APPLYING COMPENSATION TECHNIQUES ON I-VECTORS EXTRACTED FROM SHORT-TEST UTTERANCES FOR SPEAKER VERIFICATION USING DEEP NEURAL NETWORK

IL-Ho Yang, Hee-Soo Heo, Sung-Hyun Yoon, and Ha-Jin Yu

School of Computer Science, University of Seoul, Seoul, Korea

ABSTRACT

We propose a method to improve speaker verification performance when a test utterance is very short. In some situations with short test utterances, performance of i-vector/probabilistic linear discriminant analysis systems degrades. The proposed method transforms short-utterance feature vectors to adequate vectors using a deep neural network, which compensate for short utterances. To reduce the dimensionality of the search space, we extract several principal components from the residual vectors between every long utterance i-vector in a development set and its truncated short utterance i-vector. Then an input i-vector of the network is transformed by linear combination of these directions. In this case, network outputs correspond to weights for linear combination of principal components. We use public speech databases to evaluate the method. The experimental results on short2-10sec condition (det6, male portion) of the NIST 2008 speaker recognition evaluation corpus show that the proposed method reduces the minimum detection cost relative to the baseline system, which uses linear discriminant analysis transformed i-vectors as features.

Index Terms— speaker verification, i-vector, deep neural network, principal components analysis

1. INTRODUCTION

I-vector/probabilistic linear discriminant analysis (PLDA) speaker verification systems [1] show good accuracy when training and test utterances are sufficiently long. However, in many real-world applications of speaker verification, the durations of test utterances are limited. Kanagasundaram et al. reported in [2] that as the duration of a test utterance decreases, the performance of the i-vector/PLDA system degrades.

Some related studies have been conducted to avoid performance degradation caused by duration of utterances in i-vector/PLDA systems. Sarkar et al. [3] trained statistical parameters of an i-vector system using both short and long utterances to improve performance when the duration of a target speaker's training and testing utterances mismatch. Kenny et al. [4] quantified the uncertainty associated with the i-vector extraction process and propagated it into a PLDA classifier to manage duration variability properly. Cumani et al. [5] proposed a new PLDA model that exploits the uncertainty of the i-vector extraction process. In [6, 7], score calibration methods were introduced that use quality measure functions to include utterance duration in the calibration transformation. Vesnicer et al. [8] presented a duration-based weighting techniques for controlling the impact of a given i-vector to the overall statistics being computed. Kanagasundaram et al. [9] proposed a technique to improve speaker verification performance when only short utterances are available.

In this research, we propose a method to reduce the performance degradation caused by short-test utterances using a deep learning technique that transforms original test feature vectors. However, training utterances are assumed to have sufficiently long duration.

Recently, deep learning has garnered considerable attention in the wide field of machine learning. Few studies have been conducted that apply deep learning to speaker verification systems. Variani et al. [10] extracted deep-vectors (d-vectors) (instead of i-vectors) as feature vectors for text-dependent speaker verification using a deep neural network (DNN) trained to identify speakers in a development set. Lei et al. [11] introduced a novel framework of deep learning for speaker verification systems that uses a DNN phone-state recognizer to compute statistics during the i-vector extraction step. Especially, Yamamoto et al. described denoising autoencoder (DAE) based speaker feature restoration for utterances of short duration [12], which is the most similar to our research. Their method showed improved speaker verification accuracy when fused with the baseline i-vector system. However, it did not show improvement when used as a single system, and they used enrollment set in training of the DAE. Moreover, the demonstration performed only with a shallow network structure, even though their methodology can be expanded to deep structure.

Our goal in this paper comes out from there. We start from aforementioned DAE-like system with deep structure, and develop it to overcome its drawbacks. To improve the DNN compensator, residual-learning [13] and principal components analysis (PCA) are applied.

The rest of this paper is organized as follows. Section 2 describes the baseline and the proposed systems. Section 3 explains the experimental setup. We report the experimental results in Section 4 and conclude the paper in Section 5. We provide a discussion about the relation between our work and the prior works in Section 6.

2. METHODS

2.1. I-vector/PLDA system (baseline1)

We use the well-known i-vector/PLDA system in this research. Fig. 1 shows the overall procedure.

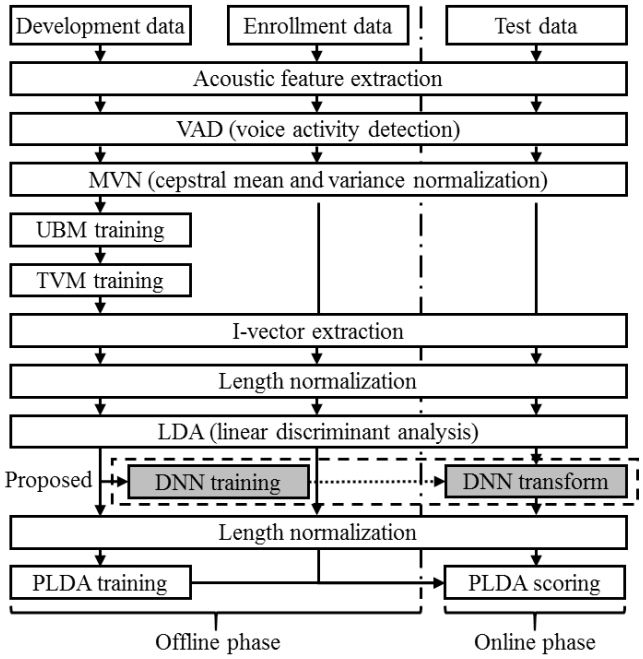

Fig. 1. Overall flow diagram

Length-normalized i-vectors are obtained by means of acoustic feature extraction, voice activity detection (VAD), and cepstral mean and variance normalization (MVN) processes. We train the universal background (UBM), total variability matrix (TVM), and probabilistic linear discriminant analysis (PLDA) models using development data. Linear discriminant analysis (LDA) is applied to length-normalized i-vectors for the compensating session and utterance variations.

An i-vector [14, 15] is extracted by means of the following equation:

$$M = m + Tw, \tag{1}$$

where M is the speaker and channel dependent GMM supervector; m is the speaker and channel independent supervector (UBM supervector); T is a rectangular matrix of low rank, which is called total variability matrix (TVM); and w is a random vector having a standard normal distribution $N(0, I)$, which is referred to as an i-vector. After i-vectors are extracted, length normalization and PLDA are applied [1, 16].

2.2. DAE-baseline (baseline2)

The DAE-baseline approach, in this research, is conceptually equivalent with [12]. But there are some differences in details. Mainly, we use deeper structure and do not use enrollment set on training of network. Another trivial details will be denoted in the next section.

In this approach, given a d-dimensional input feature vector w extracted from a short test utterance, a DNN compensated feature vector w_{dae} is defined by the following equation:

$$w_{dae} = f_\theta(w), \tag{2}$$

where f_θ denotes the DNN feed-forward procedure with parameter θ. We train the DNN with five hidden layers containing 2048 rectified linear units (ReLUs) [17] and linear output units, as shown in Fig. 2. To train this DNN, every utterance in a DNN training set is truncated to 10 s because the duration of each test utterance used in this research is 10 s. Let us represent the long- and short-utterance features as $w^{l,tr}$ and $w^{s,tr}$ respectively. The DNN is then trained to minimize the mean squared error (MSE) between each original long-utterance feature vector $w^{l,tr}$ and the DNN compensated vector $f_\theta(w^{s,tr})$ of the corresponding short-utterance feature vector $w^{s,tr}$. Fig. 3 is composed of two-dimensional examples illustrating conceptual differences between DAE-baseline and proposed methods. Each DNN output activation is directly represented by a point on each coordinate axis in the i-vector space (Fig. 3a). However, this does not work well because the given information is minimal and the search space is too large.

2.3. Applying residual-learning (proposed1)

Kim et al. introduced the residual-learning method in [13] to solve vanishing/exploding gradients problem on image super-resolution domain. According to their research, training the residual image between an input low-resolution image and target high-resolution image as an output of network is faster than the case of training the target image directly. Although their research area is different from ours, the objectives of optimization and network structures are very similar. So, we expect residual-learning technique is a helpful way to train DNN compensator.

We calculate every residual feature vector $w^{l,tr}$ - $w^{s,tr}$ in the DNN training set. The DNN is then trained by these residual feature vectors, which are used as target labels $o_{residual} = w^{l,tr} - w^{s,tr}$. The DNN compensated feature vector $w_{residual}$ is defined by the following equation:

$$w_{residual} = w + f_\theta(w). \quad (3)$$

In this case, we assume that the adequate feature vector can be represented with this additional information by the translation of a given short-utterance feature vector, and we expect the DNN can be trained more easily (see Fig. 3b).

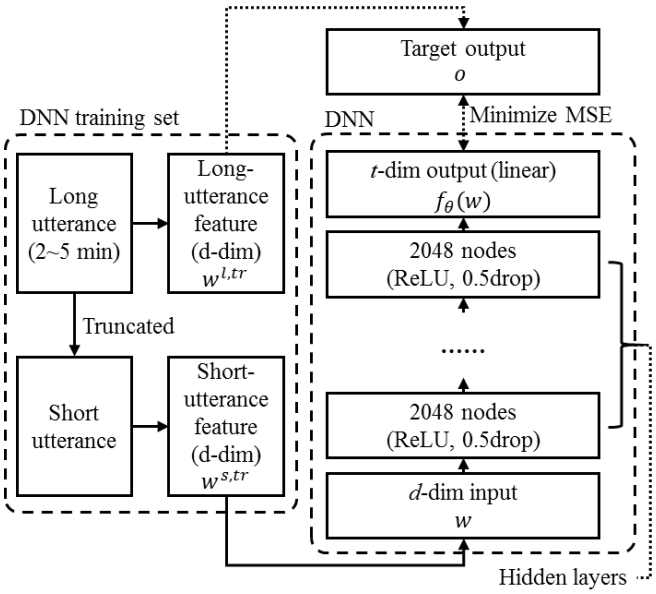

Fig. 2. DNN training diagram for short utterance compensation

2.4. Reducing the dimensionality of search space (proposed2)

Finally, we apply principal components analysis (PCA) to reduce the dimensionality of the search space during the DNN training step. The target label o_{pca} and DNN compensated feature vector w_{pca} are defined by the following equations:

$$o_{pca} = (C^{-1})^T [w^{l,tr} - w^{s,tr}], \quad (4)$$
$$w_{pca} = w + C^T f_\theta(w), \quad (5)$$

where C is a (t x d) matrix containing d-dimensional t principal components (PCs) estimated from the DNN training set. The residual vectors between every long-utterance feature vector and its corresponding short-utterance feature vector are calculated from the DNN training set, as shown in Fig. 3c. Eigenvectors are estimated from a covariance matrix of these residual vectors, as illustrated in Fig. 3d, and the t eigenvectors that correspond to the first t largest eigenvalues are selected as the PCs to reduce dimensionality. In this case, the translation after DNN feed-forwarding operates along each direction of the selected PCs, as shown in Fig. 3e.

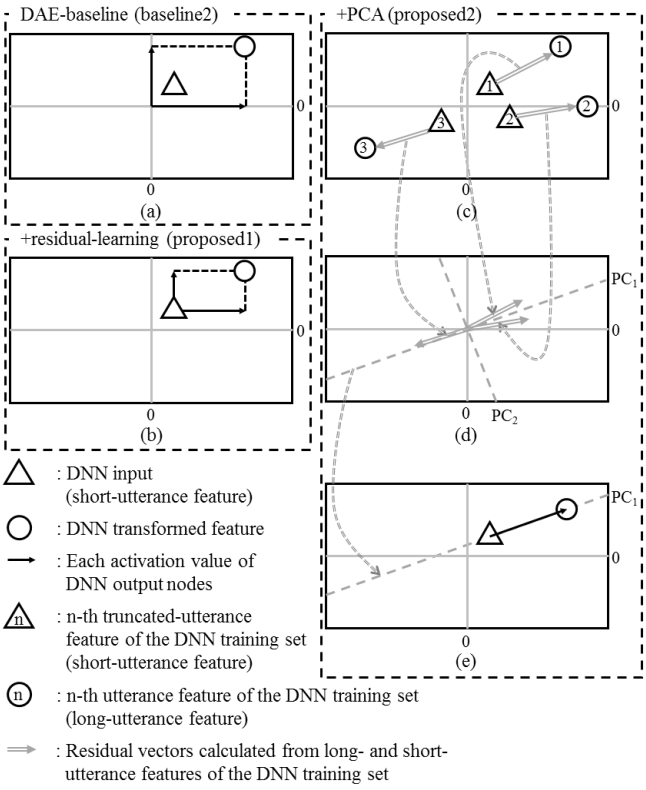

Fig. 3. Two-dimensional examples of the DAE-baseline and proposed methods
(a) DAE-baseline used in Eq. (2)
(b) Translation with additional vector used in Eq. (3)
(c) Calculation of residual vectors for PCs
(d) Estimating PCs used in Eq. (4)-(5)
(e) Translation operation used in Eq. (5)

3. EXPERIMENT SETUP

We used the NIST 2008 SRE corpus [18] to evaluate the proposed method. Specifically, det6 of a short2-10sec condition (male speakers only) were used. The short2-10sec condition was designed to evaluate speaker verification performance with a long-training utterance (approximately 2.5 min) and short-test utterance (10 s). For a development set, NIST 2004 SRE test, NIST 2005 SRE, NIST 2006 SRE training, Fisher English training and Switchboard cellular corpora were used. We used only male speaker utterances in these corpora as development data. DNN training utterances are randomly selected from whole development set with portion of 0.9. Remainders used as DNN validation set.

As acoustic features, 20 mel-frequency cepstral coefficients and their first and second derivatives were extracted. The energy-based VAD and MVN were applied and a gender dependent UBM (male only) with 2048 Gaussian components was trained. The dimensionality of total variability subspace was set to 400 and every i-vector is normalized by length normalization to have unit length.

Final dimensionality of i-vectors reduced to 150 by linear discriminant analysis. PLDA modeling and scoring were applied. We used the Kaldi [19] toolkit for these steps. For s-normalization, we randomly selected 200 utterances from the NIST 2004 SRE corpus for use as reference data.

We used Keras [20] with Theano backend engine to train the baseline and proposed DNNs. Each DNN is constructed using five hidden layers having 2048 nodes. ReLUs and linear units were used as activation functions of hidden and output layers, respectively. The loss function is the mean squared error. The learning rate decayed by a decay factor of 0.01 from an initial value of 10.0. The size of the mini-batch is 1000 and momentum is applied with a factor of 0.9. The batch normalization and dropout [21] with a rate of 0.5 for every hidden layer were applied. After DNN training, we chose the weights and biases that have minimum validation errors as final parameter θ of the DNN. When applied PCA, we set the number of PCs to in the range from 150 to 75 with an interval of 25, and the weights of five hidden layers on previous network are used as the initial weights on next network training repeatedly. For example, the final parameters of the network with 150PCs are used as the initial weights of the network with 125PCs.

We also evaluated performances in a situation when both speaker training and testing utterances are limited to 10 s. For this evaluation, each speaker training utterance in short2-10sec condition is randomly truncated to be 10 s long. In this case, unlike when only testing utterances are short, both training and testing feature vectors are compensated by DNN respectively.

4. RESULTS

Table 1 shows the minimum detection cost function (minDCF) of the experimental results. 'I-vector' denotes results of the canonical i-vector/PLDA system. '+residual-learning' and '+pca' mean the proposed methods using residual-learning and PCA respectively.

Table 1. Experiment results on NIST08 SRE short2-10sec (det6, male, minDCF)

DNN training length	method	speaker training length	
		full	truncated (10 s)
10 s	I-vector	0.0396	0.0600
	DAE-baseline	0.0616	0.0865
	+residual-learning	0.0394	0.0600
	+pca (with 150PCs)	0.0384	0.0587
	+pca (with 125PCs)	**0.0375**	**0.0574**
	+pca (with 100PCs)	**0.0375**	0.0578
	+pca (with 75PCs)	0.0377	0.0581
15 s	+pca (with 125PCs)	0.0386	0.0592
20 s		0.0388	0.0599
25 s		0.0393	0.0589

The results show that DAE-baseline performs more poorly than the canonical i-vector system. The performance of residual-learning is comparable to that of the canonical i-vector system but not sufficient, whereas PCA applied method with 125PCs shows slightly better minDCF than does the i-vector baseline system. Moreover, the proposed method showed improvement even when both speaker training and testing utterances are short. Unfortunately, equal error rates on proposed systems are comparable with canonical i-vector system but did not show significant improvement. We used various training length (15s, 20s, 25s) other than 10s to see how different truncated lengths affect the results, but the results show no advantage of using longer training utterances.

5. CONCLUSION

We proposed and demonstrated DNN structures for speaker verification to compensate for short test utterances. The DAE-baseline approach which is equivalent with [12] did not show improvement as a single system. But applying residual-learning raised the performance up to the same level of canonical i-vector system. With dimensionality reduction of search space using PCA, the proposed method gave reduced minDCF (0.0375) of 5.3% relative to the i-vector/PLDA method (0.0396).

As a future work, we are going to investigate how dimensionality reduction on the proposed method brings performance improvement. We also plan to replace the PCA to a nonlinear reduction technique, such as a deep auto-encoder [22].

6. RELATION TO PRIOR WORK

We proposed a short utterance compensating method using deep neural networks. The work by Yamamoto et al. [12] includes enrollment set in DNN training, which we do not think appropriate for practical use because we cannot train the network for every enrollment speaker. In this research, we demonstrated the deeper version of their method and we improved the performance of a system without using enrollment set on DNN training. In this process, we found that residual-learning and dimensionality reduction techniques help to improve minDCF even when both training and testing utterances are short.

7. ACKNOWLEDGEMENTS

This work was supported by the IT R&D program of MOTIE/KEIT. [10041610, The development of the recognition technology for user identity, behavior and location that has a performance approaching recognition rates of 99% on 30 people by using perception sensor network in the real environment]

8. REFERENCES

[1] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," Odyssey, June 2010.

[2] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances," Interspeech, Florence, Italy, pp. 2341–2344, August 2011.

[3] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J. F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," Interspeech, 2012.

[4] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 7649-7653, 2013.

[5] S. Cumani, O. Plchot, and P. Laface. "Probabilistic linear discriminant analysis of i-vector posterior distributions," Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 7644-7648, 2013.

[6] 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 Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2425–2438, 2013.

[7] T. Hasan, R. Saeidi, J. H. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 7663-7667, 2013.

[8] B. Vesnicer, J. Zganec-Gros, S. Dobrisek, and V. Struc, "Incorporating Duration Information into I-Vector-Based Speaker-Recognition Systems," Odyssey, pp. 241-248, 2014.

[9] A. Kanagasundaram, D. Dean, S. Sridharan, J. González-Domínguez, J. González-Rodríguez, and D. Ramos "Improving short utterance i-vector speaker verification using utterance variance modelling and compensation techniques," *Speech Communication*, vol. 59, pp. 69-82, 2014.

[10] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. González-Domínguez, "Deep neural networks for small footprint text-dependent speaker verification," Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, pp. 4052–4056, May 2014.

[11] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, pp. 1695–1699, May 2014.

[12] H. Yamamoto, and T. Koshinaka, "Denoising Autoencoder-Based Speaker Feature Restoration for Utterances of Short Duration," Interspeech, 2015.

[13] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[14] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," Odyssey, June 2010.

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

[16] D. Garcia-Romero, and C. Y. Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," Interspeech, pp. 249-252, August 2011.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," Int. Conf. on Machine Learning (ICML), Haifa, Israel, pp. 807-814, June 2010.

[18] National Institute of Standards and Technology (NIST), "The NIST year 2008 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig//tests/sre/2008/sre08_evalplan_release4.pdf, accessed October 2014.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, "The Kaldi speech recognition toolkit," IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Hawaii, US, December 2011.

[20] B. Van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio, "Blocks and fuel: Frameworks for deep learning," arXiv preprint arXiv:1506.00619, 2015.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

[22] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504-507. 2006.