Non-Linear I-vector Extraction for Speaker Recognition

Oren Barkan[1,2] and Hagai Aronowitz[1]

[1]IBM Research, Israel
[2]School of Computer Science, Tel Aviv University, Israel
orenba@il.ibm.com, hagaia@il.ibm.com

Abstract

We propose an algorithm for non-linear i-vector extraction. The algorithm is based on the manifold learning technique named Diffusion Maps (DM) and motivated by recent results that showed that the GMM supervectors reside on a low dimensional manifold. Our proposed method may further be processed using standard techniques such as Linear Discriminant Analysis (LDA), Within Class Covariance Normalization (WCCN) and Probabilistic LDA (PLDA). We demonstrate the effectiveness of our algorithm and compare its results with the state-of-the-art i-vector based PLDA algorithm on the NIST 2010 Speaker Recognition Evaluation (SRE).

Keywords - Speaker verification, Diffusion Maps, i-vectors, Non-linear dimensionality reduction, Pattern recognition.

1 Introduction

During the last few years i-vectors [3] have become the standard front-end layer in most of state-of-the-art speaker recognition systems. I-vectors are Factor Analysis based method which provides a way to map the high dimensional GMM supervectors to a relatively low dimensional vectors, named i-vectors. Factor Analysis, as a subspace learning method, assumes the data to reside in a subspace and therefore is able to capture only linear structures. However, it was not until recently that Karam et al. [2] showed that the GMM supervectors in the GMM space are lying on a low dimensional manifold and that by the use of manifold learning techniques such as graph geodesics and ISOMAP [7] it is possible to improve classification error.

In this paper we replace the misleading assumption of linearity with an assumption that the GMM supervectors reside on a low dimensional manifold and propose an alternative non-linear way for i-vector extraction we name d-vector extraction. The proposed algorithm is based on the DM framework and may further be processed using standard techniques such as PLDA [9]. We demonstrate the usefulness of our approach on the telephone core condition 5 of NIST 2010, and obtain significant error reduction.

The paper is organized as follows: In Section 2 we overview the DM framework. In Section 3 we explain in detail the proposed d-vector extraction algorithm. In Section 4 we present the experimental setup and results. In Section 5 we draw conclusions.

2 Diffusion Maps

Diffusion Maps (DM) [6] is a machine learning technique for non-linear dimensionality reduction. The method focuses on discovering the underlying manifold that the data has been sampled from.

In this method a graph affinity matrix is built which is used to generate a diffusion process. As the diffusion process progresses, it integrates local geometries to reveal geometric structures of the data at different scales. Based on the revealed geometry, one can measure the similarity between two data samples at a specific scale. A diffusion map embeds the high dimensional data in a lower-dimensional space D, such that the Euclidean distance between points in D approximates the diffusion distance in the original feature space. The dimension of D is determined by the geometric structure underlying the data, and the accuracy by which the diffusion distance is approximated.

In our setup, the high dimensional feature vectors are the GMM supervectors that represent different sessions in the GMM supervector space G, we name them as g-vectors. DM is performed in order to map the g-vectors to l-dimensional d-vectors in the diffusion space D. From now on, we will use these notations to differ between the original high dimensional feature space, and the low dimensional diffusion space. The rest of this section discusses the DM algorithm in more detail.

Given a development set of n g-vectors $\{x_i\}_{i=1}^{n} \subseteq G$ the first step in the DM algorithm is to define an affinity kernel over G. A common choice is the Gaussian kernel:

$$k(x_i, x_j) = \exp\left(-\frac{c(x_i, x_j)^2}{\sigma}\right) \tag{1}$$

where $c(x_i, x_j)$ is a metric and the σ parameter determines the scale or size of the neighborhood we trust our local similarity measure to be accurate in. In practice, σ is chosen empirically or according to prior knowledge of the geometric structure and density of the data. A method for automatic configuration of σ was proposed in [4].

In this way, we can define a full undirected graph where the g-vectors are the nodes, and the weights of the edges are determined according to the diffusion kernel in Eq. (1). We then define a random walk on this graph by converting the affinity kernel to a probability function as follows:

$$p(x_i, x_j) = \frac{k(x_i, x_j)}{\sum_{h=1}^{n} k(x_i, x_h)}$$

This results in a transition Markov matrix P in which the entry $P_{i,j} = p(x_i, x_j)$ is the probability of transition from node x_i to node x_j in a single step. In the same way, P^t is a matrix in which the entry $P^t_{i,j}$ is the probability of transition from node x_i to node x_j in t steps.

A diffusion distance after t steps is defined as follows:

$$Q_t(x_i, x_j) = \sum_{k=1}^{n} (P^t_{i,k} - P^t_{j,k})^2 .$$

A spectral decomposition of P results in a complete set of eigenvalues $1 = \lambda_0 \geq \lambda_1 \geq ... \geq \lambda_n$ and left and right eigenvectors that satisfies: $P\psi_i = \lambda_i \varphi_i$. Then, we define a mapping $M_t : \{x_i\}_{i=1}^{n} \to D$ as follows:

$$M_t(x_i) = \left[\lambda_1^t \psi_{1i}, ..., \lambda_l^t \psi_{li}\right]^T$$

where ψ_{ki} indicates the i-th element of the k-th eigenvector of P and l is the dimension of the diffusion space D. In [6]. it has been shown that for $l = m - 1$ the following equation holds:

$$\left\| M_t(x_i) - M_t(x_j) \right\|_2^2 = Q_t(x_i, x_j) .$$

This result justifies the use of squared Euclidean distance in the diffusion space. Of course in practice one should pick $l < m - 1$ according to the spectral decay of $(\lambda_i)_{i=1}^{n}$. This decay is determined by complexity of the intrinsic dimensionality of the data and the choice of the parameter σ .

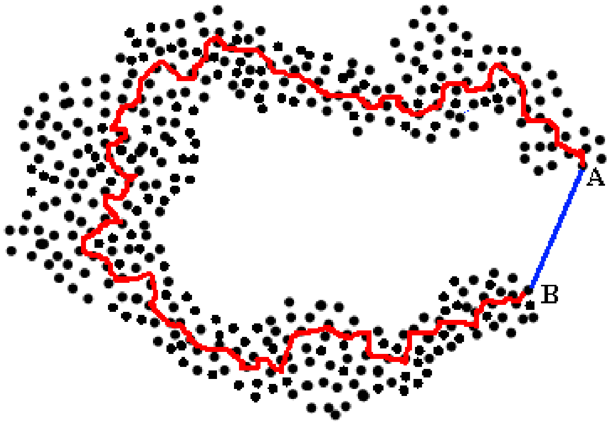

Figure 1. *As the diffusion process progresses, travelling along the blue path becomes more likely than travelling along the red one.*

Figure 1 shows an illustrative example for a diffusion process. Two alternative paths connect points on the manifold. The blue path is the longer one but is the one that follows the geometric structure of the manifold while the red path is the short one but does not follow the manifold structure. As the number of steps in the diffusion process, t, increases, the probability of travelling along the red path also increases, since it consists of many short distance jumps. However, the probability of travelling along the blue path stays always small (and becomes smaller and smaller as t increases) as it consists of long distance jumps.

2.1. Out of sample extension

So far, we addressed the situation when all g-vectors are given a-priori. However, we need to also address the situation where a new g-vector $x_{n+1} \notin \{x_i\}_{i=1}^{n}$ is introduced and we are asked to extract its corresponding d-vector. A naïve approach would be to repeat the whole process described above from scratch. Although this might be practical in offline applications, it is extremely inefficient and results in large amount overhead. Therefore, we propose the Nystrom extension [5]:

$$\psi_{k(n+1)} = \frac{1}{\lambda_k} \sum_{j=1}^{n} P_{n+1,j} \psi_{kj} . \tag{2}$$

This extension extends the eigenvectors with one additional entry corresponding to the new g-vector while it is consistent on the development set $\{x_i\}_{i=1}^{n}$. This results in an extended mapping: $M_t^{x_{n+1}}:\{x_i\}_{i=1}^{n+1}\rightarrow D$.

3 D-Vector Extraction for Speaker Recognition

Our main contribution in this work is the utilization of the DM framework for a non-linear method for i-vector extraction for speaker verification. This type of extraction results in a d-vector. The d-vector can be used independently or in conjunction with the traditional i-vector. The proposed method is divided into two steps: DM training and d-vector extraction.

3.1. DM training

In this step we train the DM model. The input to this phase is a development set of g-vectors (adapted GMM supervectors means) $\{g_i\}_{i=1}^{n}$, where each g-vector corresponds to a development session. First, following [10] we normalize each g_i as follows :

$$x_i=(w^{1/2}\otimes I_m)\Sigma_d^{-1/2}(g_i-\mu_d)$$

where μ_d is the $\{g_i\}_{i=1}^{n}$ sample mean vector, Σ_d is the $\{g_i\}_{i=1}^{n}$ diagonal sample covariance matrix, w is the vector of stacked mixture GMM weights, $\otimes$ is the Kronecker product and I_m is the identity matrix of size m, which is the g-vector dimension. Note that this type of normalization generates a new set of normalized g-vectors $\{x_i\}_{i=1}^{n}=G_d\subseteq G$. Then, by applying the DM algorithm to G_d, we learn the structure of the underlying speaker manifold that resides on G. This is done by defining a mapping $M_t:G_d\rightarrow D$ as described in Section 2. In this work we chose to use the following affinity kernel:

$$k(x_i,x_j)=\exp\left(-\frac{1-c(x_i,x_j)^2}{\sigma}\right) \tag{3}$$

where $x_i,x_j\in G$ and $c(\cdot,\cdot)$ is the cosine distance. In this way each g-vector is mapped to a corresponding l - dimensional d-vector.

The computational complexity of the training phase is reduced to the complexity of spectral decomposition of *P*. Note that the decomposition is carried out only for the first *l* eigenvectors and eigenvalues. The size of *P* is determined by the size of the development set.

3.2. D-vector extraction

As mentioned in Section 2, the mapping M_t is defined only on the domain G_d (the normalized development set). Therefore, in case of a new test g-vector, $x\in G\setminus G_d$, M_t has to be extended to $M_t^x:G_d\cup\{x\}\rightarrow D$ in order to estimate the new coordinates of x in D. For this task we use the Nystrom extension according to Eq. (2).

The computational complexity of d-vector extraction is determined by the size of the development set and the complexity of the chosen diffusion kernel.

4 Experimental Framework and Results

4.1 Front-end

The front-end we use throughout this paper 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 [1] is applied with a 300 frame window. We use a GMM order of 1024 for estimating sufficient statistics for i-vector extraction and for estimation of supervectors for d-vector extraction.

4.2. PLDA

PLDA [9,11] jointly models speaker and channel variability in the i-vector (or d-vector) space. A speaker and channel dependent i-vector (or d-vector) can be defined as

$$w=\overline{w}+\mathrm{V}y+\mathrm{U}x+\varepsilon \tag{4}$$

where w denotes the observed i-vector (d-vector), $\overline{w}$ is a global mean i-vector (d-vector), y and x are the speaker and channel factor respectively, V and U are the eigenspeaker and eigenchannel matrices. ε is a residual vector that is assumed to be distributed according to the standard normal distribution.

The PLDA model is trained on a development data for a given eigenspeaker rank and a given eigenchannel rank. In verification phase, the verification score has a closed form expression which can be found in [11].

4.3 Evaluated systems

Our baseline system is a gender-dependent i-vector based Gaussian-PLDA system inspired by [11]. We set the dimension of the i-vectors to 400. The Gaussian-PLDA backend processes length-normalized i-vectors by first applying LDA for obtaining a dimensionality reduction to 250. The PLDA model we use is configured to have 200 eigenspeakers and 200 eigenchannels. We do not apply any sort of score normalization (as we found score normalization to degrade accuracy).

The gender-dependent d-vector based PLDA system is similar to the i-vector based PLDA system, except for the substitution of the i-vectors with d-vectors. In the DM training phase we chose the following set of parameters: l (dimension) = 800, $\sigma = 6$ and $t = 1$. We found out that 300 eigenspeakers and 300 eigenchannels were optimal for our setup.

The third system fuses between the i-vector based PLDA and the d-vector based PLDA systems by applying a simple average (with equal weights) to the score level.

4.4. Datasets

We trained a gender-independent UBM on 12,711 sessions from Switchboard-II, NIST 2004 speaker recognition evaluation (SRE) and NIST-2006-SRE. For training the i-vector and d-vector extractors we used 16989 (female) and 11145 (male) telephone sessions from NIST 2004-2006 and 2008 SREs. We ran experiments on the telephone-only condition 5 of NIST-2010-SRE [12].

4.5. Results

Table 1 presents comparisons of the baseline i-vector based PLDA system with the proposed d-vector based PLDA system and the fused system. The results are measured in Equal Error Rate (EER), old-minDCF [12] and new-minDCF [12].

Table 1. *A comparison of i-vector PLDA to d-vector PLDA and the fused system on NIST-2010 telephone only condition 5.*

System	EER (%)	Old min-DCF	New min-DCF
Males			
i-vector PLDA	2.5	0.138	0.507
d-vector PLDA	2.3	0.131	0.307
Fused system	1.7	0.103	0.279
Females			
i-vector PLDA	2.7	0.132	0.431
d-vector PLDA	2.3	0.127	0.322
Fused system	2.0	0.096	0.291

Table 2 summarizes the gains we get using the d-vector based PLDA system and using the fused system compared to the baseline i-vector based PLDA system. We see that the d-vector based PLDA system improves over the baseline by an average of 11.5%, 4.5% and 32.5% for EER, old-minDCF and new-minDCF respectively, and the fused system improves over the baseline by 29%, 26.5% and 39% for EER, old-minDCF and new-min-DCF respectively.

Table 2. *Summary of the improvements for the d-vector PLDA system and the fused system compared to the baseline i-vector PLDA system.. Results are in relative improvement (%)*

Measure	d-vector PLDA system	Fused system
Males		
EER	8	32
Old min-DCF	5	25
New min-DCF	40	45
Females		
EER	15	26
Old min-DCF	4	28
New min-DCF	25	33

5 Conclusion

In this paper, we presented the d-vector extraction algorithm. This algorithm can be used as a non-linear alternative to the traditional i-vector extraction algorithm. We demonstrated the effectiveness of d-vector extraction algorithm when it is used as a front-end layer for a PLDA based speaker recognition system.

We managed to obtain reduced error rate using the d-vector based method compared to using i-vectors. The error reduction was in the range of 4-40%, depending on the gender and error measure.

Furthermore, a simple fusion of the d-vector based system and the i-vector based system resulted in significant error reductions of 25-45% compared to the baseline, depending on the gender and error measure.

.

References

[1] J. Pelecanos and S. Sridharan, “Feature warping for robust speaker verification,” in *Proc. of Speaker Odyssey Workshop*, 2001.

[2] Z. N. Karam and W. M. Campbell, “Graph-embedding for speaker recognition,” in Proc. *Interspeech*, 2010.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” *IEEE Transactions on Audio, Speech, and Lanuage Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] A. Singer, ``From graph to manifold Laplacian: the convergence rate", Applied and Computational Harmonic Analysis, 21 (1), 135-144 (2006).

[5] Fowlkes C., Belongie S., Chung F., Malik J., "Spectral Grouping Using the Nyström Method", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pp. 214-225, Feburary, 2004.

[6] R.R. Coifman and S. Lafon, "Diffusion maps". Applied and Computational Harmonic Analysis, 21(1):5–30, 2006.

[7] J.B. Tenenbaum, V. de Silva, and J.C. Langford. "A global geometric framework for nonlinear dimensionality reduction". Science, 290(5500):2319–2323, 2000.

[9] S. J. D. Prince, “Probabilistic linear discriminant analysis for inferences about identity”, in Proc. *International Conference on Computer Vision (ICCV)*, 2007.

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

[11] S. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of Ivector Length Normalization in Speaker Recognition Systems", in Proc. *Interspeech*, 2011.

[12] NIST 2010 Speaker Recognition Evaluation Plan, available online: http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf.