E–VECTORS: JFA AND I–VECTORS REVISITED

Sandro Cumani and Pietro Laface

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

ABSTRACT

Systems based on i–vectors represent the current state–of–the–art in text–independent speaker recognition. In this work we introduce a new compact representation of a speech segment, similar to the speaker factors of Joint Factor Analysis (JFA) and to i–vectors, that we call "e–vector". The e–vectors derive their name from the eigenvoice space of the JFA speaker modeling approach. Our working hypothesis is that JFA estimates a more informative speaker subspace than the "total variability" i–vector subspace, because the latter is obtained by considering each training segment as belonging to a different speaker. We propose, thus, a simple "i–vector style" modeling and training technique that exploits this observation, and estimates a more accurate subspace with respect to the one provided by the classical i–vector approach, as confirmed by the results of a set of tests performed on the extended core NIST 2012 Speaker Recognition Evaluation dataset. Simply replacing the i–vectors with e–vectors we get approximately 10% average improvement of the $C_{primary}$ cost function, using different systems and classifiers. These performance gains come without any additional memory or computational costs with respect to the standard i–vector systems.

Index Terms— Speaker Recognition, eigenvoice, Joint Factor Analysis, i–vectors, e–vectors.

1. INTRODUCTION

Systems based on i–vectors [1], and on Probabilistic Linear Discriminant Analysis (PLDA) [2] or discriminative classifiers [3, 4] represent the current state–of–the–art in text–independent speaker recognition. A speech segment is represented in this approach by a low–dimensional "identity vector" or i–vector, obtained from the statistics of a Gaussian Mixture Model (GMM) supervector [5] by a Maximum a Posteriori point estimate of a posterior distribution [6]. The i–vector model constrains the GMM supervector $\mathbf{s}$, representing both the speaker and channel characteristics of a given speech segment, to live in a single subspace according to:

$$\mathbf{s} = \mathbf{u} + \mathbf{T}\mathbf{w} \,, \tag{1}$$

where $\mathbf{u}$ 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 dimensions, respectively. The M columns of $\mathbf{T}$ are vectors spanning the "total variability" space, and $\mathbf{w}$ is a random vector of size M having a standard normal prior distribution.

In a recently proposed approach [7, 8, 9, 10, 11], the standard acoustic Universal Background Model (UBM) is replaced by a fine–grained "phonetic" UBM obtained by associating a single Gaussian to each output unit of a DNN, trained to discriminate among the states of a set of context–dependent phonetic units. For each frame, the posterior probability of the DNN states is used as the occupation probability for computing the usual statistics that allow training the UBM Gaussian parameters, and successively to extract the i–vectors.

I–vector modeling stems from the Joint Factor Analysis (JFA) approach [12, 13, 14]. JFA models the speaker and channel variability of a Gaussian supervector by means of a linear combination of eigenvoice, eigenchannel and MAP adaptated supervectors. Each of them can be represented by a low–dimensional set of factors that can be estimated according to the iterative procedure illustrated in [12]. It has, however, experimentally shown in [15] that the eigenchannel factors keep some correlation with the eigenvoice factors, i.e., they still convey some information about the speaker identity. This observation motivated the introduction of the i-vector approach as a feature extractor [16, 1], where speaker and channel variability are modeled in a single low–dimensional space spanned by the column vectors of a single matrix $\mathbf{T}$.

Although the i–vector subspace also includes channel variability, which is detrimental for speaker recognition, i–vectors have shown to provide a large performance boost over JFA–based methods. The main advantage of the i–vector representation is that the problem of intersession variability is deferred to a second stage, dealing with low–dimensional vectors rather than with the high–dimensional supervector space of the GMM supervectors. This allows training, in a low dimensional subspace, better classifiers, such as PLDA and Pairwise Support Vector Machine (PSVM) [3, 4]. Furthermore, in such low dimensional space it is possible to perform transformations that are particularly suited for enhancing the classifier performance, such as compensating development and evaluation set mismatches by means of length normalization [17, 18], or transforming i–vectors so that they better fit the classifier assumptions [19]. Finally, PLDA and PSVM models allow exploiting multiple recordings of the same speaker by simply averaging their corresponding i–vectors. This simple approach has shown to be more effective than properly estimation of multi–session likelihood ratios in PLDA.

In this work we propose a speaker modeling approach that estimates a more accurate speaker subspace with respect to the one provided by standard i–vector models. It extracts a compact representation of a speech segment, similar to i–vectors, but richer in speaker information, as confirmed by the results of a set of experiments performed on the extended core NIST SRE-2012 tests [20]. By analogy, we will refer to this representation as "eigenvoice-vector", or "e–vector" for short.

Our main idea is to extract more accurate i–vectors, relying on the eigenvoice space, which should be more accurate than the one represented by matrix $\mathbf{T}$, The novelty of our approach consists in estimating an i–vector subspace matrix so that it spans the same directions of the JFA speaker subspace. In particular, we estimate a linear transformation that allows keeping the span of the speaker–specific subspace, but at the same time allows learning a prior suited

Computational resources for this work were provided by the high performance computing clusters of HPC@POLITO (http://www.hpc.polito.it)

for i–vector extraction rather than for speaker–factor extraction. This is simply obtained by considering each training segment as belonging to a different speaker, as it is done in standard i–vector training, and re–training the i–vector subspace, starting from the JFA speaker matrix, by means of Minimum Divergence Estimation (MDE) iterations only. This corresponds to a right–multiplication of the JFA eigenvoice matrix with the Cholesky decomposition of $\mathbf{P} = \frac{1}{N}\sum_i \mathbb{E}[\mathbf{w}_i\mathbf{w}_i^T]$, where the sum extends over all the N training i–vectors, so that the empirical distribution of the e-vectors conforms to the standard normal prior [14, 2, 21]. Thus, we keep the span of the eigenvoice space, but we estimate a better model prior with the aim of better estimating the e–vector posterior.

The paper is organized as follows: Section 2 recalls the eigenvoice, JFA, and i–vector approaches. Section 3 introduces the e–vectors and their training procedure. The experimental results are presented and commented in Section 4, and conclusions are drawn in Section 5.

2. SUPERVECTOR REPRESENTATIONS

A model–based speaker adaptation approach was proposed in [22], which constrains the adapted model to be a linear combination of a small number of basis vectors obtained from a set of reference speakers. In this approach, these "eigenvoice" vectors are estimated with the objective of capturing the most important components of variation among the reference speakers. The adaptation data are used for obtaining, by means of Maximum Likelihood Eigen-Decomposition, the weights of the linear combination, leading to a low–dimensional vector representation of a new speaker in the eigenvoice space. Eigenvoice modeling, thus, aims at characterizing the speaker within the speaker subspace, and thanks to the correlations between GMM components, allows adapting also rarely observed Gaussians. This modeling approach was then also proposed for speaker recognition in [23], and in [24], where eigenvoice MAP adaptation was introduced.

In [25] the eigenvoice approach has been applied effectively to the problem of modeling intra–speaker variability, by compensating the session (channel) variability at recognition time. Finally JFA modeling was introduced in [12], where the eigenvoice model was extended to deal with intersession speaker variability, and channel mismatches between enrollment and evaluation conditions, taking care of the channel effects also in speaker enrollment. This was obtained by defining two subspaces: the speaker space represented by an eigenvoice matrix $\mathbf{V}$, and the channel space represented by an eigenchannel matrix $\mathbf{U}$. In particular, JFA models the speaker and channel dependent supervector $\mathbf{s}$ for a given speech segment as:

$$\mathbf{s} = \mathbf{u} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x} + \mathbf{D}\mathbf{z}\,, \tag{2}$$

where $\mathbf{u}$ is the UBM supervector, $\mathbf{V}$ and $\mathbf{U}$ are rectangular low rank matrices, $\mathbf{D}$ is a diagonal matrix, and $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{z}$ are the speaker, channel, and residual (or common) factors, respectively.

However, since the JFA channel factors do also contain information about the speaker identity [15], the i–vector approach has been introduced in [16, 1], where the speaker and channel dependent supervector model for a given speech segment was simplified as recalled in (1). Although the JFA speaker subspace better captures relevant speaker information, in the last years it has been shown that directly using this subspace in the JFA framework does not provide as good results as i–vector–based classifiers.

3. E–VECTORS

It is worth noting that, due to the substantial similarity of the models (2), and (1), matrix $\mathbf{T}$ training can be performed similarly to eigenvoice $\mathbf{V}$ matrix training. The only difference is that in $\mathbf{V}$ matrix estimation, the segments of a given speaker are considered as belonging to a single class, whereas in $\mathbf{T}$ matrix estimation, all segments are considered as belonging to different classes. Since matrix $\mathbf{T}$ spans both the speaker and channel subspace, and it is estimated considering each training segment as belonging to a different speaker, matrix $\mathbf{T}$ does not model the speaker subspace as well as the eigenvoice matrix $\mathbf{V}$.

On the basis of this observation, we propose a speaker modeling approach that tries to take advantage of the best of the JFA and of the i-vector techniques. We keep the i–vectors framework to exploit the low–dimensionality of a voice segment representation, but we estimate a different $\mathbf{T}$ matrix, which better accounts for the speaker space. This new matrix, $\mathbf{E}$, is similar to the $\mathbf{T}$ matrix, but it is estimated with the additional constraint that it spans the same subspace represented by the eigenvoice matrix trained on the same dataset.

The steps for training the $\mathbf{E}$ matrix are as follows:

- First a $\mathbf{V}$ matrix is trained exactly as matrix $\mathbf{T}$ is, but assuming that the segments of a given speaker belong to a single class, i.e., accumulating the sufficient statistics per speaker, rather than per speaker segment. It is worth noting that we perform this procedure as the eigenvoice estimation of [22, 24]. Alternatively, it is possible to train the $\mathbf{V}$ matrix together with matrices $\mathbf{U}$ and $\mathbf{D}$ in the JFA framework.
- In the second step, matrix $\mathbf{E}$ is initialized by $\mathbf{V}$. Then, the $\mathbf{E}$ matrix is trained considering each training segment as belonging to a different speaker, as it is done for the estimation of the $\mathbf{T}$ matrix, but applying only Minimum Divergence Estimation (MDE) iterations. MDE modifies the $\mathbf{E}$ matrix so that the empirical e–vector posterior conforms to the standard normal prior, increasing the data likelihood but leaving unchanged the span of matrix E.

The resulting e–vector model is, thus, similar to the i–vector model:

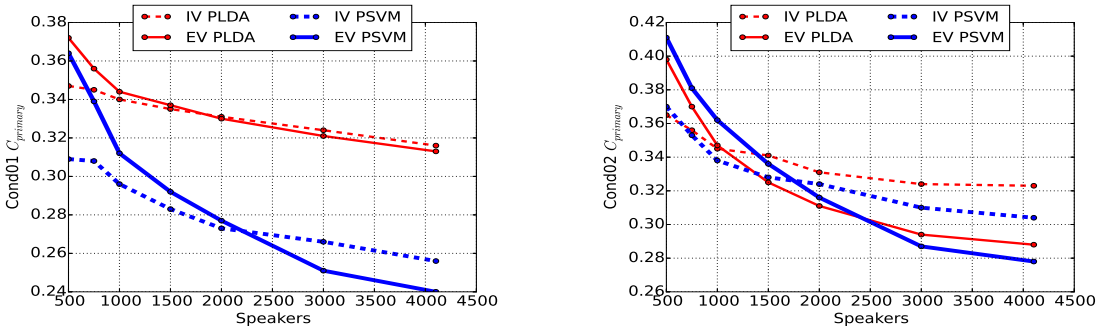
$$\mathbf{s} = \mathbf{u} + \mathbf{E}\mathbf{w}\,, \tag{3}$$

but matrix $\mathbf{E}$ better spans the speaker variability subspace. This means that e-vectors are more informative than i-vectors having the same dimensions, because some directions spanned by the $\mathbf{T}$ matrix mainly represent channel effects. The latter are reduced in matrix $\mathbf{E}$.

Since the segments of a given speaker are considered as a single class in matrix $\mathbf{V}$ training, the number of different speakers in the training corpora must be large enough to accurately model the speaker subspace. The effects of the dimension of the speaker population is analyzed in the next section, devoted to the experiments.

4. EXPERIMENTS

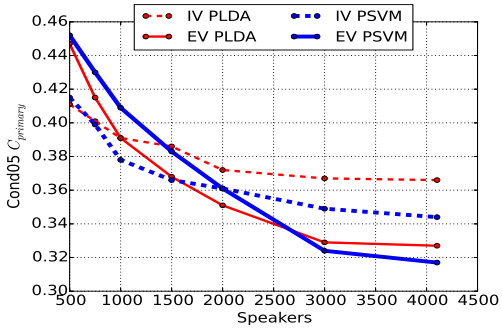
We compared the performance of i–vectors and e–vectors systems on the core extended NIST SRE 2012 tests [20]. For these experiments we used the 45-dimensional feature vectors obtained by stacking 18 cepstral (c_1-c_{18}), 19 delta (Δc_0-Δc_{18}) and 8 double–delta ($\Delta\Delta c_0$-$\Delta\Delta c_7$) parameters. We trained gender–independent i–vector and e–vector extractors, based on a 1024–component diagonal covariance UBM, estimated with data from NIST SRE 2004–2010, and additionally with the Switchboard II, Phases 2 and 3, and Switchboard Cellular, Parts 1 and 2 datasets, for a total of 79185 utterances from



(a) Interview without added noise $C_{primary}$

(b) Phone call without added noise $C_{primary}$

Fig. 1: Plots of the $C_{primary}$ of a PLDA and a PSVM systems, using i–vectors or e–vectors, as a function of the size of the number of speaker in the training set, for the first two conditions (Interview and phone call without added noise) of the core extended NIST SRE 2012 tests.



(a) Phone call in noisy environment $C_{primary}$

Fig. 2: Plots of the $C_{primary}$ of a PLDA and a PSVM systems, using i–vectors or e–vectors, as a function of the size of the number of speaker in the training set, for condition 5 (Phone call in noisy environment) of the core extended NIST SRE 2012 tests.

4103 speakers. The i–vector and e–vector dimension were both set to $d = 400$.

Classification has been performed by means of PLDA and by the PSVM classifier illustrated in [3, 4]. Gender–independent models were trained using the NIST SRE 2004–2010 datasets [20, 26], for a total of 42522 utterances of 3209 speakers.

As a first experiment, we evaluated the performance of a PLDA and of a PSVM classifier on the core extended NIST SRE 2012 tests, using i–vectors or e–vectors models trained with the speech segments of an increasing number of speakers randomly selected among the 4103 available. In Figure 1 we report the $C_{primary}$ results of these systems as a function of the number of speakers in the training set for the first two conditions, which refer to interview and phone calls without added noise, whereas Figure 2 shows the results for condition 5 (phone calls in noisy environment). The figures for the other two conditions (interview and phone calls with added noise) are similar. Looking at the plots, it is evident that the systems based on the eigenvoice subspace suffer the lack of training speakers. However, considering the PLDA $C_{primary}$ results using i–vectors and e–vectors (the dashed and solid red lines in the graphs, respectively), the systems using e–vectors estimated with more than 2000 speakers outperform the corresponding i–vector systems. The same behavior can be observed looking at the blue lines, corresponding to the PSVM systems. Thus, if the eigenvoice matrix $\mathbf{E}$ is trained with a large enough number of speakers, the extracted e–vectors are more informative about the speaker identity than the corresponding i–vectors, extracted by using matrix $\mathbf{T}$. Also evident is that a PSVM system trained with all available data is better than the corresponding PLDA system for all conditions.

Table 1 summarizes the results of a set of experiments performed on the same evaluation set, in terms of DCF08 and $C_{primary}$ cost functions [27, 20], using i–vectors or e–vectors. Table 2 shows the relative average performance improvement of the systems that we compare. In particular, its "EV vs EV" column gives the relative improvement of a system using e–vectors with respect to the same system using e–vectors. Label "DNN vs no DNN" refers to the DCF improvement given by a "phonetic" DNN–GMM system with respect to the corresponding standard acoustic GMM approach. The performance gains obtained by the PSVM model with respect to the corresponding acoustic or phonetic PLDA models are shown in column "PSVM vs PLDA", and the last column of the table, "System vs reference", reports the improvement observed for each system with respect to its reference.

The matrices $\mathbf{T}$ and $\mathbf{E}$ used for these experiments were estimated with the full set of training data previously described.

The first row of Table 1 reports the results, for all conditions, of "PLDA IV": the baseline GMM–based PLDA system using i–vectors. Its average performance is shown in the last column. The "PLDA EV" system is identical to the first one, but uses e–vectors rather than i-vectors. It improves the average decision cost functions approximately by 8%, as shown in Table 2.

We also assessed the quality of e–vectors extracted by using the DNN posteriors of the hybrid DNN/GMM architecture described in [10]. The performance of the system using i–vectors, "PLDA DNN IV", achieves 13% improvement with respect to the baseline "PLDA IV", and the corresponding e–vector system, "PLDA DNN EV", gains 14% with respect to "PLDA EV" system. The contribution of our DNN to performance gain is, thus, of the order of 13%, but an additional 10% improvement is obtained by using e–vectors rather than i–vectors also with this "phonetic" approach. Using the DNN–GMM approach and e–vectors we get approximately 22% improvement of the $C_{primary}$ with respect to the baseline "PLDA IV".

Finally, a third set of experiments was performed to verify that

Table 1: DCF08 and $C_{primary}$ cost functions results of a set of PLDA and PSVM systems, using with i–vectors (IV) or e–vectors (EV), on the core extended NIST SRE 2012 tests. Label DNN refers to the hybrid DNN–GMM framework.

System	Cond 1 interview without noise		Cond 2 phone call without noise		Cond 3 interview with added noise		Cond 4 phone call with added noise		Cond 5 phone call noisy environment		Average	
	DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}
PLDA IV	0.149	0.308	0.121	0.321	0.120	0.251	0.217	0.455	0.143	0.364	0.150	0.340
PLDA EV	0.141	0.294	0.106	0.288	0.133	0.248	0.183	0.400	0.129	0.325	0.138	0.311
PLDA DNN IV	0.146	0.279	0.100	0.264	0.118	0.229	0.184	0.396	0.120	0.305	0.134	0.295
PLDA DNN EV	0.137	0.262	0.085	0.237	0.118	0.216	0.155	0.345	0.103	0.272	0.120	0.266
PSVM IV	0.117	0.250	0.114	0.295	0.103	0.211	0.192	0.421	0.136	0.331	0.132	0.302
PSVM EV	0.111	0.234	0.106	0.265	0.095	0.192	0.164	0.375	0.127	0.303	0.121	0.274
PSVM DNN IV	0.114	0.233	0.088	0.239	0.104	0.199	0.159	0.363	0.105	0.272	0.114	0.261
PSVM DNN EV	0.108	0.218	0.077	0.209	0.101	0.185	0.141	0.320	0.093	0.238	0.104	0.234

Table 2: DCF08 and $C_{primary}$ average improvement of a set of PLDA and PSVM systems on the core extended NIST SRE 2012 tests.

System	Performance		Improvement							
	DCF08	C_{prim12}	EV vs IV		DNN vs no DNN		PSVM vs PLDA		System vs reference	
			DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}	DCF08	C_{prim12}
PLDA IV	0.150	0.340							PLDA reference	
PLDA EV	0.138	0.311	8.0%	8.5%					8.0 %	8.5%
PLDA DNN IV	0.134	0.295			10.7%	13.2%			10.7%	13.2%
PLDA DNN EV	0.120	0.266	10.4%	9.8%	13.0%	14.5%			20.0%	21.8%
PSVM IV	0.132	0.302					12.0%	11.2%	PSVM reference	
PSVM EV	0.121	0.274	8.3%	9.3%			12.3%	11.9%	8.3 %	9.3%
PSVM DNN IV	0.114	0.261			13.6%	13.6%	24.0%	23.2%	13.6%	13.6%
PSVM DNN EV	0.104	0.234	8.8%	10.3%	14.0%	14.6%	24.6%	24.8%	21.2%	22.5%

the e–vectors are also better than i–vectors also using a more accurate baseline discriminative classifier. The baseline performance of our PSVM approach [3, 4] is approximately 12% better than the baseline PLDA both for i–vectors and e–vectors. Again, the effectiveness of the e–vectors is evident with respect to the i–vectors: the former keep a 10% accuracy gain both using vectors extracted by means of the standard GMM approach or exploiting the posteriors of the hybrid DNN/GMM framework.

Overall, using e–vectors we obtain approximately 10% $C_{primary}$ improvement with respect the corresponding i–vectors systems either using a PLDA or a PSVM classifier, and 20% performance improvement with respect to the PLDA and PSVM baseline i–vector systems using DNN–GMM and e–vectors.

5. CONCLUSIONS

In this work we have verified that the eigenvoice space has more information about the speakers than i–vector subspace, because the latter includes more channel effects. To exploit this information, we have proposed a simple training procedure of the eigenvoice matrix, and introduced the e–vector, a compact representation of a speech segment, equivalent to i–vectors, but extracted exploiting the JFA speaker subspace.

E–vectors have shown to be a very good, no–cost, replacement of i–vectors for different extraction approaches and classifiers. Care has to be taken that the training corpus contains enough speakers and multiple recordings to accurately model the speaker subspace.

6. 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] 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`.

[3] S. Cumani, N. Brümmer, L. Burget, P. Laface, O. Plchot, and V. Vasilakakis, "Pairwise discriminative speaker verification in the i-vector space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1217–1227, 2013.

[4] S. Cumani and P. Laface, "Large scale training of pairwise support vector machines for speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 11, pp. 1590–1600, 2014.

[5] 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.

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

[7] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware Deep Neural Networks," in *Proceedings of ICASSP 2014*, pp. 1695–1699, 2014.

[8] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep Neural Networks for extracting Baum-Welch statistics for speaker recognition," in *Proceedings of Odyssey 2014*, pp. 293–298, 2014.

[9] D. Garcia-Romero and A. McCree, "Insights into Deep Neural Networks for speaker recognition," in *Proceedings of Interspeech 2015*, pp. 1141–1145, 2015.

[10] S. Cumani, P. Laface, and F. Kulsoom, "Speaker recognition by means of acoustic and phonetically informed GMMs," in *Proceedings of Interspeech 2015*, pp. 200–204, 2015.

[11] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified Deep Neural Network for speaker and language recognition," in *Proceedings of Interspeech 2015*, pp. 1146–1150, 2015.

[12] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[13] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 4, pp. 1448–1460, 2007.

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

[15] N. Dehak, *Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification*. PhD thesis, Ècole de Technologie Supérieure, Université du Québec, Montreal, Canada, 2009.

[16] 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.

[17] 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.

[18] 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.

[19] S. Cumani and P. Laface, "I–vector transformation and scaling for PLDA based speaker recognition," in *Proceedings of Odyssey 2016*, pp. 39–46, 2016.

[20] "The NIST year 2012 speaker recognition evaluation plan." Available at `"http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf`.

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

[22] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000.

[23] O. Thyes, R. Kuhn, P. Nguyen, and J. Junqua, "Speaker identification and verification using eigenvoices," in *Proceedings of ICSLP 2000*, pp. 242–245, 2000.

[24] S. Lucey and T. Chen, "Improved speaker verification through probabilistic subspace adaptation," in *Proceedings of EUROSPEECH 2003*, pp. 2021–2024, 2003.

[25] P. Kenny, M. Mihoubi, and P. Dumouchel, "New map estimators for speaker recognition," in *Proceedings of EUROSPEECH 2003*, pp. 2964–2967, 2003.

[26] Available at `http://www.ldc.upenn.edu/Catalog/`.

[27] NIST, "The nist year 2008 and 2010 speaker recognition evaluation plans." http://www.itl.nist.gov/iad/mig/tests/sre.