Memory-aware i-vector extraction by means of subspace factorization

MEMORY–AWARE I–VECTOR EXTRACTION BY MEANS OF SUB–SPACE FACTORIZATION

Sandro Cumani and Pietro Laface

sandro.cumani, pietro.laface@polito.it - Politecnico di Torino, Italy

ABSTRACT

Most of the state–of–the–art speaker recognition systems use i–vectors, a compact representation of spoken utterances. Since the “standard” i–vector extraction procedure requires large memory structures, we recently presented the Factorized Sub-space Estimation (FSE) approach, an efficient technique that dramatically reduces the memory needs for i–vector extraction, and is also fast and accurate compared to other proposed approaches. FSE is based on the approximation of the matrix $\mathbf{T}$, representing the speaker variability sub–space, by means of the product of appropriately designed matrices. In this work, we introduce and evaluate a further approximation of the matrices that most contribute to the memory costs in the FSE approach, showing that it is possible to obtain comparable system accuracy using less than a half of FSE memory, which corresponds to more than 60 times memory reduction with respect to the standard method of i–vector extraction.

Index Terms— Speaker Recognition, I-vectors, I-vector extraction, Probabilistic Linear Discriminant Analysis, matrix rotation.

1. INTRODUCTION

I–vectors [1], a compact representation of a Gaussian Mixture Model (GMM) supervector [2], in combination with Probabilistic Linear Discriminant Analysis (PLDA) [3, 4], allow speaker recognition systems to reach state–of–the–art performance [5, 6, 7, 8, 9, 10, 11].

Since the “standard” i–vector extraction procedure requires large memory structures and is relatively slow, several approaches have been proposed that are able to obtain either fast approximate solutions, possibly traded for lower memory costs, or accurate solutions based on a Variational Bayes (VB) formulation, at the expense of an increase of the computational load [12, 13, 14, 15, 16]. In [16] we have highlighted that the incidence of the time spent for i-vector computation in a system using large models and scoring long speaker segments is negligible compared to the importance of keeping the original accuracy and reducing memory usage. The effectiveness of the i-vector extractor is more relevant for systems dealing with short utterances [17, 18, 19, 20, 21] such as, for example, the text prompts in speaker verification [22, 23]. Saving memory is important not only for small footprint applications, but also because larger and possibly more precise models can be used. We have recently proposed an approximate i–vector extraction approach [24], the Factorized Sub-space Estimation (FSE). FSE tackles the main memory cost issue in the standard i–vector extraction: the size of the variability sub–space matrix $\mathbf{T}$, and the huge amount of memory devoted to pre–computation of the matrices, which are needed for speeding–up the i–vector computation. In [24] we have shown that our solution not only substantially improves the performance with respect to the fast, but inaccurate, eigen-decomposition approach [12], but also dramatically reduces (approximately by 35 times) the memory needed for i–vector extraction compared to other methods [1], [14, 16], which require storing the original sub–space matrix $\mathbf{T}$.

In this work, we further improve the memory requirements of the i–vector extractor module by an additional approximation of the matrices that most contribute to the FSE memory costs, and we show that it is possible to obtain comparable system accuracy using less than half of FSE memory, which corresponds to approximately 60–85 times memory reduction with respect to the standard i–vector extraction. Our experiments, performed on the extended NIST SRE2010 female tests, allow comparing different i–vector extraction approaches and appreciating the contribution of our proposal in terms of accuracy and memory costs.

The paper is organized as follows: Section 2 summarizes the i–vector representation for speaker recognition, and the standard extraction process. The FSE approach and the key idea for the optimization of its memory costs are introduced in Section 3. Section 4 details the FSE optimization steps, focusing on the factors that require more memory. The optimization of the novel data structures is introduced in Section 5. The experimental results are presented and commented in Section 6, and conclusions are drawn in Section 7.

2. I–VECTOR REPRESENTATION

The i–vector representation [1] constrains the GMM supervector $\mathbf{s}$, representing both the 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. C are the mixture components, and F is the feature dimension. The M columns of $\mathbf{T}$ are vectors spanning the variability space, and $\mathbf{w}$ is a random vector of size M having a standard normal prior distribution. $\mathbf{T}$ is multiplied for convenience by $\mathbf{\Sigma}^{\frac{1}{2}}$, where $\mathbf{\Sigma}$ denotes the block–diagonal matrix whose diagonal blocks contain the UBM covariance matrices $\mathbf{\Sigma}^{(c)}$. It is worth noting that the i–vector representation (1) is equivalent to the classical one, but takes advantage of the UBM statistics whitening introduced in [12] to simplify the i–vector computation.

Given a set of feature vectors $\mathcal{X} = \{\mathbf{x}_1, \dots \mathbf{x}_t \dots \mathbf{x}_T\}$ 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 $\mathbf{L}_\mathcal{X}$ is the precision matrix of the posterior distribution:

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

Computational resources for this work were provided by HPC@POLITO (http://www.hpc.polito.it)

In these equations, $N_{\mathcal{X}}^{(c)}$ 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 $\mathbf{T} = \left(\mathbf{T}^{(1)*}, \ldots, \mathbf{T}^{(C)*}\right)^*$, 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:

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

where $\mathbf{m}^{(c)}$ is the mean of the c–th Gaussian component of the UBM, $\mathbf{x}_t$ is the t–th feature vector in $\mathcal{X}$ and $\gamma_t^{(c)}$ is its occupation probability on the c–th Gaussian.

3. MEMORY AWARE I–VECTOR EXTRACTION

The complexity of a single i–vector computation (2) mainly depends on the computation of $\mathbf{L}_{\mathcal{X}}$ and on its inversion: it is $O(M^3)$ + $O(CFM)$ for (2) plus $O(CFM^2)$ for (3). Usually the number of Gaussian components C is greater than the sub–space dimension M, and the latter is greater that the feature dimension F. Popular settings for state–of–the-art systems are: $F = 60$, $C = 2048$, and $M = 400$. The term $O(CFM^2)$, thus, accounts for most of the computation complexity, whereas the memory demand for storing matrix $\mathbf{T}$ is $O(CFM)$. The standard i–vector extraction approach consists in pre–computing and storing all the factors $\mathbf{T}^{(c)*}\mathbf{T}^{(c)}$. This allows reducing the computational costs to $O(CM^2)$, at the expense of additional $O(CM^2)$ memory.

We have recently proposed an approximate i–vector extraction approach [24] that tackles the main memory cost issues in the standard i–vector extraction: the size of the variability sub–space matrix $\mathbf{T}$, and the size of the set of the $\mathbf{T}^{(c)*}\mathbf{T}^{(c)}$ matrices. In [16] we have shown that the computation of the i–vector covariance matrix $\mathbf{L}_{\mathcal{X}}$ is unnecessary because the solution of (2) can be obtained by means of a Conjugate Gradient approach which requires only matrix– vector multiplications involving the subspace matrix $\mathbf{T}$. In [24] we have further shown that an effective approximation of the rows of the subspace matrix $\mathbf{T}$ can be obtained as a linear combination of the atoms of a common dictionary. This approach, combined with the Conjugate Gradient approach for i–vector extraction, allows obtaining approximate i–vectors without explicitly reconstructing the original $\mathbf{T}$ matrix and without explicitly computing $\mathbf{L}_{\mathcal{X}}$. In particular, a good approximation of the $\mathbf{T}^{(c)}$ matrices can be obtained by means of the decomposition:

$$\hat{\mathbf{T}}^{(c)} = \mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q} \approx \mathbf{T}^{(c)} , \tag{5}$$

where $\mathbf{O}^{(c)}$ is an orthogonal $F \times F$ matrix, $\mathbf{\Pi}^{(c)}$ is a sparse $F \times K$ matrix having at most one non-zero element per row, and $\mathbf{Q}$ is a $K \times M$ dictionary matrix, shared among all $\hat{\mathbf{T}}^{(c)}$, including K atoms in its rows. $\hat{\mathbf{T}}^{(c)}$ is, thus, a linear combination of F scaled and rotated atoms of $\mathbf{Q}$. It is worth noting that the original FSE model does not require $\mathbf{O}^{(c)}$ to have a positive determinant. However, since multiplying by -1 one column of $\mathbf{O}^{(c)}$ and the corresponding row of$\mathbf{\Pi}^{(c)}$ does not change the result of the factorization, without loss of generality we can further impose that the matrices $\mathbf{O}^{(c)}$ are proper rotation matrices. Since the size K of the dictionary that we estimate can be selected according to memory-accuracy trade–offs, and it is usually much less than $C \times F$, our solution is able to preserve the performance with respect to the standard approach, dramatically reducing the memory needed for i–vector extraction compared to other methods [14, 16], which require storing the original sub–space matrix $\mathbf{T}$.

Analyzing the memory cost of the matrices involved in the FSE model, with typical settings for the dimensions of the UBM, of the i–vectors, and of the dictionary, it can be easily verified that most of the memory in the FSE implementation is devoted to the rotation matrices $\mathbf{O}^{(c)}$. In this work we propose to approximate each $F \times F$ matrix $\mathbf{O}^{(c)}$ by means of a sequence of J rotations, where, for each index j, the rotation planes are shared among the set of matrices $\mathbf{O}_j^{(c)}$. In particular, we approximate each matrix $\mathbf{O}^{(c)}$ as [25]:

$$\mathbf{O}^{(c)} \approx \prod_j \mathbf{U}_j \mathbf{B}_j^{(c)} \mathbf{U}_j^{-1} \tag{6}$$

where each matrix $\mathbf{U}_j$, shared among all $\mathbf{O}^{(c)}$'s, is an orthogonal rotation matrix, which identifies a set of shared rotation planes, and each $\mathbf{B}_j^{(c)}$ is a block–diagonal matrix obtained as the direct sum of Givens rotations:

$$\mathbf{B}_j^{(c)} = \begin{bmatrix} \mathbf{G}_{j,1}^{c} & & \\ & \ddots & \\ & & \mathbf{G}_{j,\lceil\frac{F}{2}\rceil} \end{bmatrix} , \tag{7}$$

where

$$\mathbf{G}_{j,i}^{(c)} = \begin{bmatrix} \cos(\theta_{j,i}^{(c)}) & -\sin(\theta_{j,i}^{(c)}) \\ \sin(\theta_{j,i}^{(c)}) & \cos(\theta_{j,i}^{(c)}) \end{bmatrix} , i = 1, \ldots, \frac{F}{2} ,$$

and $\mathbf{G}_{j,\lceil\frac{F}{2}\rceil} = 1$ if F is odd. Defining $\mathbf{A}_0 = \mathbf{U}_1$, $\mathbf{A}_J = \mathbf{U}_J^{-1}$ and $\mathbf{A}_i = \mathbf{U}_i^{-1}\mathbf{U}_{i+1}, i = 1, \ldots J-1$, expression (6) can be rewritten as:

$$\mathbf{O}^{(c)} \approx \tilde{\mathbf{O}}^{(c)} = \mathbf{A}_0 \prod_j \mathbf{B}_j^{(c)} \mathbf{A}_j . \tag{8}$$

It is worth noting that each matrix $\mathbf{A}_i$ is orthogonal. Although the terms $\mathbf{A}_i$ are not independent, in the following they will be estimated independently, with the only constraint that each matrix $\mathbf{A}_i$ is a proper rotation matrix.

The factorization (8) allows reducing the memory costs for storing the matrices $\mathbf{O}^{(c)}$ from $O(CF^2)$ to $O(CJF/2)$ for the parameters of the block–diagonal matrices, plus $O(JF^2)$ for the shared matrices. It is worth noting that the FSE i–vector extractor requires the matrices $\mathbf{O}^{(c)}$ only for computing the term $\hat{\mathbf{T}}^{(c)*}\mathbf{f}_{\mathcal{X}}^{(c)} = \mathbf{Q}^*\mathbf{\Pi}^{(c)*}\mathbf{A}_0 \prod_j \left(\mathbf{B}_j^{(c)}\mathbf{A}_j\right) \mathbf{f}_{\mathcal{X}}^{(c)}$, which can be computed without explicitly reconstructing the approximated $\tilde{\mathbf{O}}^{(c)*}$, by means of matrix by vector multiplications.

4. FACTORIZED SUB–SPACE ESTIMATION OF $\mathbf{T}^{(c)}$

The matrices $\mathbf{O}^{(c)}$, $\mathbf{\Pi}^{(c)}$, and $\mathbf{Q}$ in (5) are obtained by minimizing a weighted average square norm of the difference between each $\mathbf{T}^{(c)}$ and its approximation $\hat{\mathbf{T}}^{(c)}$. In particular, if $\omega^{(c)}$ is the weight of the c–th component of the UBM, the objective function is:

$$\min_{\{\mathbf{O}^{(c)}\},\{\mathbf{\Pi}^{(c)}\},\mathbf{Q}} \sum_c \omega^{(c)} \left\|\mathbf{T}^{(c)} - \mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right\|^2 . \tag{9}$$

where matrices $\mathbf{O}^{(c)}$ are constrained to be orthogonal, and matrices $\mathbf{\Pi}^{(c)}$ are constrained to have at most one non–zero element per row.

The optimization is performed by updating a matrix while keeping constant the others, according to the iterative sequence of optimizations illustrated in [24]. We here recall only the optimization procedure for $\mathbf{O}^{(c)}$, because we use similar considerations for the optimization of the FSE objective function with respect to the terms of the factorization given in (8).

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

The minimization of (9) with respect to $\mathbf{O}^{(c)}$ is equivalent to the maximization:

$$\max_{\{\mathbf{O}^{(c)}\}} \sum_c \omega_c \operatorname{tr}\left(\mathbf{T}^{(c)*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right) \ . \tag{10}$$

Since the optimization of each matrix $\mathbf{O}^{(c)}$ can be done independently, and recalling that the trace operator is invariant under cyclic permutations, each $\mathbf{O}^{(c)}$ can be estimated as the maximizer of:

$$\max_{\mathbf{O}^{(c)}} \operatorname{tr}\left(\mathbf{T}^{(c)*}\mathbf{O}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q}\right) = \max_{\mathbf{O}^{(c)}} \operatorname{tr}\left(\mathbf{O}^{(c)}\mathbf{Z}^{(c)}\right) \ . \tag{11}$$

where $\mathbf{Z}^{(c)} = \mathbf{\Pi}^{(c)}\mathbf{Q}\mathbf{T}^{(c)*}$.

The Von Neumann's trace inequality [26, 27] states that:

$$\left|\operatorname{tr}(\mathbf{O}^{(c)}\mathbf{Z}^{(c)})\right| \leq \sum_{i=1}^{F} \sigma_{oi}\sigma_{zi} \ , \tag{12}$$

where σ_{oi} and σ_{zi} are the sorted singular values of $\mathbf{O}^{(c)}$ and $\mathbf{Z}^{(c)}$, respectively. Since $\mathbf{O}^{(c)}$ has to be orthogonal, its singular values must be equal to 1. Thus, for any feasible solution $\mathbf{O}^{(c)}$, the objective function is bounded by:

$$\operatorname{tr}(\mathbf{O}^{(c)}\mathbf{Z}^{(c)}) \leq \sum_{i=1}^{F} \sigma_{zi} \ , \tag{13}$$

which is maximized if we can find a matrix $\mathbf{O}^{(c)}$ such that the equality holds. This condition is satisfied by the matrix:

$$\mathbf{O}^{(c)} = \mathbf{V_Z}\mathbf{U_Z^*} \ , \tag{14}$$

where $\mathbf{V_Z}$ and $\mathbf{U_Z}$ are the singular vectors of the Singular Value Decomposition of $\mathbf{Z}^{(c)} = \mathbf{\Pi}^{(c)}\mathbf{Q}\mathbf{T}^{(c)*} = \mathbf{U_Z}\mathbf{\Sigma_Z}\mathbf{V_Z^*}$. It is worth noting that this approach does not guarantee that $\mathbf{O}^{(c)}$ is a proper rotation matrix. However, as long as $\mathbf{Z}^{(c)}$ is not singular, $\det(\mathbf{O}^{(c)})$ and $\det(\mathbf{Z}^{(c)})$ have the same sign. Thus, if $\det(\mathbf{O}^{(c)}) = -1$, multiplying by -1 a column of $\mathbf{O}^{(c)}$ and the corresponding row of $\mathbf{\Pi}^{(c)}$ provides an equivalent solution with $\det(\mathbf{O}^{(c)}) = 1$, i.e., $\mathbf{O}^{(c)}$ becomes a proper rotation matrix. In practice, a proper rotation matrix can always been obtained, by simply pre–multiplying by -1 one row of matrix $\mathbf{\Pi}^{(c)}$, if necessary, so that $\mathbf{Z}^{(c)}$ has a positive determinant before the optimization step.

5. APPROXIMATION OF THE $\mathbf{O}^{(c)}$ MATRICES

In order to optimize the FSE objective function with respect to the terms of the factorizations (8) we adopt a modified coordinate descent strategy. In particular, we iteratively optimize the FSE objective function with respect to a single matrix $\mathbf{A}_i$ or $\mathbf{B}_i^{(c)}$, while keeping all the other terms fixed. For each of these matrices, the steps are similar to the ones performed in Section 4.1 for optimizing the matrices $\tilde{\mathbf{O}}^{(c)}$. We replace $\mathbf{O}^{(c)}$ by its approximation $\tilde{\mathbf{O}}^{(c)}$ in the original objective function, obtaining:

$$\sum_c \omega^{(c)} \left\| \mathbf{T}^{(c)} - \tilde{\mathbf{O}}^{(c)}\mathbf{\Pi}^{(c)}\mathbf{Q} \right\|^2 . \tag{15}$$

Since the matrices $\tilde{\mathbf{O}}^{(c)}$ are orthogonal, the minimization of (15) corresponds to the maximization of

$$\sum_c \omega_c \operatorname{tr}\left(\tilde{\mathbf{O}}^{(c)}\mathbf{Z}^{(c)}\right) = \sum_c \omega_c \operatorname{tr}\left(\mathbf{A}_0\mathbf{B}_1^{(c)}\mathbf{A}_1\mathbf{B}_2^{(c)}\cdots\mathbf{B}_J^{(c)}\mathbf{A}_J\mathbf{Z}^{(c)}\right) \ . \tag{16}$$

Using again the properties of the trace operator, the optimization of (16) with respect to one shared matrix $\mathbf{A}_j$ can be rewritten as:

$$\begin{aligned} &\max_{\mathbf{A}_j} \sum_c \omega_c \operatorname{tr}\left(\mathbf{A}_j \prod_{i>j}(\mathbf{B}_i\mathbf{A}_i)\,\mathbf{Z}^{(c)} \prod_{i<j}(\mathbf{A}_{i-1}\mathbf{B}_i)\right) = \\ &\max_{\mathbf{A}_j} \sum_c \omega_c \operatorname{tr}\left(\mathbf{A}_j\mathbf{P}_j^{(c)}\right) \ , \end{aligned} \tag{17}$$

where $\mathbf{P}_j^{(c)}$ collects all the factors of the trace, except $\mathbf{A}_j$. Since trace is a linear operator, (17) can be rewritten as:

$$\begin{aligned} \max_{\mathbf{A}_j} \sum_c \omega_c \operatorname{tr}\left(\mathbf{A}_j\mathbf{P}_j^{(c)}\right) &= \max_{\mathbf{A}_j} \operatorname{tr}\left(\mathbf{A}_j \sum_c \omega_c\mathbf{P}_j^{(c)}\right) \\ &= \max_{\mathbf{A}_j} \operatorname{tr}\left(\mathbf{A}_j\mathbf{P}_j\right) \ . \end{aligned} \tag{18}$$

Since (18) has the same form of (11), the optimization of $\mathbf{A}_j$ can be performed in analogy with the solution given by (14) for the optimization of $\mathbf{O}^{(c)}$. In order to optimize (16) with respect to $\mathbf{B}_j^{(c)}$, (16) can be rewritten as:

$$\begin{aligned} &\max_{\mathbf{B}_j^{(c)}} \sum_c \omega_c \operatorname{tr}\left(\mathbf{B}_j^{(c)} \prod_{i>j}(\mathbf{A}_{i-1}\mathbf{B}_i)\,\mathbf{A}_J\mathbf{Z}^{(c)}\mathbf{A}_0 \prod_{i<j}(\mathbf{B}_i\mathbf{A}_i)\right) = \\ &\max_{\mathbf{B}_j^{(c)}} \sum_c \operatorname{tr}\left(\mathbf{B}_j^{(c)}\mathbf{P}_j^{(c)}\right) \ , \end{aligned} \tag{19}$$

where $\mathbf{P}_j^{(c)}$ collects all the factors of the trace, except $\mathbf{B}_j^{(c)}$.
Since each term in the sum involves a single component c, every matrix $\mathbf{B}_j^{(c)}$ can be independently estimated as the maximizer of:

$$\max_{\mathbf{B}_j^{(c)}} \operatorname{tr}\left(\mathbf{B}_j^{(c)} \sum_c \mathbf{P}_j^{(c)}\right) \ , \tag{20}$$

but $\mathbf{B}_j^{(c)}$ is a block–diagonal matrix, thus:

$$\operatorname{tr}\left(\mathbf{B}_j^{(c)}\mathbf{P}_j^{(c)}\right) = \sum_{i=1}^{F/2} \operatorname{tr}\left(\mathbf{G}_{j,i}^{(c)}\mathbf{P}_{j,i}^{(c)}\right) \ , \tag{21}$$

where $\mathbf{G}_{j,i}^{(c)}$ and $\mathbf{P}_{j,i}^{(c)}$ are the i–th 2×2 blocks of $\mathbf{B}_j^{(c)}$ and $\mathbf{P}_j^{(c)}$, respectively. Each term of the sum, corresponding to the product of 2×2 block–diagonal matrices, can be independently estimated as:

$$\mathbf{G}_{j,i}^{(c)} = \arg\max_{\mathbf{G}} \operatorname{tr}\left(\mathbf{G}\mathbf{P}_{j,i}^{(c)}\right) \ , \tag{22}$$

which has again the same structure of (11). Therefore, it can be solved using the same approach.

Although not interesting for memory reduction, it is worth not-

Table 1: Results of PLDA models for the common condition 5 of the NIST SRE2010 female extended tests in terms of % EER, minDCF08×1000 and minDCF10×1000, memory occupation, and memory saving ratio, for standard i–vector extraction, FSE, and FSE with approximated $\mathbf{O}^{(c)}$ matrices. Standard Fast and Slow refer to classical i–vector extraction with or without pre–computation of the terms $\mathbf{T}^{(c)*}\mathbf{T}^{(c)}$, respectively. Label FSE–K refers to the FSE approach setting the dictionary dimension to K.

Method	Shared matrices	% EER	min DCF08	min DCF10	Memory (Mb)	Saving Ratio wrt Fast	Saving Ratio wrt FSE	Average time per utterance (ms)
Standard Fast	-	3.5	167	547	767.2	-	-	109
Standard Slow	-	3.5	167	547	140.6	-	-	1250
FSE–3.5K	-	3.6	172	514	21.9	35.1	-	23
FSE–3.5K	15	3.8	180	545	8.9	86.5	2.5	78
FSE–3.5K	20	3.7	181	531	9.8	78.2	2.2	100
FSE–5K	-	3.5	171	522	24.2	31.7	-	28
FSE–5K	15	3.6	178	537	11.2	68.7	2.2	83
FSE–5K	20	3.6	175	549	12.1	63.4	2.0	104

ing that by increasing the number of shared matrices it is possible to obtain a perfect decomposition of any set of orthogonal matrix. This claim is supported by the findings in [28, 29, 30], where Givens reduction is used for QR factorization. From these works it follows that any $n \times n$ rotation matrix ($n > 1$) can be written as the product of $\frac{n(n-1)}{2}$ elementary Givens rotations, which can be arranged as a sequence of at most $2n - 3$ independent rotations (i.e., rotations taking place on orthogonal rotation planes). Since each of such rotations can be represented as a permutation of a direct sum of Givens rotations, any $n \times n$ orthogonal matrix can be represented as in (6) where the matrices $\mathbf{U}_j$ are fixed (and correspond to the permutations of the independent Givens rotations), using at most $2n - 3$ factors.

6. EXPERIMENTS: SETTINGS AND RESULTS

A gender–independent i–vector extractor was trained based on a 2048–component diagonal covariance gender–independent UBM, and on a gender-independent $\mathbf{T}$ matrix, both trained using NIST SRE 04/05/06 datasets. The acoustic features are 45–dimensional Mel frequency cepstral coefficients. In particular, we extracted, every 10 ms, 18 MFCCs and the frame log–energy on a 25 ms sliding Hamming window. This 19–dimensional feature vector was subjected to short time mean and variance normalization using a 3 s sliding window, and a 45-dimensional feature vector was obtained by appending the delta and the first 7 double delta coefficients computed on a 5 frame window. The i-vector dimension was fixed to $d = 400$. Gender–dependent PLDA models were trained, with full–rank channel factors, and 200 dimensions for the speaker factors, using the NIST SRE datasets, and additionally the Switchboard II, Phases 2 and 3, and Switchboard Cellular, Parts 1 and 2 datasets. The i–vectors of the PLDA models were whitened and L_2 normalized after Within Class Covariance Normalization (WCCN) [31] has been applied. The WCCN transformations were trained on the same datasets used for training the PLDA models.

Our classifier for these experiments is based on Gaussian PLDA, implemented according to the framework illustrated in [6]. The presented scores are not normalized. FSE was trained according to the original procedure [24], and the approximations of the orthogonal matrices $\mathbf{O}^{(c)}$ was introduced only in the last iteration. Table 1 summarizes the performance of the evaluated approaches on the female part of the extended telephone condition (common condition 5) in the NIST 2010 evaluation. The recognition accuracy is given in terms of percent Equal Error Rate (EER) and Minimum Detection Cost Functions (×1000) defined by NIST for the 2008 (minDCF08) and 2010 (minDCF10) evaluations [32]. Table 1 also shows the memory occupation, and memory saving ratio, for standard i–vectors, FSE with dictionary dimensions $K = 3500$ and $K = 5000$, and FSE with approximated $\mathbf{O}^{(c)}$ matrices. FSE, using the settings $F = 45$, $C = 2048$, $M = 400$ and $K = 3500$ reduces the memory cost of i–vector extraction by approximately 35 times compared to the standard approach, with comparable accuracy. The i–vectors, obtained by the Conjugate Gradient procedure illustrated in [24], are computed faster than the standard method. Increasing the number of dictionary entries to 5000 gives not statistically significant improvement on these tests. As far as the approximated $\mathbf{O}^{(c)}$ approach is concerned, using $K = 3500$ dictionary entries and $J = 15$ shared matrices, we save about 2.5 times memory, with only a small performance degradation (3–7% relative) with respect to the corresponding FSE–3.5K models, whereas increasing the number of shared matrices to $J = 20$, reduces the performance degradation to less than 5% relative, with a memory reduction of 2.3 times. Increasing the accuracy of the approximated $\hat{\mathbf{T}}^{(c)}$, by increasing the number of dictionary entries to $K = 5000$ and the number of shared matrices to $K = 20$, we get EER and DCF10 results that are similar to the ones of the standard i–vector extraction, and only a 4.5% relative increase in DCF08, while the memory requirements are reduced 60 times with respect to the corresponding standard models. The memory size of the models is extremely small, and thus suitable for embedded systems. Using 20 shared matrices has of course a computational overhead: the i–vector extraction time with respect to plain FSE increases as shown in Table 1, but it remains of the order of magnitude of the standard PLDA technique.

7. CONCLUSIONS

An approximation of the orthogonal matrices that most contribute to the memory costs of the Factorized Sub-space Estimation approach has been presented, based on a sequence of rotations performed on a small set of shared planes. It has been shown that in spite of these additional approximations, which allow a dramatic reduction of the memory needed for i–vector extraction, the accuracy of the recognizer is not harmed. Larger and more precise models can be thus easily stored in low–memory devices. Future work will be devoted to alternative approaches for the estimation of these approximations.

8. REFERENCES

[1] 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.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian Mixture Models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 31–44, 2000.

[3] 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.

[4] 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`.

[5] P. Matějka, O. Glembek, F. Castaldo, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, “Full-covariance ubm and heavy-tailed plda in i–vector speaker verification,” in *Proceedings of ICASSP 2011*, pp. 4828–4831, 2011.

[6] N. Brümmer and E. de Villiers, “The speaker partitioning problem,” in *Proc. Odyssey 2010*, pp. 194–201, 2010.

[7] M. Senoussaoui, P. Kenny, N. Brummer, and P. Dumouchel, “Mixture of PLDA models in i-vector space for gender–independent speaker recognition,” in *Proceedings of INTERSPEECH 2011*, pp. 25–28, 2011.

[8] J. Villalba and N. Brümmer, “Towards fully bayesian speaker recognition: Integrating out the between-speaker covariance,” in *Proceedings of INTERSPEECH 2011*, pp. 505–508, 2011.

[9] T. Hasan and J.H.L.Hansen, “Acoustic factor analysis for robust speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 842–853, 2013.

[10] V. Hautamaki, T. Kinnunen, F. Sedlak, K. Lee, B. Ma, and H. Li, “Sparse classifier fusion for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1622–1631, 2013.

[11] B. Srinivasan, L. Yuancheng, D. Garcia-Romero, D. Zotkin, and R. Duraiswami, “A symmetric kernel partial least squares framework for speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1415–1423, 2013.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] A. Larcher, P. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J. Bonastre, “I-vectors in the context of phonetically-constrained short utterances for speaker verification,” in *Proceedings of ICASSP 2012*, pp. 4773–4776, 2012.

[18] S. Cumani, O. Plchot, and P. Laface, “Probabilistic Linear Discriminant Analysis of i–vector posterior distributions,” in *Proceedings of ICASSP 2013*, pp. 7644–7648, 2011.

[19] P. Kenny, T. Stafylakis, P. Ouellet, M. Alam, and P. Dumouchel, “PLDA for speaker verification with utterances of arbitrary duration,” in *Proceedings of ICASSP 2013*, pp. 7649–7653, 2013.

[20] B. J. Borgstrom and A. McCree, “Discriminatively trained bayesian speaker comparison of i-vectors,” in *Proceedings of ICASSP 2012*, pp. 7659–7662, 2013.

[21] S. Cumani, O. Plchot, and P. Laface, “On the use of i-vector posterior distributions in probabilistic linear discriminant analysis,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 846–857, 2014.

[22] A. Aronowitz, “Text dependent speaker verification using a small development set,” in *Proceedings of Odyssey 2012*, pp. 312–316, 2012.

[23] A. Larcher, K. A. Lee, B. Ma, and H. Li, “Phonetically-constrained plda modeling for text-dependent speaker verification with multiple short utterances,” in *Proceedings of ICASSP 2013*, pp. 7673–7677, 2012.

[24] S. Cumani and P. Laface, “Factorized sub-space estimation for fast and memory effective i-vector extraction,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 248–259, 2013.

[25] D. C. Youla, “A normal form for a matrix under the unitary congruence group,” *Canad. J. Math.*, vol. 13, pp. 694–704, 1961.

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

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

[28] W. M. Gentleman, “Error analysis of QR decomposition by Givens transformations,” *Linear Algebra and Appl.*, vol. 10, pp. 189–197, 1975.

[29] A. H. Sameh and D. J.Kuck, “On stable parallel linear system solvers,” *J. ACM*, vol. 25, no. 1, pp. 81–91, 1978.

[30] J. J. Modi and M. R. B. Clarke, “An alternative Givens ordering,” *Numer. Math.*, vol. 43, pp. 83–90, 1984.

[31] A. Hatch, S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for SVM-based speaker recognition,” in *Proceedings of ICSLP 2006*, pp. 1471–1474, 2006.

[32] “The NIST year 2010 speaker recognition evaluation plan.” Available at `http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf`.