Deep Learning Backend for Single and Multisession i-Vector Speaker Recognition

Omid Ghahabi and Javier Hernando

***Abstract*—The lack of labeled background data makes a big performance gap between cosine and Probabilistic Linear Discriminant Analysis (PLDA) scoring baseline techniques for i-vectors in speaker recognition. Although there are some unsupervised clustering techniques to estimate the labels, they cannot accurately predict the true labels and they also assume that there are several samples from the same speaker in the background data that could not be true in reality. In this paper, the authors make use of Deep Learning (DL) to fill this performance gap given unlabeled background data. To this goal, the authors have proposed an impostor selection algorithm and a universal model adaptation process in a hybrid system based on deep belief networks and deep neural networks to discriminatively model each target speaker. In order to have more insight into the behavior of DL techniques in both single- and multisession speaker enrollment tasks, some experiments have been carried out in this paper in both scenarios. Experiments on National Institute of Standards and Technology 2014 i-vector challenge show that 46% of this performance gap, in terms of minimum of the decision cost function, is filled by the proposed DL-based system. Furthermore, the score combination of the proposed DL-based system and PLDA with estimated labels covers 79% of this gap.**

***Index Terms*—Deep learning, deep neural network, deep belief network, i-vector, speaker recognition.**

I. Introduction

THE recent compact representation of speech utterances known as i-vector [1] has become the state-of-the-art in the text-independent speaker recognition. There are two common scoring techniques to decide if two i-vectors belong to a same speaker namely cosine and Probabilistic Linear Discriminant Analysis (PLDA) [2], [3]. PLDA scoring leads to a superior performance but with the cost of need to speaker-labeled background data. Moreover, it needs several samples for each background speaker spoken in different session conditions to work efficiently. One of the recent challenges in speaker recognition, which was organized by the National Institute of Standards and Technology (NIST), has been how to fill the performance gap between these two common scoring techniques when no labeled background data is available [4]. Although there are some unsupervised automatic labeling techniques like those proposed in [5], [6], they cannot appropriately estimate the true labels and also they assume that there are several samples from a same speaker in the background data which could not be true in reality. PLDA with estimated labels performs reasonably well [5], [6], but the results are still far from that of PLDA with actual labels [7].

On the other hand, the success use of Deep Learning (DL) in speech processing, specifically in speech recognition (e.g., [8]–[12]), has inspired the community to make use of DL techniques in speaker recognition as well. Both generative approaches, like Restricted Boltzmann Machines (RBM) and Deep Belief Networks (DBN), and discriminative ones, like Deep Neural Networks (DNN), have been used for this purpose. A possible use of DL techniques in speaker recognition is to combine them with the state-of-the-art i-vector approach. Two kinds of combination have been considered. DL techniques have been used in the i-vector extraction process [13]–[17] or applied on i-vectors as a backend [18]–[23].

DNNs have been used in the i-vector extraction algorithm for two main goals. First, the Universal Background Model (UBM) is replaced by a DNN, which is typically trained for acoustic modeling in speech recognition [13], [14], [16], [24], [25]. Second, conventional spectral features are replaced or appended by so-called DNN bottleneck features [15], [16]. A significant performance gain is reported in both cases but it is shown that appending bottleneck features to spectral ones and using Gaussian UBM as the acoustic model will lead to higher quality i-vectors [15], [16].

Besides, after i-vector computation, DL techniques can be used for different purposes. For example, different combinations of RBMs have been proposed in [18], [19] to classify i-vectors and in [20] to learn speaker and channel factor subspaces in a PLDA simulation. RBMs in [26] and DNNs in [27] are used to increase the discrimination power of i-vectors given speaker-labeled background data. In [21]–[23] DBNs have been integrated in an adaptation process to provide a better initialization for DNNs in order to have discriminative target models. There are also some attempts to extract compact representations of speech signals given spectral features [28]–[30] and GMM supervectors [31].

In this work, the authors make use of deep architectures for backend i-vector classification in order to fill the performance

Manuscript received June 15, 2016; revised October 27, 2016 and January 8, 2017; accepted January 16, 2017. Date of publication February 8, 2017; date of current version March 1, 2017. This work was supported in part by the Spanish Project DeepVoice (TEC2015-69266-P) and in part by the European project CAMOMILE (PCIN-2013-067). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bin Ma.

The authors are with the TALP Research Center, Department of Signal Theory and Communications, Universitat Politecnica de Catalunya—BarcelonaTech, Barcelona 08034, Spain (e-mail: omid.ghahabi@upc.edu; javier.hernando@upc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2661705

2329-9290 © 2017 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

gap between the cosine (unlabeled-based) and PLDA (labeled-based) scoring baseline systems given unlabeled background data. As in [21], [22], the authors take advantage of unsupervised learning of DBNs to train a global model referred to as Universal DBN (UDBN) and DNN supervised learning to model each target speaker discriminatively. To provide a balanced training, an impostor selection algorithm and to cope with few training data, a UDBN-adaptation process is proposed.

Compared to [21], [22], deep architectures with different number of layers are explored for both single and multi-session speaker enrollment tasks. The parameters of the global model are normalized before adaptation. Normalization is just scaling down the parameters but it facilitates the training of the networks specifically where more than one hidden layer is used. The top layer pre-training proposed in [21] is not used in this work. The reason is that it emphasizes on the top layer connection weights and avoids the lower hidden layers to learn enough from the input data. This fact is of more importance when more hidden layers are used. In addition, new experiments based on unsupervised labeling techniques for PLDA [6] are performed in this paper as a potential baseline system when no labeled background data is available.

The preliminary experiments are performed on NIST SRE 2006 [32] to show the effect of each contribution. Taking advantage of the conclusions obtained on the preliminary experiments, another set of experiments are carried out on the newer and more challenging database NIST 2014 i-vector challenge [4]. Experimental results performed on 2014 i-vector challenge show that the proposed DL-based system fills 46% of the performance gap between cosine and oracle PLDA scoring systems in terms of minDCF which is similar to the PLDA scoring results obtained with unsupervised estimated labels. The score combination of the proposed DL-based system and PLDA with estimated labels fills 79% of this gap.

The rest of the paper is organized as follows. Section II gives a brief background overview about i-vectors, PLDA, and deep learning techniques used in experiments. Section III presents the proposed DL-based backend for i-vector classification. Section IV describes the proposed impostor selection algorithms in order to have a balanced training. Section V shows how we will cope with the few amount of data for the training of each target model. Sections VI and VII discuss the experimental results obtained on NIST SRE 2006 and NIST 2014 i-vector challenge, respectively. Section VIII concludes the paper.

II. Background

A. i-Vector and PLDA

It is shown that a Gaussian Mixtures Model (GMM) adapted from a Universal Background Model (UBM) can represent the feature vectors of a speech signal adequately [33]. If the mean vectors of the adapted GMM are stacked to build the supervector $\boldsymbol{s}$, it can be further modeled as follows [1],

$$\boldsymbol{s} = \boldsymbol{s}_u + \boldsymbol{T}\boldsymbol{x} \tag{1}$$

where $\boldsymbol{s}_u$ is the speaker- and session-independent mean supervector typically from UBM, $\boldsymbol{T}$ is the total variability matrix,

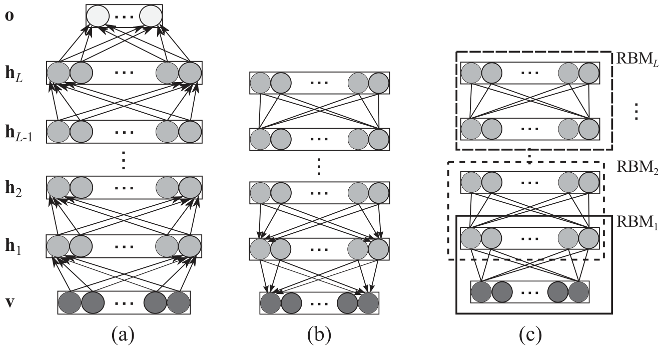

Fig. 1. (a) DNN, (b) DBN, and (c) DBN training/DNN pre-training.

and $\boldsymbol{x}$ is a low rank vector of latent variables. The mean of the posterior distribution of $\boldsymbol{x}$ is referred to as i-vector $\boldsymbol{\omega}$ [1]. This posterior distribution is conditioned on the Baum-Welch statistics of the given speech utterance. The $\boldsymbol{T}$ matrix is trained using the Expectation-Maximization (EM) algorithm given the centralized Baum-Welch statistics from background speech utterances. In other words, one can say that an i-vector is a low rank vector, typically between 400 and 600, representing a speech utterance. More details can be found in [1].

Two main scoring techniques for i-vectors are cosine [1], [34] and Probabilistic Linear Discriminant Analysis (PLDA) [2], [3]. PLDA is a more efficient technique which performs scoring along with session variability compensation. Since i-vectors are of sufficiently low dimension, a modified version of PLDA proposed in [3] is typically used. It assumes that each i-vector can be decomposed as,

$$\boldsymbol{\omega} = \boldsymbol{m} + \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2}$$

where $\boldsymbol{m}$ is a global offset, the columns of $\boldsymbol{\Phi}$ are eigenvoices, $\boldsymbol{\beta}$ is a latent vector having a standard normal prior, and the residual vector $\boldsymbol{\epsilon}$ is normally distributed with zero mean and a full covariance matrix. The model parameters are estimated from a large collection of speaker-labeled background data using an EM algorithm as in [2]. Within and between class i-vector covariance matrices, depending only on the model parameters, are stored and used for scoring.

B. Deep Learning

Deep Learning (DL) refers to a branch of machine learning techniques which attempts to learn high level features from data. Since 2006 [35], [36], DL has become a new area of research in many applications of machine learning and signal processing. Various deep learning architectures have been used in speech processing (e.g., [11], [12], [37]–[39]). Deep Neural Networks (DNN), Deep Belief networks (DBN), and Restricted Boltzmann Machines (RBM) are three main techniques we have used in this work to discriminatively model each target speaker given input i-vectors.

DNNs are feed-forward neural networks with multiple hidden layers (Fig. 1(a)). They are trained using discriminative back-propagation algorithms given class labels of input vectors. The training algorithm tries to minimize a loss function between the class labels and the outputs. For classification tasks, cross-entropy is often used as the loss function and the softmax is commonly used as the activation function at the output layer [40]. Typically, the parameters of DNNs are initialized with small random numbers. Recently, it has been shown that there are more efficient techniques for parameter initialization [41]–[43]. One of those techniques consists in initializing DNN with DBN parameters, which it is often referred to as unsupervised pre-training or just hybrid DBN-DNN [9], [44]. It has been empirically shown that this pre-training stage can set the weights of the network closer to an optimum solution than random initialization [41]–[43].

DBNs are generative models with multiple hidden layers of stochastic units above a visible layer which represents a data vector (Fig. 1(b)). The top two layers are undirected and the other layers have top-down directed connections to generate the data. There is an efficient greedy layer wised algorithm to train DBN parameters [36]. In this case, DBN is divided in two-layer sub-networks and each one is treated as an RBM (Fig. 1(c)). When the first RBM built on visible units is trained, its parameters are frozen and the outputs are given to the RBM above as input vectors. This process is repeated until the top two layers are reached.

RBMs are generative models constructed from two undirected layers of stochastic hidden and visible units. RBM training is based on maximum likelihood criterion using the stochastic gradient descent algorithm [9], [36]. The gradient is estimated by an approximated version of the Contrastive Divergence (CD) algorithm which is called CD-1 [35], [36]. More theoretical and practical details can be found in [35], [36], [45]. The whole training algorithm is given in [31].

In all of these networks, it is possible to update the parameters after processing each training example, but it is often more efficient to divide the whole input data (batch) into smaller size batches (minibatch) and to update the parameters by averaging the gradients over each minibatch. The parameter updating procedure is repeated when the whole available input data is processed. Each iteration is called an epoch.

III. Proposed Deep Learning Backend for i-Vectors

The success use of i-vectors in speaker recognition and DL techniques in speech processing applications has encouraged the research community to combine those techniques for speaker recognition. Two kinds of combination can be considered. DL techniques can be used in the i-vector extraction process, or applied as a backend.

In this work, DL technology is used as a backend in which a two-class hybrid DBN-DNN is trained for each target speaker to increase the discrimination between target i-vector/s and the i-vectors of other speakers (non-targets/impostors) (Fig. 2). Proposed networks are initialized with speaker-specific parameters adapted from a global model, which is referred to as Universal Deep Belief Network (UDBN). Then the cross-entropy between the class labels and the outputs is minimized using the back-propagation algorithm.

DNNs usually need a large number of input samples to be trained efficiently. As a general rule, deeper networks require more input data. In speaker recognition, target speakers can be enrolled with only one sample (single session task) or multiple samples (multi-session task). In both cases, the number of target samples is very limited. A network trained with such limited data is highly probable to be overfitted. On the other hand, the number of target and impostor samples will be highly unbalanced, i.e., one or some few target samples against thousands of impostor samples. Learning from such unbalanced data will result in biased DNNs towards the majority class. In other words, DNNs will have a much higher prediction accuracy over the majority class.

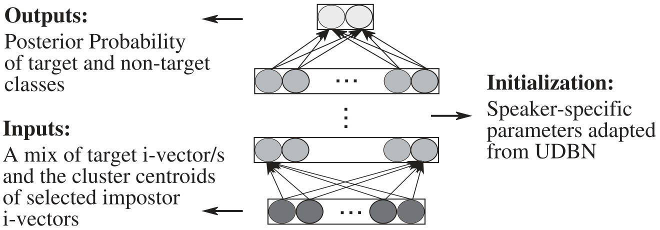
Outputs:
Posterior Probability of target and non-target classes
Inputs:
A mix of target i-vector/s and the cluster centroids of selected impostor i-vectors
Initialization:
Speaker-specific parameters adapted from UDBN

Fig. 2. Proposed deep learning architecture for training of each speaker model.

Fig. 3 shows the block diagram of the proposed approach to discriminatively model target speakers. Two main contributions have been proposed in this work to tackle the above problems. The balanced training block attempts to decrease the number of impostor samples and, on the contrary, to increase the number of target ones in a reasonable and effective way. The most informative impostor samples for target speakers are first selected by the proposed impostor selection algorithm. Afterwards, the selected impostors are clustered and the cluster centroids are considered as final impostor samples for each target speaker model. Impostor centroids and target samples are then divided equally into minibatches to provide balanced impostor and target data in each minibatch.

On the other hand, the DBN adaptation block is proposed to compensate the lack of input data. As DBN training does not need any labeled data, the whole background i-vectors are used to build a UDBN. The parameters of the UDBN are then adapted to the balanced data obtained for each target speaker. At the end, given the target/impostor labels, the adapted DBN and the balanced data, a DNN is discriminatively trained for each target speaker. These two contributions are described in more details in the following sections.

IV. Balanced Training

As speaker models in the proposed method will be finally discriminative, they need both positive and negative data as inputs. Nevertheless, the problem is that the amount of positive and negative data are highly unbalanced in this case, which leads to biasing towards the majority class. Some of the straightforward ways to deal with unbalanced data problem are explored in [46]–[48] [49], [50]. A commonly used method is data sampling. The data of the majority class is undersampled and, on the contrary, the data of the minority class is oversampled. The effectiveness of these techniques is highly dependent on the data structure.

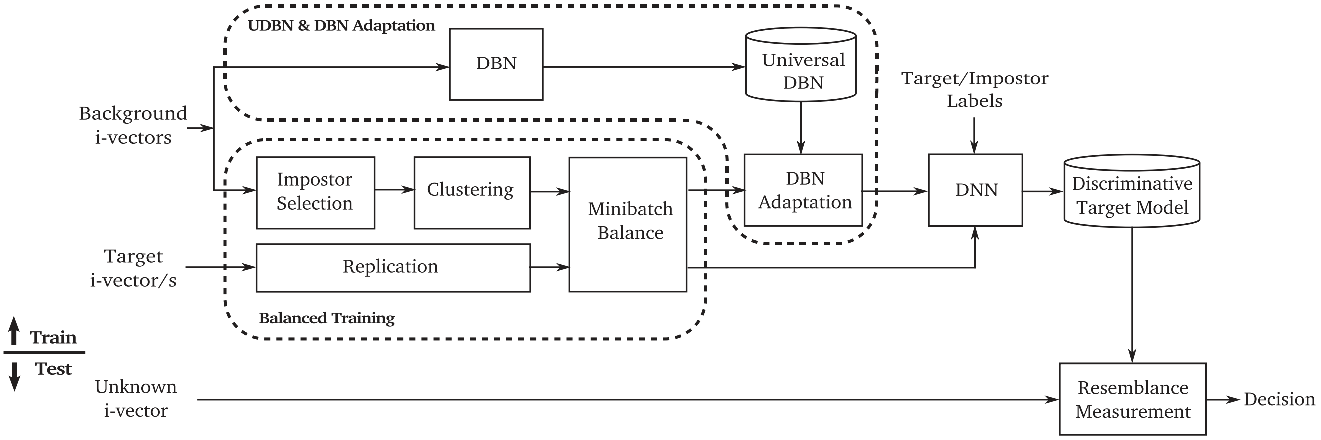

Fig. 3. Block-diagram of the train/test phases of the proposed deep learning backend for i-vectors.

In the proposed approach shown in Fig. 3, the amount of impostors is decreased in two steps, namely selection and clustering. On the other hand, the amount of target samples is increased by either replication or combination. After that, balanced target and impostor samples are distributed equally among minibatches.

A. Impostor Selection and Clustering

The objective is to decrease the large number of negative samples in a reasonable way. Our proposal has two main steps. First, only those impostor i-vectors which are more informative for the training dataset are selected. Informative impostor means, in this case, the impostor which is not only representative to a given target but also is statistically close to other targets in the dataset. For some real applications, it could makes sense to select those impostors who are globally close to all enrolled speakers. When the target speakers are changed significantly, the selected impostors could be re-selected according to the new target dataset. Second, as the number of selected impostor samples is still high in comparison to the number of target ones, they are clustered by the k-means algorithm using the cosine distance criterion. The centroids of the clusters are then used as the final negative samples.

The selection method is inspired from a data-driven background data selection technique proposed in [51]. In that technique given all available impostor supervectors, a Support Vector Machine (SVM) classifier is trained for each target speaker. The number of times each impostor is selected as a support vector, in all training SVM models, is called impostor support vector frequency [51]. Impostor examples with higher frequencies are then selected as the refined impostor dataset. However, SVM training for each target speaker would be computationally costly. Moreover, as our final discriminative models will be DNNs, it would not be worth to employ this technique as such. Instead, we have proposed to use cosine similarity as an efficient and a fast criterion for comparing i-vectors. We compare each target i-vector with all impostor i-vectors in the background data set. Those N impostors which are close to each target i-vector are treated like support vectors in [51]. Then the κ impostors with higher frequencies are selected as the most informative impostors. The N and κ selected impostors are referred to as local and global selected impostors in this work. The parameters N and κ are determined experimentally. The whole algorithm is shown in Fig. 4 and can be summarized as follows,

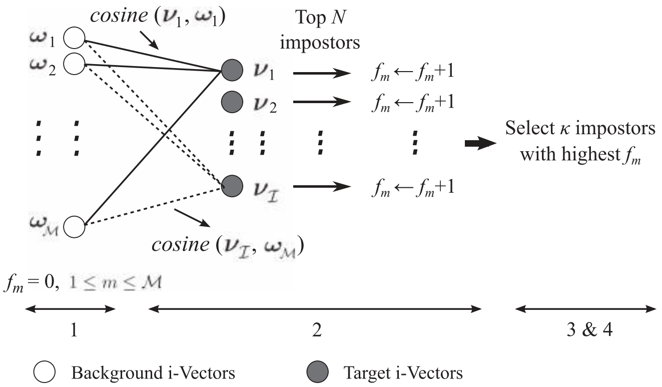
$cosine\,(\boldsymbol{\nu}_1, \boldsymbol{\omega}_1)$
Top N impostors
$\boldsymbol{\omega}_1$
$\boldsymbol{\omega}_2$
$\boldsymbol{\omega}_{\mathcal{M}}$
$\boldsymbol{\nu}_1$
$\boldsymbol{\nu}_2$
$\boldsymbol{\nu}_{\mathcal{I}}$
$f_m \leftarrow f_m+1$
$f_m \leftarrow f_m+1$
$f_m \leftarrow f_m+1$
Select κ impostors with highest f_m
$cosine\,(\boldsymbol{\nu}_{\mathcal{I}}, \boldsymbol{\omega}_{\mathcal{M}})$
$f_m = 0,\ 1 \le m \le \mathcal{M}$
1
2
3 & 4
○ Background i-Vectors
● Target i-Vectors

Fig. 4. Steps of the proposed impostor selection algorithm.

1) Set impostor frequencies $f_m = 0$ for impostor i-vectors $\boldsymbol{\omega}_m$, $1 \le m \le \mathcal{M}$
2) For each target i-vector $\boldsymbol{\nu}_i$, $1 \le i \le \mathcal{I}$
    a) Compute $cosine\,(\boldsymbol{\nu}_i, \boldsymbol{\omega}_m)$, $1 \le m \le \mathcal{M}$
    b) Select the N impostors with the highest scores
    c) For the selected impostors $f_m \leftarrow f_m + 1$
3) Sort impostors in descending order based on their f_m
4) Select the first κ impostors as the final ones.

where $cosine\,(\boldsymbol{\nu}_i, \boldsymbol{\omega}_m)$ is the cosine score between target i-vector $\boldsymbol{\nu}_i$ and the impostor i-vector $\boldsymbol{\omega}_m$ in the background dataset, $\mathcal{M}$ and $\mathcal{I}$ are the number of impostor and target i-vectors, respectively. Note, in the case of multi-session target enrollment, the average of the available i-vectors per each target speaker will be considered in the above algorithm. The final selected impostors could be only local, global, or a pooling of both of them. If local or pooling are used, the computational cost would be higher as the k-means clustering should be run for each target model independently.

We have proposed a similar algorithm in [23] in which the selection process is only dependent on the background data. A

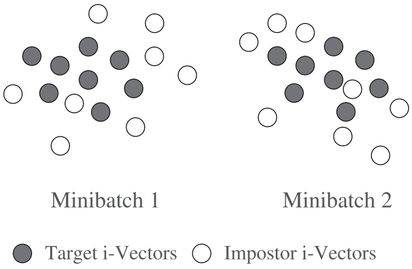

Fig. 5. An example of proposed balanced training for DNNs in multi-session speaker verification task. In each minibatch the same target i-vectors but different impostors are shown to DNNs.

randomly selected subset from the background data is used in the above algorithm rather than the target training database. In order to make the process statistically more reliable, the whole process is repeated several times and the impostor frequencies are accumulated over all iterations. The full algorithm can be found in [23]. It was shown that this algorithm performs similar to the first algorithm which uses the training target set in the selection process when the background database is large enough [23].

B. Target Replication or Combination

In order to balance positive and negative samples, the number of target samples is increased as many as the number of impostor cluster centroids obtained in Section IV-A. In the single session enrollment task, the i-vector of each target speaker is simply replicated as many as the number of cluster centroids. Replicated target i-vectors will not act exactly the same as each other in the pre-training process of DNNs due to the sampling noise created in RBM training [45]. Moreover, in both adaptation and supervised learning stages the replicated versions make the target and impostor classes having the same weights when the network parameters are being updated. In multi-session task, the available i-vectors of each target speaker can be combined, i.e., the average of every n i-vectors is considered as a new target i-vector.

Once the number of positive and negative samples are balanced, they are distributed equally among minibatches. In other words, each minibatch contains the same number of impostors and targets. If target samples in the multi-session task are not combined, the same target samples but different impostor ones are shown to the network in each minibatch (Fig. 5). The optimum numbers of impostor clusters and minibatches will be determined experimentally in Sections VI and VII.

V. Universal DBN and Adaptation

Unlike DNNs, which need labeled data for training, DBNs do not necessarily need such labeled data as inputs. Hence, they can be used for unsupervised training of a global model referred to as Universal DBN (UDBN) [21]. UDBN is trained by feeding background i-vectors from different speakers. The training procedure is carried out layer by layer using RBMs as described in Section II-B. As the input i-vectors are real-valued, a Gaussian-Bernoulli RBM (GRBM) [9], [45] is used to train the connection weights between the visible and the first hidden layer units. The rest of the connection weights are trained with Bernoulli-Bernoulli RBMs.

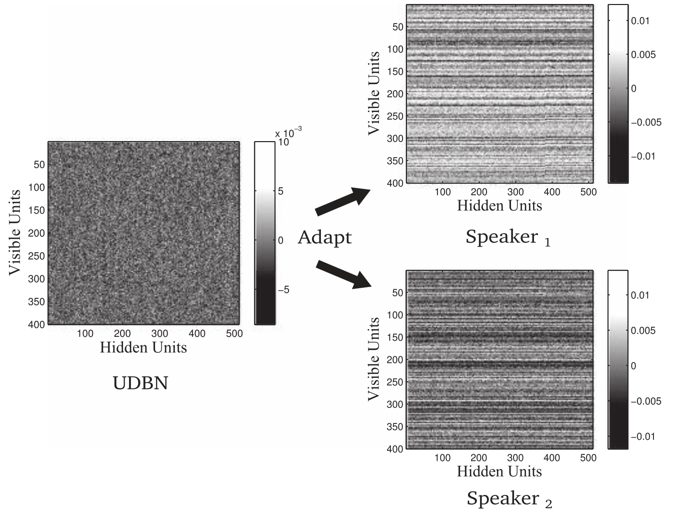

Fig. 6. Comparison of the adapted connection weights between the visible and the first hidden units for two different speakers.

It is shown that pre-training techniques can initialize DNNs better than simply random numbers [41]–[43]. However, when a few input samples are available, just pre-training may not be enough to achieve a good model. In this case, we have proposed in [21] to adapt UDBN parameters to the balanced data obtained for each target speaker. Adaptation is carried out by training a DBN which is initialized by the parameters of the UDBN given the balanced data of each target speaker. Adapted DBNs are then used as an initialization for the final DNN target models. In order to avoid overfitting, only a few iterations will be considered for adaptation. It is supposed that UDBN can learn both speaker and channel variabilities from the background data. Therefore, UDBN will provide a more meaningful initial point for DBNs than a simple random initialization. The study in [42] has shown that pre-training is robust with respect to the random initialization seed. The use of UDBN parameters makes target models almost independent from the random seeds.

In contrast to [21], [22], in this work we normalize the UDBN parameters before adaptation. Normalization is carried out by simply scaling down the maximum absolute value of connection weights to 0.01. In this way, connection weights will have a dynamic range similar to that typically used for random initialization. Additionally, bias terms are multiplied by 0.01 to be closer to zero. This is because the bias terms are usually set to zero when the connection weights are randomly initialized. UDBN parameter normalization facilitates the training of the networks specifically where more than one hidden layer is used. In this way, the same learning rates and the number of epochs tuned for random initialized DNNs can also be used for adapted DNNs in the supervised learning stage.

Fig. 6 shows the comparison of the adapted UDBN connection weights, between the input layer and the first hidden layer, for two different speakers. As it can be seen in this figure, speaker-specific initial points are set by the adaptation process for each DNN target model. Given target/impostor labels, the minibatch stochastic gradient descent back-propagation is then carried out for fine-tuning. The softmax and the logistic sigmoid will be the activation functions of the top label layer and the other hidden layers, respectively.

We have proposed to compute the output scores in Log Posterior Ratio (LPR) forms as,

$$\Lambda(\text{target}|\boldsymbol{\omega}) = \log P(\text{target}|\boldsymbol{\omega}) - \log P(\text{non-target}|\boldsymbol{\omega}) \quad (3)$$

where $P(\text{target}|\boldsymbol{\omega})$ and $P(\text{non-target}|\boldsymbol{\omega})$ are, respectively, the posterior probability of the target and non-target classes given the test i-vector $\boldsymbol{\omega}$. LPR computation helps to Gaussianize the true and false score distributions which can be useful for score fusion.

In addition, to make the fine-tuning process more efficient a momentum factor is used to smooth out the updates, and the weight decay regularization is used to penalize large weights. The top layer pre-training proposed in [21] is not used in this work. The reason is that it gives the emphasis on the top layer connection weights and avoids the lower layers, closer to the input, to learn enough from the input data. This fact will be more important when higher number of hidden layers are used.

VI. Experiments on NIST SRE 2006

NIST SRE 2006 [32] is used to show the effect of each proposed contribution shown in Fig. 3 for both single and multi-session speaker verification tasks. In these experiments, we have built the whole system from scratch including Voice Activity Detection (VAD) and feature and i-vector extraction. Taking advantage of the conclusions of this section, the NIST 2014 i-vector challenge database [4] is used in Section VII to compare the performance of the proposed system with the most recent state-of-the-art baseline systems.

A. Baseline and Database

The whole core test condition of SRE 2006 is used as a single session task and 8 conversation side training condition is used as the multi-session task. In both cases, training and test signals have approximately two-minute total speech duration. There are 816 target models and 51,068 trials in the single session and 699 target models and 31,080 trials in the multi-session task. Speech signals with the two-minute approximate duration from NIST SRE 2004 and 2005 are used as the background data containing 6,063 speech signals from 1,070 distinct speakers.

Frequency Filtering (FF) features [52] are used in these experiments. FFs, like Mel Frequency Cepstral Coefficient (MFCC), are decorrelated version of log Filter Bank Energies (FBE) [52]. It has been shown that FF features achieve a performance equal to or better than MFCCs [52]. Features are extracted every 10 ms using a 30 ms Hamming window. The number of static FF features is 16 and along with delta FF and delta log energy, 33-dimensional feature vectors are built. Before feature extraction, speech signals are subject to an energy-based silence removal process. The gender-independent UBM is represented as a diagonal covariance, 512-component GMM. All the i-vectors are 400-dimensional. The i-vector extraction process is carried out using ALIZE open source software [53]. UBM, $\boldsymbol{T}$ matrix, and PLDA parameters are trained using the same background data. PLDA baseline systems are gender-independent with a 250-dimensional speaker space. For PLDA experiments, i-vectors are length normalized. Performance is evaluated using Detection Error Tradeoff (DET) curves, Equal Error Rate (EER), and the minimum of the Decision Cost Function (minDCF) defined as follows [32],

$$DCF(t) = 0.1 \times P_M\,(t) + 0.99 \times P_{FA}\,(t) \quad (4)$$

where the miss rate P_M is the relative number of target trials decided incorrectly, the false alarm rate P_{FA} is the relative number of non-target trials decided incorrectly, and t is the threshold for which DCF is computed.

B. Single Session Experiments

For DNN experiments, the size of hidden layers is set to 512. DNNs with up to three hidden layers are explored in all experiments. We do not go further than three layers because of few amount of data and increasing the computational complexity without more significant gain. The number of minibatches and the number of impostor centroids are set experimentally to 3 and 12, respectively. Each minibatch will include four impostor centroids and four replicated target samples. It is worth noting that compared to speech recognition in which the amount of training data is typically very high, the size and the number of minibatches are much less in this application. However, the gradient is still stable and training works very well.

As a DNN baseline system, we train a DNN for each target speaker using the whole impostor background data and random initialization. In this case, the whole background i-vectors are clustered using the k-means algorithm and the centroids are considered as impostor samples. In this work, we use the uniform distribution $\mathcal{U}\,(0, 0.01)$ for random initialization as the experimental results showed that it achieves slightly better performance than the normal distribution $\mathcal{N}\,(0, 0.01)$ used in the prior work [21]. We tune the parameters of the networks and keep them fixed in all other experiments. DNN-3L will stand for a three hidden layer DNN.

The two parameters N and κ, the number of local and global selected impostors in the proposed impostor selection method, need to be determined experimentally. Fig. 7 illustrates the variability of EER in terms of these two parameters for one hidden layer DNNs. The similar behavior can be observed for minDCF curves. DNN examples shown in this figure are initialized randomly. Based on this figure, for DNN-1L we set N and κ to 10 and 2,000, respectively. Similar curves are plotted for other networks and N is set to 10 for all of them and κ is set to 300 and 500 for DNN-2L and DNN-3L, respectively.

Experimental results showed that the main improvement due to the adaptation process comes from the adaptation of the connection weights between the input layer and the first hidden layer for all DNNs. The adaptation of the other layers has no

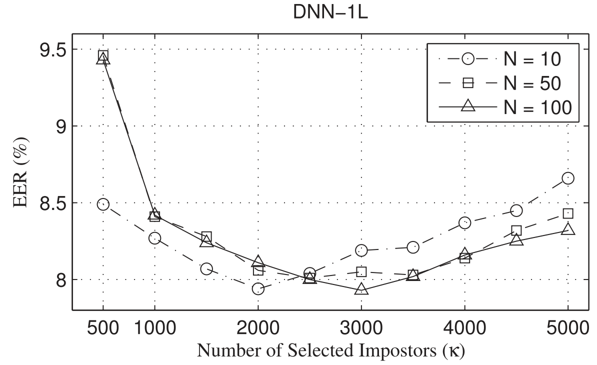

Fig. 7. Determination of the parameters of the proposed impostor selection algorithm for one hidden layer DNNs. N and κ are, respectively, the number of local and global nearest impostor i-vectors to target i-vectors.

TABLE I
THE EFFECT OF EACH PROPOSED IDEA OF FIG. 3 ON THE PERFORMANCE OF THE PROPOSED DNN SYSTEMS

Impostor Selection	Adaptation	EER (%) # Hidden Layers			minDCF ($\times 10^4$) # Hidden Layers		
		1	2	3	1	2	3
–	–	8.55	7.76	7.59	381	353	351
✓	–	8.06	7.12	7.09	360	327	326
–	✓	7.43	7.47	7.45	339	343	339
✓	✓	**6.81**	6.97	6.99	315	317	313
Fusion with cosine		6.83	**6.88**	**6.64**	**308**	**309**	**299**
Fusion with PLDA		4.98	5.03	**4.76**	253	**248**	**230**

Results are Obtained on the **Core Test Condition** of NIST SRE 2006. The Cosine and PLDA Baseline Systems Achieve (EER=7.18%, minDCF=324) and (EER=4.78%, minDCF=253), Respectively.

significant impact on the performance. In order to decrease the probability of overfitting during the adaptation, a separate network is adapted to each minibatch and then the parameters of the obtained networks are averaged.

Table I summarizes the effect of each proposed contribution. Impostor selection improves the performance to a great extent for all the networks. We have tried global, local, and the pooling of global and local selected impostors before k-means clustering and the best performance was obtained by using only global selected impostors. The biggest improvement due to the adaptation process is observed in DNNs with one hidden layer. The best results are obtained using both impostor selection and adaptation techniques which show an 8-20% and 10-17% relative improvements in terms of EER and minDCF, respectively, compared to the baseline DNNs. The biggest relative improvements are achieved on DNN-1L. The last two rows of the table show the fusion of DNN systems with the cosine (EER=7.18%, minDCF=0.0324) and PLDA (EER=4.78%, minDCF=0.0253) baseline systems. Scores of each system are first mean and variance normalized and then simply summed. The fusion of the cosine baseline and DNN systems improves the results and DNN-3L achieves the best results corresponding to an 8% relative improvement for both EER and minDCF in comparison to the cosine scoring baseline system. Nevertheless, only DNN-3L scores can improve the PLDA results specifically for minDCF by 9% relative improvement. We have also combined the scores of DNNs with different number of hidden layers, but no gain is observed.

TABLE II
THE EFFECT OF EACH PROPOSED IDEA OF FIG. 3 ON THE PERFORMANCE OF THE PROPOSED DNN SYSTEMS

Impostor Selection	Adaptation	EER (%) # Hidden Layers			minDCF ($\times 10^4$) # Hidden Layers		
		1	2	3	1	2	3
–	–	4.58	4.58	4.38	208	213	217
✓	–	4.02	4.07	3.86	183	201	194
–	✓	4.24	4.30	4.20	202	207	202
✓	✓	3.68	3.83	3.50	170	189	172
Fusion with cosine		**3.61**	**3.77**	**3.45**	**161**	**169**	**162**
Fusion with PLDA		2.46	2.62	2.36	111	121	112

Results are obtained on NIST SRE 2006, **8-Session Enrollment** Task. The Cosine and PLDA Baseline Systems Achieve (EER=4.2%, minDCF=191) and (EER=2.27%, minDCF=105), Respectively.

C. Multi-Session Experiments

The same configuration used for the single session task is also applied for the multi-session one. The number of minibatches is set to 3. In each minibatch, all 8 target i-vectors accompanying with 8 impostor cluster centroids are shown to the network. Therefore, the size of each minibatch and the total number of impostor clusters will be 16 and 24, respectively. As the combination of the i-vectors of each target speaker did not help the training of the networks, we replicated the target i-vectors in every minibatch as it was shown in Fig. 5. We train the networks with the same parameters tuned for the single session experiments.

Results are summarized in Table II. Around 12% relative improvements are achieved in all DNNs employing impostor selection technique proposed in this work. With the same parameters obtained for the single session task, we re-selected the impostors for the new multi-session data set. The adaptation process improves the performance up to 8%. As in the single session task, adaptation is more effective for one-hidden-layer DNNs. For all the networks, only the parameters of the first hidden layer are adapted because no more improvement was observed adapting the other layers. The best results are obtained with DNN-3L when the two proposed techniques are combined. It shows more than 20% relative improvements of EER and minDCF in comparison to the baseline three-layer DNNs.

The proposed three-hidden-layer DNNs show a performance between the cosine (EER=4.2%, minDCF=0.0191) and PLDA (EER=2.27%, minDCF=0.0105) baseline systems, with more than 17% and 10% relative improvements in terms of EER and minDCF, respectively, compared to the cosine scoring. Fusion with the cosine baseline system improves the results in all cases, but no improvement is observed by combination with PLDA scores.

VII. Experiments on NIST 2014 i-Vector Challenge

The full database provided in the NIST 2014 speaker recognition i-vector challenge [4] is used for the experiments in this section. Rather than speech signals, i-vectors are given directly by NIST in this challenge to train, test, and develop the speaker recognition systems. This enables system comparison more readily with consistency in the front-end and in the amount and type of the background data [4]. For this challenge, speaker recognition systems are evaluated in two phases: when the speaker labels of the background data are not known and when they are known to the systems. The cosine and PLDA scoring techniques are used by NIST as the baseline systems when unlabeled and labeled background data are available, respectively. The goal of this evaluation is to see how other techniques can fill the performance gap between these two baseline systems when no labeled background data is available.

A. Baseline and Database

Conventional telephone speech recordings from NIST SRE 2004 to 2012 are used to compute i-vectors for this challenge [7]. Unlike NIST SRE 2006 experiments, in which the duration of speech signals for each i-vector was approximately 2 minutes, in this challenge i-vectors are extracted from speech utterances of varying duration with a mean of 39.6 seconds. Three sets of 600-dimensional i-vectors are provided: development, train, and test consisting of 36,572, 6,530, and 9,634 i-vectors, respectively. The number of target speaker models is 1,306 and for each of them five i-vectors are available. Each target model will be scored against all the test i-vectors and, therefore, the total number of trials will be 12,582,004. Trials are divided by NIST into two randomly selected subsets: a progress subset (40%), and an evaluation subset (60%). The performance is evaluated using a minDCF obtained by [4],

$$DCF(t) = P_M\,(t) + 100 \times P_{FA}\,(t) \tag{5}$$

Two main baseline systems are considered in this work when the background i-vectors are not labeled: cosine and PLDA with estimated labels. The PLDA with actual labels is also used as an oracle system for comparison. In all of them, i-vectors are whitened and length normalized prior to evaluation and the average i-vector per each target speaker is used as a single target model. Only for the cosine baseline system the average i-vectors are again length normalized as it is shown that for the PLDA systems re-normalization affects the performance [7]. Both PLDA systems are gender-independent with a 400-dimensional speaker space. In order to have the best PLDA with actual labels, those background i-vectors extracted from speech signals shorter than 30 seconds are discarded before PLDA training [7]. For the PLDA with estimated labels, a two stage unsupervised clustering technique is used to estimate the speaker labels of the background data. The first stage of the clustering algorithm is similar to the Mean Shift based algorithm proposed in [54] and used successfully in this challenge in [6]. In the second stage, the closer clusters obtained in the first stage are combined. In both stages, i-vectors are joined based on the cosine similarity considering a threshold which is set to 0.29 in our experiments as in [6]. At the end, only clusters contained no less than 4 and no more than 50 i-vectors are selected. As in [6], those i-vectors with less than 20 seconds of speech are discarded before PLDA training in this case. It is possible to train a PLDA with the estimated labels and repeat the two stage unsupervised clustering algorithm with the PLDA similarity measurement, but it would be time consuming and no significant gain will be observed in practice. The experimental results for this baseline system show a comparable performance to those reported in [6] and [5].

B. Multi-Session Experiments

The same architecture as in SRE 2006 multi-session experiments has been used for these experiments with some modification. The size of hidden layers is set to 400. Each minibatch consists of 5 impostor centroids and 5 target samples. The total number of impostor centroids is 15 for each target model. Since DNN-1L and DNN-3L worked better than DNN-2L in SRE 2006 experiments, we only implement these two networks for the NIST i-vector challenge. DNN-1L and DNN-3L are trained with the learning rates of 0.002 and 0.07 and with the number of epochs of 30 and 300, respectively. Momentum and weight decay are set, respectively, to 0.9 and 0.001 for all DNNs. The whole unlabeled background i-vectors are used for UDBN training. The learning rate and the number of epochs for UDBN training are set to 0.02 and 200 for GRBM, and to 0.06 and 120 for the rest of RBMs, respectively. Momentum, weight decay, and the minibatch size are set, respectively, to 0.9, 0.0002, and 100 for all RBMs. For DNN-3L we adapted only the first two layers. The learning rate and the number of epochs of adaptation are set, respectively, to 0.001 and 10 for the first layer and to 0.0001 and 20 for the second layer.

As it was discussed in Section IV-A, when the background data set is big enough like in this challenge, the results will be only slightly better if the training data set is used in the selection algorithm. On the other hand, as a general rule of this challenge the use of training data is not allowed for impostor selection. Therefore, in order to have a fair comparison with the results of other participating sites, we use only the background i-vectors in the impostor selection algorithm (Section IV-A).

As in SRE 2006 experiments, we have tried global, local, and the pooling of global and local selected impostors before k-means clustering and the best performance was obtained by pooling. For global impostor selection, κ and N are set to 4,500 and 100 for both DNN-1L and DNN-3L, respectively. The algorithm is iterated 20 times. Afterwards, the global selected impostors are pooled with 500 local impostors for each target speaker before k-means clustering.

Table III compares the performance of the proposed DNN systems with other baseline systems in terms of minDCF and EER, and Figs. 8 and 9 compares them in all operating points in terms of DET curves. Circles in the figures show the operating points corresponding to minDCFs. It is worth noting that in NIST 2014 i-vector challenge the performance of the systems were evaluated only in terms of minDCF. However, we have also included EERs in the table for better comparison. As it can

TABLE III
COMPARISON OF THE PERFORMANCE OF THE PROPOSED DNN SYSTEM WITH OTHER BASELINE SYSTEMS ON NIST 2014 I-VECTOR CHALLENGE

Unlabeled Background Data	Progress Set		Evaluation Set	
	EER (%)	minDCF	EER (%)	minDCF
[1] cosine	4.78	0.386	4.46	0.378
[2] PLDA (Estimated Labels)	3.85	0.300	3.46	0.284
[3] Proposed DNN-1L	5.13	0.327	4.61	0.320
[4] Proposed DNN-3L	4.55	0.305	4.11	0.300
Fusion [2] & [4]	**2.99**	**0.260**	**2.70**	**0.243**
Labeled Background Data				
[5] PLDA (Actual Labels)	2.23	0.226	2.01	0.207
Fusion [2] & [5]	2.04	0.220	1.85	0.204
Fusion [4] & [5]	2.13	0.221	2.00	0.196
Fusion [2] & [4] & [5]	**1.88**	**0.204**	**1.74**	**0.190**

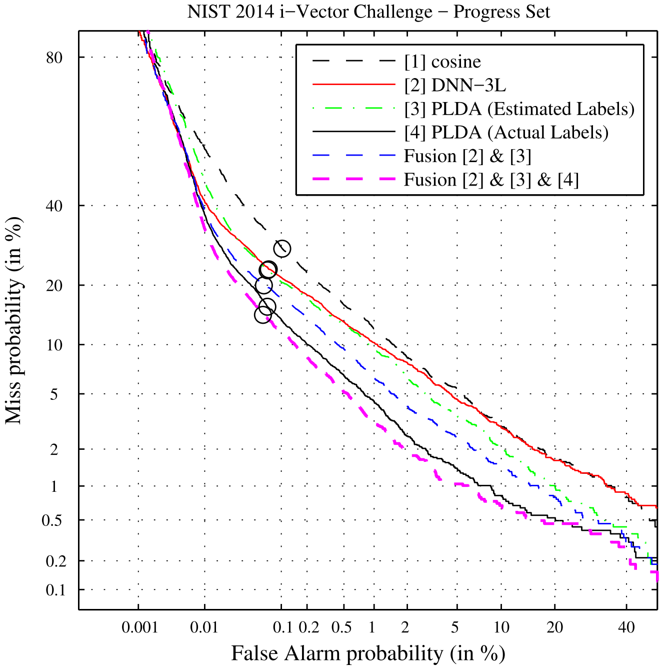

Fig. 8. Comparison of the performance of the proposed DNN-3L system with other baseline systems on the progress set of NIST 2014 i-vector challenge.

be seen in the table, the proposed DNN-3L performs better than DNN-1L, as it was concluded from SRE 2006 experiments. The proposed DNN-3L system achieves comparable performance to PLDA with estimated labels in terms of minDCF (with 21% relative improvement compared to cosine scoring), but lower performance in terms of EER. In other words, as it is shown in Figs. 8 and 9, the proposed DNN-3L system performs closer to PLDA with actual labels than to cosine for lower False Alarm (FA) probabilities. For higher FA probabilities, it is the other way around. The proposed DNN and PLDA with actual labels achieve the same performance for FA probability around 0.01, and for lower than 0.01 the proposed DNN system outperform the PLDA with actual labels. This can be seen as an advantage of the proposed system since having better performance in lower FA probabilities is more important for higher security purposes.

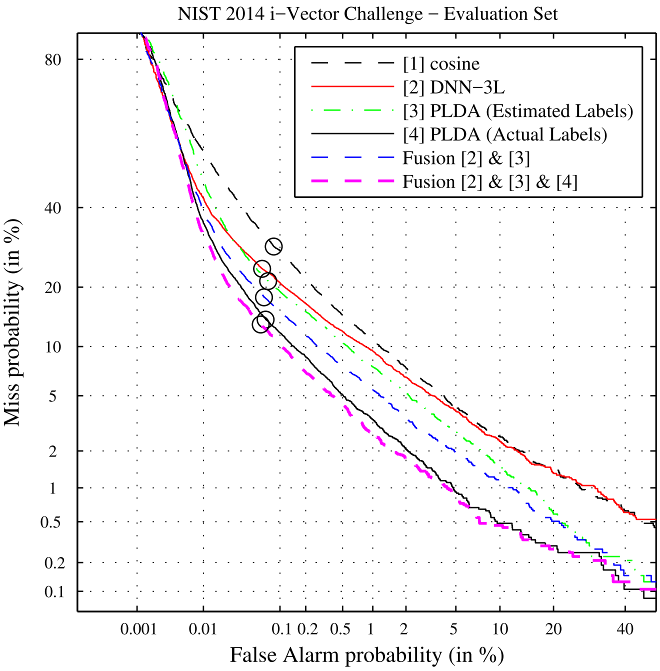

Fig. 9. Comparison of the performance of the proposed DNN-3L system with other baseline systems on the evaluation set of NIST 2014 i-vector challenge.

The interesting point is that the combination of the DNN-3L and PLDA with estimated labels in the score level improves the results to a great extent in all operating points. The score fusion is carried out using BOSARIS toolkit [55]. The combination weights are trained on the progress trial set and used for the evaluation set. The resulting relative improvement compared to cosine baseline system is 36% in terms of minDCF on the evaluation set. This improvement with no use of background labels is considerable compared to 45% relative improvement which can be obtained by PLDA with actual labels.

Although the use of speaker labels for the background data has not been the goal of this work, it would be interesting to see how the proposed DL-based backend and PLDA with estimated labels can help the oracle PLDA system, which uses the actual labels. As it can be seen in Table III, in both cases of DNN-3L and PLDA with estimated labels, the combination with oracle PLDA improves the results. This improvement is higher in terms of EER for PLDA with estimated labels and higher in terms of minDCF for DNN-3L systems. Nevertheless, the combination of all three systems achieves the best performance, corresponding to 8% and 13% relative improvement in terms of minDCF and EER, respectively, compared to the PLDA with actual labels.

VIII. CONCLUSION

A hybrid architecture based on Deep Belief Networks (DBN) and Deep Neural Networks (DNN) has been proposed in this work to discriminatively model each target speaker for i-vector speaker verification. The main objective has been to fill the performance gap between the cosine and the oracle PLDA scoring systems when no labeled background data is available. Two main contributions have been proposed to make DNNs more efficient in this particular task. Firstly, the most informative impostor i-vectors have been selected and clustered to provide a balanced training. Secondly, each DNN has been initialized with the speaker specific parameters adapted from a global model, which has been referred to as Universal DBN (UDBN). In order to have more insight into the behavior of these techniques in both single and multi-session speaker enrollment tasks, the experiments have been carried out in both scenarios. Experiments were performed on NIST SRE 2006, mainly for development, and on NIST 2014 i-vector challenge, mainly for evaluation. It was shown that the proposed hybrid system fills approximately 46% of the performance gap between the cosine and the oracle PLDA scoring systems in terms of minDCF. Although the proposed system still does not outperform the baseline PLDA with estimated labels, their score fusion is highly effective and covers 79% of this gap. The reason that the proposed system still does not outperform the baseline PLDA system could be that it does not explicitly compensate the session variability as it is carried out in PLDA. Thus, it is expected that adding some explicit session modeling to the proposed hybrid model could improve the performance, but it has been beyond the scope of this paper.

References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] S. J. D. Prince and J. H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. 2007 IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[3] P. Kenny, “Bayesian speaker verification with heavy tailed priors,” in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, 2010.

[4] NIST, “The NIST speaker recognition i-vector machine learning challenge,” 2014. [Online]. Available: http://nist.gov/itl/iad/mig/upload/sre-ivectorchallenge_2013-11-18_r0.p df

[5] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, “Hierarchical speaker clustering methods for the NIST i-vector challenge,” in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, 2014, pp. 254–259.

[6] S. Novoselov, T. Pekhovsky, and K. Simonchik, “STC speaker recognition system for the NIST i-vector challenge,” in *Proc. Odyssey, Speaker Lang. Recog. Workshop*, 2014, pp. 231–240.

[7] C. Greenberg *et al.* “The NIST 2014 speaker recognition i-vector machine learning challenge,” in *Proc. Odyssey, Speaker Lang. Recog. Workshop*, 2014, pp. 224–230.

[8] A. Mohamed, D. Yu, and L. Deng, “Investigation of full-sequence training of deep belief networks for speech recognition,” in *Proc. Interspeech*, 2010, pp. 2846–2849.

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[10] A. Mohamed, G. E. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[11] G. Hinton *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[12] A. Senior, H. Sak, and I. Shafran, “Context dependent phone models For LSTM RNN acoustic modelling,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4585–4589.

[13] Y. Lei, N. Scheffer, L. Ferre, and M. Mclaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1695–1699.

[14] 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*, 2014, pp. 293–298.

[15] M. Mclaren, Y. Lei, and L. Ferre, “Advances in deep neural network approaches to speaker recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4814–4818.

[16] F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, Oct. 2015.

[17] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, “Deep feature for text-dependent speaker verification,” *Speech Commun.*, vol. 73, pp. 1–13, Oct. 2015.

[18] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, “Preliminary investigation of Boltzmann machine classifiers for speaker recognition,” in *Proc. Odyssey*, 2012, pp. 109–116.

[19] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, “First attempt of Boltzmann machines for speaker verification,” in *Proc. Odyssey*, 2012, pp. 117–121.

[20] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, “PLDA using gaussian restricted Boltzmann machines with application to speaker verification,” in *Proc. Interspeech*, 2012, pp. 1692–1695.

[21] O. Ghahabi and J. Hernando, “Deep belief networks for i-vector based speaker recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 1700–1704.

[22] O. Ghahabi and J. Hernando, “i-vector modeling with deep belief networks for multi-session speaker recognition,” in *Proc. Odyssey*, 2014, pp. 305–310.

[23] 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* (Lecture Notes in Artificial Intelligence). Berlin, Germany: Springer, Nov. 2014.

[24] W. M. Campbell, “Using deep belief networks for vector-based speaker recognition,” in *Proc. Interspeech*, 2014, pp. 676–680.

[25] D. Garcia-Romero, Xiaohui Zhang, A. McCree, and D. Povey, “Improving speaker recognition performance in the domain adaptation challenge using deep neural networks,” in *Proc. 2014 IEEE Spoken Lang. Technol. Workshop*, Dec. 2014, pp. 378–383.

[26] S. Novoselov, T. Pekhovsky, K. Simonchik, and A. Shulipa, “RBM-PLDA subsystem for the NIST i-vector challenge,” in *Proc. Interspeech*, 2014, pp. 378–382.

[27] Y. Z. Isik, H. Erdogan, and R. Sarikaya, “S-vector: A discriminative representation derived from i-vector for speaker verification,” in *Proc. Eur. Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 2097–2101.

[28] V. Vasilakakis, S. Cumani, and P. Laface, “Speaker recognition by means of deep belief networks,” in *Proc. Biometric Technol. Forensic Sci.*, 2013.

[29] E. Variani, Xin Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in *Proc. 2014 IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 4052–4056.

[30] P. Safari, O. Ghahabi, and J. Hernando, “From features to speaker vectors by means of restricted Boltzmann machine adaptation,” in *Proc. Odyssey*, 2016, pp. 366–371.

[31] O. Ghahabi and J. Hernando, “Restricted Boltzmann machine supervectors for speaker recognition,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4804–4808.

[32] NIST, “The NIST year 2006 speaker recognition evaluation plan,” 2006. [Online]. Available: http://www.nist.gov/speech/tests/spk/2006/index.htm

[33] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digit. Signal Process.*, vol. 10, no. 1, pp. 19–41, Jan. 2000.

[34] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, “Cosine similarity scoring without score normalization techniques,” in *Proc. Odyssey, Speaker Lang. Recognit. Workshop*, 2010, pp. 71–75.

[35] G. E. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[36] G. E. Hinton, S. Osindero, and Y-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, May 2006.

[37] Z-H. Ling, L. Deng, and D. Yu, “Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2129–2139, Oct. 2013.

[38] X-L. Zhang and J. Wu, “Deep belief networks based voice activity detection,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 697–710, Apr. 2013.

[39] Tara N. Sainath *et al.* “Deep convolutional neural networks for Large-scale speech tasks,” *Neural Netw.*, vol. 64, pp. 39–48, Apr. 2015.

[40] Z.-H. Ling et al. “Deep learning for acoustic modeling in parametric speech Generation: A systematic review of existing techniques and future trends,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 35–52, May 2015.

[41] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, “Exploring strategies for training deep neural networks,” *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, Jun. 2009.

[42] E. Dumitru, P. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, “The difficulty of training deep architectures and the effect of unsupervised pre-training,” in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, pp. 153–160.

[43] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?” *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Mar. 2010.

[44] L. Deng and D. Yu, *Deep Learning: Methods and Applications*. Delft, The Netherlands: Now Publishers, Jun. 2014.

[45] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” in *Neural Networks: Tricks of the Trade* (Lecture Notes in Computer Science, 7700). Berlin, Germany: Springer, Jan. 2012, pp. 599–619.

[46] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[47] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, “Cost-sensitive learning methods for imbalanced data,” in *Proc. 2010 Int. Joint Conf. Neural Netw.*, Jul. 2010, pp. 1–8.

[48] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “Supervised neural network modeling: An empirical investigation into learning from imbalanced data with labeling errors,” *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 813–830, May 2010.

[49] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Inform. Sci.*, vol. 250, pp. 113–141, Nov. 2013.

[50] S. Barua, M. M. Islam, X. Yao, and K. Murase, “MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[51] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, “Data-driven background dataset selection for SVM-based speaker verification,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1496–1506, Aug. 2010.

[52] C. Nadeu, D. Macho, and J. Hernando, “Time and frequency filtering of filter-bank energies for robust HMM speech recognition,” *Speech Commun.*, vol. 34, no. 12, pp. 93–114, Apr. 2001.

[53] A. Larcher et al. “ALIZE 3.0 open source toolkit for state-of-the-art speaker recognition,” in *Proc. Interspeech*, 2013, pp. 2768–2771.

[54] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, “A study of the cosine distance-based mean shift for telephone speech diarization,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 217–227, Jan. 2014.

[55] N. Brummer and E. Villiers, “BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing,” 2011. [Online]. Available: https://sites.google.com/site/bosaristoolkit/

Omid Ghahabi received the M.Sc. degree in electrical engineering from Shahid Beheshti University, Tehran, Iran, in 2009. He is currently working toward the Ph.D. degree at the Technical University of Catalonia (UPC), Barcelona, Spain. From 2009 to 2011, he was in the Speech Processing Group, Research Center for Intelligent Signal Processing, Tehran, Iran. Between 2011 and 2016, he was a Researcher in the Speech Processing Group, Signal Theory and Communications Department, UPC. Since late 2016, he has been in the EML European Media Laboratory GmbH, Heidelberg, Germany, as a Speech Technologist. His research interests include, but not limited to, speaker recognition and diarization, speech signal processing, and deep learning. He is the author and coauthor of several journal and conference papers on these topics. He is a member of the Research Center for Language and Speech Technologies and Applications, Barcelona, Spain.

Javier Hernando received the M.S. and Ph.D. degrees in telecommunication engineering from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1988 and 1993, respectively. Since 1988, he has been in the Department of Signal Theory and Communications, UPC, where he is currently a Full Professor and the Director of the Research Center for Language and Speech. During the academic year 2002–2003, he was a Visiting Researcher in the Panasonic Speech Technology Laboratory, Santa Barbara, CA, USA. He has led the UPC team in several European, Spanish, and Catalan projects. His research interests include robust speech analysis, speech recognition, speaker verification and localization, oral dialogue, and multimodal interfaces. He is the author or coauthor of about 200 publications in book chapters, review articles, and conference papers on these topics. He received the 1993 Extraordinary Ph.D. Award of UPC.