DIFFUSION MAPS FOR PLDA-BASED SPEAKER VERIFICATION

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

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

ABSTRACT

During the last few years, i-vectors have become an important component in most state-of-the-art speaker recognition systems. I-vector extraction is based on an assumption that GMM supervectors reside on a low dimensional space, which is modeled using Factor Analysis. In this paper we replace the above assumption with an assumption that the GMM supervectors reside on a low dimensional manifold and propose to use Diffusion Maps to learn that manifold. The learnt manifold implies a mapping of spoken sessions into a modified i-vector space which we call d-vector space. D-vectors can further be processed using standard techniques such as LDA, WCCN, cosine distance scoring or Probabilistic Linear Discriminant Analysis (PLDA). We demonstrate the usefulness of our approach on the telephone core conditions of NIST 2010, and obtain significant error reduction.

Index Terms— Speaker verification, Diffusion Maps, i-vectors, non-linear dimensionality reduction

1. INTRODUCTION

During the last few years i-vectors have become the standard front-end layer in most of state-of-the-art speaker verification systems. In [3] the authors showed that most of the speaker variability in the high dimensional GMM space may be captured by a low dimensional subspace named as the Total Variability space.

Therefore, the i-vector framework provides a way to map the high dimensional GMM supervectors to a relatively low dimensional vectors, named i-vectors. In a common setup, i-vectors are used as a front-end processing which is followed by a subsequent chain of linear projections, LDA and WCCN [3]. Recently, an even more successful technique named PLDA [9, 10] has been introduced to the speaker recognition community and currently i-vector extraction followed by PLDA is regarded as an extremely robust and accurate framework for speaker verification.

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 [8] it is possible to improve classification error. A further attempt for non-linear dimensionality reduction for speaker recognition has been done in [4]. In that work, the authors used a manifold learning technique named Diffusion Maps (DM) [6], however they abandoned the GMM framework. Instead, they used 78 dimensional feature vectors consisting of MFCC and delta MFCC mean, variance, min and max statistics extracted from a session.

In this paper we 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. 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 Evaluation data.

Our work differs from the previous ones in several aspects. First, unlike [8], we choose to use the DM framework due to its better modeling of relations between data points and the relatively easy way it can be extended to new data points using geometric harmonics [7]. Secondly, as opposed to the approach presented in [4] we do use the GMM framework as a baseline representation of the data and consequently use a more appropriate metric function than a simple Euclidean distance in order to model the relations between the GMM supervectors. Lastly, in both [4] and [8], the experiments focused on clustering and data mining tasks which are based on the assumption that the evaluation data or part of it is given a priori to the recognition system. Hence, these setups are not suitable for the scenario when the evaluation data is unknown and is given only on test time.

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

2. DIFFUSION MAPS

Diffusion Maps (DM) [6] is a machine learning technique for non-linear dimensionality reduction. Different from other dimensionality reduction methods such as principle component analysis (PCA), multi-dimensional scaling (MDS), and factor analysis (FA), DM is a non-linear method that focuses on discovering the underlying manifold that the data has been sampled from.

In this method an affinity matrix is built which is used to generate a diffusion process. As the diffusion process progress, it integrates local geometry 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 note 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 metric $c(x_i, x_j)$ over G. Then this metric should be converted to a similarity measure. A common approach for this type of conversion is using the Gaussian kernel:

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

where 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. Therefore, for an intricate, non linear and dense structure, σ should be set to a small value, while for a sparse structure large values might be considered as more suitable.

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 (1). We then define a random walk on this graph by converting the similarity measure to a probability function as follows: $p(x_i, x_j) = k(x_i, x_j) / z(x_i)$ where $z(x_i) = \sum_{j=1}^{n} k(x_i, x_j)$. The next step is to define 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 $P_{i,j}^t$ is the probability of transition from node x_i to node x_j in t steps.

Based on the above construction, a diffusion distance after t steps is defined as follows: $Q_t(x_i, x_j) = \sum_{k=1}^{n} (P_{i,k}^t - P_{kj}^t)^2$. By spectral decomposition of P we get a complete set of eigenvalues $1 = \lambda_0 \geq \lambda_1 \geq ... \geq \lambda_n$ and left and right eigenvectors satisfying: $P\psi_i = \lambda_i \varphi_i$. We then define a mapping $M_t : \{x_i\}_{i=1}^n \to D$ according to: $M_t(x_i) = \left[\lambda_1^t \psi_{1i}, ..., \lambda_l^t \psi_{li}\right]^T$, 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. It has been shown [6] 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 justified the use of squared Euclidean distance in the diffusion space. Of course in practice one should pick $l < m - 1$ according to the decay of $\{\lambda_i\}_{i=1}^n$. This decay is related to the complexity of the intrinsic dimensionality of the data and the choice of the parameters σ and t.

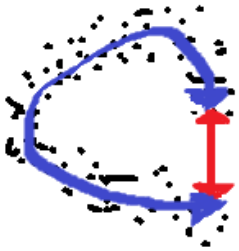

Figure 1: *Travelling along the blue path (which follows the intrinsic geometry of the manifold) has higher probability than travelling along the red path as the diffusion process progress.*

Figure 1 shows an illustrative example for a diffusion process. Two alternative paths are connecting between points on the manifold. The blue path is the longer one but is the one which 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 blue path also increases, since it consists of many short distance jumps. However, the probability of travelling along the red path stays always small (and become smaller and smaller as the t increases) as it is consists of a long distance jumps.

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 the beginning. Although this might be practical in offline applications, it is extremely inefficient and results in large amount overhead. This out-of-sample extension problem is well-studied in the manifold learning community and recently, many successful methods [5] have been proposed to alleviate it. In this work we chose to use the Geometric harmonic [7] approach which is based on Nystrom extension:

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

This extension extends each of 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: $M_t^{x_{n+1}} : \{x_i\}_{i=1}^{n+1} \to D$.

3. D-VECTOR EXTRACTION FOR SPEAKER VERIFICATION

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. This d-vector can be used independently or in conjunction with the traditional i-vector. The proposed method is divided into two phases: DM training and d-vector extraction.

3.1. DM training

In this phase we train the DM model. The input to this phase is a development set of g-vectors (GMM supervectors means) $\{g_i\}_{i=1}^{n}$, where each g-vector corresponds to a development session. First, following [13] we normalize each g_i as follows: $x_i = \Sigma_d^{-1/2}(w_i^{1/2} \otimes I_m)(g_i - \mu_d)$ 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_i is the vector of stacked mixture GMM weights corresponding to the *i*-th g-vector, $\otimes$ 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 in G. This is done by defining a mapping $M_t : G_d \to D$ as described in Section 2. In this work we chose to use the following diffusion kernel:

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

where $x_i, x_j \in G$ and $c(\cdot,\cdot)$ is the cosine distance [3]. 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 the transition matrix *P*. Note that the decomposition is carried out only for the first *l* eigenvectors and eigenvalues, for a selected parameter *l*. The size of *P* is determined by the size of the development set.

3.2. D-vector extraction

As explained 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 $M_t^x : G_d \cup \{x\} \to D$ in order to estimate the new coordinates of x in D. For this task we use the geometric harmonic technique described in Section 2. Therefore, as already explained in Section 2, the diffusion distance between a pair of g-vectors in G can be approximated by the squared Euclidean distance between the corresponding pair of d-vectors in D.

The computational complexity of d-vector extraction is determined by the size of the development set and the complexity of the chosen diffusion kernel. Given a new g-vector the extension is done according to (2). Therefore it is necessary to compute a new row in the transition matrix which is corresponding to the probability of jumping from the new test g-vector to each of the development g-vectors separately. It is important to clarify that other extension methods can be also considered. For example, the method proposed in [5] have been proved to be more robust and computational efficient than the one used in our paper.

4. EXPERIMENTAL SETUP 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 supervectors for d-vector extraction.

4.2. Gaussian-PLDA

PLDA 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

$$w = \overline{w} + \mathrm{V}y + \mathrm{U}x + \varepsilon \quad (4)$$

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 I-vector PLDA baseline system

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 (600 dimensional i-vectors did not show improvement on our setup). 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).

4.4. D-vector PLDA system

The gender-dependent d-vector based PLDA system is similar to the i-vector based PLDA system described in the previous section 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 $t = 20$. We found out that 300 eigenspeakers and 300 eigenchannels were optimal for our setup.

4.5. The fused system

We fuse both the i-vector based PLDA system and the d-vector based PLDA system by doing a simple average (with equal weights) on the score level.

4.6. 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 three telephone-only core conditions of NIST-2010-SRE (5, 6, and 8).

4.7. Results

Tables 1-3 present comparisons of the baseline i-vector based PLDA system with the proposed d-vector based PLDA system. The results are measured in Equal Error Rate (EER) , old-minDCF [12] and new-minDCF [12] respectively.

Table 1. *A comparison of i-vector PLDA to d-vector PLDA on NIST-2010 telephone only conditions. Results are in EER (%)*

System	Condition 5	Condition 6	Condition 8
males			
i-vector PLDA	2.5	4.9	1.0
d-vector PLDA	2.3	2.9	1.6
Fused system	1.7	2.2	0.8
females			
i-vector PLDA	2.7	6.0	2.2
d-vector PLDA	2.3	4.4	2.2
Fused system	2.0	3.3	1.7

Table 2. *A comparison of i-vector PLDA to d-vector PLDA on NIST-2010 telephone only conditions. Results are in old min-DCF*

System	Condition 5	Condition 6	Condition 8
males			
i-vector PLDA	0.138	0.231	0.073
d-vector PLDA	0.131	0.192	0.045
Fused system	0.103	0.128	0.033
females			
i-vector PLDA	0.132	0.244	0.087
d-vector PLDA	0.127	0.224	0.065
Fused system	0.096	0.199	0.056

Table 3. *A comparison of i-vector PLDA to d-vector PLDA on NIST-2010 telephone only conditions. Results are in new-min-DCF.*

System	Condition 5	Condition 6	Condition 8
males			
i-vector PLDA	0.507	0.769	0.109
d-vector PLDA	0.307	0.696	0.192
Fused system	0.279	0.617	0.150
females			
i-vector PLDA	0.431	0.758	0.242
d-vector PLDA	0.322	0.814	0.238
Fused system	0.291	0.781	0.179

Table 4 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 18.5%, 13.5% and 9% for EER, old-minDCF and new-minDCF respectively, and the fused system improves over the baseline by 40%, 32% and 18.5% for EER, old-minDCF and new-min-DCF respectively.

Table 4. *Summary of the improvements for the d-vector PLDA system and the fused system compared to the baseline i-vector PLDA system. Results for the separate conditions are averaged. Results are in relative improvement (in %)*

Measure	d-vector PLDA system	Fused system
males		
EER	19	44
Old min-DCF	17	40
New min-DCF	14	24
females		
EER	18	36
Old min-DCF	10	24
New min-DCF	4	13

5. CONCLUSION

In this paper we introduced the use of Diffusion Maps for the application of i-vector extraction for speaker verification systems. 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 verification system.

We managed to obtain reduced error using the d-vector based method compared to using i-vectors. The error reduction was in the range of 4-19%, 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 error reductions of 13-44% compared to the baseline, depending on the gender and error measure.

.

6. 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] Y. Michalevsky, R. Talmon, and I. Cohen. "Speaker identification using diffusion maps", in Proc. *Eusipco*, 2011.

[5] A. Bermanis, A. Averbuch, and R.R. Coifman. "Multiscale data sampling and function extension", Applied and Computational Harmonic Analysis, 2012. DOI:10.1016/j.acha.2012.03.002.

[6] R.R. Coifman and S. Lafon, "Diffusion maps". Applied and Computational Harmonic Analysis, 21(1):5–30, 2006.
[7] R.R. Coifman and S. Lafon, "Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions", Applied and Computational Harmonic Analysis, 21(1):31–52, 2006.
[8] 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] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, 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.
[13] W. Campbell, Z. Karam, "Simple and Efficient SpeakerComparison using Approximate KL Divergence", in Proc. *Interspeech*, 2010.