ADVANCED B-VECTOR SYSTEM BASED DEEP NEURAL NETWORK AS CLASSIFIER FOR SPEAKER VERIFICATION

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

School of Computer Science, University of Seoul, Korea

ABSTRACT

Few studies on speaker verification have directly used a deep neural network (DNN) as a classifier. It is difficult to directly apply a DNN as a discriminative model to speaker-verification tasks because the training data for each speaker are very limited. Therefore, a b-vector has been proposed to solve the problem. However, the DNN with the b-vectors showed lower performance than the conventional i-vector probabilistic linear-discriminant analysis (PLDA) system.
In this paper, we propose an improved version of the b-vector DNN system, which incorporates the background speakers' information into the DNN. In this study, each input feature is paired with a representative background speaker's feature vectors, and a b-vector is extracted from each pair; thus, feeding background information into the DNN. We confirmed that the performance improvements of the proposed system compensate for the shortcomings of conventional b-vectors in experiments carried out using the National Institute of Standards and Technology 2008 Speaker-Recognition Evaluation tests.

Index Terms— speaker verification, b-vector, DNN

1. INTRODUCTION

In recent years, deep neural networks (DNNs) have been applied to automatic speech-recognition (ASR) studies, and considerably improved the performance [2, 3]. However, performance improvement was difficult when using a DNN as a classifier in speaker-verification tasks. Two problems arising in the application process may cause this difficulty.

The first problem is a lack of data for target speakers. It is a well-known fact that supervised DNN training requires a large amount of labeled data. However, in reality, collecting sufficient data from a target speaker is very difficult. The second is that we need as many models as there are speakers. Models for each target speaker are required for speaker verification, as in a Gaussian supervector support-vector machine (GSV-SVM) or a Gaussian-mixture model universal background model (GMM-UBM) [4, 5]. However, training DNNs for every possible target speaker is too costly.

Because of these problems, few speaker-verification studies have directly applied DNNs as classifiers. For example, some studies used DNNs to calculate the Baum-Welch statistics needed for i-vector extraction [6, 7] and another study used a DNN to transform the i-vectors [8]. On the other hand, the b-vector system [1] directly applies the DNN as a classifier to speaker verification; however, the b-vector system had lower performance than expected for speaker verification. Through various experiments using the b-vectors, we found that the b-vector system has higher performance than the i-vector cosine similarity system, but lower performance than the i-vector probabilistic linear-discriminant analysis (PLDA) system.
Therefore, in this paper, we examine the problems that cause the performance degradation of the b-vectors and attempt to find an appropriate solution for the DNNs. Then, we investigate the applicability of DNNs as classifiers in speaker verification. In Sections 2 and 3, we describe the i-vector PLDA system and the conventional b-vector system, respectively. Section 4 describes the proposed system. Finally, Sections 5, 6, and 7 describe the experiments, conclusions, and future work, respectively.

2. I-VECTOR SYSTEM

An identity vector (i-vector) is a state-of-the-art front-end technique in speaker recognition; this effective factor-analysis method can extract an identity vector from an utterance [9]. The total variability matrix used for extracting i-vectors contains the class-dependent and independent subspaces in the original space of the utterance supervectors. Generally, Gaussian-mixture model (GMM) supervectors are used as the original utterance supervectors.

We can represent high-dimensional input vectors using low-dimensional i-vectors, as in equation (1).

$$\mathbf{M} = \mathbf{m} + \mathrm{T}\mathbf{w}, \tag{1}$$

where $\mathbf{M}$ is an input utterance vector, $\mathbf{m}$ is the global mean vector (generally a universal-background model (UBM) supervector), T is the total variability matrix containing utterance dependent subspaces, and $\mathbf{w}$ is the i-vector. Probabilistic linear-discriminant analysis (PLDA) is a factor-analysis technique first proposed in [10], which was shown to perform well for modeling and scoring i-vectors [11]. Class-dependent and independent factors contained in i-vectors can be analyzed using the PLDA model. The PLDA

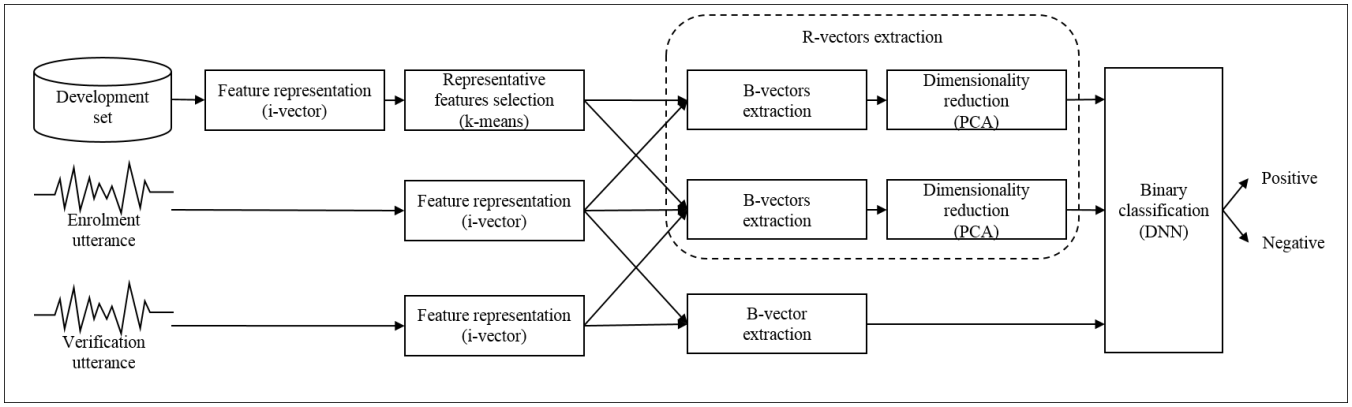

Figure 1. Flow of the complete rb-vector system containing the extraction of r-vectors

model represents the input i-vectors using the sum of several terms, as in equation (2).

$$\mathbf{w} = \boldsymbol{\mu} + \Phi \boldsymbol{h_i} + \Psi \boldsymbol{s_{ij}} + \boldsymbol{\epsilon}, \quad (2)$$

where $\mathbf{w}$ is an input i-vector, $\boldsymbol{\mu}$ is the overall mean vector of the training data set, Φ is the matrix containing the subspaces of between-class factors, $\boldsymbol{h_i}$ is the position of class $\boldsymbol{i}$ in the between-class subspace, Ψ is the matrix containing subspaces of the within-class factor, $\boldsymbol{s_{ij}}$ is the position of class $\boldsymbol{i}$ and utterance $\boldsymbol{j}$ in the within-class subspace, and $\boldsymbol{\epsilon}$ is a residual-noise term.

3. B-VECTOR SYSTEM

The b-vector system [1] is proposed for solving speaker verification tasks as binary-classification problems. In this system, a pair of utterances is represented by a b-vector that describes the relationship between them. Each utterance is generally represented as an i-vector, and a b-vector is generated using a concatenation of the results of binary operations, such as the addition or subtraction of two i-vectors.

For example, a b-vector can be calculated from i-vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ by concatenating the following vectors.

$$\boldsymbol{b}_a = \boldsymbol{w}_1 \oplus \boldsymbol{w}_2, \quad (3)$$
$$\boldsymbol{b}_m = \boldsymbol{w}_1 \otimes \boldsymbol{w}_2, \quad (4)$$
$$\boldsymbol{b}_s = |\boldsymbol{w}_1 \ominus \boldsymbol{w}_2|, \quad (5)$$
$$\boldsymbol{b}_r = |\log|\boldsymbol{w}_1| \ominus \log|\boldsymbol{w}_2||, \quad (6)$$

where $\oplus$, $\otimes$ and $\ominus$ are the element-wise addition, multiplication and subtraction operations, respectively.

The b-vectors do not contain intuitive information that can be utilized for speaker recognition. However, a binary classifier, such as an SVM or DNN, can classify them, whether the b-vector is made from two vectors from the same speaker or from different speakers. The entire b-vector system is summarized as feature extraction regarding the relationship between two utterances, and binary classification for speaker verification. In [1], a b-vector system that uses a DNN as a classifier was proposed. However, the performance of the b-vector system was lower than expected. In our empirical experiments, the b-vector system showed higher performance than the i-vector cosine-similarity system (i-vector CSS), whereas the system had lower performance than the i-vector PLDA system. In the next section, we introduce a system expected to improve the performance of the b-vector system.

4. PROPOSED SYSTEM

We considered that the lack of background information in the b-vectors might cause the performance degradation of the conventional system. In the conventional system, a b-vector is extracted using only two i-vectors. Therefore, the b-vector is missing information about the complete i-vector distribution in the total variability space, and the background speakers. For example, the GMM-UBM system uses two models, which represent a target speaker and a background speaker. By calculating the likelihood ratio from the two models, the GMM-UBM system facilitates the consideration of not only the information about the target speaker but also about the background speakers. In addition, the i-vector PLDA system considers information about the entire i-vector space by extracting speaker factors from a development i-vector set.

We devised a method to add information about the background speakers into the b-vector system. The devised method uses vectors that **refer** to information about the background speakers (called r-vectors for convenience), with the b-vector as an input to the classifier.

The r-vectors can be extracted using the following steps. The first step is to extract features (typically i-vectors) from the development utterances, an enrollment, and a verification utterance (user data). The second step is to select representative feature vectors from the development feature set. Clustering techniques, such as k-means, can be used to select representative feature vectors. The third step is to extract two r-vectors using two user feature vectors (from the enrollment and a verification utterance) and a representative vector from the background (development) set.

Each of the two user-feature vectors is paired with a representative background vector, and a b-vector is extracted from each pair. Principal component analysis (PCA) is applied to the b-vectors extracted from the user and background vectors to reduce the dimensionality. Finally, an r-vector is made by concatenating the dimensionality-reduced b-vectors from the two pairs. It is possible that the lack of background information in the b-vectors mentioned above is mitigated by using the r-vectors, because the r-vectors represent the relation between the background speakers' vectors and the new user's vectors. The proposed system uses a DNN as a classifier, and the concatenation of the b-vectors and r-vectors as an input to the DNN. The complete proposed system is depicted in Figure 1.

In this section, we introduced a simple process for extracting r-vectors. Any r-vectors can be used in the proposed rb-vector system, if they contain information about the relationship between a user and background utterances.

5. EXPERIMENTS

5.1. Database

All the experiments in this study were carried out using the male portion of the core condition (short2-short3) in the National Institute of Standards and Technology (NIST) 2008 Speaker-Recognition Evaluation (SRE) tests [12]. Table 1 shows all corpora that were used to estimate the UBM, total variability matrix (TVM), linear discriminant analysis (LDA), and PLDA models.

Table 1. Databases used as the development set

	UBM	TVM	LDA	PLDA
Fisher English Training	O	O		
NIST SRE 2004	O	O	O	O
NIST SRE 2005	O	O	O	O
Switchboard Cellular		O	O	
Switchboard-2		O	O	

5.2. i-vector PLDA system

60-dimensional feature vectors (19 Mel-frequency cepstral coefficients (MFCC) + energy + Δ + ΔΔ) were extracted using a 25-ms window with 10-ms shifts, and then mean and variance normalization (MVN) was applied.

A gender-dependent UBM, containing 2048 Gaussian components, and a TVM with dimensionality 400 were trained, both with 10 iterations. LDA was applied to reduce the i-vector to 150 dimensions. Length normalizations were applied to the i-vector before and after applying the LDA. For the baseline system, a PLDA model was estimated using the dimensionality-reduced i-vectors. An open-source speech and speaker-recognition toolkit, Kaldi, was used for the baseline system [13].

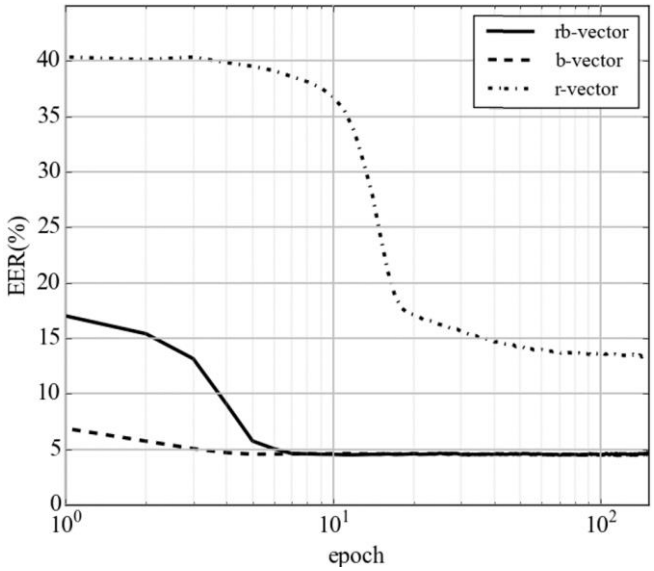

Figure 2. EERs of the DNN systems for each epoch

5.3. b-vector extraction

i-vectors without length normalization were used to extract the b-vectors. The b-vectors were a concatenation of the results of the element-wise addition, multiplication, subtraction, and ratio operations of two input i-vectors. Therefore, a 600-dimensional b-vector was extracted for each two (150-dimensional) i-vectors.

5.4. r-vector extraction

The r-vectors were extracted as explained in the previous section. First, 30 representative background i-vectors were selected from the development set using the k-means algorithm. The input i-vectors were paired with the representative-background i-vectors, and the b-vectors were extracted from each pair. PCA was applied to reduce the dimensionality of these b-vectors. The dimensionality-reduced 10-dimensional b-vectors were concatenated to comprise the r-vector. Finally, 300-dimensional r-vectors were extracted from each input i-vector.

5.5. DNN training

The DNN has five hidden layers. Each hidden layer includes 2048 fully connected neurons activated by a hyperbolic-tangent function. The DNN was trained using 1200-dimensional rb-vectors. Approximately 200,000 rb-vectors were extracted from the development set. For each training epoch, the DNN was trained with a learning rate of 0.01, and a "drop-out" technique was applied. All DNNs were trained in the Theano environment [14, 15]

Table 2. Performance in EER and minDCF of the PLDA and DNN systems (NIST08, short2-short3, male set)

		Avg	DET1	DET2	DET3	DET4	DET5	DET6	DET7	DET8
EER(%)	i-vector PLDA	4.30	**5.79**	0.81	**5.73**	7.01	4.98	**4.67**	3.14	2.30
	b-vector DNN	4.42	6.38	**0.40**	6.56	6.42	5.46	4.90	**2.94**	2.30
	Rb-vector DNN	**3.98**	5.85	**0.40**	5.98	**4.63**	**4.17**	5.46	3.23	**2.21**
minDCF	i-vector PLDA	**.0197**	**.0266**	**.0008**	**.0261**	**.0287**	.0200	**.0249**	.0172	.0138
	b-vector DNN	.0206	.0282	.0028	.0287	.0302	.0238	.0251	**.0147**	**.0112**
	Rb-vector DNN	**.0197**	.0268	**.0008**	.0273	.0292	**.0188**	.0276	.0160	**.0112**

For the DNN to learn the r-vectors and b-vectors equally well, we applied a pre-training technique that is different from conventional pre-training. First, the DNN was trained using only r-vectors, by setting the b-vector values to 0. After the 50th epoch, the b-vectors were used with the r-vectors to train the DNN, with a learning rate of 0.1.

5.6. Results

Figure 2 shows the experimental evaluation results of the conventional b-vector system and the proposed r-vector system. The graph shows the average equal-error rate (EER) of all DETs in the SRE 2008 core condition of each system. In the graph, the b-vector system has an EER lower than 5%, and this performance converges before the 10th epoch. The r-vector system shows a relatively high EER because it contains only supplemental information; the performance converges after the 100th epoch.

The rb-vector system, which uses b-vectors and r-vectors together, shows performance similar to the b-vector system. This result could indicate that the information contained in the r-vector was ignored because the performance converged before sufficiently training the r-vector (until about the 100th epoch). More precisely, the DNN in the rb-vector system is trained to ignore the r-vectors, of which the discernment is low before the 10th epoch. To solve the problem, we introduced the pre-training concept so that the DNN can better combine the r-vector and b-vector information.

Table 2 shows the experimental evaluation results of the i-vector PLDA system, the b-vector system, and the rb-vector system. The average performance of the proposed rb-vector system is higher than the conventional b-vector system and the i-vector PLDA, based on the EER. Based on the minimum decision-cost function (minDCF), the average performance of the rb-vector system is similar to the i-vector PLDA system, and higher than the b-vector system.

6. CONCLUSIONS

In this paper, we addressed the problems of the conventional b-vector system and proposed the rb-vector system as a solution. We evaluated the proposed system using the NIST SRE 2008 core condition. The results of the experimental evaluation showed that the relative error reduction of the proposed system over that of the i-vector PLDA system was 7.44%, based on the average EER.

The contributions of this paper related to prior work are as follows. A direction for improving the conventional b-vector system was proposed. This direction included using vectors containing information about the background speakers, along with the b-vectors. In addition, the possibility of DNNs as classifiers for speaker verification was confirmed. Therefore, many DNN studies can more directly be applied to speaker-verification studies.

7. FUTURE WORKS

Research on the proposed system is still in its early stages. Therefore, additional work will be needed to establish the rb-vector system as a stable speaker-verification method.

Firstly, the b-vectors must be optimized. Presently, the utterance pairs are arbitrarily comprised. However, it will be possible to apply support vectors to the process of extracting r-vectors. Through experimentation, we found that training the r-vectors is a relatively long process, compared to the b-vectors. To solve this problem, we will use the tandem features of the r-vectors, as in [8]. Finally, the DNN in the rb-vector system can be optimized using various techniques, such as pre-training or max-out nodes.

8. ACKNOWLEDGMENTS

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]

9. REFERENCES

[1] H. S. Lee, Y. Tso, Y. F. Chang, H. M. Wang and S. K. Jeng, "Speaker verification using kernel-based binary classifiers with binary operation derived features," *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1660-1664, 2014.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing*, pp. 30-42, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine*, pp. 82-97, 2012.

[4] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett*, vol. 13, no. 5, pp. 308-311, 2006.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19-41, 2000.

[6] Y. Lei, N. Scheffer, L. Ferrer and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1695-1699, 2014.

[7] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," *Odyssey: Speaker Lang. Recognit. Workshop*, 2014.

[8] F. Richardson, D. Reynolds, and N. Dehak, "A Unified Deep Neural Network for Speaker and Language Recognition," arXiv preprint arXiv:1504.00923.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing*, pp. 788-798, 2011.

[10] S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision–ECCV,* pp. 531-542, 2006.

[11] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," *Odyssey: Speaker Lang. Recognit. Workshop*, 2010.

[12] The NIST Year 2008 Speaker Recognition Evaluation Plan, http://www.itl.nist.gov/iad/mig/tests/sre/2008.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[14] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. "Theano: new features and speed improvements," *NIPS 2012 deep learning workshop*.

[15] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio, "Theano: A CPU and GPU Math Expression Compiler," *Proceedings of the Python for Scientific Computing Conference (SciPy),* June 30 - July 3, 2010.