I-Vector DNN Scoring and Calibration for Noise Robust Speaker Verification

Zhili TAN and Man-Wai MAK

Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University

eddy.zhili@connect.polyu.hk, enmwmak@polyu.edu.hk

Abstract

This paper proposes applying multi-task learning to train deep neural networks (DNNs) for calibrating the PLDA scores of speaker verification systems under noisy environments. To facilitate the DNNs to learn the main task (calibration), several auxiliary tasks were introduced, including the prediction of SNR and duration from i-vectors and classifying whether an i-vector pair belongs to the same speaker or not. The possibility of replacing the PLDA model by a DNN during the scoring stage is also explored. Evaluations on noise contaminated speech suggest that the auxiliary tasks are important for the DNNs to learn the main calibration task and that the uncalibrated PLDA scores are an essential input to the DNNs. Without this input, the DNNs can only predict the score shifts accurately, suggesting that the PLDA model is indispensable.

Index Terms: Deep learning, speaker verification, score calibration, multi-task learning, noise robustness

1. Introduction

Since 2011, i-vectors [1] together with probabilistic linear discriminant analysis (PLDA) [2, 3] have been the state-of-the-art methods for speaker verification. In 2014, the deep neural network (DNN)-based i-vectors that incorporate phonetic information [4] further improve speaker verification performance.

Because of its success, a lot of effort has been made to improve the noise and duration robustness of the i-vector/PLDA framework in recent years. For example, attempts have been made to enhance and restore speech in the feature domain [5] using factor analysis and in the spectral domain [6, 7] or i-vector space [8, 9] using denoising autoencoders (DAE) [10]. Improving noise robustness of PLDA models is another direction. Hasan *et al.* [11] and Garcia-Romero *et al.* [12] trained a PLDA model by pooling speech from multiple conditions, and Li and Mak [13, 14] modeled the noise-level variability in utterances by introducing an SNR factor and an SNR subspace into the PLDA model. In [15], Mak *et al.* advocated that utterances of different SNR levels will not only cause the i-vectors to fall on different regions of the i-vector spaces but also change the orientation of the speaker subspace. A mixture PLDA model with mixture alignments determined by the SNR level of utterances [15] or by their i-vectors [16] was then derived to model the SNR-dependent i-vectors.

Observing that adverse acoustic conditions and duration variability in utterances could have detrimental effect on PLDA scores, a number of score calibration methods have been proposed to compensated for the effect by modeling it as a shift in the PLDA scores. While some of these methods only compensate for the duration mismatch between the i-vector pair during PLDA scoring [17, 18, 19], there are techniques also taking the SNR mismatch into account [20, 21]. In [22], the shift is assumed to follow a Gaussian distribution with mean and variance dependent on the speech quality. On the other hand, the score shift in [23, 24] is assumed to be simple functions (bilinear transformation and cosine distance) of the two quality vectors derived from the i-vectors involved in the scoring.

In [21], a quality measure function (QMF) was proposed to compensate for the score shift caused by background noise:

$$S' = w_0 + w_1 S + w_2 \text{SNR}_{tst} + w_3 \text{SNR}_{tgt}, \quad (1)$$

where S is a PLDA score, SNR_{tst} and SNR_{tgt} are the SNR of the test and target utterances, respectively, and w_i's are calibration weights. As background noise can distort the i-vectors, which in turn will shift the PLDA scores, it is more intuitive to estimate the score shift directly from the i-vectors rather than from the SNR of target and test utterances. This motivates us to develop DNN-based score calibration methods.

In [25], we proposed to estimate the score shifts by multi-task DNNs using noisy i-vector pairs and their corresponding PLDA scores as input. Moreover, instead of expressing the score shifts as a linear function of SNRs, we used the SNRs of training utterances as part of the target outputs and applied multi-task learning to guide the network to produce the ideal score shifts or clean scores. In this paper, we extend the multi-task DNNs in [25] in three respects. First, in addition to using SNR as target outputs, we also use utterance duration and same-speaker and different-speaker hypotheses as target outputs. Second, we compute the posterior odds of same-speaker and different-speaker hypotheses and use the odds as verification scores. Third, we explore the potential of replacing the PLDA model by a multi-task DNN which receives i-vector pairs as input only. Results based on noise contaminated speech in NIST 2012 SRE suggest that the multi-task DNNs can effectively calibrate the scores produced by a PLDA model, leading to superior performance as compared to the conventional linear calibration method.

2. DNN Score Calibration/Scoring

2.1. Estimating Score Shifts by DNNs

A DNN can be trained to estimate the appropriate score shift given the target and test i-vector pairs $\boldsymbol{x}_{tgt}$ and $\boldsymbol{x}_{tst}$ and the uncalibrated PLDA score S:

$$\text{DNN}_1(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}, S) \approx \delta_{score}, \quad (2)$$

where δ_{score} denotes the ideal score shift that will bring S to the clean score S_{cln} as if both $\boldsymbol{x}_{tgt}$ and $\boldsymbol{x}_{tst}$ were derived from clean utterances. Given the estimated score shift, the calibrated score can be computed as:

$$S'_1 = S + \text{DNN}_1(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}, S). \quad (3)$$

During the calibration stage, the DNN and the PLDA scor-

This work was supported by the RGC of the Hong Kong SAR, Project Nos. PolyU 152068/15E and PolyU 152518/16E.

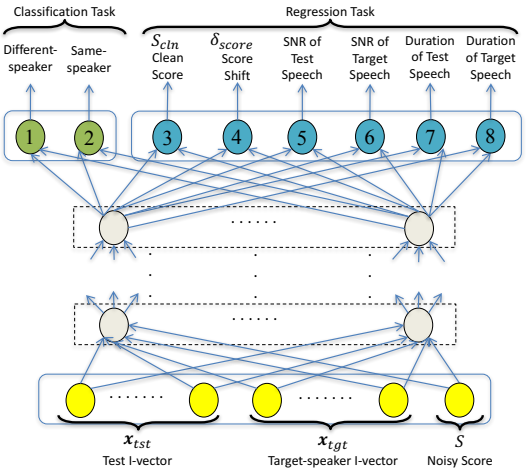

Figure 1: *DNN with classification and regression tasks.*

ing function receive the same i-vector pair, where the former computes the score shift, δ_{score}, and the latter computes the *noisy* PLDA score S. By substituting Eq. 2 into Eq. 3, we have $S_1' \approx S + \delta_{score} = S_{cln}$, i.e., we recover the clean score.

2.2. Recovering Clean PLDA Scores by DNNs

In the above methods, scores are calibrated by shifting and scaling. However, if the clean scores can be directly restored, the estimation of score shifts seems to be redundant. To make the calibrated scores close to the ideal clean scores, we can use a DNN to model the complex relationship between the i-vector pairs, noisy scores (S), and the clean scores (S_{cln}) as follows:

$$S_2' = \text{DNN}_2(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}, S) \approx S_{cln}. \quad (4)$$

The DNN is trained to produce clean scores given i-vector pairs and their corresponding noisy scores as input.

The DNNs in Eq. 2 and Eq. 4 have hundreds of input nodes but only one output node. Their goal is to learn a regression task to produce the desired score shifts or clean scores. During training, the squared errors in the output node will need to be propagated to hundreds of nodes in both the hidden and input layers. Our experience is that having a single source of errors makes the backpropagation (BP) of error gradients very inefficient. One possible solution is to introduce some auxiliary tasks for the network to learn. In the literature, this is known as multi-task learning [26, 27]. Therefore, a multi-task DNN with auxiliary information in the output layer may help to improve the learning efficiency.

Fig. 1 shows a DNN that uses multi-task learning to learn not only the main task but also some auxiliary tasks. The main task is to produce the score shifts and the calibrated scores, and the auxiliary tasks are to predict the SNRs and durations of target-speaker's and test utterances and the same-speaker and different-speaker posteriors. These auxiliary tasks are selected because they have great influence on the i-vectors and PLDA scores. To leverage multi-task learning, the network is trained to achieve two different tasks: regression and classification. There are 6 output nodes in the regression task and 2 output nodes in the classification task. Note that the main task (Nodes 3 and 4) is part of the regression task and that the auxiliary tasks involve both classification (Nodes 1 and 2) and regression (Nodes 5–8). The regression part of the DNN uses linear output nodes and minimum mean squared error as the optimization criterion, whereas the classification part uses softmax outputs and cross-entropy as the optimziation criterion. Note that the input and target values in the regression task are subject to z-normalization.

During training, given an i-vector pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and a noisy uncalibrated PLDA score S of these two i-vectors, the DNN is trained to output a target vector $\mathbf{t}_{ij}$:

$$\text{DNN}_3(\boldsymbol{x}_i, \boldsymbol{x}_j, S) \approx \mathbf{t}_{ij} = \left[\underbrace{[y_1, y_2]}_{\text{Classification}}, \underbrace{[S_{cln}, \delta_{score}, \text{SNR}_i, \text{SNR}_j, \text{Dur}_i, \text{Dur}_j]}_{\text{Regression}} \right], \quad (5)$$

where SNR_i and SNR_j are the SNRs of the two utterances, Dur_i and Dur_j are their durations, S_{cln} is the clean score if both utterances were clean, and δ_{score} is the ideal score shift. Also, $[y_1, y_2] = [1, 0]$ if the two utterances are from the same speaker; otherwise $[y_1, y_2] = [0, 1]$.

During score calibration, only the clean scores (Node 3) and the score shifts (Node 4) produced by the DNN will be used:

$$S_3' = \text{DNN}_{3,cln}(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}, S) \approx S_{cln} \quad (6)$$

$$S_4' = S + \text{DNN}_{3,shift}(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}, S) \approx S + \delta_{score} = S_{cln}, \quad (7)$$

where $\boldsymbol{x}_{tgt}$ and $\boldsymbol{x}_{tst}$ are the target-speaker's and test-speaker's i-vectors, respectively. To make the score shifts compatible with S, they are subject to inverse z-normalization. More specifically, $\text{DNN}_{3,shift}$ is the inverse z-norm of output Node 4.

Nodes 1 and 2 give the posterior probabilities (p^+, p^-) of same-speaker and different-speaker hypotheses, which can be leveraged to give the posterior odds:

$$S_5' = \log\left(\frac{\text{DNN}_{3,cf}(\boldsymbol{x}_{tst}, \boldsymbol{x}_{tgt}, S)[1]}{\text{DNN}_{3,cf}(\boldsymbol{x}_{tst}, \boldsymbol{x}_{tgt}, S)[2]}\right) = \log\left(\frac{p^+}{p^-}\right), \quad (8)$$

where $\text{DNN}_{3,cf}(\boldsymbol{x}_{tst}, \boldsymbol{x}_{tgt}, S)[i]$, $i = 1, 2$, represents Node i in Fig. 1. Then, S_5' can be used as verification scores.

2.3. DNN Scoring Machine

It is of interest to train a multi-task DNN *without* using the noisy PLDA scores as input. The advantage of this approach is that the PLDA model is not necessary during the scoring stage, i.e., given an i-vector pair, we can obtain the approximated clean score or score shift from the DNN's outputs. We refer to the resulting DNN as DNN scoring machine.

During training, given an i-vector pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the DNN scoring machine is trained to achieve both the classification and regression tasks using $\mathbf{t}_{ij}$ as the target vector:

$$\text{DNN}_4(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx \mathbf{t}_{ij} = \left[\underbrace{[y_1, y_2]}_{\text{Classification}}, \underbrace{[S_{cln}, \delta_{score}, \text{SNR}_i, \text{SNR}_j, \text{Dur}_i, \text{Dur}_j]}_{\text{Regression}} \right], \quad (9)$$

where the elements of $\mathbf{t}_{ij}$ have the same definitions as those in Eq. 5. Note that Eqs. 9 and 5 differ in the input only. Architecturally, this is equivalent to removing the input node S in Fig. 1.

After training, only the clean scores (*cln*) and the score shifts (*shift*) produced by the multi-task DNN will be used:

$$S_6' = \text{DNN}_{4,cln}(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}) \approx S_{cln}, \quad (10)$$

$$S_7' = S + \text{DNN}_{4,shift}(\boldsymbol{x}_{tgt}, \boldsymbol{x}_{tst}) \approx S_{cln}. \quad (11)$$

Table 1: *Performance of various score calibration methods on CC4 of NIST 2012 SRE (male, core task) with test utterances contaminated with different levels of babble noise. For the DNN-based method, the network in Fig. 1 was trained by using different combinations of auxiliary tasks, including classification (Cls, Nodes 1–2), score shift (SS, Node 4), SNR of target and test utterances (SNR, Nodes 5–6), and duration of target and test utterances (Dur, Nodes 7–8).*

Score Calibration Method	Auxiliary tasks of DNN	Original			15dB			6dB			0dB		
		EER	minDCF	actDCF	EER	minDCF	actDCF	EER	minDCF	actDCF	EER	minDCF	actDCF
Baseline (no calibration)	N/A	1.56	0.218	0.855	2.27	0.225	0.778	2.29	0.276	0.749	5.37	0.753	0.779
SNR-dep Score Shift (Eq. 1)	N/A	1.68	0.209	0.780	2.24	0.215	0.770	2.28	0.269	0.811	5.35	0.754	0.794
Recover Clean Scores by DNN (Eq. 6)	None	14.61	0.910	1.000	15.41	0.881	1.000	17.05	0.942	1.000	21.44	1.002	1.000
	Cls	1.57	0.194	0.643	2.29	**0.211**	0.572	2.25	0.246	0.582	3.65	0.438	0.603
	Cls + SS	1.59	0.193	0.595	2.26	**0.211**	0.527	2.24	0.243	0.531	3.55	0.425	0.548
	Cls + SS + SNR	**1.50**	**0.189**	**0.517**	**2.21**	**0.211**	**0.455**	**2.16**	0.248	**0.470**	**3.48**	**0.409**	**0.516**
	Cls + SS + SNR + Dur	1.56	0.193	0.763	2.31	0.212	0.700	2.21	**0.239**	0.716	3.58	0.430	0.744

Table 2: *Performance of various DNN-based score calibration methods on CC4 of NIST 2012 SRE (male, core task) with test utterances contaminated with different levels of babble noise.*

Score Calibration Method	Original		15dB		6dB		0dB	
	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
Estimate SNR-dep Score Shift (Eq. 1)	1.68	0.209	**2.24**	0.215	2.28	0.269	5.35	0.754
Recover Clean Scores by DNN (Eq. 6)	1.56	0.193	2.31	0.212	**2.21**	0.239	3.58	0.430
Estimate Score Shifts by DNN (Eq. 7)	**1.54**	**0.192**	2.30	0.211	**2.21**	**0.238**	3.57	0.428
Use Posterior Odds as Scores (Eq. 8)	1.70	0.193	2.25	**0.210**	2.23	0.245	**3.56**	**0.426**

Similar to DNN_3, we may also use the posterior probabilities produced by DNN_4 to compute the posterior odds and use them as verification scores:

$$S'_8 = \log\left(\frac{\text{DNN}_{4,cf}(\boldsymbol{x}_{tst}, \boldsymbol{x}_{tgt})[1]}{\text{DNN}_{4,cf}(\boldsymbol{x}_{tst}, \boldsymbol{x}_{tgt})[2]}\right) = \log\left(\frac{p^+}{p^-}\right). \quad (12)$$

3. Experiments

3.1. Speech Data and Acoustic Features

Evaluations were conducted on the NIST 2012 SRE under Common Condition 4 (CC4, male). Speech files from NIST 2005–2010 SREs were used as development data. Speech regions were extracted by using a two-channel voice activity detector [28]. A 60-dim vector comprising energy, MFCCs, and their first and second derivatives was extracted every 10ms.

To obtain the performance under noisy conditions, we used the FaNT tool [29] to add babble noise to the target-speaker utterances and test utterances at an SNR of 15dB, 6dB, and 0dB, respectively. Therefore, we have four groups of training utterances and four groups of test utterances, with the first group being the original utterances and the last three groups having SNRs close to 15dB, 6dB, and 0dB, respectively. Hereafter, we refer to these 4 groups as SNR groups.

3.2. DNN Training

To train the multi-task DNNs, we used the i-vectors derived from the clean utterances and the 3 groups of noise contaminated utterances to give a rich set of clean scores S_{cln}, noisy scores S, and score shifts δ_{score}. We formed utterance pairs from the clean and noise contaminated groups. When both utterances in a pair come from the clean group, we treated their PLDA score as clean, i.e., S_{cln}. If any of the utterances in the pair is from the noise contaminated groups, we treated their PLDA scores as noisy, i.e., S. For each utterance pair, their clean PLDA score, ideal score shift, SNRs, durations, and class (same-speaker or different-speaker) were used as the target values for DNN training. This procedure gives us 1.5 million input/output pairs for same-speaker utterance pairs and 400 million different-speaker utterance pairs for training.

Restricted Boltzmann machines with 256 hidden nodes were trained layer-by-layer [30, 31], resulting in 4 hidden layers for each DNN. The output layer was initialized with small random weights. Then we applied 300 iterations of backpropagation (BP) to minimize the cross entropy in the classification task with a learning rate of 0.005 and to minimize the mean squared error in the regression task with a learning rate of 0.05. Both the inputs and desired regression outputs of the DNNs were preprocessed by z-normalization.

3.3. Denoising Senone I-vectors and PLDA Model

We used a senone i-vector/PLDA system [7] to produce the uncalibrated scores. The 500-dimensional senone i-vectors were whitened by within-class covariance normalization (WCCN) [32] and length normalization [33], followed by linear discriminant analysis (LDA) to reduce the dimension to 200 and variance normalization by WCCN [34].

The PLDA model was trained by using the utterances from the 4 SNR groups mentioned in Section 3.1 and the i-vectors derived from the microphone utterances (interview speech) of the same set of target speakers in NIST 2006–2010 SREs. All of the *calibrated* scores are subject to further calibration to produce true likelihood-ratio scores using the Bosaris toolkit [35].

4. Results and discussion

4.1. Importance of Uncalibrated Scores in DNN Input

Fig. 2 plots the l^2-norm of the weight vectors corresponding to the strength of connections between the input layer and the first hidden layer. The figure suggests that the connection strength between the uncalibrated score S and the first hidden layer is much stronger than that between the i-vector pair and the first hidden layer. This means that the uncalibrated PLDA scores play an important role in this DNN.

4.2. Importance of the Classification Task

To highlight the importance of the classification task, we trained two multi-task DNNs, one with the classification task and one without the classification task. Then, we observed the mean

Table 3: *Performance of multi-task DNNs without using the noisy PLDA scores as input. The test conditions are the same as Table 2.*

Row	Scoring Method	Score Calibration Method	Original		15dB		6dB		0dB	
			EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
1	PLDA	SNR-dep Score Shift (Eq. 1)	1.68	0.209	2.24	0.215	2.28	0.269	5.35	0.754
2	Recover Clean Score (Eq. 10)	N/A	2.51	0.308	3.33	0.311	3.02	0.349	**3.61**	**0.456**
3	PLDA	Score Shift (Eq. 11)	**1.39**	**0.166**	**2.14**	**0.192**	**1.96**	**0.230**	3.80	0.571
4	Posterior Odds (Eq. 12)	N/A	3.37	0.415	4.22	0.410	4.52	0.445	5.54	0.660

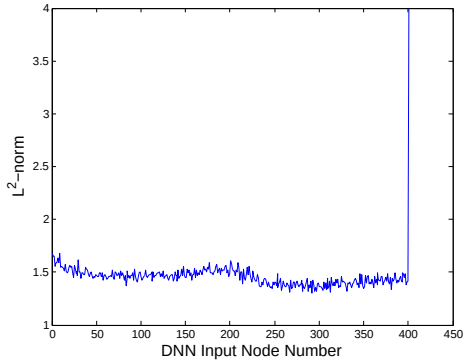

Figure 2: *The l^2-norm of the weight vectors in the bottom layer of Fig. 1. Each input node number corresponds to one weight vector representing the strength of that particular input to the first hidden layer. Of the 401 input nodes, the first 400 correspond to the pre-processed i-vector pairs, and the last one corresponds to the noisy PLDA score.*

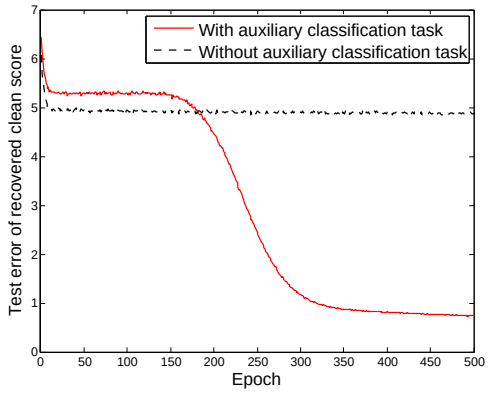

Figure 3: *The mean squared test error between the recovered clean scores and the true clean score for 500 epochs of BP.*

squared error between the clean test scores and the recovered scores (from Node 3 of Fig. 1) during the course of multi-task training. Fig. 3 shows the test error against the iteration number for 500 epochs of BP. The result shows that although the classification task leads to a higher test error at the beginning, it can guide the DNN to find a better solution after 200 epochs. Therefore the classification task plays an important role in the training of the multi-task DNNs.

4.3. Results on NIST 2012 SRE

Table 1 shows the performance of various score calibration methods, including the SNR-dependent score shift (Eq. 1)[1] and DNN-based methods. For the latter, the clean PLDA scores were recovered from noisy PLDA scores (Eq. 6) using multi-task DNNs trained with different auxiliary tasks. All of the networks received i-vector pairs and PLDA scores as input.

The 3rd row in Table 1 represents the situation where all of the auxiliary tasks have been removed, which results in a single-task DNN. The performance of this single-task DNN is significantly poorer than that of the baseline. The poor performance is attributed to the inability of the network to recover the clean scores, as Fig. 3 suggests. Comparing the third and fourth rows, the auxiliary classification task can assist the network to estimate the ideal clean scores, leading to comparable performance to the baseline. The SNR information improves the robustness significantly. The duration information, however, is not helpful, as evident by the slight performance degradation after adding duration to the auxiliary task.

Table 2 shows the performance of the multi-task DNN with noisy PLDA score as input shown in Fig. 1. Results show that the three variant of calibrations defined in Eqs. 6, 7 and 8 have similar performance under all SNR conditions. But all of them outperform the baseline (Eq. 1), especially at 0dB. Interestingly, although the posterior odds are derived from the outputs of an auxiliary task, they perform quite well and are better than the baseline in most cases.

Table 3 shows the performance of the DNN scoring machine described in Section 2.3. Evidently, under clean and moderately noisy conditions ($\geq$ 6dB), the performance is good only when PLDA scoring is used. The DNN scoring machine outperforms the baseline and others only when the noise level is very high (0dB). This means that without using the noisy PLDA scores as input, the DNN is not able to recover the clean scores (Row 2). The reliability of the posterior probability outputs is also questionable (Row 4). This is to be expected because according to Fig. 2 and the discussions in Section 4.1, the noisy uncalibrated scores play an important role in recovering the clean scores. However, Table 3 shows that even without the noisy PLDA scores as input, the DNN is still able to estimate the score shift accurately, leading to the best performance in Row 3. But, bear in mind that Row 3 requires the noisy PLDA scores S during the scoring stage, as suggested in Eq. 11. This means that the PLDA model is still indispensable.

5. Conclusions

This paper proposes several DNN-based score calibration algorithms, where the calibrated scores, score shifts and posteriors of same-speaker and different-speaker hypotheses were estimated. The three usages of the multi-task DNNs have very close performance if the noisy scores are used as part of the inputs. Without the uncalibrated PLDA scores as input, the DNNs can only estimate the score shifts with sufficient accuracy to improve performance. In summary, the best performance can be achieved if multi-task DNNs are used for calibrating the scores produced by the PLDA model.

[1] We used the FoCal toolkit to find the weights of the QMF in Eq. 1.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

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

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, Brno, Czech Republic, June 2010.

[4] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 105–116, 2016.

[5] T. Hasan and J. Hansen, "Acoustic factor analysis for robust speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 842–853, 2013.

[6] J. H. Liu, W. Q. Zheng, and Y. X. Zou, "A robust acoustic feature extraction approach based on stacked denoising autoencoder," in *Proc. BigMM*, 2015, pp. 124–127.

[7] Z. L. Tan, Y. K. Zhu, M. W. Mak, and B. Mak, "Senone i-vectors for robust speaker verification," in *Proc. ISCSLP*, Oct. 2016.

[8] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Proc. Odyssey*, 2016.

[9] S. Novoselov, T. Pekhovsky, O. Kudashev, V. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," *Proc. Interspeech*, pp. 214–218, September 2015.

[10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[11] T. Hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Bořil, and J. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," in *Proc. ICASSP*. IEEE, 2013, pp. 6783–6787.

[12] 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*. IEEE, 2012, pp. 4257–4260.

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

[14] ——, "SNR-invariant PLDA with multiple speaker subspaces," in *Proc. ICASSP*. IEEE, 2016, pp. 5565–5569.

[15] 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.

[16] N. Li, M.-W. Mak, and J.-T. Chien, "DNN-driven mixture of PLDA for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1371–1383, 2017.

[17] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *Proc. ICASSP*, May 2013, pp. 7663–7667.

[18] Q. Hong, L. Li, M. Li, L. Huang, L. Wan, and J. Zhang, "Modified-prior PLDA and score calibration for duration mismatch compensation in speaker recognition system," in *Proc. Interspeech*, 2015.

[19] A. Shulipa, S. Novoselov, and Y. Matveev, "Scores calibration in speaker recognition systems," in *Speech and Computer: 18th International Conference*, Budapest, Hungary, August 2016, pp. 596–603.

[20] M. I. Mandasari, R. Saeidi, M. McLaren, and D. A. van Leeuwen, "Quality measure functions for calibration of speaker recognition systems in various duration conditions," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2425–2438, Nov 2013.

[21] M. I. Mandasari, R. Saeidi, and D. A. van Leeuwen, "Quality measures based calibration with duration and noise dependency for speaker recognition," *Speech Communication*, vol. 72, pp. 126 – 137, 2015.

[22] A. O. J. Villalba, A. Miguel and E. Lleida, "Bayesian networks to model the variability of speaker verification scores in adverse environments," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2327–2340, 2016.

[23] A. Nautsch, R. Saeidi, C. Rathgeb, and C. Busch, "Robustness of quality-based score calibration of speaker recognition systems with respect to low-SNR and short-duration conditions," in *Proc. Odyssey*, 2016, pp. 358–365.

[24] L. Ferrer, L. Burget, O. Plchot, and N. Scheffer, "A unified approach for audio characterization and its application to speaker recognition," in *Proc. Odyssey*, 2012, pp. 317–323.

[25] Z. L. Tan, M. W. Mak, and B. Mak, "DNN-based score calibration with multi-task learning for noise robust speaker verification," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, submitted.

[26] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[27] D. Chen and B. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 7, pp. 1172–1183, 2015.

[28] 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, 2014.

[29] H. Hirsch, "FaNT-filtering and noise adding tool," 2005.

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

[31] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, vol. 19, p. 153, 2007.

[32] M. McLaren, M. Mandasari, and D. Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," in *Proc. Odyssey*, 2012, pp. 55–61.

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

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

[35] N. Brümmer and E. de Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.