Covariance Based Deep Feature for Text-Dependent Speaker Verification

Shuai Wang, Heinrich Dinkel, Yanmin Qian, and Kai Yu(✉)

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, SpeechLab, Department of Computer Science and Engineering, Brain Science and Technology Research Center, Shanghai Jiao Tong University, Shanghai, China
kai.yu@sjtu.edu.cn

Abstract. *d*-vector approach achieved impressive results in speaker verification. Representation is obtained at utterance level by calculating the mean of the frame level outputs of a hidden layer of the DNN. Although mean based speaker identity representation has achieved good performance, it ignores the variability of frames across the whole utterance, which consequently leads to information loss. This is particularly serious for *text-dependent* speaker verification, where within-utterance feature variability better reflects text variability than the mean. To address this issue, a new covariance based speaker representation is proposed in this paper. Here, covariance of the frame level outputs is calculated and incorporated into the speaker identity representation. The proposed approach is investigated within a joint multi-task learning framework for *text-dependent* speaker verification. Experiments on RSR2015 and RedDots showed that, covariance based deep feature can significantly improve the performance compared to the traditional mean based deep features.

Keywords: Deep features · Text-dependent speaker verification
Speaker recognition · *d*-vector · *j*-vector · Covariance discrimination

1 Introduction

Speaker verification (SV) is the task of verifying the identity of a certain person by means of his voice. Considering the restriction on the spoken text, speaker verification can be classified into two categories, text-dependent and text-independent. Moreover the duration of the model registration is detrimental to the user experience, although a tough challenge, short enrollment utterances are preferred over long ones. In short utterance environments, the additional

This work has been supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002102 and the China NSFC projects (No. U1736202 and No. 61603252). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

© Springer Nature Switzerland AG 2018
Y. Peng et al. (Eds.): IScIDE 2018, LNCS 11266, pp. 231–242, 2018.
https://doi.org/10.1007/978-3-030-02698-1_20

xpjiang@mail.ccnu.edu.cn

information provided by the text gives the text-dependent SV an edge over the text-independent SV, while simultaneously being less convenient for the user. In recent research, deep neural network (DNN) was applied to speaker verification [1–5] and was critically acclaimed.

After a fast deep neural network training algorithm was published in [6,7], progressively more researchers turned their focus to deep learning. Motivated by the powerful non-linear learning ability, researchers started using DNN as a feature extractor, e.g. as a bottleneck feature in speech recognition [8,9] and language identification [10]. Moreover, bottleneck features can be used together with traditional features in a tandem manner, which can be seen in [11,12].

Other than using DNN as a feature extractor, some researchers extract model representations directly out of a DNN. In google's work [13], *d*-vector is proposed to produce speaker model and test utterance representations, subsequently cosine distance is used to calculate a score between model and test utterance *d*-vectors. Motivated by google's work, *j*-vector based on a multi-task learning framework was proposed in [14] and achieved significant improvements. In this paper, we introduce and survey the usage of covariance based representations into the frameworks of both *d*-vector and *j*-vector. Different scoring methods are then applied to the covariance based representations. Experiments manifest that the proposed covariance based representations surpass mean based approaches on RSR2015 [15] and RedDots [16] data-sets respectively.

The remainder of the paper is organized as follows, Sect. 2 briefly introduces previous works. Section 3 introduces our own *v*-vector. Section 4 showcases our experiment design and result analysis. Finally Sect. 5 concludes this paper.

2 Speaker Representation Using DNN

In [13], Google proposed to use a neural network to extract frame level vectors from commonly used cepstral features (PLP, FBANK, MFCC). The outputs of the last hidden layer are derived and averaged to get utterance-level representations (*d*-vector).

Based on Google's work, a multi-task framework which learns both speaker identity and text information is proposed in [14]. In this framework, the output nodes consist of both speakers and texts, where two types of multi-task joint training can be considered: speaker + phrase, speaker + phone. The architecture is shown in Fig. 1 (speaker + phrase as an example).

3 Covariance Based Deep Feature

Although mean based speaker identity representation has achieved good performance, it ignores the variability of frames across the whole utterance, which consequently leads to information loss. This is particularly serious for text-dependent speaker verification, where within-utterance feature variability better reflects text variability than the mean. To address this issue, a new covariance based speaker representation is proposed in this paper.

xpjiang@mail.ccnu.edu.cn

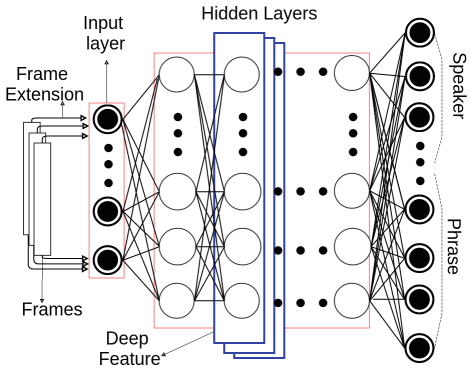

Fig. 1. *j*-vector approach

In [17] it was shown that covariance is a key factor to cluster narrow-band, wide-band and background (noise, music, silence) speech into the respective categories. Furthermore, in another related field of research, speaker anti-spoofing [18,19], it can be seen that covariance based discrimination for binary classification outperforms conventional mean based features. Moreover, covariance has been used in computer vision for human detection tasks to represent human descriptors as an important feature [20,21].

3.1 *v*-vector

In this paper we introduce covariance based vectors within the *d/j*-vector framework. After extracting the utterance-level representation $\mathbf{u}$ containing N frames with dimension D from speaker s, different types of vectors can be extracted by calculating the mean vector ($\boldsymbol{m}$) and covariance matrix ($\boldsymbol{\Sigma}$).

$$\boldsymbol{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{u}_i \tag{1}$$

$$\boldsymbol{\Sigma} = cov(\mathbf{u}) \tag{2}$$

$$\boldsymbol{v}_{\mathtt{f}} = \left[\boldsymbol{\Sigma}_{11}, \boldsymbol{\Sigma}_{12}, \ldots, \boldsymbol{\Sigma}_{1\mathtt{D}}, \boldsymbol{\Sigma}_{22}, \boldsymbol{\Sigma}_{23}, \ldots, \boldsymbol{\Sigma}_{\mathtt{DD}}\right]^\top \tag{3}$$

$$\boldsymbol{v}_{\mathtt{d}} = \left[\boldsymbol{\Sigma}_{11}, \boldsymbol{\Sigma}_{22}, \boldsymbol{\Sigma}_{33}, \ldots, \boldsymbol{\Sigma}_{\mathtt{DD}}\right]^\top \tag{4}$$

Since *covariance* matrices are symmetric, only the upper triangular part contains information, thus we obtain the $\boldsymbol{v}_{\mathtt{f}}$(full covariance vector) as a concatenation of the upper triangular part according to Eq. 3, the dimension of $\boldsymbol{v}_{\mathtt{f}}$ is $\frac{\mathtt{D(D+1)}}{2}$. $\boldsymbol{v}_{\mathtt{d}}$(diagonal covariance vector) is defined as the diagonal of $\boldsymbol{\Sigma}$ according to Eq. 4 and has a dimension of D. In terms of the methods of extracting vectors, we annotate the mean vector $\boldsymbol{m}$ as *m*-vector, $\boldsymbol{v}_{\mathtt{f}}$ and $\boldsymbol{v}_{\mathtt{d}}$ as *v*-vector. Both vectors can be extracted in *d*-vector or *j*-vector framework.

It should be stressed that we denote *d/j* vector in terms of the network types (multi-task or not), while *m/v*-vector in terms of the method of generating utterance-level representations from frame-level.

xpjiang@mail.ccnu.edu.cn

3.2 Scoring Methods

Having obtained the speaker model and test-utterance representations from the respective neural network, scoring leads to obtain a classification metric to either accept or reject a certain testcase.

CDS (Cosine Distance Scoring) is a simple but effective scoring method successfully used in the *i*-vector framework [22–24]. Cosine distance scoring is a dot product between test vector, $\mathbf{w}$ and speaker model mean, μ_m

$$score_w^m = \frac{\mathbf{w}^T \mu_m}{\|\mathbf{w}\| \, \|\mu_m\|} \tag{5}$$

GC (Gaussian Classifier) is a classical classifier following a generative manner, speaker identity vector representations(eg. *i*-vector) are modeled by a Gaussian distribution, where full covariance matrix is shared across all speakers. For an test vector representation $\mathbf{w}$, we evaluate the log likelihood score against the target m,

$$ln(\mathbf{w}|m) = \mathbf{w}^T \mathbf{\Sigma}^{-1} \mu_m - \frac{1}{2}(\mathbf{w}^T \mathbf{\Sigma}^{-1} \mathbf{w} + \mu_m^T \mathbf{\Sigma}^{-1} \mu_m) + const \tag{6}$$

where μ_m is the mean vector for speaker m, $\mathbf{\Sigma}$ is the common covariance matrix and *const* is a speaker- and test vector-independent constant. Furthermore, the speaker-independent part and constant can be neglected and we get

$$ln(\mathbf{w}|m) = \mathbf{w}^T \mathbf{\Sigma}^{-1} \mu_m - \frac{1}{2} \mu_m^T \mathbf{\Sigma}^{-1} \mu_m \tag{7}$$

PLDA (Probabilistic Linear Discriminant Analysis) [25] uses a generative approach to score utterances [26,27]. A PLDA estimator is trained on the background data, which will be used to transform and score the test utterances against the target models. situation [14].

4 Experiments

4.1 Experiments on RSR2015

Experimental Setup and Baseline. RSR2015 Part 1 consists of overall 300 speakers, whereas 143 are females and 157 are males. The whole set is divided into background (*bkg*), development (*dev*) and evaluation (*eval*) subsets (Table 1).

Bkg and *dev* data is merged to obtain an extended training data set consisting of 194 speakers and 52244 utterances. The evaluation part is split into enrolement part and test part. The test set encompasses 1568008 tests, which is divided into 19052 true speaker tests and 1548956 impostor tests.

xpjiang@mail.ccnu.edu.cn

Table 1. Subset definition of RSR2015 part 1

Subset	# Female speaker	# Male speaker	# Total
bkg	47	50	97
dev	47	50	97
eval	49	57	106

The neural network was trained using 39-dimensional PLP features, which were extended by a frame window of 5 at left and right. Network initialization was done using a 6 hidden-layer, 1024 neuron RBM network. Sigmoid was used as the activation function.

The complete DNN comprises of 8 layers, 1 input layer with 429 (11×39) neurons, 6 hidden layers with 1024 neurons and a single output layer having 194 output neurons (one for each speaker) using the *d*-vector approach and 224 (194 speakers + 30 phrases) using the *j*-vector approach.

During the enrolement and evaluation phase, we feed forward one sample at a time into the network to acquire a 1024-dimensional representation, while the bottleneck features were extracted with a 45 dimensional representation. We used the 2nd, 4th and 7th layer (corresponding to the 1st, 3rd, 6th hidden layer) outputs as valid representations in our experiments.

Finally while scoring the output vectors, we first normalize the mean and variances against the before applying cosine distance. PLDA is trained for 20 iterations with a within-covariance smoothing factor[1] of 0.5. It's notable that PLDA here doesn't reduce the vector dimension. Moreover we apply z-norm [28] on the PLDA scores to further enhance the performance.

The GMM-UBM baseline (Table 2) follows a gender-independent approach [29], where 39-dimensional PLP features were used as input. A DNN based VAD was applied to all the features to filter silent segments out. Finally z-norm [28] was utilized on the scores, using 300 impostor utterances.

A GMM based *i*-vector baseline (Table 2) is also provided, in which *bkg* data is used to train the $\boldsymbol{T}$ matrix and PLDA classifier. 400-dim *i*-vectors are used.

Table 2. Baselines for RSR2015, in % EER

Method	EER
GMM-UBM	1.10
i-vector	1.39

[1] In order to get a good estimate of the within-class covariance, the product of this parameter and between-class covariance is adding to the within-class covariance.

xpjiang@mail.ccnu.edu.cn

The Comparison of Deep Features. Following google's work [1], we first extracted our features from the last hidden layer (7th layer) and indeed, the result outperforms the GMM-UBM baseline. We denote $\boldsymbol{v}_{\mathtt{d}}$ as the diagonal covariance vector, $\boldsymbol{m}$ as the common d-vector baseline and $\boldsymbol{m} \oplus \boldsymbol{v}_{\mathtt{d}}$ denotes the score fusion of $\boldsymbol{m}$ and $\boldsymbol{v}_{\mathtt{d}}$.

Table 3. 7th layer results, in % EER

Deep feature	d-vector			j-vector		
	GC	PLDA	CDS	GC	PLDA	CDS
$\boldsymbol{m}$	0.79	2.90	15.95	0.12	1.15	9.35
$\boldsymbol{v}_{\mathtt{d}}$	0.81	1.95	9.28	**0.07**	0.71	4.74
$\boldsymbol{m} \oplus \boldsymbol{v}_{\mathtt{d}}$	0.63	1.99	9.62	0.07	0.73	4.51

As we can see (Table 3), cosine distance is largely outperformed by GC and PLDA, hence we decided to exclude cosine distance out of the future experiments. Furthermore we see that the covariance based vector consistently surpasses the traditional mean based method.

Moreover, performance of the full covariance vector $\boldsymbol{v}_f$ inside the j-vector framework was investigated. To make the vector length comparable, bottleneck features of 45 dimensions were extracted from the last hidden layer to get 1035-dim ($\boldsymbol{v}_f$) according to Eq. 3 (it's also hard to use 1024-dim feature for $\boldsymbol{v}_f$). We denote D as the extracted feature dimension and v-dim as the dimension of the covariance vector. The comparison between the full and diagonal vectors can be found in Table 4. The full covariance approach ($\boldsymbol{v}_f$) performs poorly inside the j-vector framework, so further research was discontinued.

Table 4. Full v.s. diagonal covariance vector, 7th layer j-vector, in % EER

Covariance	D	v-dim	GC	PLDA
$\mathtt{Full}(\boldsymbol{v}_f)$	45	1035	2.62	3.90
$\mathtt{Diagonal}(\boldsymbol{v}_d)$	1024	1024	**0.07**	**0.71**

Layer Comparison. The covariance based deep feature is investigated with the different positions of the DNN, i.e. the deep features are extracted from the different hidden layers (the 2nd, 4th and 7th layer in this paper). As we can observe in Table 5, the best results are achieved in layer 4.

xpjiang@mail.ccnu.edu.cn

Table 5. Layer-wise comparison, in % EER

Deep feature	Layer	*d*-vector		*j*-vector	
		GC	PLDA	GC	PLDA
$\boldsymbol{m}$	2	0.20	0.19	0.15	1.07
	4	0.14	1.18	**0.08**	0.95
	7	0.79	2.90	0.12	1.15
$\boldsymbol{v}_{\mathtt{d}}$	2	0.10	0.85	0.07	0.64
	4	0.11	0.82	**0.05**	0.55
	7	0.81	1.95	0.07	0.71
$\boldsymbol{m} \oplus \boldsymbol{v}_{\mathtt{d}}$	2	0.13	0.75	0.09	0.59
	4	0.12	0.83	**0.05**	0.56
	7	0.63	1.99	0.06	0.72

Open-Set Condition Analysis. Despite GC being the best result throughout, this is partly due to the test only having a closed set of speakers. Thus, by removing 1/4 enrolement speakers and corresponding test-cases (which means there are 1/4 speakers present in test set are not present in the enrolement set), we simulate real-life conditions to accurately estimate GC's performance (Table 6). We can see that in open set cases, PLDA relatively looses 10% accuracy, while GC looses 540%, compared to closed set cases.

Table 6. 4th layer deep feature results after removing 1/4 enrolement speakers (gender balanced), in % EER

Deep feature	*d*-vector		*j*-vector	
	GC	PLDA	GC	PLDA
$\boldsymbol{m}$	0.61	1.31	0.47	1.16
$\boldsymbol{v}_{\mathtt{d}}$	0.54	0.95	**0.32**	0.72

Error Analysis. To further figure out where the performance gain comes from, an analysis on error types is given in Table 7. Here, we chose the best result, *j*-vector of the 4th layer to analyse the error pattern.

In text-dependent tasks there exist three kinds of impostors:

1. The enrolled speaker speaks a wrong utterance
2. An impostor speaks a correct utterance
3. An impostor speaks a wrong utterance

xpjiang@mail.ccnu.edu.cn

In real applications, error type 3 occurs the most and fortunately is the easiest one to detect. In fact in the test sets defined by RSR2015, this kind of error occupies nearly 90% of all test cases, we completely neglect these test cases because otherwise the EER will be extremely low.

Table 7. Err. distribution of false accepts (4th layer *j*-vector)

Error type	Deep feature	# Trials	# Err	Err. rate (%)
Speaker	$\boldsymbol{m}$	996448	13562	1.36
	$\boldsymbol{v}_{\mathrm{d}}$		8943	0.89
Text	$\boldsymbol{m}$	552508	1235	2.2
	$\boldsymbol{v}_{\mathrm{d}}$		123	0.22

From the Table 7, we can observe that the error rate (PLDA) is reduced by relatively 90% for text and 35% for speaker[2], respectively. We clarify the assumption that covariance does relate to text much more closely than mean. Since the improvement on speaker is not as significant, we can infer that this approach should also work on text-independent speaker verification tasks, but the improvement will not be as perceivable.

4.2 Experiments on RedDots Database

The RedDots project was initiated, with collaboration from multiple sites, as a follow-up to a special session during INTERSPEECH 2014 [16]. In this section, experiments on RedDots 2015 Quarter 4, part 1 will be discussed.

Experimental Setup and Baseline. RedDots only provided the the enrolement (1133 utterances) and test data (4726 utterances), thus we used the identical deep feature extractor as in Sect. 4.1, which leads to both channel and context mismatch. After applying VAD, the dataset was truncated, resulting in 1131 enrolement utterances and 4680 test utterances. The truncated test set encompasses 1275424 tests including 3850 true speaker and 1271574 impostor tests.

The baseline (Table 8) was run using the same configuration as in the corresponding RSR experiments, except that for *i*-vector, we use GC instead of PLDA since PLDA gives much worse performance than the GMM-UBM baseline. (*Bkg* data is borrowed from RSR2015 database, Sect. 4.1, leading to both channel and context mismatch.) Following the same metric with experiments on RSR2015, we didn't separate three types of error explicitly and only compute the overall EER.

[2] Speaker errors happen when an impostor speaker utters the correct text, is accepted, while text errors happen when an enrolled speaker utters the wrong text is accepted.

xpjiang@mail.ccnu.edu.cn

Table 8. Baselines for RedDots

Method	EER (%)
GMM-UBM	2.45
i-vector	3.30

The pattern of different layers on RedDots is the same as that on RSR2015, thus only the best results achieved in the 4th layer are presented in the following sections.

Results and Analysis. As can be perceived in Table 9, the *d/j*-vector(using GC) does not show commensurate performance on RedDots, which is a totally mismatched corpus (Besides the channel and context mismatch between RSR2015 *bkg* and RedDots data, RedDots also contains channel mismatch between enrolement and test data). However, it can still be observed that the newly proposed *v*-vector within our previous *j*-vector framework significantly outperforms the traditional mean based vector, which demonstrates the superiority and robustness of the new method and the best system is also slightly better than the baseline, even in a totally mismatched scenario.

Table 9. 4th layer results on RedDots, in % EER

Deep feature	*d*-vector	*j*-vector
$\boldsymbol{m}$	6.91	6.03
$\boldsymbol{v}_{\mathrm{d}}$	5.04	**2.36**

We also give an analysis on false accept error distribution on RedDots, as can be observed in Table 10, error rate is reduced by more than 60% on text and 27% on speaker, which agrees with the observation on RSR2015 in Sect. 4.1. Another interesting observation, which also exists in the RSR2015 experiments, is that covariance based deep feature works better for *j*-vector, which is consistent with that *j*-vector framework takes more text information into consideration.

Table 10. Err. distribution of false accepts (4th layer *j*-vector)

Error type	Deep feature	# Trials	# Err	Err. rate (%)
Speaker	$\boldsymbol{m}$	123703	16941	13.7
	$\boldsymbol{v}_{\mathrm{d}}$		12446	10.06
Text	$\boldsymbol{m}$	34658	2506	7.23
	$\boldsymbol{v}_{\mathrm{d}}$		997	2.87

xpjiang@mail.ccnu.edu.cn

Additionally, the utterance lengths in RedDots vary from 75 to 375 frames(after VAD), most of them are shorter than those in RSR2015. To further explore the impact of the utterance length, an analysis of the false reject error rate is given in Fig. 2. We can conclude that errors mainly happen when the utterances are rather short (less than 200 frames). Moreover, *v*-vector performs better than *i*-vector on utterances longer than 150 frames.

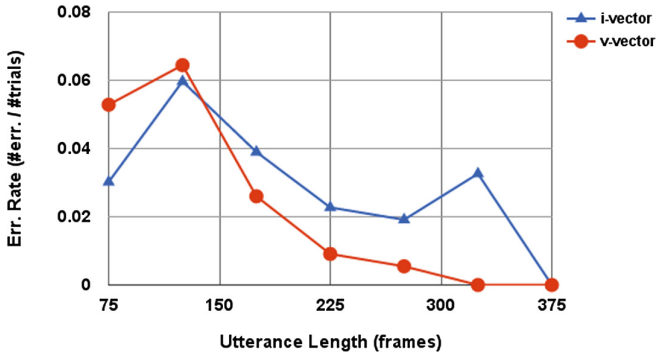

Fig. 2. False reject Err. rate w.r.t Utt length

Short utterances contain less text information and covariance matrices can not be estimated accurately with only a few frames, which explains why errors are more tending to occur when the utterances are shorter than 200 frames.

5 Conclusion

We proposed two kinds of covariance based approaches for deep feature extraction. While the diagonal covariance vector beats the mean based deep feature on both the RSR2015 and RedDots *text-dependent* speaker verification tasks, the full covariance vector is not yet applicable and needs some further research. We show that covariance based deep features are more capable of capturing text variability and perform better when incorporated into the joint multi-task learning framework. However, this approach's performance degrades when it comes to utterances shorter than 200 frames. For such short utterances, we will try to use some adaptation techniques rather than directly estimate the representation in the future work. Furthermore, other fusion techniques can be investigated.

References

1. Chen, K., Salman, A.: Learning speaker-specific characteristics with a deep neural architecture. IEEE Trans. Neural Netw. **22**(11), 1744–1756 (2011)
2. Heigold, G., Moreno, I., Bengio, S., Shazeer, N.: End-to-end text-dependent speaker verification. arXiv preprint arXiv:1509.08062 (2015)
3. Chen, Y.-H., Lopez-Moreno, I., Sainath, T.N., Visontai, M., Alvarez, R., Parada, C.: Locally-connected and convolutional neural networks for small footprint speaker recognition. In: INTERSPEECH (2015)

xpjiang@mail.ccnu.edu.cn

4. Lei, Y., Ferrer, L., McLaren, M., et al.: A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1695–1699. IEEE (2014)
5. Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K.: Deep feature for text-dependent speaker verification. Speech Commun. **73**, 1–13 (2015)
6. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
7. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Computat. **18**(7), 1527–1554 (2006)
8. Yu, D., Seltzer, M.L.: Improved bottleneck features using pretrained deep neural networks. In: INTERSPEECH, vol. 237, p. 240 (2011)
9. Grézl, F., Karafiát, M., Kontár, S., Cernocky, J.: Probabilistic and bottle-neck features for lvcsr of meetings. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, pp. IV–757. IEEE (2007)
10. Matejka, P., et al.: Neural network bottleneck features for language identification. In: Proceedings of IEEE Odyssey, pp. 299–304 (2014)
11. Fu, T., Qian, Y., Liu, Y., Yu, K.: Tandem deep features for text-dependent speaker verification. In: INTERSPEECH, pp. 1327–1331 (2014)
12. Richardson, F., Reynolds, D., Dehak, N.: Deep neural network approaches to speaker and language recognition. IEEE Sig. Process. Lett. **22**(10), 1671–1675 (2015)
13. Variani, E., Lei, X., McDermott, E., Lopez Moreno, I., Gonzalez-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4052–4056. IEEE (2014)
14. Chen, N., Qian, Y., Yu, K.: Multi-task learning for text-dependent speaker verification. In: INTERSPEECH (2015)
15. Larcher, A., Lee, K.A., Ma, B., Li, H.: Text-dependent speaker verification: classifiers, databases and RSR2015. Speech Commun. **60**, 56–77 (2014)
16. Lee, K.A., et al.: The RedDots data collection for speaker recognition. In: INTERSPEECH (2015)
17. Hain, T., Johnson, S., Tuerk, A., Woodland, P., Young, S.: Segment generation and clustering in the HTK broadcast news transcription system. In: Proceedings of 1998 DARPA Broadcast News Transcription and Understanding Workshop, pp. 133–137 (1998)
18. De Leon, P.L., Pucher, M., Yamagishi, J., Hernaez, I., Saratxaga, I.: Evaluation of speaker verification security and detection of hmm-based synthetic speech. IEEE Trans. Audio Speech Lang. Process. **20**(8), 2280–2290 (2012)
19. Chen, L.-W., Guo, W., Dai, L.-R.: Speaker verification against synthetic speech. In: 7th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 309–312. IEEE (2010)
20. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on Riemannian manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
21. Yao, J., Odobez, J.-M.: Fast human detection from videos using covariance features. Technical report, Idiap (2007)
22. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011)
23. Kenny, P., Boulianne, G., Dumouchel, P.: Eigenvoice modeling with sparse training data. IEEE Trans. Speech Audio Process. **13**(3), 345–354 (2005)

xpjiang@mail.ccnu.edu.cn

24. Kenny, P.: A small footprint i-vector extractor. In: Odyssey, pp. 1–6 (2012)
25. Prince, S.J., Elder, J.H.: Probabilistic linear discriminant analysis for inferences about identity. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8. IEEE (2007)
26. Kenny, P., Stafylakis, T., Ouellet, P., Alam, M.J., Dumouchel, P.: PLDA for speaker verification with utterances of arbitrary duration. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7649–7653. IEEE (2013)
27. Matějka, P., et al.: Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4828–4831. IEEE (2011)
28. Auckenthaler, R., Carey, M., Lloyd-Thomas, H.: Score normalization for text-independent speaker verification systems. Digit. Sig. Process. **10**(1), 42–54 (2000)
29. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digit. Sig. Process. **10**(1), 19–41 (2000)

xpjiang@mail.ccnu.edu.cn