(c)*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right) \qquad \text{s.t.} \quad \mathbf{O}^{(c)^*}\mathbf{O}^{(c)} = \mathbf{I} \tag{15}$$

Since the trace operator is invariant under cyclic permutations, we can rewrite the argument of (15) as:

$$\mathrm{tr}\left(\mathbf{T}^{(c)*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right) = \mathrm{tr}\left(\mathbf{O}^{(c)}\mathbf{Z}\right) \ ,$$

where $\mathbf{Z} = \mathbf{\Pi}^{(c)}\mathbf{Q}\mathbf{T}^{(c)*}$. The Von Neumann's trace inequality [13, 14] states that:

$$\left|\mathrm{tr}(\mathbf{O}^{(c)}\mathbf{Z})\right| \leq \sum_{i=1}^{F} \sigma_{oi}\sigma_{zi} \ ,$$

where σ_{oi} and σ_{zi} are the sorted i–th singular values obtained by Singular Value Decomposition (SVD) of $\mathbf{O}^{(c)}$ and $\mathbf{Z}$, respectively. Since $\mathbf{O}^{(c)}$ has to be orthonormal, its singular values have to be equal to 1, thus for any feasible solution $\mathbf{O}^{(c)}$ the objective function is bounded by

$$\left|\mathrm{tr}(\mathbf{O}^{(c)}\mathbf{Z})\right| \leq \sum_{i=1}^{F} \sigma_{zi} \ , \tag{16}$$

and can therefore be maximized if we find a matrix $\mathbf{O}^{(c)}$ such that the singular values of $\mathbf{O}^{(c)}\mathbf{Z}$ and $\mathbf{Z}$ are exactly the same. This condition is satisfied by matrix:

$$\mathbf{O}^{(c)} = \mathbf{V}_{\mathbf{Z}} U^*_{\mathbf{Z}} \ , \tag{17}$$

where $\mathbf{Z} = \mathbf{U}_{\mathbf{Z}}\mathbf{\Sigma}_{\mathbf{Z}}\mathbf{V}^*_{\mathbf{Z}}$ is a SVD of $\mathbf{Z}$. This can be verified substituting (17) in the left hand term of (16):

$$\begin{aligned}\left|\mathrm{tr}(\mathbf{O}^{(c)}\mathbf{U}_{\mathbf{Z}}\mathbf{\Sigma}_{\mathbf{Z}}\mathbf{V}^*_{\mathbf{Z}})\right| &= |\mathrm{tr}(\mathbf{V}_{\mathbf{Z}}\mathbf{U}^*_{\mathbf{Z}}\mathbf{U}_{\mathbf{Z}}\mathbf{\Sigma}_{\mathbf{Z}}\mathbf{V}^*_{\mathbf{Z}})| = \\ |\mathrm{tr}(\mathbf{V}_{\mathbf{Z}}\mathbf{\Sigma}_{\mathbf{Z}}\mathbf{V}^*_{\mathbf{Z}})| &= |\mathrm{tr}(\mathbf{\Sigma}_{\mathbf{Z}})| = \sum_{i=1}^{F}\sigma_{zi} \ .\end{aligned} \tag{18}$$

5.3. Matrix $\mathbf{\Pi}^{(c)}$ optimization

Considering again (13), the optimization can be done independently for each $\mathbf{\Pi}^{(c)}$ considering constants $\mathbf{O}^{(c)}$ and $\mathbf{Q}$, as:

$$\min_{\mathbf{\Pi}^{(c)}} \left[\mathrm{tr}\left(\mathbf{Q}^*\mathbf{\Pi}^{(\mathbf{c})^*}\mathbf{\Pi}^{(c)}\mathbf{Q}\right) - 2\,\mathrm{tr}\left(\mathbf{T}^{(c)^*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right)\right] \ . \tag{19}$$

Defining $\mathbf{A}^{(c)} = \mathbf{O}^{(c)^*}\mathbf{T}^{(c)}\mathbf{Q}^*$, and noting that although the dimension of $\mathbf{Q}\mathbf{Q}^*$ is huge, we need only its diagonal because $\mathbf{\Pi}^{(\mathbf{c})^*}\mathbf{\Pi}^{(c)}$ is diagonal, the arguments of objective function (19) can be rewritten as:

$$\sum_{f=1}^{F}\left[\sum_{k=1}^{K} q_k^2\left(\boldsymbol{\pi}_f^{(c)^*}\boldsymbol{\pi}_f^{(c)}\right)_{k,k} - 2\,\mathrm{tr}\left(\boldsymbol{\pi}_f^{(c)*}A_f^{(c)}\right)\right] \ , \tag{20}$$

where q_k^2 is the k–th element of the diagonal of $\mathbf{Q}\mathbf{Q}^*$, and $\boldsymbol{\pi}_f^{(c)}$ is the f-th row of $\mathbf{\Pi}^{(c)}$.

Since the terms in the summation of (20) can be factorized with respect to the rows $\boldsymbol{\pi}_f^{(c)}$, we can optimize each $\boldsymbol{\pi}_f^{(c)}$ independently.

Since the row vector $\boldsymbol{\pi}_f^{(c)}$ should have at most a single non-zero element v_k, with index k, the optimal index k_f^{opt} and its corresponding value v_f^{opt} are obtained as:

$$k_f^{opt} = \arg\max_k \frac{A_{f,k}^{(c)2}}{q_k^2} \qquad v_f^{opt} = \frac{A^{(c)}_{f,k^{opt}}}{q^2_{k^{opt}}} \ . \tag{21}$$

5.4. Initialization of matrices Q and $\mathbf{O}^{(c)}$

In order to initialize the dictionary matrix $\mathbf{Q}$ and all the matrices $\mathbf{O}^{(c)}$, we compute the SVD of each matrix $\omega^{(c)}\mathbf{T}^{(c)}$ as:

$$\omega^{(c)}\mathbf{T}^{(c)} = \mathbf{U}^{(c)}\boldsymbol{S}^{(c)}\mathbf{V}^{(c)^*}$$

Matrix $\mathbf{O}^{(c)}$ is then initialized by the corresponding matrix $\mathbf{U}^{(c)}$. Matrix $\mathbf{Q}$ is initialized by pooling together the rows of the matrices $\mathbf{V}^{(c)^*}$ and keeping only the rows corresponding to the largest pooled singular values.

Matrix $\mathbf{\Pi}^{(c)}$ does not need to be initialized because, given $\mathbf{Q}$ and $\mathbf{O}^{(c)}$, $\mathbf{\Pi}^{(c)}$ can be computed by (21) as illustrated in Section 5.3. In our experiments, 10 iterations of alternate optimizations of $\mathbf{\Pi}^{(c)}$ and $\mathbf{O}^{(c)}$ are performed before a new matrix $\mathbf{Q}$ is estimated keeping fixed $\mathbf{\Pi}^{(c)}$ and $\mathbf{O}^{(c)}$. This procedure is repeated for 40 iterations.

It is worth noting that matrix $\mathbf{\Pi}^{(c)}$ is full rank only if it has a single non–zero element per row, located in different columns. Our optimization procedure introduced in Section 5.3, however, is not able to directly impose such constraints on matrix $\mathbf{\Pi}^{(c)}$, and it may happen that an estimated matrix is not full rank. Although this does not affect sensibly the recognition accuracy, it does not allow to fully exploit the potential of the method. We developed, thus, a further optimization procedure for obtaining full rank $\mathbf{\Pi}^{(c)}$ matrices, which is not described here due to the page limitations for this paper.

6. I-vector extraction

Using the approximated $\hat{\mathbf{T}}^{(c)}$ of (11), the posterior distribution precision matrix $\mathbf{L}_{\mathcal{X}}$ in (3) can be computed as:

$$\begin{aligned}\hat{\mathbf{L}}_{\mathcal{X}} &= \mathbf{I} + \sum_c N_{\mathcal{X}}^{(c)}\mathbf{Q}^*\mathbf{\Pi}^{(c)^*}\mathbf{O}^{(c)^*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q} \\ &= \mathbf{I} + \mathbf{Q}^*\sum_c N_{\mathcal{X}}^{(c)}\mathbf{\Pi}^{(c)^*}\mathbf{\Pi}^{(c)}\mathbf{Q} \ .\end{aligned} \tag{22}$$

From (2):

$$\hat{\mathbf{L}}_{\mathcal{X}}\hat{\mathbf{w}}_{\mathcal{X}} = \sum_c \hat{\mathbf{T}}^{(c)*}\mathbf{f}_{\mathcal{X}}^{(c)} = \mathbf{Q}^*\sum_c \mathbf{\Pi}^{(c)^*}\mathbf{O}^{(c)^*}\mathbf{f}_{\mathcal{X}}^{(c)} \ .$$

Since matrix $\hat{\mathbf{L}}_{\mathcal{X}}$ is symmetric and positive definite, this linear system of equations can be solved by the Conjugate Gradient (CG) method. Since at each iteration n, the updates in this algorithm are based on the residual: $\mathbf{r}_n = \mathbf{c} - \hat{\mathbf{L}}\hat{\mathbf{w}}_n$, it is possible to reduce the high storage demands and the costs due to the computation and inversion of matrix $\hat{\mathbf{L}}_{\mathcal{X}}$, because it appears in the residual multiplied by $\hat{\mathbf{w}}_n$ [11]. Moreover, since matrices $\mathbf{\Pi}^{(c)}$ are sparse, much less operations are needed to compute $\hat{\mathbf{L}}\hat{\mathbf{w}}_n$ with respect to the full CG–based i–vector extractor [11].

Due to the space limitation we cannot give a report of the complexity analysis of various i–vector extraction approaches, but memory and computation costs can be compared in Table 2, which summarizes the results of the set of experiments performed to validate the factorized sub–space approach.

7. Experimental settings

Since this work was focused on memory and computational costs of i–vector extraction, we did not devote particular care to select the best combination of features, techniques, and training data that allow obtaining the best performance. Thus, we tested our systems only on the female part of the tel-tel extended NIST 2010 evaluation trials [15], which is known to be more difficult, thus more often compared in the literature.

We did experiments using the "standard", the Variational Bayes, the eigen–decomposition, and the factorized sub–space i–vector extraction techniques, with systems having the same front–end, based on cepstral features. In particular, we extracted, every 10 ms, 19 Mel frequency cepstral coefficients and the frame log-energy on a 25 ms sliding Hamming window. This 20–dimensional feature vector was subjected to short time mean and variance normalization using a 3 s sliding window, and a 60-dimensional feature vector was obtained by appending the delta and double delta coefficients computed on a 5–frame window. We trained a gender-independent UBM, modeled by a diagonal covariance 2048-component GMM, and also a gender-independent $\mathbf{T}$ matrix using only the NIST SRE 04/05/06 datasets. The i-vector dimension was fixed to 400 for all the experiments, and the i–vectors were length–normalized [16]. The classifier is based on Gaussian PLDA, implemented according to the framework illustrated in [3] with i–vectors. We trained models with full–rank channel factors, using 120 dimensions for the speaker factors. The PLDA models have been trained using the same NIST datasets, and additionally the Switchboard II, Phases 2 and 3, and Switchboard Cellular, Parts 1 and 2 datasets.

Table 2 summarizes the performance of the evaluated approaches on the female part of the extended telephone condition in the NIST 2010 evaluation. The recognition accuracy is given in terms of Equal Error Rate (EER) and Minimum Detection Cost Functions defined by NIST for the 2008 (minDCF08) and 2010 (minDCF10) evaluations [15].

The baseline results, corresponding to the standard i–vector extraction, were obtained 14 times faster than the corresponding slow approach. However, the latter requires only 188 MB for storing matrix $\mathbf{T}$, whereas the former needs 5 times more memory to store the terms $\mathbf{T}^{(c)^*}\mathbf{T}^{(c)}$ required to speed–up the computation of (3). The approximate i–vector extraction based on eigen-decomposition [7] is extremely fast and requires almost the same amount of memory required for the accurate slow approach. However, it is not able to reach the accuracy of the baseline system.
The Variational Bayes system [11] is able to get the same results of the baseline systems, it is approximately 1.3 times slower than the standard approach, but it uses only 1/4 of its memory, slightly more than the fast, but inaccurate, eigen–decomposition approach.

Table 2: *%EER, minDCF08×1000 and minDCF10×1000 for the female NIST SRE2010 extended tel–tel condition using different i–vector extraction approaches. FSE–K–S refers to the Factorized sub–space estimation approach with a dictionary of K atoms, and stopping threshold S.*

System	Memory (MB)	time ratio	(%) EER	min DCF8	min DCF10
Fast baseline	815	4.70	3.59	181	566
Slow baseline	188	66.54	3.59	181	566
Variational Bayes	221	5.27	3.53	183	572
Eigen-dec.	191	1.00	4.27	201	692
FSE-2k-10	32	0.76	3.70	191	575
FSE-2k-100	32	0.57	4.04	194	581
FSE-3.5k-10	35	0.97	3.76	195	551
FSE-3.5k-100	35	0.68	4.08	201	588
FSE-5k-10	38	1.16	3.49	185	580
FSE-5k-100	38	0.78	3.69	191	606
FSE-10k-10	48	2.30	3.56	185	584
FSE-10k-100	48	1.43	3.76	190	589

I–vector extraction with our Factorized Sub–space Estimation (FSE) approach has been tested by training four systems, based on different dictionary dimensions: $K = 2000, 3500, 5000,$ and 10000, respectively. The i–vectors were obtained by the Conjugate Gradient procedure illustrated in Section 6 and in [11], stopping the iterations when the CG residual is less than two different thresholds $S = 10^2$ or $S = 10^1$, respectively. The results show that the FSE performance is always better than the eigen–decomposition approach, and depending on the dimension of the dictionary it can reach an accuracy comparable to the standard approach. FSE dramatically reduces the memory cost of i–vector extraction by 20 times compared to the standard approach, but also by 5 times compared to the other memory aware approaches. It is also extremely fast: faster than the standard method, and even faster than the eigen–decomposition approach for large UBM models and small dictionary sizes.
The small CG-2K systems perform surprisingly well, considering that they use $1/5$ of the memory of the eigen–decomposition approach, but obtains results similar to the standard technique.

8. Conclusions

A new approach has been presented that accurately approximates the components of the total variability matrix by means of a linear combination and rotation of the atoms of a dictionary. The use of a common dictionary not only allows the memory required to be reduced with respect to the standard approach, but also with respect to the eigen–decomposition technique, which cannot avoid storing the $\mathbf{T}$ matrix. Our approach is not as fast as the eigen–decomposition technique, but allows obtaining accurate i–vectors and results, and requires substantially less memory than any other technique.

Although this optimization is particularly useful for small footprint applications, it can be also relevant for speaker identification and verification applications, where the duration of the available speaker segments is short.

9. Acknowledgements

We would like to thank Daniele Colibro and Claudio Vair from Nuance, for useful discussions and support.

10. References

[1] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, and P. Ouellet, “Support Vector Machines versus fast scoring in the low-dimensional total variability space for speaker verification,” in *Proceedings of Interspeech 2009*, pp. 1559–1562, 2009.

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

[3] P. Kenny, “Bayesian speaker verification with Heavy–Tailed priors,” in *Keynote presentation, Odyssey 2010, The Speaker and Language Recognition Workshop*, 2010. Available at `http://www.crim.ca/perso/patrick.kenny/kenny\_Odyssey2010.pdf`.

[4] S. J. D. Prince and J. H. Elder, “Probabilistic Linear Discriminant Analysis for inferences about identity,” in *Proceedings of 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[5] S. Cumani, N. Brümmer, L. Burget, and P. Laface, “Fast discriminative speaker verification in the i–vector space,” in *Proceedings of ICASSP 2011*, pp. 4852–4855, 2011.

[6] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, “Discriminatively trained Probabilistic Linear Discriminant Analysis for speaker verification,” in *Proceedings of ICASSP 2011*, pp. 4832–4835, 2011.

[7] O. Glembek, L. Burget, P. Matějka, M. Karafiát, and P. Kenny, “Simplification and optimization of i–vector extraction,” in *Proceedings of ICASSP 2011*, pp. 4516–4519, 2011.

[8] H. Aronowitz and O. Barkan, “Efficient approximated i–vector extraction,” in *Proceedings of ICASSP 2012*, pp. 4789–4792, 2012.

[9] P. Kenny, “A small footprint i-vector extractor,” in *Proceedings of Odyssey 2012*, pp. 1–6, 2012.

[10] S. Cumani, P. Laface, and V. Vasilakakis, “Memory and computation effective approaches for i–vector extraction,” in *Proceedings of Odyssey 2012*, pp. 7–13, 2012.

[11] S. Cumani and P. Laface, “Memory and computation trade-offs for efficient i-vector extraction,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 934–944, 2013.

[12] P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” in *Technical report CRIM-06/08-13*, 2005.

[13] L. Mirsky, “A trace inequality of John von Neumann,” *Monatshefte fű Mathematik*, vol. 79, no. 4, pp. 303–306, 2000.

[14] R. D. Grigorieff, “A note on von Neumann’s trace inequality,” *Math. Nachr.*, vol. 151, pp. 327–328, 1991.

[15] Available at `http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10\_evalplan.r6.pdf`.

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