EFFICIENT APPROXIMATED I-VECTOR EXTRACTION

Hagai Aronowitz, Oren Barkan

IBM Research – Haifa, Israel

hagaia@il.ibm.com, orenba@il.ibm.com

ABSTRACT

I-vectors are currently widely used by state-of-the-art speech processing systems for tasks such as speaker verification and language identification. A shortcoming of i-vector-based systems is that the i-vector extraction process is computationally expensive. In this paper we propose an efficient method to extract i-vectors approximately. The method normalizes the GMM counts to be similar across sessions. We validate our method empirically for the speaker verification task on five different datasets, both text independent and text dependent. A significant speedup was obtained with a very small degradation in accuracy compared to the standard exact method.

Index Terms— efficient speaker recognition, i-vectors, approximated i-vectors extraction

1. INTRODUCTION

Recently, i-vector-based systems have become popular for speech processing systems such as speaker recognition [1] and language identification [2]. I-vectors provide a way to map audio sessions into a low-dimensional feature vector while retaining most of the relevant information.

The computational resources needed for estimating i-vectors in recognition-time are not negligible and are significantly larger than the corresponding resources required for the Nuisance Attribute Projection (NAP) algorithm which has been shown to provide accurate results [3] with a relatively low computation complexity. Joint Factor Analysis (JFA) [4] which is another state-of-the-art method for speaker recognition is originally more computationally complex than the i-vector approach. However, lately [5] a method named JFAlight was introduced and has managed to significantly reduce the time complexity in recognition-time with the cost of a very small degradation in accuracy. Therefore, the i-vector approach is no longer computationally efficient compared to JFA.

In [6] two novel simplifications were introduced for i-vector extraction. Both simplifications managed to speed up runtime by factors of 10-25 but in the expense of a significant degradation in accuracy (at least 17% in EER and 16% in DCF_{new}).

In this paper we propose to improve on the works reported in [5] and [6]. Using our method we manage to maintain the speedup factor obtained in [6] while keeping the accuracy with only a small degradation.

This paper is organized as follows: Section 2 reviews related background for i-vector extraction. Section 3 introduces the proposed method for i-vector extraction. Section 4 describes the experimental setup. Section 5 reports the results. Finally, Section 6 concludes the paper.

2. I-VECTOR BASED SPEECH PROCESSING

In the i-vector framework, a session is first represented by its zero and first order statistics under a GMM-UBM (Gaussian mixture model - universal background model) framework. The basic assumption is that a speaker and channel dependent supervector of stacked GMM means denoted by s can be modeled as:

$$s = m + \mathrm{T}w \tag{1}$$

where m is the UBM supervector, T is a low-rank matrix of bases spanning a subspace covering most of the variability in the supervector space, and $\boldsymbol{w}$ is an M-dimensional vector having a standard normal distribution. For each session, the i-vector is the MAP point estimate of the latent variable $\boldsymbol{w}$.

The UBM and the hyper-parameter T are estimated from a development data in a process described in [7]. We define the following terms: C is the GMM order, K is the feature vector size, Σ_c is the covariance matrix for Gaussian c, Σ is a $CK \times CK$ diagonal matrix, whose diagonal blocks are Σ_c.

For the zero and first order statistics we define the following definitions: N_c is the data count for Gaussian c for a particular session, N is a $CK \times CK$ diagonal matrix, whose diagonal blocks are $N_c\mathrm{I}_K$ (I_K is $K \times K$ identity matrix), and F is a $CK \times 1$ vector, obtained by concatenating the first order GMM statistics for a particular session.

2.1. Exact i-vector extraction

For a given session X with zero order statistics N and first order statistics F, the MAP estimate for $\boldsymbol{w}$ is given by [1]

$$w_{MAP} = \mathrm{L}^{-1}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}F \tag{2}$$

with L defined as

$$\mathrm{L} = \mathrm{I} + \mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}N\mathrm{T} \tag{3}$$

The computational complexity of calculating the w_{MAP} is $O(CKM + CM^2 + M^3)$ and is dominated by the complexity of computing the value of L which is $O(CM^2)$.

2.2. Efficient approximated i-vector extraction: Related work

In [6], two simplification methods were proposed for speeding up i-vector extraction. The most successful method named *i-vector*

extractor orthogonalization was able to obtain a speedup factor of 25 with the cost of a 17% relative degradation in EER and a 16% relative degradation in DCF. In this paper we do not use this method.

The other method named *constant GMM component alignment* takes the following approximation:

$$\begin{aligned} \mathrm{L} &= \mathrm{I} + N_{tot}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\frac{N}{N_{tot}}\mathrm{T} \\ &= \mathrm{I} + N_{tot}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}W\mathrm{T} \\ &\approx \mathrm{I} + N_{tot}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{W}\mathrm{T} \end{aligned} \quad (4)$$

with N_{tot} defined as the total data count (length) of session X, W is defined as a $CK \times CK$ diagonal matrix, and contains the ML estimates for the GMM weights (zero order statistics divided by the total length). $\overline{W}$ is an estimate of the expected value of W over the entire dataset. In [6] $\overline{W}$ is obtained by taking the UBM weights. In our implementation we estimate $\overline{W}$ by averaging W over the dev set. The outcome of using Eq. (4) reduces the time complexity of estimating L from $O(CM^2)$ to $O(M^2)$. The time complexity of the whole[1] extraction process reduces to $O(CKM+M^3)$ which is reported to give a speedup factor of 14 with the cost of a 24% relative degradation in EER and a 29% relative degradation in DCF. We denote this approximation method with w_{CGCA}.

In [5] a method quite similar to *constant GMM component alignment* named JFAlight was introduced for the sake of efficient approximated JFA factors estimation. The approximation taken was:

$$\mathrm{L} \approx \mathrm{I} + \mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{N}\mathrm{T} \quad (5)$$

where $\overline{N}$ is an estimate of the expected value of N over the entire dataset and was in practice estimated by averaging N over the dev set. Note that the approximation in Eq. (5) is less accurate than the approximation in Eq. (4) but enables the computation and inversion of L offline, which in the i-vector framework can reduce the recognition-time computation to

$$w_{JFAlight} = \mathbf{A}F \quad (6)$$

where $\mathbf{A}$ is a $M \times CK$ matrix defined as:

$$\mathbf{A} = \mathrm{L}^{-1}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1} \quad (7)$$

with L taken from the approximation in Eq. (5). The outcome of using this method for JFA resulted in a speedup factor of 100 with a relative degradation of up to 5% in accuracy (depending on the dataset). In the context of i-vector extraction, the time complexity of the whole extraction process reduces to $O(CKM)$.

2.3. Speaker recognition in i-vector space

Extracted i-vectors (400 dimensioal) are length-normalized [8] in order to become more Gaussian. We then follow the approach described in [9]. We apply LDA (Linear Discriminant Analysis) to reduce the dimensionality to 250, and then use WCCN (within class covariance normalization). We use cosine-based similarity scoring and normalize using ZT-norm which we found to be slightly superior to s-norm.

3. PROPOSED METHOD

Our starting point is the two methods described in subsection 2.2. As we report in section 5 (and as reported in [6]), these methods cause a significant degradation in accuracy. We propose several modifications to these methods aimed at improving the obtained accuracy.

We begin by rewriting the MAP estimate for the i-vector $\boldsymbol{w}$ as follows:

$$w_{MAP} = \left(\mathrm{I} + \mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}N\mathrm{T}\right)^{-1}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}Ns \quad (8)$$

with s defined as the maximum likelihood (ML) estimate for the GMM supervector associated with the session: $s = N^{-1}F$. The formulation in Eq. (8) implies that if the zero order statistics are approximated in the computation of L, they should be approximated in the same manner in the term Ns. Therefore, we propose to modify the *constant GMM component alignment* method described in Eq. (4) to compute w_1

$$w_1 = \left(\mathrm{I} + N_{tot}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{W}\mathrm{T}\right)^{-1}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{W}W^{-1}F \quad (9)$$

and we further modify the approximation described in Eqs. (5-7) to compute w_2

$$w_2 = \mathbf{A}_2 N^{-1}F \quad (10)$$

with $\mathbf{A}_2 = \left(\mathrm{I} + \mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{N}\mathrm{T}\right)^{-1}\mathrm{T}^{t}\boldsymbol{\Sigma}^{-1}\overline{N}$

3.1. Gender dependency

In our experiments we have observed that it is best to compute the average Gaussian occupancy matrix $\overline{N}$ (or the average Gaussian weights matrix $\overline{W}$) separately for males and females. In recognition time, we do not assume that the gender is known a priori. Instead for a given session we automatically select $\overline{N}$ (or $\overline{W}$) by comparing the session dependent statistics (N or W) to the corresponding gender dependent statistics and choosing the most similar. The distance measure we used was the Euclidean distance.

3.2. Task dependency

In our experiments we have observed that it is important to estimate the average statistics ($\overline{N}$ or $\overline{W}$) from a dataset that matches the characteristics of the test data. Otherwise, we have observed a significant degradation in accuracy.

[1] Excluding the calculation of GMM statistics

3.3. Score normalization

The approximation methods we investigate in this paper are expected to be applied in recognition time. In most of our experiments we assume that the enrollment sessions are processed without approximations. In order to avoid a mismatch in the normalization sessions, we apply the approximations on the Z-norm sessions (but not on the T-norm sessions).

3.4. LDA and WCCN retraining

The results we report are with LDA and WCCN trained on i-vectors extracted with the exact method. However, we observed that retraining of LDA and WCCN on approximated i-vectors reduces accuracy degradation. We do not choose to use this approach in general because training LDA and WCCN requires large amounts of data which we don't have when we are working on non-NIST tasks and according to subsection 3.2, estimating the average statistics ($\overline{N}$ or $\overline{W}$) from matched development sets is essential.

3.5. Soft approximation

In order to cope with a possible degradation in accuracy we propose a flexible tradeoff between accuracy and speed. We exemplify our method on the $w_{JFAlight}$ approximation method (Eq. 5-7). We propose the following modified approximation:

$$w_{Soft\ JFAlight} = L_{soft}^{-1}\mathrm{T}^t\boldsymbol{\Sigma}^{-1}F \tag{11}$$

$$\mathrm{L}_{\mathrm{soft}} \approx \mathrm{I} + \mathrm{T}^t\boldsymbol{\Sigma}^{-1}\overline{N}\mathrm{T} + \sum_{c\in\Delta}\left(N_c - \overline{N}_c\right)\left(\mathrm{T}_c^t\boldsymbol{\Sigma}^{-1}\mathrm{T}_c\right) \tag{12}$$

where T_c is a K×M sub-matrix of T corresponding to the c mixture component such that $\mathrm{T} = \left(T_1^t \cdots T_C^t\right)^t$, and Δ is a set of Gaussian components for which we use the exact calculation instead of the approximation. Note that if Δ is empty than Eq. (11) is equal to $w_{JFAlight}$, and if Δ consists of all the Gaussian components, Eq. (11) reduces to the exact method. We can therefore set through Δ a trade-off between accuracy and speed. The time complexity of computing $w_{Soft\ JFAlight}$ is $O(CKM+|\Delta|M^2+M^3)$. We define Δ to be session dependent by computing for each Gaussian component the session dependent expected approximation error E_c and selecting the top R percentile.

$$\mathrm{E}_c \approx \left|N_c - \overline{N}_c\right|\left\|\mathrm{T}_c^t\boldsymbol{\Sigma}^{-1}\mathrm{T}_c\right\| \tag{13}$$

4. EXPERIMENTAL SETUP

4.1. Datasets

We trained the UBM and matrix T on 12,711 sessions from Switchboard-II, NIST 2004 speaker recognition evaluation (SRE) and NIST-2006-SRE. We used a subset of these sessions for ZT-score normalization for our NIST 2008 experiments.

We ran experiments on five datasets. The first dataset is a subset of the NIST-2008-SRE (the short2-short3 condition). We limited our experiments on telephone trials only. The male experiments consist of 5,838 target trials and 435,142 impostor trials. The female experiments consist of 11,312 target trials and 1,231,732 impostor trials.

In order to assess the accuracy of the approximation methods on short sessions we conducted experiments on four other datasets which were collected by the Wells Fargo Bank within the framework of a proof of technology (POT) [10]. The datasets consist of 750 recorded WF employees. Each dataset is partitioned into a development dataset (200 speakers) and an evaluation dataset (550 speakers). Each speaker has 2 sessions using a landline phone and 2 sessions using a cellular phone. The data collection was accomplished over a period of 4 weeks.

Four different authentication conditions were defined for the WF POT. In the first authentication condition named *global*, a common text is used for both enrollment and verification. In the second condition named *speaker* a user (speaker) dependent password (assumed to be known to the imposters) is used for both enrollment and verification. The third condition named *prompted* is a condition in which during the verification stage the user is instructed to speak a prompted text. Enrollment for the *prompted* condition uses speech corresponding to text different than the prompted verification text. Finally, in the *text independent* condition the user is enrolled by reading a fixed text (shared among all speakers) and verified by saying utterances such as user's full name, user's work phone number, user's zip code, etc.

The WF POT experiments consist of approximately 6,500 target trials and 75,000 impostor trials per authentication condition. About 75% of the target trials are channel mismatched (landline vs. cellular) and 25% of the target trials are channel matched. The WF POT development data (800 sessions) is used for ZT-score normalization.

4.2 Front-end

The front-end is based on Mel-frequency cepstral coefficients (MFCC). An energy based voice activity detector is used to locate and remove non-speech frames. The final feature set consists of 12 cepstral coefficients augmented by 12 delta and 12 delta-delta cepstral coefficients extracted every 10ms using a 32ms window. Feature warping is applied with a 300 frame window. We use a GMM order of 1024.

5. RESULTS

Table 1 presents a comparison of the exact i-vector extraction method to the w_{CGCA}, $w_{JFAlight}$, w_1 and w_2 approximation methods on the NIST-2008 dataset. Error rates for males and females are averaged. For w_{CGCA} and $w_{JFAlight}$ we observe a significant degradation in accuracy of ~20% in EER and 12% in minDCF. For the proposed w_1 and w_2 methods we see no significant degradation in accuracy. We choose to use w_2 which obtained the best performance for the rest of our experiments.

Table 1. *A comparison of the proposed approximation methods on the NIST-2008 dataset.*

Dataset	EER in (%)	minDCF (× 10^5)	Runtime (in sec)	Speedup factor
Exact	1.80	848	0.76	-
w_{CGCA} [6]	2.12	949	0.07	11[2]
$w_{JFAlight}$ [5]	2.16	941	0.03	25
w_1	1.78	873	0.07	11
w_2	1.79	874	0.03	25

Table 2 presents results for the w_2 approximation method compared to exact i-vector extraction for the four WF conditions. We can see that for the *global*, *speaker* and *prompted* conditions we observe no significant degradation due to using the w_2 approximation. However, for the TI condition we do observe a significant degradation of 10%.

Table 2. *A comparison of the w_2 approximation method to the exact method for the WF datasets.*

Dataset	Exact EER in (%)	w_2 EER in (%)
WF *global*	3.04	3.07
WF *speaker*	3.60	3.65
WF *prompted*	7.18	7.15
WF *TI*	2.53	2.79

The soft approximation method introduced in subsection 3.5 was evaluated on the WF *TI* condition. The results are presented in Table 3. A reasonable tradeoff between accuracy and speed can be obtained using R=5% which gives a relative degradation of 2.8% with a speedup factor of 5.

Table 3. *Soft approximation for the WF TI dataset.*

Dataset	EER in (%)	Runtime (in sec)	Speedup factor
Exact	2.53	0.76	-
w_2	2.79	0.03	25
w_2 , R=1%	2.73	0.09	8
w_2 , R=2%	2.67	0.11	7
w_2 , R=5%	2.60	0.15	5
w_2 , R=10%	2.58	0.26	3

6. CONCLUSIONS

In this paper we have proposed a method for efficient approximated extraction of i-vectors. The method manages to speed up i-vector extraction (excluding sufficient statistics calculation which is relatively fast) by a factor of 25. Speedup is obtained by normalizing the GMM counts for each session to be similar across sessions of the same gender and the same task characteristics. This normalization enables doing most of the calculations offline.

For NIST 2008, we observed no degradation in EER using the proposed method, and observed a very small degradation in minDCF. These accurate results are explained by the fact that NIST sessions are relatively long therefore it is not essential to use accurate values of Gaussian components in the MAP estimation process (Eq. 8).

The WF POT datasets contain significantly shorter sessions. Fortunately, for the *global*, *speaker* and *prompted* conditions, we observed only a small degradation in accuracy (if any). For the WF *TI* condition we did observe a significant degradation in EER (10%). We think that the degradation is due to the combination of the shortness of the test sessions (17 sec in average) and the severe textual content and length mismatches between the development data used to estimate the average Gaussian counts and the test data. We managed to reduce most of this degradation using the soft approximation approach in the cost of a reduced speedup factor.

[2] We obtain a different speedup factor that reported in [6] because we use a different GMM order and a different feature dimension are different than in [6]

7. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," IEEE *Trans. on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788 - 798, 2010.

[2] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek and T. Svendsen, "iVector Approach to Phonotactic Language Recognition", in Proc. *Interspeech*, 2011.

[3] W. Campbell, Z. Karam, "Simple and Efficient Speaker Comparison using Approximate KL Divergence", in Proc. *Interspeech*, 2010.

[4] P. Kenny, G. Boulianne, P. Ouellet and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker. recognition," in IEEE *Trans. on Audio, Speech and. Language Processing*, vol. 15, pp. 1435-1447, 2007.

[5] H. Aronowitz, O. Barkan, "New Developments in Joint Factor Analysis for Speaker Recognition", in Proc. *Interspeech*, 2011.

[6] O. Glembek, L. Burget, P. Matejka, M. Karafiat, P. Kenny, "Simplification and Optimization of I-Vector Extraction", in Proc. *ICASSP*, 2011.

[7] P. Kenny, "Joint factor analysis of speaker and session variability: theory and algorithms", technical report CRIM-06/08-14, 2006.

[8] D. Garcia-Romero and C. Y. Espy-Wilson "Analysis of I-vector Length Normalization in Speaker Recognition systems", in Proc. *Interspeech* 2011.

[9] N. Dehak , R. Dehak, J. Glass, D. Reynolds, P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques", in Proc. *Speaker Odyssey*, 2010.

[10] H. Aronowitz, R. Hoory, J. Pelecanos, D. Nahamoo, "New Developments in Voice Biometrics for User Authentication", in Proc. *Interspeech*, 2011.