An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech

Mohammed Senoussaoui[1,2], *Patrick Kenny*[2], *Najim Dehak*[3], *Pierre Dumouchel*[1,2]

[1]École de Technologie Supérieur (ÉTS) Canada
[2]Centre de Recherche Informatique de Montréal (CRIM) Canada
[3]Spoken language system, CSAIL –MIT, Cambridge USA
{mohammed.senoussaoui, patrick.kenny, pierre.dumouchel}@crim.ca;
najimdcsail@mit.edu

Abstract

It is widely believed that speaker verification systems perform better when there is sufficient background training data to deal with nuisance effects of transmission channels. It is also known that these systems perform at their best when the sound environment of the training data is similar to that of the context of use (test context). For some applications however, training data from the same type of sound environment is scarce, whereas a considerable amount of data from a different type of environment is available. In this paper, we propose a new architecture for text-independent speaker verification systems that are satisfactorily trained by virtue of a limited amount of application-specific data, supplemented with a sufficient amount of training data from some other context.

This architecture is based on the extraction of parameters (i-vectors) from a low-dimensional space (total variability space) proposed by Dehak [1]. Our aim is to extend Dehak's work to speaker recognition on sparse data, namely microphone speech. The main challenge is to overcome the fact that insufficient application-specific data is available to accurately estimate the total variability covariance matrix. We propose a method based on Joint Factor Analysis (JFA) to estimate microphone eigenchannels (sparse data) with telephone eigenchannels (sufficient data).

For classification, we experimented with the following two approaches: Support Vector Machines (SVM) and Cosine Distance Scoring (CDS) classifier, based on cosine distances. We present recognition results for the part of female voices in the interview data of the NIST 2008 SRE. The best performance is obtained when our system is fused with the state-of-the-art JFA. We achieve 13% relative improvement on equal error rate and the minimum value of detection cost function decreases from 0.0219 to 0.0164.

1. Introduction

In the last decade, many approaches have been tested to improve performance of speaker recognition systems. The most popular are those based on generative models, like Gaussian Mixture Models based on Universal Background Model (GMM-UBM) [2]. Other generative models such as Eigenvoices, Eigenchannels and the most powerful one, the Joint Factor Analysis (JFA) [3], have built on the success of the GMM-UBM approach.

Recently, Dehak [1] proposed a feature extractor inspired from the joint factor analysis. Unlike JFA which models separately between-speaker and within-speaker variability in a high dimension space of supervectors, Dehak's idea consists in finding a low dimensional subspace of the GMM supervector space, named the *total variability space* that represents both speaker and channel variability. The vectors in the low-dimensional space are called i-vectors.

In [1], the i-vector features were tested on the 2008 NIST speaker recognition evaluation (SRE) telephone data. The i-vectors are smaller in size to reduce the execution time of the recognition task while maintaining recognition performance similar to that obtained with JFA. A key ingredient to the success of this approach was the enormous quantity of telephone data used to extract the i-vector feature set.

Our objective in this paper is to test the i-vector representation on the interview data of 2008 NIST speaker recognition evaluation (SRE) using Dehak's methods.

The main problem of this task is the small amount of microphone data at our disposal to extract the i-vector features. Indeed, this limited quantity does not allow a robust estimation of the total variability covariance matrix. To overcome this problem of lack of data, we propose to supplement the microphone data with telephone data, that is 10 times greater in size. Using this augmented corpus, we solve the estimation problem in a manner similar to [3].

This paper is organized as follow. Section 2 explains how the total variability features are processed in the case of abundant data (telephone speech). In Section 2, we discuss how to take advantage of telephone data to construct an i-vector extractor for microphone speech. The fourth and the fifth sections are dedicated, respectively, to the channel compensation and classification methods used in our experiments. Section 6 reports experimental results on the NIST 2008 interview data and the last section presents conclusions.

2. i-vector feature extraction

The main idea in traditional JFA, introduced by Kenny [3], is to find two subspaces which represent the speaker- and channel-variabilities, respectively. Dehak's experiments [1] show that JFA is only partially successful in separating speaker and channel variabilities. He found that the channel space contains some information that can be used to distinguish between speakers. For this reason, Dehak proposed a single space that models the two variabilities and named it the total variability space.

Dehak's basic assumption is that a given speaker- and channel-dependent GMM supervector M can modeled as follows:

$$M = m + Tw \tag{1}$$

where m is a speaker- and channel-independent supervector (UBM supervector is a good estimate of m), T is a low rank matrix, which represents a basis of the reduced total variability space and w is a standard normal distributed vector. T is named the total variability matrix; the components of w are the total factors and they represent the coordinates of the speaker in the reduced total variability space. These feature vectors are referred to as *identity vectors* or i-vectors for short. The feature vector associated with a given recording is the MAP estimate of w, whose calculation is explained in Kenny's Proposition 1 [4]. We denote it by $\hat{w}$. The matrix T is estimated using the EM algorithm described in Kenny's Proposition 3 [4].

3. i-vector extraction in a context of sparse microphone speech data

As mentioned previously, robust estimation of the total variability matrix T in (1) requires a large amount of data, whereas we have relatively little microphone data at our disposal. The main contribution of this paper is to show how to use telephone data in addition to microphone data to estimate a new total variability matrix (we name it N) that is suitable for speaker recognition on microphone speech. We adopt a modified version of the method proposed in [3] to deal with the *cross-channel condition* of the 2006 NIST SRE. In that paper, it was shown how to estimate supplementary eigenchannels on microphone development data and append them to eigenchannels estimated on telephone speech. Section IV in Kenny's article [3] presents results obtained with this method.

First, we estimate a gender dependent matrix T of rank R_{tel} using only telephone data, by assuming the supervector M associated with a telephone recording has the form represented in (1).

Second, we estimate a gender dependent matrix T' of rank R_{mic} using only microphone data, by assuming the supervector M associated with a microphone recording has the form

$$M = m + T\hat{w} + T'w' \tag{2}$$

where w' is a standard normally distributed random vector.

Finally, we combine (1) and (2) to produce a feature extractor which can be used for microphone speech as well as telephone speech. In a companion paper [5], this representation was also tested on the telephone speech using probabilistic methods. That is, we assume that the supervector M associated with a recording has the form

$$M = m + Nx \tag{3}$$

where $N = [T\ T']$[1] is the new total variability matrix of rank $R = R_{tel} + R_{mic}$, x is a random vector of dimension $D = R$ having a standard normal distribution. So the new feature vector associated with a recording is the MAP estimate of x.

The estimation procedure of the total matrix N is illustrated in Figure 1:

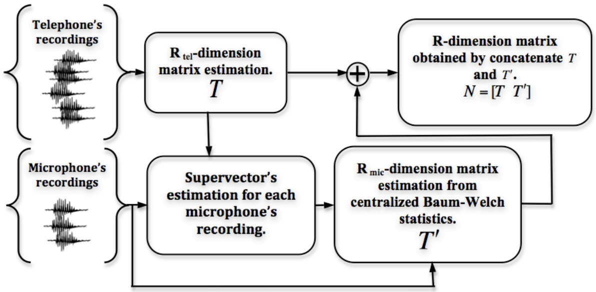
Telephone's recordings
Microphone's recordings
R tel-dimension matrix estimation. T
Supervector's estimation for each microphone's recording.
R mic-dimension matrix estimation from centralized Baum-Welch statistics. T'
R-dimension matrix obtained by concatenate T and T'. N = [T T']

Figure 1: The block diagrams showing the estimation of the matrix N.

4. Channel compensation

I-vectors are extracted in a way that makes no distinction between channel and speaker variability. So the problem of channel variability has to be dealt with in constructing classifiers for speaker recognition using i-vectors as features.

We use a combination of Linear Discriminant Analysis (LDA) and Within Class Covariance Normalization (WCCN). This combination was successfully tested with NIST telephone data in [1].

4.1. Linear Discriminant Analysis

LDA is a supervised method, intended mainly for dimensionality reduction. The power of LDA lies in the fact that it uses class labels to find the optimal mapping to the reduced space. This mapping minimizes the within

[1] The square bracket notation in $N = [T\ T']$ represents the concatenation of the pair of matrices.

class variability and simultaneously maximizes the between classes variability. To do this, LDA is based on the optimization of the Fisher objective function:

$$J(u) = \frac{u^t S_b u}{u^t S_w u} \tag{4}$$

where S_b and S_w represent, respectively, the between classes covariance matrix and the within class covariance matrix; u is a given space direction and t is the transposed symbol. Given a training set of S speakers and n_s utterances per speaker, the optimization of Fisher's criterion (4) consists in solving the generalized eigenvalue problem given by:

$$S_b u = \lambda S_w u \tag{5}$$

The projection matrix A is given by the eigenvectors associated with the greatest eigenvalues. The formulas used to calculate S_b and S_w are as follows:

$$S_b = \sum_{s=1}^{S} (x_s - \bar{x})(x_s - \bar{x})^t \tag{6}$$

$$S_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{o=1}^{n_s} (x_o^s - \bar{x_s})(x_o^s - \bar{x_s})^t \tag{7}$$

where x_o^s is the o^{th} observation of speaker s (in our case x_o^s is an i-vector), $\bar{x}_s = \frac{1}{n_s} \sum_{o=1}^{n_s} x_o^s$ is the mean of the observations of speaker s and $\bar{x}$ represents the mean of all instances in the training set. However, in the case of i-vectors the mean $\bar{x}$ is equal to zero, due to the assumption that the i-vectors are normally distributed with zero mean vector and identity covariance matrix.

4.2. Within Class Covariance Normalization

WCCN was introduced by Andrew Hatch [6] in the context of SVM classifiers. This method proposes to use the inverse of the within class covariance matrix to normalize the linear kernel. To avoid confusion with the LDA's within class covariance, we refer to this matrix as W. This matrix is calculated in a manner similar to that given in (7). The purpose of the WCCN is to minimize the error expectation of false alarm and false rejection in the training stage of a linear kernel SVM.

In our systems, W is calculated in the projected space of the LDA and is used with cosine kernel [1] (as we will see in later sections).

4.3. Using telephone data to estimate the normalization matrices of LDA and WCCN

As in the case of T-matrices, we need to use telephone data in addition to microphone data to estimate LDA projections and WCCN matrices. We have deployed two strategies to do so: pooling and weighting.

- The first one consists in the use of all microphone and telephone i-vectors to estimate the LDA matrices S_b and S_w, and the WCCN (or equivalently W^{-1}). We refer to this scheme as strategy of *pooling*.
- In the second strategy, we begin by estimating LDA matrices from telephone i-vectors, then we similarly estimate other LDA matrices from the microphone data. Thereafter, we calculate the final LDA matrices by a weighted average of microphone and telephone LDA matrices. More specifically, we estimate the between-class covariance S_b from telephone S_{btel} and microphone S_{bmic} between-class covariance matrices as follows:

  $$S_b = P_{tel} S_{btel} + P_{mic} S_{bmic}$$

  where P_{tel} and P_{mic} are the weights associated with the microphone and telephone between-class covariances, respectively. These weights are set empirically. The same weighting approach is used for the within-class covariance matrix of WCCN. We refer to this scheme as strategy of *weighting*.

5. Classification methods

In this section we briefly describe the two classification methods used for this work: Support vector machines and Cosine Distance Scoring.

5.1. Support Vector Machines

Since its introduction by Vapnik [7], SVM has become widely used in statistical learning. A SVM is a supervised binary linear classifier which finds, among all possible linear hyperplane separators, the one which maximizes the margin between two labeled classes of data. The classification function f associated with the optimal hyperplane separator H is given by:

$$\begin{aligned} f : \mathbb{R}^N &\to \mathbb{R} \\ x &\mapsto f(x) = w^t x + b \end{aligned} \tag{8}$$

where x is an instance vector, w and b are respectively the weights and bias vector estimated during the training stage. For a given test instance x_t, the classification is based on the $sign$ of the function of hyperplane separator f:

$$Classification(x_t) : sign(f(x_t) = w^t x_t + b) \tag{9}$$

As defined, SVM is a linear classifier. In this context a linear separation is equivalent to using a linear kernel as follows:

$$k(x_1, x_2) = \langle x_1, x_2 \rangle \tag{10}$$

where $\langle x_1, x_2 \rangle$ is the dot product of the instance vectors x_1 and x_2.

The introduction of kernel functions to classical SVM classifiers allows more complex problems to be solved, as revealed by the good performances obtained by Dehak [1] for the *core condition* and the *10sec-10sec* contexts of NIST telephone data. Accordingly, we carry out our experiments with the use of the cosine kernel. The cosine kernel between two observation vectors x_1 and x_2 is given by:

$$k(x_1, x_2) = \frac{\langle x_1, x_2 \rangle}{\|x_1\| \|x_2\|} \tag{11}$$

At first glance, one can observe that the cosine kernel is just a *norm*-normalized version of the linear kernel. However, this *norm*-normalization compensates a dilatation of data in the kernel space, that in turn has a positive effect on channel compensation in the case of i-vectors.

Dehak and al. found that LDA alone is not sufficient to offset the channel effect. The best results were obtained when LDA was followed by WCCN, as explained in Section 4. Specifically, for two given i-vectors x_1 and x_2, we first project i-vectors using the LDA projection matrix. Then we normalize the cosine kernel by the inverse of the WCCN matrix, as follows:

$$k(x_1, x_2) = \tag{12}$$
$$\frac{(A^t x_1)^t W^{-1} (A^t x_2)}{\sqrt{(A^t x_1)^t W^{-1} (A^t x_1)}.\sqrt{(A^t x_2)^t W^{-1} (A^t x_2)}}$$

where A and W are respectively LDA and WCCN projection matrices.

This kernel normalization is applied in both classifiers, support vector machine and cosine distance scoring.

5.2. Cosine Distance Scoring

The SVM classifier uses target data and a set of impostors data to train a target model, and thereafter determines the sign of the function f to make the final decision. Unlike SVM, the Cosine Distance Scoring classifier calculates the target i-vector using its training data in the enrollment stage, which implies that no target model is required. In the test stage, a score is calculated by taking the *cosine distance* between the test instance and the target one. This score is compared to a decision threshold θ in order to make the final decision according to:

$$score(x_{target}, x_{test}) = \frac{\langle x_{target}, x_{test} \rangle}{\|x_{target}\| \|w_{test}\|} \gtrless \theta \tag{13}$$

Results reported by Dehak and al. [1] show that the use of CDS classifier for telephone speech yields better performances than those obtained by SVM, especially for the *10sec-10sec* condition of the NIST 2008 SRE.

6. Experiments and results

In this section, we first describe our experimental set-up, followed by performance results.

6.1. Experimental set-up

6.1.1. Universal Background Model

We use gender-dependent UBMs containing 2048 Gaussians. This UBM is trained using LDC releases of Switchboard II, Phases 2 and 3; Switchboard Cellular, Parts 1 and 2; and NIST 2004-2005 Speaker Recognition Evaluation (SRE). The gender-dependent joint factor analysis models are trained on the same quantities of data as the UBM training. Speech parameters are represented by a vector of dimension 60 of Mel Frequency Cepstral Coefficients (MFCC) i.e. static MFCC, delta and double delta.

6.1.2. Joint Factor Analysis

In this work, we will be presenting results obtained by the application of JFA on interview speech of the NIST's condition *short2-short3*. The underlying theory of JFA is beyond the scope of this article, but interested readers can refer to Kenny and al. [3] for details. For our experiments, we use a joint factor analysis configuration which is made up of 300 speaker factors and 100 channel factors computed from the same data as those used to train the UBM. One hundred (100) additional channel factors are computed from all NIST microphone speech data.

6.1.3. Total variability matrix N

As mentioned in Section 3, the computation of the matrix N proceeds as follows. First, we compute a 400-dimension matrix T from LDC releases of Switchboard II, Phases 2 and 3; Switchboard Cellular, Parts 1 and 2; and NIST 2004 and 2005 SRE (i.e. telephone speech). Second, a 200-dimension matrix T' is estimated from all NIST microphone data (i.e. NIST 05, NIST 06 and 2008 interview development microphone data). Finally, we append T to T' to obtain a variability matrix N of dimension 600, which is also the dimension of our i-vectors.

6.1.4. Support Vector Machines and Cosine Distance Scoring

To build the SVM classifier, we use the normalized cosine kernel shown by equation (12) to carry out the kernel space transformation in the SVM systems.

Similarly, for the CDS classifier, we use normalized cosine distances according to (13).

6.1.5. Score normalization

We use 200 *t-norm* female models and 1007 *z-norm* female utterances taken separately from NIST 05 and NIST 06 microphone data. All SVM scores are *z-normalized* scores while CDS scores are *zt-normalized.*

6.1.6. Voice activity detection for interview data

To eliminate the silence parts in the microphone speech recordings, we used the transcripts of the microphone data provided by NIST as a proxy for voice activity detection. This fails on some files so the results we present are incomplete.

6.2. Results

In this section, we present performance results only for the female part of the NIST *short2-short3* task since the error rate on the corpus of females is lower than males and that the behavior of experiments is similar for both corpora. Specifically, we focus on cases where all auxiliary microphones used for recording the training corpus are different from those used for recording the test corpus (det3). For evaluation, we use the *Equal Error Rate* (EER) and the minimum of NIST's *Detection Cost Function* (DCF) as metrics for performance evaluation.

We performed a total of three experiments. The first uses microphone data only. The second uses both microphone and telephone data. Finally, the last experiment fuses several systems trained using telephone and microphone data.

The objective of this experiment is to obtain a performance benchmark for systems driven with little data from the same environmental context. As shown in Table1, the best performance is obtained by a SVM classifier that achieves an EER of 5.18% and a minDCF of 0.0242. We also observe that the rotation of the i-vector space by LDA brings more discrimination than CDS (i.e. a space which minimizes intra-speaker variability and maximizes inter-speaker variability) for the same dimension of 600.

Table 1: *EER, minimum DCF and optimal dimension for SVM and CDS systems based on microphone data only. The i-vectors are estimated as summarized in Section 3.*

	EER	minDCF	Dim
SVM	**5.18%**	**0.0242**	600
CDS	6.13%	0.0345	600

In this first experiment, we showed results obtained with SVM and CDS classifiers for microphone data only. A first experiment with both microphone and telephone data shows that a JFA classifier surpasses the performance obtained so far with an EER of 3.97% and a minDCF of 0.0219 (as shown in the first line of Table 3). This observation motivates us to use telephone data in addition to microphone data in estimating the LDA projections and WCCN matrices.

In the second experiment, we tested two strategies to use both microphone and telephone data as discussed in Section 4.3: pooling and weighting. As our results show (Table 2), the best strategy applied to both types of data is that of weighting.The best performance by EER metric is obtained with a SVM. The dimension of the SVM is then 400, the weights are $P_{mic} = 0.2$ and $P_{tel} = 0.8$ with an EER of 4.57%. The best performance by minDCF metric is as well obtained with a SVM (minDCF of 0.0219). For this case, the SVM is of dimension 350 and the weights are set to $P_{mic} = 0.3$ and $P_{tel} = 0.7$.

In the second experiment, it is worth noting that the parametric dimension has been reduced to 350, compared to 600 in first experiment. Furthermore, according to the minDCF metric, the SVM classifier yields a better performance (0.0217) than the best obtained so far (0.0219 with JFA). However, under the EER metric, the JFA classifier performs better than a SVM one (3.97% compared to 4.57%). These observations motivated us to devise a third experiment that fuses a JFA classifier with SVM and CDS classifiers. The merging of classifiers is carried out using the FoCal toolkit presented in [8]. In this experiment, we train the JFA classifier on microphone speech under the same conditions, namely NIST *short2-short3*. Furthermore, for SVM and CDS classifiers, we have reused the same size and weight as those in the second experiment. Table 3 shows the performance achieved in this case.

Table 2: *EER, minimum DCF and optimal dimension for systems based on SVM and CDS classifiers, using microphone and telephone i-vectors to compensate channel effects with the pooling and weighting methods summarized in Section 4.3.*

		EER	minDCF	Dim
pooling	SVM	**6.44%**	**0.0338**	400
	CDS	7.13%	0.0400	400
weighting	SVM	4.68%	**0.0217**	350
		4.57%	0.0220	400
	CDS	5.46%	0.0305	400

Compared to a single JFA classifier, fused systems offer better performance under both metrics. According to the minDCF metric, a SVM-JFA classifier improves results by decreasing the minDCF from 0.0219 to 0.0164. According to the EER metric, a CDS-JFA improves performance by a relative 13%.

Table 3: *EER and minimum DCF for a system based on JFA classifier and other fused systems.*

	EER	minDCF
JFA	**3.97%**	**0.0219**
SVM-JFA	3.47%	**0.0164**
CDS-JFA	**3.44%**	0.0178

7. Conclusion

In this paper, we have presented a new way to use microphone and telephone speech in order to design an i-vector extractor that is suitable to work on microphone speech as well as telephone speech. We have also shown how we could add telephone speech to further compensate channel effects with LDA and WCCN in our i-vector space. The best results are obtained when these systems are fused with the classical joint factor analysis. We achieved 13% relative improvement on equal error rate and the minimum value of detection cost function decreases from 0.0219 to 0.0164.

8. References

[1] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *INTERSPEECH*, Brighton, UK, Sept 2009.

[2] D. A. Reynolds, *A gaussian mixture modeling approach to text-independent speaker identification*, Ph.D. thesis, Georgia Institute of Technology, August 1992.

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

[4] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Process. (USA)*, vol. 13, no. 3, pp. 345 – 54, May 2005.

[5] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey Speaker and Language Recogntion Workshop, Brno, Czech Republic,*, June 2010.

[6] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," *INTERSPEECH 2006 and 9th International Conference on Spoken Language Processing - ICSLP*, vol. 3, pp. 1471 – 1474, 2006.

[7] V.N. Vapnik, *Statistical Learning Theory*, 1998.

[8] N. Brummer and J. du Preez, "Application-independent evaluation of speaker detection," *Comput. Speech Lang. (UK)*, vol. 20, no. 2-3, pp. 230 – 75, April 2006.