FAST APPROXIMATE I-VECTOR ESTIMATION USING PCA

Mohamed Kamal Omar

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
mkomar@us.ibm.com

ABSTRACT

The i-vector representation has become increasingly popular in speaker and language recognition systems. The estimation of the projection matrix of the i-vector model is usually performed using the iterative expectation maximization (EM) algorithm. This work presents a novel approach to estimate the projection matrix of the i-vector representation and to estimate the i-vector representation for each utterance. In this approach, we formulate the estimation of the projection matrix as a principal component analysis (PCA) problem. Using the relation between PCA and a linear Gaussian model trained using the EM algorithm, we show that an approximate solution of the i-vector estimation can be obtained as the solution of a PCA problem. We evaluate the performance of our approximate i-vector estimation on the language recognition task of the robust automatic transcription of speech (RATS) project. The proposed approach reduces by 50% relative the computational time required to estimate the i-vector projection matrix and by 42% relative the computational time to estimate the i-vector representation compared to the standard EM-based approach to i-vector estimation. In addition, our experiments show improvements up to 29% relative in language recognition performance in terms of equal error rate compared to the standard EM-based i-vector estimation.

Index Terms— i-vector estimation, language recognition, PCA, EM algorithm

1. INTRODUCTION

Language recognition is an essential preprocessing step of audio streams to determine how to direct the audio for further processing. For example, the language detection system may enhance the customer service experience by facilitating the direction of the call to an agent with knowledge of the spoken language. One of the most popular approaches for reducing the dimension of the utterance representation in language and speaker recognition is the i-vector approach [1]. In this approach, a projection matrix is estimated based on a linear Gaussian model using maximum likelihood estimation (MLE). The expectation maximization (EM) algorithm is usually used to estimate the projection matrix iteratively [1, 2]. The EM algorithm is guaranteed to reach only a local maximum. This makes the estimated projection matrix sensitive to the initialization point of the algorithm. The estimation of the i-vector projection matrix also involves the estimation of the inverse of a positive semi-definite matrix in the low-dimensional subspace for each utterance in each iteration. This increases the computational cost of estimating the projection matrix using large training data. There are many recently suggested approaches to reduce the computational cost [3, 4] of the i-vector training and estimation and to reduce the memory cost using variational Bayes approaches [5, 6]. The approach in [3], for example, provides large savings in the i-vector training and extraction time, but there is a significant loss in the performance due to this simplification.

In this work, we propose an algorithm for estimating an approximate i-vector projection matrix which significantly reduces the computational time required for the estimation. In addition, this algorithm, unlike the i-vector model and its previous approximations, does not have the issues of sensitivity to initialization and making assumptions inconsistent with the assumptions made by the UBM model. This work builds on the relation between PCA and linear Gaussian models which was established by the work in [7, 8]. We construct a PCA problem which produces a projection matrix that approximates the i-vector projection matrix. Not only the estimation of the projection matrix is speedup by this approach, but also the estimation of the low-dimensional vector no longer requires the estimation of the inverse of a matrix for each utterance.

We evaluate the performance of the proposed approach in the context of the language recognition task of the Robust Automatic Transcription of Speech (RATS) program. The program targets audio analytics on highly distorted radio-frequency channels [9].

In the next section, the proposed fast approximate i-vector estimation approach is introduced. The experiments performed to evaluate the different techniques are described in Section 3. Finally, Section 4 contains a discussion of the results.

2. THE PROPOSED APPROACH

In this section, we discuss our approach to fast approximate estimation of the i-vector projection matrix using PCA. First, we provide a brief motivation and introduction to the relation between MLE of the projection matrix of a linear Gaussian model and the estimation of the PCA projection matrix as introduced in [7]. Then we discuss the application of this relation to the problem of estimating the i-vector projection matrix. We derive a PCA problem which its solution provides an approximate solution of the i-vector projection matrix estimation problem.

2.1. Motivation

The i-vector approach has been proven to be very successful in many applications including language [10] and speaker recognition [1], and speaker adaptation in speech recognition [11]. However, there are many aspects of the model which can be improved:

1. Estimating the i-vector projection matrix requires the estimation of the inverse of the posterior precision matrix of the hidden vector given the observations for each utterance. This can be computationally expensive for a large training data set.

2. The solution of the i-vector projection estimation is obtained using the EM algorithm which converges to a local maximum which is not guaranteed to correspond to the principal eigenvectors of the supervector covariance matrix [2].
3. Therefore the solution of the i-vector projection matrix estimation is sensitive to the initialization point.
4. The assumptions of the i-vector model are inconsistent with the assumptions of the UBM model; In the case of the UBM, the frame-based observation vectors are assumed to be independent and identically distributed (iid) with a GMM probability density function. While for the i-vector model, the frame-based observation vectors are in general not iid and becomes iid only conditioned on the hidden vector. This may explain why, as reported in [2], jointly updating the UBM covariance matrices and the i-vector projection matrix does not improve the performance.

In this work, we develop an approximation of the i-vector setup, called f-vector for fast vector, which:

1. does not require the estimation of the inverse of a matrix for each utterance during the projection matrix estimation or during the estimation of the projected vector,
2. is obtained by solving an eigenvalue decomposition problem which converges to a global minimum. Therefore the solution is not sensitive to the initialization point,
3. corresponds to the principal eigenvectors of the input sample covariance matrix,
4. does not make explicit assumptions about the probabilistic model in the original supervector space other than the assumptions of the UBM model.

2.2. PCA and linear Gaussian models

Principal component analysis is viewed in [7] as a limiting case of a particular class of linear Gaussian models. In linear-Gaussian models, the observation vector is assumed to be produced as a linear transformation of some lower-dimensional latent vector z plus additive Gaussian noise. The latent variables are assumed to be independent and identically distributed according to a unit variance spherical Gaussian probability density functions. Denoting the transformation by the P matrix, and the noise vector by v, the generative model can be written as

$$x = Pz + v, z \sim \mathcal{N}(0, I), v \sim \mathcal{N}(0, R). \quad (1)$$

Principal component analysis is a limiting case of the linear-Gaussian model as the covariance of the noise R becomes infinitesimally small and equal in all directions. i.e. $R = \lim_{\epsilon \to 0} \epsilon I$. This leads to a PCA solution using the EM algorithm by taking the limit of the solution for the linear Gaussian model as the covariance matrix of the noise term goes to zero. In this case, the EM updates are [7]

$$Z = \left(P^T P\right)^{-1} P^T X, \quad (2)$$

$$P^{new} = XZ^T \left(ZZ^T\right)^{-1}, \quad (3)$$

where X is the matrix of all observed data, and Z is the matrix of hidden vectors for all observed data.

2.3. I-Vector and linear Gaussian models

The i-vector model is [1, 2]

$$r_u = Ty_u + m, \; y_u \sim \mathcal{N}(0, I), \quad (4)$$

$$r_u = \left[r_u^1 r_u^2 \dots r_u^C\right]^T, \quad (5)$$

$$r_u^c = \begin{cases} \frac{1}{n_u^c} s_u^c & \text{if } n_u^c > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$s_u^c = \sum_{i=1}^{n_u} \gamma_c^{iu} \left(o_{iu} - \mu_c\right), \quad (7)$$

where T is the i-vector projection matrix, m is the residual noise vector, y_u is the latent vector, C is the number of Gaussian components in the UBM, n_u^c is the number of observations from utterance u assigned to the Gaussian component c, γ_c^{iu} is the posterior probability of the UBM Gaussian component c given the ith observation vector from utterance u, o_{iu} is the ith observation vector of utterance u, n_u is the number of observation vectors of utterance u, and μ_c is the mean vector of the cth Gaussian component of the UBM.

The update equations for the i-vector projection matrix estimation are [1, 2]

$$T^c = \sum_u s_u^c E[y_u^T] \left(\sum_u n_u^c E[y_u y_u^T]\right)^{-1}, \quad (8)$$

$$E[y_u] = L_u^{-1} T^T \Sigma^{-1} s_u, \quad (9)$$

$$L_u = I + T^T \Sigma^{-1} N_u T, \quad (10)$$

where T^c is the matrix which consists of the d rows of the T matrix which correspond to the UBM Gaussian component c, $E[y_u]$ denotes the posterior expectation of y_u given the observations of utterance u, $E[y_u y_u^T]$ is the posterior expectation of $y_u y_u^T$ given the observations of utterance u, L_u^{-1} is the posterior covariance of y_u given the observations of utterance u, N_u is a $dC \times dC$ diagonal matrix of the zeroth order statistics with the count of observations of the utterance u assigned to each Gaussian component c in the UBM repeated d times on the diagonal, d is the dimension of the frame-based observation vector, Σ is a diagonal covariance matrix of dimension $dC \times dC$ with the elements on the diagonal coming from staking the diagonal covariance matrices of the UBM.

For the model in Equations 4-7 to have a common posterior covariance matrix of the hidden vector y_u across all utterances, the posterior covariance matrix of the hidden vector y_u should be independent of the observation count of each utterance. This can be achieved by modeling instead of r_u another random vector, $t_u = N_u^{\frac{1}{2}} r_u$. The i-vector model for this random vector, t_u, is

$$t_u = Mh_u + q, \quad (11)$$

where M is the i-vector projection matrix, q is the residual noise vector, h_u is the latent vector.

The update equations for the i-vector projection matrix estimation of the model in Equation 11 are

$$M^c = \sum_u n_u^{c\,\frac{1}{2}} s_u^c E[h_u^T] \left(\sum_u n_u^c E[h_u h_u^T]\right)^{-1}, \quad (12)$$

$$E[h_u] = F^{-1} M^T \Sigma^{-1} N_u^{-\frac{1}{2}} s_u, \quad (13)$$

$$F = I + M^T \Sigma^{-1} M, \quad (14)$$

where M^c is the matrix which consists of the d rows of the M matrix which correspond to the UBM Gaussian component c, $E[h_u]$ denotes the posterior expectation of h_u given the observations of utterance u, $E[h_u h_u^T]$ is the posterior expectation of $h_u h_u^T$ given the observations of utterance u, F^{-1} is the posterior covariance of h_u given the observations of utterance u.

If we contrast the EM-based update equations of PCA in Equations 2 and 3 with the update equations for i-vector estimation in Equations 12-14, we notice that the correspondence of the two problems can be improved by using the following transforms

$$P = \Sigma^{-\frac{1}{2}} M, \quad (15)$$

$$x_u = N_u^{\frac{1}{2}} \Sigma^{-\frac{1}{2}} r_u. \quad (16)$$

$$z_u = \left[I + \left[P^T P \right]^{-1} \right] E[h_u]. \quad (17)$$

Substituting these values into the update equation of $E[h_u]$ in Equation 13, we get

$$z_u = \left[P^T P \right]^{-1} P^T x_u, \quad (18)$$

which is exactly in the same form as Equation 2 for updating the hidden vector in the EM-based PCA algorithm.

The dependency of the update equation of the i-vector projection matrix in Equation 12 on the zeroth order statistics is canceled out once the expressions for $E[h_u]$ and $E[h_u h_u^T]$ in terms of s_u are substituted. The only approximation needed to map the update equation of the projection matrix of the i-vector model in Equation 12 to the EM update equations for the PCA projection matrix in Equation 3 is to approximate the $E[h_u h_u^T]$ term by neglecting the posterior covariance of the hidden vector h_u given the observations of utterance u compared to the term function of $E[h_u]E[h_u^T]$ i.e.

$$E[h_u h_u^T] \approx E[h_u] E[h_u^T] \left[I + \left[P^T P \right]^{-1} \right]. \quad (19)$$

It should be noted that this approximation does not change the low-dimensional subspace spanned by the projection matrix but only change the weight of the different directions in this subspace. This approximation enables us to achieve a PCA-based model-free formulation which makes no assumptions about the model in the original feature space which may contradict the UBM assumptions. This is in contrast with the original i-vector model formulation and its previous approximations.

Applying this approximation, we get the following update equation for the projection matrix P,

$$P_{new} = \sum_u x_u z_u^T \left(\sum_u z_u z_u^T \right)^{-1}, \quad (20)$$

which matches Equation 3 for updating the hidden vector in the EM-based PCA algorithm.

The estimation of the approximate i-vector projection matrix for t_u is now reduced to the estimation of the PCA projection matrix of the vector $x_u = N^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} s_u$ and then multiplying the solution by $\Sigma^{\frac{1}{2}}$.

It is interesting to note the connection between the formulation derived here with the work in [12]. Based on that work, it can be shown that the l_2 distance between the normalized supervectors, $x_u = N_u^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} s_u$, of any two utterances is an approximation of the KL-divergence between the ML models of the utterances based on the UBM. These ML models of the utterances have the Gaussian component weights replaced with the normalized zeroth-order statistics of the utterance and the Gaussian component means replaced with the first order statistics conditioned on the UBM. This interpretation is very useful given that the PCA projection attempts to minimize the least square reconstruction error.

3. EXPERIMENTS

In this section, We evaluate and compare the performance of three projection techniques:

1. **PCA:** The projection matrix is estimated using the PCA objective function by eigen decomposition of the supervector sample covariance matrix.
2. **I-Vector:** The projection matrix is estimated using the EM algorithm with the ML objective function as described in [1].
3. **F-Vectors:** This is the proposed approach in this work. The projection matrix is estimated using the PCA objective function by eigen decomposition of the sample covariance matrix of the normalized vector, x_u, described in Equation 16.

3.1. Implementation

The training and test data for the experiments reported here use the LDC releases of the RATS LID data [9]. These consist of speech recordings from previous NIST-LRE telephone recordings as well as other RATS telephone recordings passed through eight noisy communication channels. The language recognition data is comprised of samples from five target languages and ten imposter languages. The five target languages are Levantine-Arabic, Farsi, Dari, Pashto and Urdu. The training data consists of about 270 hours of audio recorded over each radio channel. The systems are evaluated on two test sets: RATS dev2 and IBM test sets. The RATS dev2 data consists of approximately 83 hours divided equally across the 8 channels. While the IBM test set consists of approximately 478 hours divided equally across the 8 channels.

In the first set of experiments, we use acoustic features estimated using the power normalized cepstral coefficients (PNCC) algorithm [13]. The PNCC feature vector consists of 57 elements: 19 cepstral coefficients and their delta and delta-delta. In another set of experiments, we use features estimated using the frequency domain linear prediction (FDLP) algorithm [14]. In this case, the FDLP feature vector consists of 42 elements: 14 cepstral coefficients and their delta and delta-delta.

In all the experiments reported here, the GMM-UBM is trained using 43607 2-minute recordings from the eight channels. The GMM consists of 1024 diagonal-covariance Gaussian components and is trained using MLE. The three projection techniques evaluated here are trained using 74116 2-minute utterances and 74116 30-second utterances. In the case of i-vector estimation, we use the PCA projection matrix as an initial estimate of the i-vector projection matrix in the EM algorithm. This setup seemed to give us the best performance compared to random initialization. The final language scores are generated by six 5th order polynomial kernel SVMs. Each SVM is trained using one-versus-all setup to generate scores for one of the possible classes. The six classes represent the five target languages in addition to an imposter class representing all imposter languages. We use 82398 recordings divided equally across all durations to train each 5th order polynomial kernel SVM.

Table 1. Comparing the EER of PNCC systems on the RATS dev2 test set using the standard PCA, standard i-vector, and the f-vector representations with 400 dimensions

System	2 minutes	30 seconds	10 seconds	3 seconds
PCA	3.2	5.5	9.5	16.9
i-vector	3.3	5.5	8.9	15.6
f-vector	3.0	5.1	8.1	14.9

Table 2. Comparing the EER of PNCC systems on the RATS dev2 test set using the standard PCA, standard i-vector, and the f-vector representations with 800 dimensions

System	2 minutes	30 seconds	10 seconds	3 seconds
PCA	2.9	5.3	9.7	16.8
i-vector	3.4	5.4	8.8	15.8
f-vector	3.2	5.0	7.9	15.3

3.2. Results

In the first set of experiments, we use PNCC-based systems and set the dimension of the low-dimensional space after projection to 400. We compare the performance of the three projection techniques on both the RATS dev2 test set and the IBM test set. As shown in Table 1 for the 400-dimension systems, the systems using standard PCA projection and i-vector projection have very similar performance on the long durations: 2 minutes, and 30 seconds. While on the short durations: 10 seconds and 3 seconds, the performance of the i-vector system is significantly better than the standard PCA-based system. The results show small improvement on all durations from using the f-vector system compared to using the i-vector system. The improvements are relatively larger on the 2-minute and 30-second tasks compared to on the 10-second and 3-second tasks. Table 2 shows the results for the three projection techniques with 800 dimensions on the RATS dev2 test set. The relative performance of the different techniques seems to be consistent with the 400 dimension case. However, the PCA-based system seems to benefit more from doubling the dimension on the 2-minute task compared to the other two techniques. On the 10-second and 3-second tasks, still both the i-vector and f-vector systems significantly outperform the PCA system. The f-vector system slightly outperforms the i-vector system across all durations as in the 400 dimension case.

In Table 3, the results of the three 400-dimension PNCC systems on the IBM test set are reported. The performance of the i-vector and f-vector systems is significantly better than the PCA system on the IBM test set across all durations except for the 3-seconds task. On which the performance of the PNCC i-vector and f-vector systems is only slightly better than the PCA system. The results show also that the 400-dimension PNCC f-vector system significantly outperforms the corresponding i-vector system on the 2-minutes task and slightly outperforms the corresponding i-vector system on the other tasks.

In Tables 4 and 5, the performance of the FDLP system with the f-vector representation is compared to the corresponding systems with the PCA and i-vector representations on the RATS dev2 test set

Table 3. Comparing the EER of the PNCC systems on the IBM test set using the standard PCA, standard i-vector, and the f-vector representations with 400 dimensions

System	2 minutes	30 seconds	10 seconds	3 seconds
PCA	2.7	4.3	7.3	14.5
i-vector	2.2	3.4	6.5	14.0
f-vector	1.3	3.2	6.3	13.4

Table 4. Comparing the EER of the FDLP systems on the RATS dev2 test set using the standard PCA, standard i-vector, and the f-vector representations with 400 dimensions

System	2 minutes	30 seconds	10 seconds	3 seconds
PCA	3.7	6.0	10.7	17.8
i-vector	3.4	6.0	9.2	16.2
f-vector	3.2	5.6	8.5	15.4

Table 5. Comparing the EER of the FDLP systems on the IBM test set using the standard PCA, standard i-vector, and the f-vector representations with 400 dimensions

System	2 minutes	30 seconds	10 seconds	3 seconds
PCA	2.8	4.3	7.9	15.9
i-vector	1.6	3.5	7.0	14.9
f-vector	1.3	3.3	6.7	13.7

and the IBM internal test set respectively. The results are slightly worse than the corresponding PNCC systems. But still the relative performance of the three systems is similar. One difference is that with the FDLP frontend, the PCA system is slightly worse on the RATS dev2 2-minutes task compared to the other two systems.

Finally, we compare the processing time required to estimate each of the three dimensionality reduction techniques in absolute terms and in terms of a percentage of the time required to estimate the 400-dimensions i-vector projection matrix in Table 6. The results are obtained using C++ binary code for i-vector estimation and Matlab code for PCA and f-vector estimation. All experiments are run with multithreading enabled and with approximately 10 threads per instance on 1.5 GHz machine with 24 cores. The table shows that both f-vector and PCA approaches save more than half the time required to estimate the i-vector projection matrix independent of the dimension of the projected vector. It should be noted that 42% relative savings are also achieved in the process of estimating the low-dimensional representation of the utterances at test time. Since for both PCA and f-vector, this involves a matrix-vector multiplication. While in case of the i-vector, it requires the estimation of the inverse of the posterior precision matrix of the hidden vector.

4. DISCUSSION

In this paper, we derived an approximation of the standard i-vector representation which is estimated by solving a PCA problem. This approach avoids issues withe i-vector model and training algorithm such as sensitivity to initialization, and inconsistency with the assumptions of the UBM model. The proposed f-vector approach also reduces the processing time required to estimate the projection matrix by a factor of more than half. This comes with slight but consistent improvement in the performance compared to the standard i-vector approach.

Table 6. Comparing the training time of the standard PCA, standard i-vector, and the f-vector representations with PNCC frontend

System	time in sec.	% relative to the 400-d i-vector
400-dim. PCA	2856	48.13
400-dim. i-vector	5934	100.0
400-dim. f-vector	2955	49.8
800-dim. PCA	11709	197.32
800-dim. i-vector	25827	435.27
800-dim. f-vector	12411	209.15

5. REFERENCES

[1] Najim Dehak, Patrick Kenny, Rda Dehak, Pierre Dumouchel, and Pierre Ouelle, "Front End Factor Analysis for Speaker Verification," in *IEEE Transactions On Audio, Speech, and Language Processing*, vol. 19, no. 4, May 2011.

[2] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel, "Eigenvoice Modeling with Sparse Training Data," in *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[3] Ondrej Glembek, Luk Burget, Patrick Kenny, Martin Karafit, and Pavel Matejka, "Simplification and Optimization of I-Vector Extraction," in *Proceedings of ICASSP*, pp. 4516–4519, 2011.

[4] Sandro Cumani, Pietro Laface, and Vasileios Vasilakakis, "Memory and Computation Effective Approaches for I-Vector Extraction," in *Proc. of Speaker Odyssey*, June 2012.

[5] Sandro Cumani, and Pietro Laface, "Factorized Sub-Space Estimation for Fast and Memory Effective I-vector Extraction," in *IEEE/ACM Transactions On Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 248–259, 2014.

[6] Patrick Kenny, "A small footprint i-vector extractor" in *Proc. of Speaker Odyssey*, June 2012.

[7] Sam Roweis, "EM Algorithms for PCA and SPCA," in *Advances in Neural Information Processing Systems*, pp. 626–632, 1998.

[8] Michael Tipping, and Chris Bishop, "Probabilistic Principal Component Analysis," in *Journal of the Royal Statistical Society, Series B*, pp. 611–622, 1999.

[9] Kevin Walker, and Stephannie Strassel, "The RATS Radio Traffic Collection System," in *Proc. of Speaker Odyssey*, June 2012.

[10] Najim Dehak, Pedro Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language Recognition via I-vectors and Dimensionality Reduction," *Proceedings of InterSpeech*, pp. 857–860, 2011.

[11] Vishwa Gupta, Patrick Kenny, Pierre Ouellet, and Themos Stafylakis, "I-vector-based Speaker Adaptation of Deep Neural Networks for French Broadcast Audio Transcription," in *Proceedings of ICASSP*, pp. 6334–6338, 2014.

[12] William Campbell, Douglas Sturim, Douglas Reynolds, and Alex Solomonoff, "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation," in *Proceedings of the ICASSP*, pp. 97–100, 2006.

[13] Chanwoo Kim and Richard M. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition," in *Proceedings of ICASSP*, pp. 4101–4104, 2012.

[14] Kyu Han, Sriram Ganapathy, Ming Li, Mohamed Omar, and Shrikanth Narayanan, "TRAP Language Identification System for RATS Phase II Evaluation," *Proceedings of InterSpeech*, Lyon, France, 2013.