Odyssey 2016
21-24 Jun 2016, Bilbao, Spain

From Features to Speaker Vectors by means of Restricted Boltzmann Machine Adaptation

Pooyan Safari, Omid Ghahabi, Javier Hernando

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain
pooyan.safari@tsc.upc.edu,{omid.ghahabi,javier.hernando}@upc.edu

Abstract

Restricted Boltzmann Machines (RBMs) have shown success in different stages of speaker recognition systems. In this paper, we propose a novel framework to produce a vector-based representation for each speaker, which will be referred to as RBM-vector. This new approach maps the speaker spectral features to a single fixed-dimensional vector carrying speaker-specific information. In this work, a global model, referred to as Universal RBM (URBM), is trained taking advantage of RBM unsupervised learning capabilities. Then, this URBM is adapted to the data of each speaker in the development, enrolment and evaluation datasets. The network connection weights of the adapted RBMs are further concatenated and subject to a whitening with dimension reduction stage to build the speaker vectors. The evaluation is performed on the core test condition of the NIST SRE 2006 database, and it is shown that RBM-vectors achieve 15% relative improvement in terms of EER compared to i-vectors using cosine scoring. The score fusion with i-vector attains more than 24% relative improvement. The interest of this result for score fusion yields on the fact that both vectors are produced in an unsupervised fashion and can be used instead of i-vector/PLDA approach, when no data label is available. Results obtained for RBM-vector/PLDA framework is comparable with the ones from i-vector/PLDA. Their score fusion achieves 14% relative improvement compared to i-vector/PLDA.

1. Introduction

Gaussian Mixture Models (GMMs) are the basis of many state-of-the-art speaker modeling techniques. They are used in an adaptation process in the conventional GMM-UBM method for speaker recognition. By concatenating the mean vectors obtained from Maximum a Posteriori (MAP) adapted GMMs, a high-dimensional vector called supervector is formed. These high-dimensional supervectors can be converted into lower-dimensional vectors by means of an effective Factor Analysis (FA) technique renowned as i-vector [1]. These i-vectors can be employed in classification for speaker recognition using cosine distance similarity or Probabilistic Linear Discriminant Analysis (PLDA) [2, 1, 3].

Restricted Boltzmann Machines (RBMs) are generative models able to efficiently learn via unsupervised learning algorithms. They have recently shown success in applications such as audio and speech processing (e.g., in [4, 5, 6]). In speaker recognition, they were used to extract features [7], and speaker factors [8], and to classify i-vectors [9, 10]. They have been utilized in an adaptation process [11, 12, 13, 14], to further discriminatively model target and impostor speakers. RBMs have been recently used in DBNs as a pre-training stage to extract Baum-Welch statistics for i-vector and supervector extraction [15, 16]. RBMs were used in [17] prior to PLDA, as a transformation stage of i-vectors, to build a more suitable discriminative representation for the supervised classifier. It is also worth noting that recently different methods have been proposed to incorporate Deep Neural Networks (DNNs) into the context of speaker recognition. In [18, 19], DNNs were used to extract an enriched vector representation of speakers for text-dependent speaker verification. In [20], DNNs were employed to extract a more discriminative vector from i-vector. They have been used in [21] to collect sufficient statistics for i-vector extraction. There were also few attempts addressing methods to produce alternative vector-based speaker representation using RBMs, some of which, show moderate success compared to i-vector [22, 23].

Motivated by the representational power of RBMs, we propose a novel framework to produce a vector-based representation for each speaker, which is referred to as RBM-vector. This new framework maps the speaker spectral features to a single fixed-dimensional vector conveying speaker-specific information. Although there were attempts trying to use RBMs to produce a vector-based representation of speakers [22, 23], this work is distinct from different perspectives. In [22, 23] statistical properties of the hidden units likelihoods were used to build the vector representation for speakers. However, in this paper, the connecting weights between hidden and visible units are used to form the RBM-vectors. Furthermore, in [23], an RBM is used to transform the GMM supervectors to smaller-dimensional ones which can be later used for classification with cosine similarity. However, in this work, the proposed method is directly applied to the speech spectral features, without using GMM-UBM approach to produce supervectors.

Taking advantage of RBM unsupervised learning algorithm, a global model is trained, which will be referred to as Universal RBM (URBM). This URBM model is further adapted to each speaker's data. The connection weights of each adapted RBM model are concatenated to produce RBM-vectors. The produced RBM-vectors are classified using cosine distance or PLDA approach. The experimental results show that with cosine scoring, RBM-vector outperforms conventional i-vector by 15% relative improvement, which achieves to more than 24% by score fusion. In the PLDA framework, the performance of RBM-vector is comparable to i-vector, while their score fusion attains 14% relative improvement compared to using only i-vector.

This work has been supported by the Spanish Ministry of Economy and Competitiveness under contract PCIN-2013-067, the Generalitat de Catalunya under contract 2014-SGR-1660, and by the project TEC2015-69266-P (MINECO/FEDER, UE).

doi: 10.21437/Odyssey.2016-53

2. Background

Over the past few years, i-vector approach as a fixed- and low-dimensional representation of speech utterances has dominated the speaker recognition technologies [1]. In this method, a Factor Analysis (FA)-based approach is applied to supervectors. One of the most successful supervectors is obtained by concatenating the D-dimensional mean vectors of an M-mixture adapted GMM. In other words, each variable length utterance is represented by a $(M \times D)$-dimensional GMM supervector. The speaker- and session-dependent GMM supervectors are further used to train the total variability matrix [1]. This low-rank matrix is trained using Baum-Welch statistics of all available speech utterances. Total variability matrix is adopted to estimate a low-dimensional vector representation for each speaker, called i-vector. The obtained i-vectors can be classified by simply using cosine similarity, which is an effective distance metric when no speaker label is available for development data [1], or employing a more complex classifier such as PLDA [24, 3].

On the other hand, representational abilities of Restricted Boltzmann Machines (RBMs) along with their simple unsupervised learning algorithm raise them as a potential alternative to the previously mentioned methods. RBMs are generative models composed of two fully connected visible and hidden layers, where there is no intra-layer connection between units as illustrated in Fig. 1a. Units can be stochastic binary or Gaussian real-valued. They can be trained using an approximated version of Contrastive Divergence (CD) algorithm called CD-1 [25]. CD-1 is a three-step procedure (Fig. 1b). First, given the visible units values, the values of the hidden units are computed with their posterior probability distribution. In the second stage, given the values of the hidden units, the visible units values are reconstructed. It should be mentioned that the hidden unit likelihoods are converted to binary values before being used in the second stage [26]. In the third stage, once more the hidden values are computed given the reconstructed values of the visible units. After completing this procedure, the network weights will be modified by:

$$\Delta w_{ij} \approx -\alpha \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recons} \right) \qquad (1)$$

where α is the learning rate, w_{ij} is the connection weight between visible unit i and hidden unit j, $\langle v_i h_j \rangle_{data}$ and $\langle v_i h_j \rangle_{recons}$ denote the expectations when the hidden state values are driven, respectively, from the input data and the reconstructed one. This process is iterated until the algorithm converges. Each iteration is called an epoch. In order to accelerate the parameter updating process, it is recommended to divide the whole training dataset into smaller ones, called mini-batches [26].

3. Proposed Method

In this paper, we propose a new framework using RBMs, to produce an alternative vector-based representation for speakers, referred to as RBM-vectors. Figure 2 shows the block diagram of the RBM-vector extraction process. Employing speech spectral features, an RBM model is trained based on the background data. It is then adapted to the data of each speaker in order to build a model per speaker. These models are adopted to form RBM-vactors. These vectors can be further used for speaker verification using cosine distance metric or conventional compensation strategies such as PLDA. The overall process can be considered as three main stages, namely Universal RBM training, RBM adaptation, and vector extraction using parameters of the adapted RBM models. All these stages will be addressed in detail in the following subsections.

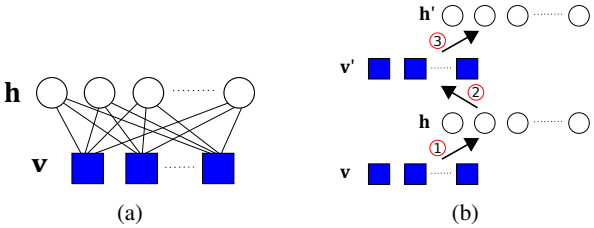

Figure 1: RBM (a), and RBM training using CD-1 algorithm (b).

3.1. Universal RBM

The first step towards building the proposed speaker vector representation is to train a universal model based on all available background data, which conveys the speaker-independent information. This is carried out by training a single RBM given the spectral features extracted from all background utterances. The binary hidden units are chosen for the RBM. However, due to the fact that the features are real-valued data, we use Gaussian real-valued units for observed variables. The CD-1 algorithm for Gaussian-Bernoulli RBMs works under the assumption that the inputs have zero mean and unit variance [26]. Therefore, a cepstral mean-variance normalization (CMVN) is applied to the features of each utterance prior to RBM training (Fig. 2). The obtained RBM model is referred to as Universal RBM (URBM). URBM represents the general, speaker-independent model. It is assumed that URBM is able to learn both speaker and session variabilities from the background data. It should be built using whole available background samples (feature vectors) in order to cover a wide range of speaker and channel variabilities. However, due to resource limitations we randomly select as many background feature vectors as possible for training.

3.2. RBM adaptation

In order to build a speaker-specific model for each speaker, it is proposed to incorporate speaker-dependent information into the obtained universal model (URBM). This is carried out by means of an adaptation stage. For each speaker, the speaker adaptation is performed by training an RBM model with a few number of iterations using data samples of the corresponding speaker. The parameters of this RBM model, such as weights and biases, are initialized by the ones obtained from the URBM. In other words, the URBM is adapted to the data of each speaker. The idea of this kind of adaptation has also shown success in [11, 12, 13, 14] to initialize the parameters of DNNs for classification purposes.

Figure 3 shows the weight matrices for URBM along with its adapted versions for two randomly selected speakers. This speaker adaptation, modifies the weights of the universal model. As it can be seen from the Fig. 3, the weights of the two selected speakers after adaptation are distinct one from another. It should be noted that in comparison to URBM training, fewer number of epochs is used for the adaptation procedure. This is important in order to avoid overfitting. This also makes the training time much less than what is needed for training a speaker-specific model without adaptation.

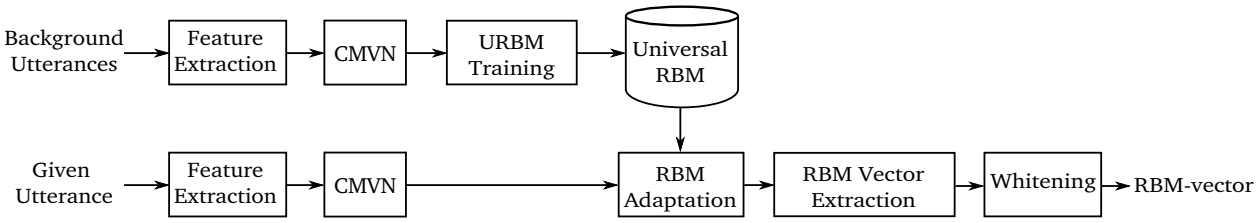

Figure 2: Block diagram showing different stages of RBM-vector extraction process. CMVN is a speaker dependent cepstral mean-variance normalization.

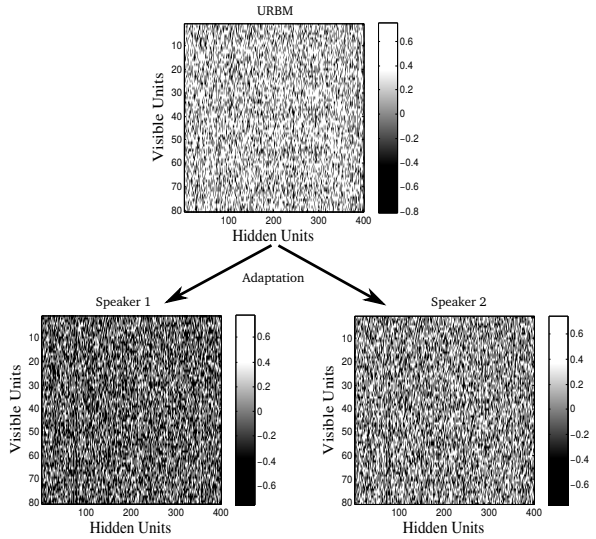

Figure 3: Comparison of the adapted weight matrices. The URBM weight matrix, obtained using the background data, is adapted to the data of two different speakers. It shows that the connection weights convey the speaker-specific information.

3.3. RBM vector extraction

Once the adaptation step is completed, an RBM model is assigned to each speaker. There may be different scenarios available in order to build a fixed-dimensional vector representation per speaker using these models. One naive approach is to feed the data of each speaker into its corresponding RBM model, and then use the statistical properties of the outputs to form a vector. In this method it is assumed that the outputs of each model, which are posterior distributions, convey speaker-specific information and the estimates of the first and second order moments can lead us to a vector-based representation.

For each speaker model the mean and variance of its outputs are computed and concatenated to build a single vector. In order to improve the discrimination power of these vectors, it is proposed to transform the outputs prior to the mean and variance computations. We have tried some different transformations among which we have chosen the logarithm function as in [23]. The logarithm function maps the output of the sigmoid function, that is within 0 and 1, to a broader interval from $-\infty$ to 0. The obtained vectors will be mean-normalized prior to whitening. The whitening with dimension reduction is carried out by means of Principle Component Analysis (PCA). These whitened vectors are now ready for classification purposes with cosine similarity or PLDA. Experimental results showed that using higher-order moments could not considerably improve the performance for this kind of vectors.

Another proposed idea in this work is to utilize the RBM model parameters such as hidden-visible connection weights $\mathbf{W}$ and biases to build the speaker vector. As illustrated in Fig. 3, the weights carry speaker-specific information which are distinct enough one from another, to be used for speaker recognition. The rows of the weight matrix along with the bias vectors are concatenated to form a high-dimensional RBM-vector. The obtained vectors will be subject to a mean-normalization prior to PCA whitening with dimension reduction. PCA is trained using background speaker vectors and then applied to all the background, target, and test vectors. Whitening transformation rotates the original data to the principal component space in which the rotated data components are less correlated,

$$\mathbf{\Lambda}_{L\times M} = \left(\mathbf{S}_{1:L\times 1:L} + \varepsilon\right)^{-1/2} \mathbf{U}_{1:L\times M} \qquad (2)$$

where $\mathbf{\Lambda}$ is the transformation matrix which is multiplied by the original data for whitening and dimension reduction, $\mathbf{U}$ is the matrix of eigenvectors, $\mathbf{S}$ is the diagonal matrix of the corresponding eigenvalues, M and L are the values for the dimension of original and shortened vectors, respectively. A small constant of ε is added, for regularization, to avoid large values in practice. The values for L and ε must be set experimentally to optimize the results.

The output of the whitening stage is called RBM-vector and similar to i-vector can be used for speaker verification using cosine similarity or PLDA. In the next section, it will be shown that using weights to build the speaker-specific vectors results in a more discriminative vector-based representation for speaker verification compared to the first scenario as mentioned earlier. It is also shown that this representation is able to outperform the conventional i-vector approach.

4. Experimental Results

4.1. Database and setup

From our experience in [14], it has been decided to work on the spectral features instead of Filter-Bank Energy (FBE) features. Frequency Filtering (FF) [27] features, have been used as spectral features in this work. FF features, like MFCCs, are a decorrelated version of FBEs [27]. It has been shown that FF features achieve equal or better performance than MFCCs [27]. They are extracted every 10 ms with a 30 ms Hamming window. The size of static FF features is 16. Before feature extraction, speech signals are subject to an energy-based silence removal process. All the features are mean-variance normalized per each utterance. The whole core test condition of the NIST 2006 SRE evaluation [28] is considered in all the experiments. It comprises 816 target speakers, with $51,068$ trials. Each signal consists of about two minutes of speech. Performance is

evaluated using the Equal Error Rate (EER) and the minimum Decision Cost Function (minDCF) calculated using $C_M = 10$, $C_{FA} = 1$, and $P_T = 0.01$.

The performance of the proposed approach is compared with the i-vector baseline system using either cosine similarity scoring or PLDA. The gender-independent UBM is represented as a diagonal covariance, 512-component GMM. ALIZE open source software [29] is used to extract 400-dimensional i-vectors. The development data includes 6125 speech files collected from NIST 2004 and 2005 SRE corpora. It is worth noting that in the case of NIST 2005 only the speech files of those speakers which do not appear in NIST 2006 database are used. The same development data is used to train UBM, T matrix, whitening matrices, PLDA, and URBM. The PLDA for the i-vector/PLDA baseline is trained with 15 iterations and the number of eigenvoices is empirically set to 250. It should be mentioned that both i-vectors and RBM-vectors are length-normalized prior to training PLDA.

Since in this work it is necessary to train a network per speaker, we try to reduce the computational complexity considering only 5 neighbouring frames (2-1-2) of the features in order to compose 80-dimensional feature inputs for the networks. All the RBMs used in this paper comprise 400 hidden units. The fixed momentum and weight decay for both URBM and adapted RBMs are set to 0.91 and 0.0002, respectively. The URBM is trained by a learning rate of 0.0001 with 200 epochs. The minibatch size of 100 is used for training both URBM and RBM adaptation. The URBM should be trained based on all available background data which is here about 60 million feature vectors. However, due to the resource limitations we have done a random sample selection prior to training and reduced the number of feature vectors to 8 million. As mentioned in section 3, the URBM model is adapted to the data of each speaker to build the speaker-specific models. This adaptation process is carried out by 5 epochs of CD-1 algorithm and a learning rate of 0.005. For each adapted RBM, the connection weights and biases are concatenated. These obtained vectors are further sent to a whitening with dimension reduction stage using PCA to decrease the dimension and the correlation between RBM-vector components. As mentioned in section 3.3, a regularization constant ε is considered as a hyperparameter for whitening to avoid numerical instability. This value is set to 0.0005 for all the experiments reported in this section. The optimum weights for the score-level fusion has been set heuristically. For cosine scoring, these weights were set to 0.35 and 0.65 for i-vector and RBM-vector, respectively. In the case of PLDA, they were set to 0.65 and 0.35 for i-vector and RBM-vector, respectively.

4.2. Results

RBM-vectors of different sizes have been evaluated using cosine similarity. The results are shown in Table 1. The performance of RBM-vector of size 400 is comparable to the i-vector of equal length. The 600-dimensional RBM-vectors slightly perform better than our baseline. By increasing the length of the RBM-vector to 800, we achieve 6% relative improvement. However, by increasing the size to 2000, it outperforms the conventional i-vector by 15% relative improvement. The last row in the table shows the score-level fusion of the i-vector technique and RBM-vector of size 2000, which is more than 24% relative improvement compared to using only i-vector. This is important particularly when no data label is available to perform supervised compensation techniques such as PLDA. It should be mentioned here that RBM-vectors perform much better than

Table 1: Comparison of the i-vector with different RBM-vectors in terms of EER%, minDCF, and vector dimension. Results obtained on the core test condition of NIST SRE 2006 using cosine scoring. The fusion is applied on score level.

Technique	EER (%)	minDCF
i-vector (400)	7.01	0.0324
RBM-vector (400)	7.26	0.0341
RBM-vector (600)	6.77	0.0327
RBM-vector (800)	6.58	0.0320
RBM-vector (2000)	**5.98**	**0.0289**
Fusion i-vector (400) & RBM-vector (2000)	**5.30**	**0.0278**

the first and second order moments of the RBM outputs. Using the mean of the RBM transformed outputs, as explained in section 3.3 to form speaker vector, resulted in an EER of 11.26%. This vector was further concatenated with the variance vector and resulted in an EER of 11.07%, which was not considerable improvement. This shows that the speaker-specific information lies more within the interaction of visible and hidden units, which is here considered as connection weights.

The dimension of the vectors are reduced to the length of L as mentioned in the previous section. This dimension affects the EER percentage. Figure 4 shows the impact of the size of RBM-vector on the performance of the system in terms of EER. The figure has been plotted upto $L = 4000$, however it is possible to obtain slightly better results at the cost of longer vectors, either using alone or in combination with i-vector as score fusion.

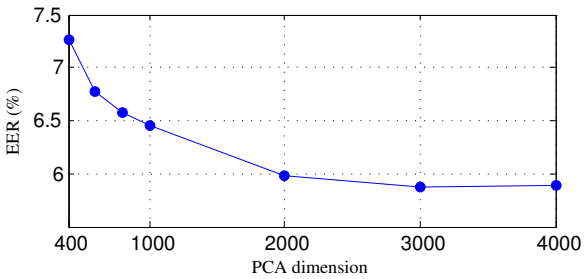

Figure 4: The impact of the size of RBM-vector on the performance of the system in terms of EER%. The performance are evaluated using cosine scoring.

PLDA is also applied to RBM-vectors and the results have been reported in Table 2. The PLDA is trained with 15 iterations and the number of eigenvoices are empirically set to 250, 350, 400, for RBM-vectors of sizes 400, 600, 800, respectively. The RBM-vectors are subject to length normalization prior to PLDA training. Using i-vector/PLDA shows an improvement of about 30% compared to i-vector/cosine framework. Comparing

Table 2: Comparison of the performance of PLDA with i-vector, and RBM-vectors of different dimensions, in terms of EER%, and minDCF. Results were obtained on the core test condition of NIST SRE 2006 evaluation. The fusion is applied on the score level.

Technique	EER (%)	minDCF
i-vector (400)	4.90	0.0263
RBM-vector (400)	5.55	0.0277
RBM-vector (600)	5.15	0.0276
RBM-vector (800)	5.42	0.0266
i-vector (400)+RBM-vector (600) Fusion	**4.21**	**0.0230**

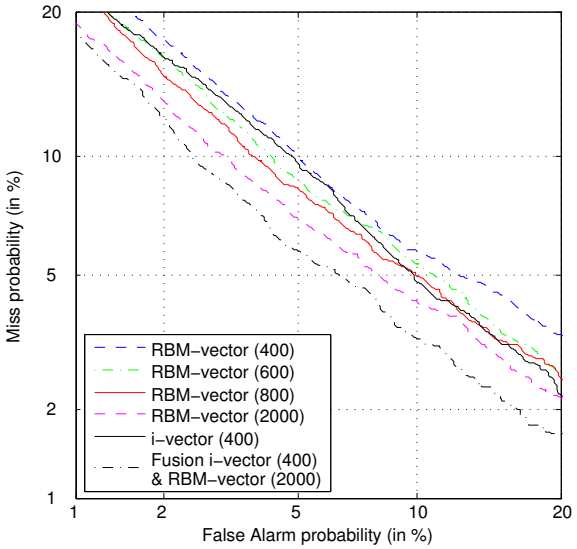

Figure 5: Comparison of DET curves for the proposed RBM-vectors with i-vector. The size of the RBM-vector is given in parenthesis. The score fusion of i-vector with RBM-vector of length 2000 is also illustrated. Results obtained on the core test condition of NIST SRE 2006 evaluation using cosine similarity.

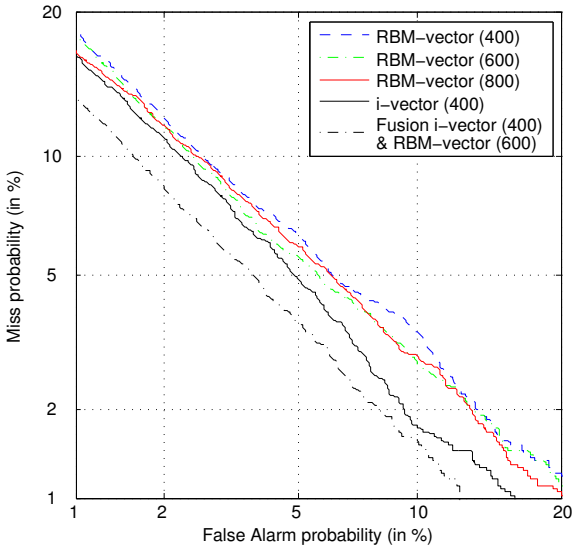

Figure 6: Comparison of DET curves for the proposed RBM-vectors with i-vector. The size of the RBM-vector is given in parenthesis. The score fusion of i-vector with RBM-vector of length 600 is also illustrated. Results obtained on the core test condition of NIST SRE 2006 evaluation using PLDA.

the results obtained by RBM-vector/PLDA framework with the ones from RBM-vector/cosine shows a relative improvement of 24%, 24%, and 18% for RBM-vectors of dimensions 400, 600, and 800, respectively. This reveals that PLDA as a compensation technique, is more suitable for i-vectors than RBM-vectors. This proposes a potential research direction to find more suitable compensation techniques for RBM-vectors.

Figure 5 compares the Detection Error Trade-off (DET) curves of RBM-vectors of different sizes with the i-vector baseline system in terms of cosine scoring. According to this figure, the RBM-vectors of size 400 works close to i-vector in the lower false alarm rate region, however, in the higher false alarm rate region they diverge from each other. The 600- and 800-dimensional RBM-vectors perform better than i-vector in the lower false alarm rates. It can be seen that RBM-vector of length 2000 consistently outperforms the i-vector in a wide range of operating points, especially those in the lower false alarm regions. The result obtained by the score fusion of our baseline with RBM-vector of size 2000 is also illustrated. It can be concluded that RBM-vector is better suited with lower false alarm rate regions compared to i-vector.

The PLDA performance of different RBM-vectors and i-vector are compared in Fig. 6, in terms of DET-curves. As it is illustrated, i-vector/PLDA performs better than RBM-vector/PLDA, in a wide range of working points. However, RBM-vector/PLDA converge to i-vector/PLDA, in lower false alarm rate regions. It should be mentioned that fusing the PLDA scores of i-vector with RBM-vector of size 600, consistently outperforms the i-vector/PLDA in all the operating points, particularly those in the lower false alarm areas.

5. Conclusions

We train a global model which is referred to as Universal RBM (URBM). The URBM is adapted to the data of each speaker in development, enrolment, and evaluation datasets. The connection weights of each adapted RBM model are concatenated to form a high-dimensional vector per speaker. This vector is further subject to a PCA whitening with dimension reduction in order to have a low-dimensional representation for each utterance, called here RBM-vector. This new fixed-dimensional vector conveys speaker-specific information and can be used in speaker recognition. The preliminary results on the core test condition of the NIST SRE 2006 database show that this new vector representation outperforms the conventional i-vector using cosine similarity by 15% relative improvement. The fusion with i-vector using cosine can improve more than 24%. The importance of this fusion arises from the fact that both vectors are produced in an unsupervised fashion, and can be used instead of i-vector/PLDA, when no data label is available. As expected, using PLDA instead of cosine similarity, improves the performance of RBM-vectors by 24% relative improvement in terms of EER. Finally, when fusing the RBM-vector/PLDA scores with the ones obtained by i-vector/PLDA a further improvement of 14% is attained compared to using only i-vector/PLDA.

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] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.

[3] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Keynote presentation, Odyssey Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.

[4] A-R. Mohamed and G.E. Hinton, "Phone Recognition us-

ing Restricted Boltzmann Machines," in *Proc. ICASSP*, 2010.

[5] N. Jaitly and G.E. Hinton, "Learning a Better Representation of Speech Soundwaves using Restricted Boltzmann Machines," in *Proc. ICASSP*, 2011.

[6] Z-H. Ling, L. Deng, and D. Yu, "Modeling Spectral Envelopes using Restricted Boltzmann Machines and Deep Belief Networks for Statistical Parametric Speech Synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2129–2139, 2013.

[7] H. Lee, P. Pham, and A.Y. Ng, "Unsupervised Feature Learning for Audio Classification using Convolutional Deep Belief Networks," in *Advances in neural information processing systems*, 2009.

[8] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "PLDA using Gaussian Restricted Boltzmann Machines with Application to Speaker Verification," in *Proc. Interspeech*, Portland, USA, September 2012.

[9] T. Stafylakis and P. Kenny, "Preliminary Investigation of Boltzmann Machine Classifiers for Speaker Recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Singapore, June 2012.

[10] M. Senoussaoui, N. Dehak, P. Kenny, and R. Dehak, "First Attempt of Boltzmann Machines for Speaker Verification," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Singapore, June 2012.

[11] O. Ghahabi and J. Hernando, "Deep Belief Networks for i-Vector Based Speaker Recognition," in *Proc. ICASSP*, Florence, Italy, May 2014.

[12] O. Ghahabi and J. Hernando, "i-Vector Modeling with Deep Belief Networks for Multi-Session Speaker Recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Joensuu, Finland, June 2014.

[13] O. Ghahabi and J. Hernando, "Global Impostor Selection for DBNs in Multi-Session i-Vector Speaker Recognition," in *Advances in Speech and Language Technologies for Iberian Languages*, pp. 89–98. Springer International Publishing, 2014.

[14] P. Safari, O. Ghahabi, and J. Hernando, "Feature Classification by means of Deep Belief Networks for Speaker Recognition," in *Proc. EUSIPCO*, Nice, France, August 2015.

[15] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep Neural Networks for Extracting Baum-Welch Statistics for Speaker Recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Joensuu, Finland, June 2014.

[16] W.M. Campbell, "using Deep Belief Networks for Vector-Based Speaker Recognition," in *Proc. Interspeech*, Singapore, May 2014.

[17] S. Novoselov, T. Pekhovsky, K. Simonchik, and A. Shulipa, "RBM-PLDA Subsystem for the NIST i-Vector Challenge," in *Proc. Interspeech*, Singapore, May 2014.

[18] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep Neural Networks for Small Footprint Text-Dependent Speaker Verification," in *Proc. ICASSP*, Florence, Italy, May 2014.

[19] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep Feature for Text-Dependent Speaker Verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[20] Y.Z. Isik, H. Erdogan, and R. Sarikaya, "S-Vector: A Discriminative Representation Derived from i-Vector for Speaker Verification," in *Proc. EUSIPCO*, Nice, France, August 2015.

[21] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A Novel Scheme for Speaker Recognition using a Phonetically-Aware Deep Neural Network," in *Proc. ICASSP*, Florence, Italy, May 2014.

[22] V. Vasilakakis, S. Cumani, P. Laface, and P. Torino, "Speaker Recognition by means of Deep Belief Networks," in *Proc. Biometric Technologies in Forensic Science*, 2012.

[23] O. Ghahabi and J. Hernando, "Restricted Boltzmann Machine Supervectors for Speaker Recognition," in *Proc. ICASSP*, Brisbane, Australia, April 2015.

[24] S.J.D. Prince and J.H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proc. ICCV 2007*, Rio de Janeiro, Brazil, Oct. 2007.

[25] G.E. Hinton, S. Osindero, and Y-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[26] G.E. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," *Momentum*, vol. 9, no. 1, 2010.

[27] C. Nadeu, Dušan Macho, and J. Hernando, "Time and Frequency Filtering of Filter-Bank Energies for Robust HMM Speech Recognition," *Speech Communication*, vol. 34, no. 1, pp. 93–114, 2001.

[28] "The NIST Year 2006 Speaker Recognition Evaluation Plan," Tech. Rep., 2006.

[29] A. Larcher, J-F. Bonastre, B. Fauve, K.A. Lee, C. Lévy, H. Li, J.S.D. Mason, and J-Y. Parfait, "ALIZE 3.0 - Open Source Toolkit for State-of-the-art Speaker Recognition," in *Proc. Interspeech*, Lyon, France, August 2013.