DEEP NEURAL NETWORK DRIVEN MIXTURE OF PLDA FOR ROBUST I-VECTOR SPEAKER VERIFICATION

Na Li[1]*, Man-Wai Mak*[1]*, and Jen-Tzung Chien*[2]

[1]Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR
[2]Dept. of Electrical and Computer Engineering, National Chiao Tung University, Taiwan
lina011779@126.com, enmwmak@polyu.edu.hk

ABSTRACT

In speaker recognition, the mismatch between the enrollment and test utterances due to noise with different signal-to-noise ratios (SNRs) is a great challenge. Based on the observation that noise-level variability causes the i-vectors to form heterogeneous clusters, this paper proposes using an SNR-aware deep neural network (DNN) to guide the training of PLDA mixture models. Specifically, given an i-vector, the SNR posterior probabilities produced by the DNN are used as the posteriors of indicator variables of the mixture model. As a result, the proposed model provides a more reasonable soft division of the i-vector space compared to the conventional mixture of PLDA. During verification, given a test trial, the marginal likelihoods from individual PLDA models are linearly combined by the posterior probabilities of SNR levels computed by the DNN. Experimental results for SNR mismatch tasks based on NIST 2012 SRE suggest that the proposed model is more effective than PLDA and conventional mixture of PLDA for handling heterogeneous corpora.

***Index Terms*—** Speaker verification; i-vector; mixture of PLDA; deep neural networks; SNR mismatch.

1. INTRODUCTION

I-vectors have become a popular feature representation for most state-of-the-art text-independent speaker verification systems. By defining a total variability (TV) space, the posterior means of the latent variables of a factor analyzer [1] are considered as the fixed-length i-vectors of the corresponding utterances with different durations. However, in addition to the speaker-specific information, other undesirable information (e.g. session, channel, additive noise, and so on) is also involved in the i-vectors. More recently, probabilistic LDA (PLDA) [2] has become a common backend for i-vector based speaker verification. Given a large collection of i-vectors with speaker labels, PLDA shows a powerful data-driven mechanism to separate speaker information from other undesired variability.

More recently, researchers have done much work on using DNNs for speaker verification [3, 4, 5, 6, 7, 8, 9, 10]. It has been found that direct applications of DNNs to speaker verification have not achieved significant performance gain. A more promising strategy is to incorporate DNNs into i-vector extraction. For example, Lei *et al.* [7] demonstrated that replacing the universal background model (UBM)—which is essentially a speaker-independent Gaussian mixture model (GMM)— with a phonetically-aware DNN for computing the frame posterior probabilities produces significant improvements compared to the standard UBM/i-vector framework. In this DNN/i-vector framework, a phonetically-aware DNN trained for automatic speech recognition (ASR) is used to softly align speech frames to senone categories. Such alignments facilitate the comparison of speakers as if they pronounced the same content.

While considerable progresses have been made in i-vector extraction, how to develop a noise robust backend classifier remains a challenge. Garcia-Romero *et al*. [11] employed multi-condition training to train multiple PLDA models with tied speaker factors, one for each condition. A robust system was then constructed by combining all of the individual PLDA models according to the posterior probability of each condition. In [12], Villalba and Lleida proposed a multi-channel simplified PLDA (MCSPLDA). It is a kind of mixture model in which each channel condition (SNR level) is modeled by one channel subspace together with a channel-dependent shift while speaker variability is modeled by a single speaker subspace. The sharing of speaker subspace across all noise conditions requires the assumption that speaker variability are noise-level invariant, which may not be the case in very noisy environments because of the Lombard effects. This may be the reason why MCSPLDA can only achieve performance comparable to the conventional PLDA.

Based on the observation that i-vectors derived from utterances having similar SNR tend to cluster together in the i-vector space [13], we have proposed two approaches for

This work was in part supported by The RGC of Hong Kong SAR, Grant No. PolyU 152068/15E and 152117/14E.

noise robust speaker verification. One is the enhanced SNR-invariant PLDA [13] where multiple SNR-dependent speaker subspaces were introduced. The other one is SNR-dependent mixture of PLDA [14]. Unlike the conventional mixture of factor analyzers [15, 16] where the posteriors of the indicator variables depend on the data samples, the posteriors of the indicator variables in [14] depend on the SNRs of the utterances. One common characteristic of these two approaches is that the SNR of each test utterance should be estimated when computing the verification score. Although estimating the SNR of an utterance is not difficult, this requirement limits the application of the approaches to handling SNR mismatch only.

Inspired by the clustering phenomenon mentioned above and the success of DNN/i-vector extraction, we propose a DNN driven mixture of PLDA for robust speaker verification. In this framework, the posterior probabilities of SNR[1] given an i-vector are used as the posteriors of the indicator variables in the mixture of PLDA to guide the training of the mixture model, where the SNR posteriors are obtained from the SNR-aware DNN. In the testing stage, given the i-vectors of the target and test speakers, the SNR posteriors obtained from the DNN are used to linearly combine the marginal likelihoods of different PLDA mixtures. Therefore, unlike the SNR-dependent mixture of PLDA, the actual SNR of the target and test utterances are not necessary, only their SNR posterior probabilities are needed.

2. DNN-DRIVEN MIXTURE OF PLDA

2.1. Training SNR-aware DNN

The SNR-aware DNN aims to provide supervisory information to assist the clustering of the i-vectors into SNR-dependent groups during the training of the PLDA mixture models. It is believed that a more "crispy" division of the i-vectors can ensure that each mixture can focus on a narrow range of SNR. To this end, the network should be able to produce posterior probabilities of SNR given i-vectors as input. Fig. 1 shows the structure of such network. It accepts i-vectors as input and produces outputs in 1-of-K format so that each output node represents one SNR range. The DNN comprises several layers of restricted Boltzmann machines (RBMs) [17, 18] trained by the contrastive divergency algorithm [19, 20]. It is believed that this pre-training step can bring the network to a stage that gives better generalization from training data [21]. After pre-training, a softmax output layer is put on the top RBM and the whole network is fine-tuned by the backpropagation algorithm that minimizes the cross-entropy between the desired outputs and actual outputs. After training, the DNN can produce the posterior probabilities of SNR groups given an input i-vector.

[1]Here, we assume that the continuous SNR can be divided into a number of discrete SNR ranges.

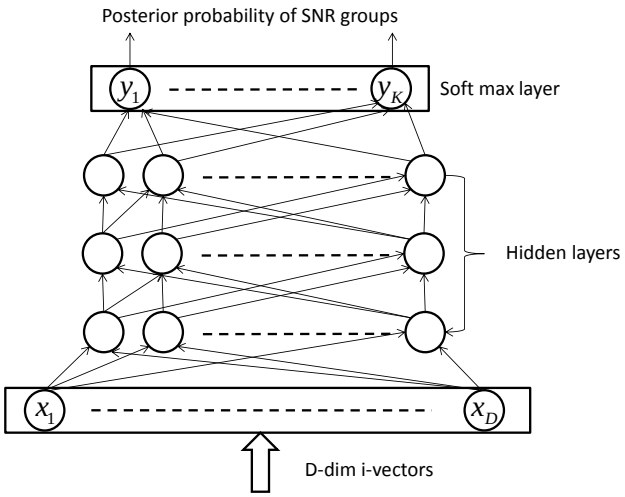

Fig. 1. Schematic diagram of the SNR-aware DNN.

2.2. Generative Model

The training of conventional mixture of PLDA (SNR-independent mPLDA in [14]) is equivalent to unsupervised clustering of i-vectors into a number of Gaussians, each with a different speaker subspace (PLDA model). Because SNR information is ignored during training, noisy i-vectors could be assigned to the clean mixture component, and clean i-vector could be aligned to the noisy component. To minimize these mis-assignments, we propose to turn the unsupervised clustering to a supervised one by incorporating the SNR information of utterance during model training. More specifically, an SNR-aware DNN is trained according to Section 2.1. For each utterance, an i-vector is extracted and presented to the DNN. The network outputs (posteriors of SNR groups) are then used as the posterior of indicator variables in the mixture model so that the clusters in the i-vector space are more dependent on the SNR levels.

Given a training i-vector $\mathbf{x}_{ij}$ from the *j*-th session of the *i*-th speaker, the posterior probability of the *k*-th SNR group obtained from the SNR-aware DNN is

$$\gamma_{\mathbf{x}_{ij}}(y_{ijk}) \equiv P(y_{ijk} = 1|\mathbf{x}_{ij}, \underline{\mathbf{w}}) \tag{1}$$

where y_{ijk} is an indicator variable specifying which of the mixture component is responsible for generating the observation $\mathbf{x}_{ij}$ and $\underline{\mathbf{w}}$ represents the weights of the SNR-aware DNN. With these definitions, the i-vectors are modeled by a mixture of K PLDA models:

$$\begin{aligned} &p(\mathbf{x}_{ij}) \\ &= \sum_{k=1}^{K} \int P(y_{ijk} = 1|\mathbf{x}_{ij}, \underline{\mathbf{w}}) p(\mathbf{x}_{ij}|\mathbf{z}, y_{ijk} = 1, \theta_k) p(\mathbf{z}) d\mathbf{z} \\ &= \sum_{k=1}^{K} \gamma_{\mathbf{x}_{ij}}(y_{ijk}) \mathcal{N}(\mathbf{x}_{ij}|\mathbf{m}_k, \mathbf{V}_k\mathbf{V}_k^\top + \mathbf{\Sigma}_k), \end{aligned} \tag{2}$$

where $\mathbf{z}$ is the speaker factor which is tied across all mixture components, $\mathbf{m}_k$, $\mathbf{V}_k$, and $\mathbf{\Sigma}_k$ represent the mean, the speaker subspace, and the covariance matrix of the k-th SNR group, respectively. The parameters of the model in Eq. 2 are denoted as $\boldsymbol{\theta} = \{\mathbf{m}_k, \mathbf{V}_k, \mathbf{\Sigma}_k\}_{k=1}^K$. We assumed that the speaker variability is modeled by $\mathbf{V}_k\mathbf{V}_k^\top$ and that the session variability is modeled by $\mathbf{\Sigma}_k$, where $k = 1, \ldots, K$.

2.3. EM Algorithm and Likelihood Ratio Scores

Denote $\mathcal{Y} = \{y_{ijk}\}_{k=1}^K$ as the set of latent indicator variables specifying which of the K factor analyzers should be selected based on the SNRs of training utterances. Specifically, $y_{ijk} = 1$ if the k-th PLDA model produces $\mathbf{x}_{ij}$, and $y_{ijk} = 0$ otherwise. Given a set of D-dim length-normalised [22] i-vectors $\mathcal{X} = \{\mathbf{x}_{ij}; i = 1, \ldots, S; j = 1, \ldots, H_i\}$. The parameters $\boldsymbol{\theta}$ can be learned from a training set using maximum likelihood estimation. Given an initial value $\boldsymbol{\theta}$, we aim to find a new estimate $\boldsymbol{\theta}'$ that maximizes the following auxiliary function:

$$
\begin{aligned}
Q(\boldsymbol{\theta}'|\boldsymbol{\theta}) &= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{\ln p(\mathcal{X},\mathcal{Y},\mathcal{Z}|\boldsymbol{\theta}')\middle|\mathcal{X},\boldsymbol{\theta}\right\} \\
&= \mathbb{E}_{\mathcal{Y},\mathcal{Z}}\left\{\sum_{ijk} y_{ijk}\ln\left[p(y_{ijk}=1|\boldsymbol{\theta}')p(\mathbf{x}_{ij}|\mathbf{z}_i,\boldsymbol{\theta}')p(\mathbf{z}_i)\right]\middle|\mathcal{X},\boldsymbol{\theta}\right\}
\end{aligned}
\tag{3}
$$

The EM formulations that maximize $Q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ are identical to Eq. 15 and Eq. 16 in [14], excepting for replacing the posterior expectations of indicator variables y_{ijk} given the SNRs $\mathcal{L} = \{\ell_{ij}\}$ by the DNN outputs. More specifically, we replace $\langle y_{ijk}|\mathcal{L}\rangle$ in [14] by $\gamma_{\mathbf{x}_{ij}}(y_{ijk})$ in Eq. 1.

During scoring, given a target-speaker i-vector $\mathbf{x}_s$ and a test i-vector $\mathbf{x}_t$, the log-likelihood ratio score will be identical to Eq. 13 and Eq. 17 in [14], excepting for the following replacements:

$$
\begin{aligned}
\gamma_{\ell_s,\ell_t}(y_{k_s}, y_{k_t}) &\leftarrow \gamma_{\mathbf{x}_s}(y_{k_s})\gamma_{\mathbf{x}_t}(y_{k_t}) \\
\gamma_{\ell_s}(y_{k_s}) &\leftarrow \gamma_{\mathbf{x}_s}(y_{k_s}) \\
\gamma_{\ell_t}(y_{k_t}) &\leftarrow \gamma_{\mathbf{x}_t}(y_{k_t})
\end{aligned}
$$

where ℓ_s and ℓ_t denote the SNR of the target-speaker's utterance and the test utterance, respectively. Specifically, the scoring function is given by Eq. 4 shown on the next page, where α is a scalar to avoid taking exponential of very large negative numbers, $\hat{\mathbf{\Lambda}}_{k_sk_t} = \hat{\mathbf{V}}_{k_s}\hat{\mathbf{V}}_{k_t}^\mathsf{T} + \hat{\mathbf{\Sigma}}_{k_sk_t}$, $\mathbf{\Lambda}_{k_s} = \mathbf{V}_{k_s}\mathbf{V}_{k_s}^\mathsf{T} + \mathbf{\Sigma}_{k_s}$, $\hat{\mathbf{\Sigma}}_{k_sk_t} = \mathrm{diag}\{\mathbf{\Sigma}_{k_s}, \mathbf{\Sigma}_{k_t}\}$, and $\mathcal{D}(\mathbf{x}\|\mathbf{y})$ is the Mahalanobis distance between $\mathbf{x}$ and $\mathbf{y}$. We set $\alpha = 5$ in this work.

Table 1. SNR ranges in dB for different numbers of SNR groups (K).

K	Group 1	Group 2	Group 3	Group 4	Group 5
2	$(-\infty, 20]$	$(20, \infty)$	–	–	–
3	$(-\infty, 8]$	$(8, 20]$	$(20, \infty)$	–	–
4	$(-\infty, 8]$	$(8, 14]$	$(14, 20]$	$(20, \infty)$	–
5	$(-\infty, 4]$	$(4, 8]$	$(8, 14]$	$(14, 20]$	$(20, \infty)$

3. EXPERIMENTAL SETUP

3.1. Speech Data and Front-End Processing

We divided the speech data into three categories: (1) development data, (2) enrollment data, and (3) test data.

- *Development Data*: The microphone and telephone speech files from NIST 2005–2008 SREs were used as development data to train the gender-dependent UBMs and total variability matrices. Babble noise was added to each telephone speech files – excluding speakers with less than two utterances – in NIST 2006–2010 SREs at an SNR of 6dB and 15dB. As a result, the original and noisy telephone speech in NIST 2006–2010 SREs and microphone speech in NIST 2008–2010 SREs were used to train the subspace projection matrices, PLDA models, PLDA mixture models, and the SNR-aware DNN. The speaker labels in the development data were obtained from the target-speaker table files in NIST 2012 SRE.[2]

- *Test Data*: All test data were extracted from NIST 2012 SRE, as defined by the `core.ndx` file in the evaluation plan. This paper focuses on common conditions (CCs) 4 and 5 of the evaluation plan.

- *Enrollment Data*: The enrollment data not only comprise the conversations as defined by the speaker-table files in NIST 2012 SRE but also comprise the noise corrupted telephone conversations of target speakers, at SNR of 6dB and 15dB. All of the 10-second utterances and summed-channel utterances formed by mixing the speech of two channels were removed from the target segments. These sentences were not used for enrollment because short utterances do not contain sufficient speech for reliable estimation of i-vectors and summed-channel utterances contain the speech of two speakers. But we ensure that each target speaker has at least one utterance for enrollment.

[2]Starting from 2012 SRE, it is legitimate to use target speakers as development data. In fact, the speakers in the target-speaker table are speakers from 2006–2010 SREs.

$$S_{\text{DNN-mPLDA}}(\mathbf{x}_s, \mathbf{x}_t) = \frac{\sum_{k_s=1}^{K}\sum_{k_t=1}^{K}\gamma_{\mathbf{x}_s}(y_{k_s})\gamma_{\mathbf{x}_t}(y_{k_t})\exp\left\{-\frac{1}{2}\log|\alpha\hat{\mathbf{\Lambda}}_{k_s k_t}| - \frac{1}{2}\mathcal{D}\left(\left[\mathbf{x}_s^{\mathsf{T}}\ \mathbf{x}_t^{\mathsf{T}}\right]^{\mathsf{T}} \| \left[\mathbf{m}_{k_s}^{\mathsf{T}}\ \mathbf{m}_{k_t}^{\mathsf{T}}\right]^{\mathsf{T}}\right)\right\}}{\left[\sum_{k_s=1}^{K}\gamma_{\mathbf{x}_s}(y_{k_s})\exp\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_s}| - \frac{1}{2}\mathcal{D}\left(\mathbf{x}_s\|\mathbf{m}_{k_s}\right)\right\}\right]\left[\sum_{k_t=1}^{K}\gamma_{\mathbf{x}_t}(y_{k_t})\exp\left\{-\frac{1}{2}\log|\alpha\mathbf{\Lambda}_{k_t}| - \frac{1}{2}\mathcal{D}\left(\mathbf{x}_t\|\mathbf{m}_{k_t}\right)\right\}\right]} \quad (4)$$

Table 2. Performance of PLDA, SI-mPLDA, SD-mPLDA, and DNN-mPLDA on CC4 and CC5 of NIST 2012 SRE core set.

Method	No. of Mixtures	Male				Female			
		CC4		CC5		CC4		CC5	
		EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
PLDA	–	3.49	0.308	2.97	0.290	3.14	0.353	2.47	0.346
SI-mPLDA	2	3.49	0.303	3.04	0.300	3.11	0.350	2.55	0.340
	3	3.31	0.302	3.06	0.286	3.02	0.351	2.41	0.345
	4	3.31	0.299	2.93	0.288	3.00	0.354	2.60	0.332
	5	3.52	0.301	3.48	0.303	3.04	0.355	2.71	0.355
SD-mPLDA	2	3.37	0.307	2.92	0.298	3.13	0.359	2.50	0.344
	3	3.06	0.315	2.80	**0.276**	**2.65**	**0.331**	2.38	0.324
	4	3.20	0.311	2.87	0.284	2.88	0.334	2.38	0.347
	5	3.24	0.321	2.87	0.287	2.77	**0.331**	2.46	0.332
DNN-mPLDA	2	**2.95**	**0.296**	2.86	0.282	2.77	0.346	2.38	0.326
	3	3.03	0.305	**2.73**	0.279	2.77	0.339	**2.36**	0.333
	4	3.10	0.319	2.78	0.278	2.79	0.347	2.38	0.329
	5	3.19	0.306	2.87	0.278	3.09	0.359	2.51	**0.323**

A two-channel voice activity detector (VAD) [23, 24] was applied to detect the speech regions of each utterance. 19 Mel frequency cepstral coefficients together with log energy plus their 1st- and 2nd-derivatives were extracted from the speech regions as detected by the VAD, followed by cepstral mean normalization [25] and feature warping [26] with a window size of 3 seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

3.2. Training of DNN and PLDA models

I-vectors were extracted based on gender-dependent UBMs with 1024 mixtures and total variability matrices with 500 total factors. The i-vectors for training the SNR-aware DNN were divided into *K* groups according to the measured SNRs of the utterances.[3] The SNRs of the whole training set were divided into K SNR intervals, as shown in Table 1. The k-th group comprises the i-vectors whose corresponding utterances have SNR falling in the k-th SNR interval. The numbers of the i-vectors in each group are comparable. The DNN classifier comprises 500 Gaussian input nodes and three hidden layers, each having 150 sigmoidal hidden units. Back propagation with mini-batch conjugate gradient descent with a batch size of 100 was performed in the pre-training stage. In the fine-tuning stage, conjugate gradient descent was used to minimize the cross-entropy loss for 30 epochs.

For training the PLDA models, similar to [28], we applied within-class covariance normalization (WCCN) [29] to whiten the i-vectors, followed by length normalization (LN) to reduce the non-Gaussian behavior of the 500-dimensional i-vectors. Then, LDA was applied to reduce intra-speaker variability and emphasize discriminative information. This procedure projects the i-vectors onto a 200-dimensional subspace so that the amount of training data should be sufficient to estimate the PLDA parameters. Then different types of PLDA models with 150 latent speaker factors were trained. These models include the following.

- *PLDA*: Conventional Gaussian PLDA.
- *SI-mPLDA*: SNR-independent mixture of PLDA in [14] where the posterior $\gamma_{\mathbf{x}_{ij}}(y_{ijk})$ in Eq. 4 is replaced by the prior probability of the k-th mixture.
- *SD-mPLDA*: SNR-dependent mixture of PLDA in [14] where the posterior $\gamma_{\mathbf{x}_{ij}}(y_{ijk})$ in Eq. 4 is obtained from a 1-dimensional (1-D) GMM modeling the SNR distribution.

[3] We modified the FaNT tool to measure the SNR. For detail, see [27].

- *DNN-mPLDA*: The proposed DNN-driven mixture of PLDA where the posterior $\gamma_{\mathbf{x}_{ij}}(y_{ijk})$ is obtained from an SNR-aware DNN that uses i-vector as input.

4. RESULTS AND ANALYSIS

We evaluated the performance of different systems using equal error rate (EER) and minimum normalized DCF (minDCF) [30].

The experiments on CC4 and CC5 shown in Table 2 and Table 3 aim to investigate the performance of different models under noise conditions.[4] Results in Table 2 show that both SD-mPLDA and DNN-mPLDA outperform the PLDA and SI-PLDA. In most cases, the proposed DNN-mPLDA performs better than SD-mPLDA. PLDA performs worse than other PLDA mixture models. The reason is that PLDA uses a single model to deal with a wide range of SNR, whereas the mixture models use a specific mixture component to deal with a much smaller range.

In contrast to SI-mPLDA, SNR information is used for assisting the clustering of i-vectors during the training of SD-mPLDA and DNN-mPLDA, which results in more proper i-vector clusters and better SNR-dependent subspace modeling. Another important advantage of SD-mPLDA and DNN-mPLDA is that the verification scores are calculated by combining the PLDA scores using the posterior probabilities of the indicator variables which are dependent on the test utterances. This leads to a very flexible mixture mechanism. On the other hand, the mixture weights in SI-mPLDA model are determined based on the training i-vectors only. Once the weights are calculated, they are used as the prior for all the mixtures. As a result, the mixture weights are independent of the test utterances during scoring. As the same combination weights are used regardless of the characteristics of the test utterance, the SI-PLDA is very inflexible.

The main difference between SD-mPLDA and DNN-mPLDA is that the former computes the posterior probabilities of y_{ijk} according to a 1-D GMM that models the SNR distribution and the latter computes the posteriors via an SNR-aware DNN using i-vector as input. Another difference is that SD-mPLDA relies on SNR information of the test utterances but DNN-mPLDA does not need such information. This trait makes DNN-mPLDA a more general model compared to SD-mPLDA.

To investigate the robustness of different models, we added different levels of noise to the test segments in CC4 and CC5 of NIST 2012 SRE to make the SNR distribution of test segments different from that of the training segments. Recall from Section 3.2 and Table 1 that the training segments comprise the original clean segments and noise contaminated segments with a wide range of SNR. The results in Table 3 show that the proposed model performs better than other models when the SNR distributions of training and test utterances are very different.

[4] The results in Table 2 are slightly different from those in our earlier paper (Table III in [14]) because we re-run all experiments and sped up the EM by training the 1-D GMM and the mixture model separately.

5. CONCLUSION

This paper proposes a new approach to training and scoring PLDA mixture models for robust speaker verification. A DNN is discriminatively trained using the SNR information embedded in the training data. This DNN is used to guide the training stage of PLDA mixture models, making each mixture can precisely model one cluster in the i-vector space. In the testing stage, the verification scores are computed by combining the PLDA scores with utterance-dependent weights. Experimental results suggest PLDA mixture models that leverage SNR information implicitly embedded in i-vectors significantly outperform those mixture models that do not make use of such information.

6. REFERENCES

[1] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification.," in *Proc. Interspeech*, 2009, pp. 1559–1562.

[2] S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007, pp. 1–8.

Table 3. Performance of PLDA, SI-mPLDA, SD-mPLDA, and DNN-mPLDA on CC4 and CC5 of NIST 2012 SRE (core set) for female speakers. K is set to 3 for mixture models.

Method	CC4 (6dB)		CC4 (15dB)		CC5 (6dB)		CC5 (15dB)	
	EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
PLDA	3.16	0.403	2.81	0.360	6.05	0.589	3.48	0.410
SI-mPLDA	3.14	0.396	3.00	0.360	5.96	0.572	3.49	0.402
SD-mPLDA	3.07	0.412	2.62	**0.349**	**5.86**	**0.560**	3.31	**0.385**
DNN-mPLDA	**2.97**	**0.379**	**2.58**	**0.349**	5.87	0.565	**3.28**	0.389

[3] S. Yaman, J.W. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition.," in *Odyssey*, 2012, vol. 12, pp. 105–108.

[4] T. Yamada, L.B. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of DNN," in *Proc. Interspeech*, 2013, pp. 3661–3664.

[5] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *Proc. ICASSP*, 2014, pp. 1700–1704.

[6] E. Variani, X. Lei, E. McDermott, Ignacio Lopez M., and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 4052–4056.

[7] 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*, 2014, pp. 1695–1699.

[8] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proc. Interspeech*, 2015, pp. 1151–1155.

[9] D. Garcia-Romero and A. McCree, "Insights into deep neural networks for speaker recognition," in *Proc. Interspeech*, 2015, pp. 1141–1145.

[10] X.J. Zhao, Y.X. Wang, and D.L. Wang, "Deep neural networks for cochannel speaker identification," in *Proc. ICASSP*, 2015, pp. 4824–4828.

[11] D. Garcia-Romero, X. Zhou, and C.Y. Espy-Wilson, "Multi-condition training of Gaussian PLDA models in i-vector space for noise and reverberation robust speaker recognition," in *Proc. ICASSP*, 2012, pp. 4257–4260.

[12] J. Villalba and E. Lleida, "Handling i-vectors from different recording conditions using multi-channel simplified PLDA in speaker recognition.," in *Proc. ICASSP*, 2013, pp. 6763–6767.

[13] N. Li and M.W. Mak, "SNR-invariant PLDA with multiple speaker subspaces," in *Proc. ICASSP*, 2016, pp. 2317–2321.

[14] M.W. Mak, X.M. Pang, and J.T. Chien, "Mixture of PLDA for noise robust i-vector speaker verification," *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 24, no. 1, pp. 130–142, 2016.

[15] G. McLachlan and D. Peel, "Mixtures of factor analyzers," *Finite Mixture Models*, pp. 238–256, 2000.

[16] Z. Ghahramani and G.E. Hinton, "The EM algorithm for mixtures of factor analyzers," Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, 1996.

[17] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

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

[19] G. E Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[20] G. Hinton, S. Osindero, and Y.W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[21] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[22] D. Garcia-Romero and C.Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011, pp. 249–252.

[23] M.W. Mak and H.B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer, Speech and Language*, vol. 28, no. 1, pp. 295–313, Jan 2014.

[24] H.B. Yu and M.W. Mak, "Comparison of voice activity detectors for interview speech in NIST speaker recognition evaluation," in *Interspeech*, 2011, pp. 2353–2356.

[25] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, no. 6, pp. 1304–1312, Jun. 1974.

[26] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Odyssey: The Speaker and Language Recognition Workshop*, Crete, Greece, Jun. 2001, pp. 213–218.

[27] N. Li and M.W. Mak, "SNR-invariant PLDA modeling in nonparametric subspace for robust speaker verification," *IEEE/ACM Trans. on Audio, Speech and Language Processing*, vol. 23, no. 10, pp. 1648–1659, 2015.

[28] M. McLaren, M.I. Mandasari, and D.A. Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," in *Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012, pp. 55–61.

[29] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *ISCSLP*, Pittsburgh, PA, USA, Sep. 2006, pp. 1471–1474.

[30] NIST, "The NIST year 2012 speaker recognition evaluation plan," *http://www.nist.gov/itl/iad/mig/sre12.cfm*, 2012.