Speaker Embedding Extraction with Multi-feature Integration Structure

Zheng Li[1], Hao Lu[2], Jianfeng Zhou[1], Lin Li[1], Qingyang Hong[2]
[1]School of Electronic Science and Engineering, Xiamen University, China
[2]School of Information Science and Engineering, Xiamen University, China
E-mail:{lilin, qyhong}@xmu.edu.cn

***Abstract*—Recently x-vector has achieved a promising performance of speaker verification task and becomes one of the mainstream systems. In this paper, we analyzed the feature engineering based on the x-vector structure, and proposed a multi-feature integration method to further improve the feature representation of speaker characteristic. The proposed multi-feature integration method could be implemented in two ways, with the symmetric branches and the asymmetric branches, respectively, to incorporate different types of acoustic features in one neural network. While each branch processed one type of acoustic features on the frame level, the outputs of the two branches for each frame were spliced together as a super vector before being input into the statistics pooling layer. The experiments were executed on the VoxCeleb1 data set, and the results showed that the proposed multi-feature integration method obtained a 22.8% relative improvement over the baseline in EER value.**

Index Terms: speaker verification, x-vector, multi-feature, feature engineering

I. Introduction

Feature engineering is a critical segment in an automatic speaker verification (ASV) system, which aims to capture internal characteristic of speaker identities with speech utterances. In the past decades, the raw acoustic features such as Mel frequency cepstral coefficients (MFCC) [1] and perceptual linear prediction (PLP) [2] were widely utilized to represent the speaker's characteristic parameter vectors. With such acoustic features, statistical models, such as GMM-UBM (Gaussian mixture model - universal background model) [3], were built to verify the speaker identities. On the other hand, the channel-dependent statistic representations, i-vector [4], would be obtained as well to map variable-length utterances into fixed-length vectors.

Recent years, deep learning based algorithms have merged increasingly. Y. Lei et al. [5] proposed DNN i-vector by introducing phonetic information into the i-vector model with a DNN model. Then E. Variani et al. [6] introduced embeddings concept into ASV task by using a neural network to extract discriminative vectors of speakers called d-vector. Soon after, deep feature [7] has been proposed to extract more informative speaker characteristics related embeddings and recurrent neural networks (RNN) [8], convolutional neural networks (CNN) [9] have been proposed to build an end-to-end speaker verification system. Inspired by [6], a significant breakthrough has been made known as x-vector [10] which was based on a time delay neural network (TDNN) [11] and a statistics pooling layer. The 'NIST Baseline Systems for 2018 Speaker Recognition Evaluation' [12] released from NIST contained the x-vector as one of baseline systems (another was the i-vector). Lately, D. Snyder et al. [13] used data augmentation to improve the robustness of x-vector and K. Okabe et al. [14] modified the statistics pooling layer of x-vector framework. What should be noted is that most neural networks mentioned above are based on single acoustic feature such as MFCC.

However, during the MFCC computation, some details were discarded with the compression of Mel filter banks and discrete cosine transform (DCT). Those discarded information may be advantageous in some tasks. To deal with such issues, it is common to train manifold systems by different kinds of acoustic features such as MFCC and FBank, and then make score fusion to achieve a better performance [15].

In this paper, we propose the a multi-feature integration method to utilize complementary acoustic features into a single x-vector system, which would extract more speaker discriminative information from raw speech. We realized two input network branches to process two kinds of acoustic features respectively, and the outputs of the branches were spliced together on frame level before the statistics pooling layer. These input network branches could have the same structure, named as the symmetric branches, or even various types of neural network structures (such as TDNN and ResNet [16]), named as the asymmetric branches, to learn more complementary representation.

The contributions of our work are as follows:

1. Exploring the potential of utilizing multi-feature to improve the single ASV system's performance.

2. Discussing the integration strategy, especially on which layer it would be better to execute the multi-feature integration between two input network branches.

3. Investigating the feasibility of the combination of various types of deep neural networks for different acoustic features.

The rest of this paper is organized as follow: Section 2 describes some technologies related to our work and Section 3 reveals the details of the proposed multi-feature integration methods. Section 4 introduces the experimental settings and Section 5 discusses the results. Finally, Section 6 concludes the paper.

II. RELATED WORK

This section demonstrates the related works of taking advantage of acoustic features more effectively in ASV task.

K. S. R. Murty et al. [17] indicated that there were some information missed, named the residual phase, in comparison with the information presented in the conventional MFCC. To tackle this problem, they proposed an algorithm with LP to capture residual phase as an additional feature. Systems that they built on residual phase feature performed far worse than systems on conventional MFCC, but after the fusion of two systems, the better EER was obtained.

Z. Li et al. [18] presented a method to utilize complementary acoustic features by concatenating multiple features to attain a new feature and implemented the ASV system with GMM-UBM [3]. To reduce the dimensions of the new combined feature and to avoid redundancy after combination, LDA and feature-domain latent factor analysis (fLFA) were used.

S. Yaman et al. [19] introduced a bottleneck feature extraction model which performed slightly worse than MFCC but it was complementary in score-level fusion. M. McLaren et al. [20] proposed the tandem feature based on the combination of bottleneck feature and basic acoustic feature to improve the performance of i-vector.

The ASVspoof2019 officially [21] revealed the baseline systems with two kinds of features (LFCC [22] and CQCC [23]) and two baseline systems shared a same GMM classifier. The results given show that two features have advantages in different attack tasks. Due to the complementary of acoustic features in ASVspoof, we explored multi-feature integration and multi-task learning [15] in the challenge and achieved comprising results. However in [15], we only implemented the proposed symmetric branches with the 5th hidden layer as the stitching layer and no attempts were completed on ASV task. In this paper, we further analyze the potentiality of multi-feature in the speaker verification task.

The works aforementioned indicated that training a speaker discriminative information extractor with only one kind of acoustic feature is not enough and it is necessary to compensate information discarded in single acoustic feature.

III. MULTI-FEATURE INTEGRATION STRUCTURES

The proposed multi-feature integration structures are based on the mature x-vector system without tedious modification but it could compensate the omitted details in acoustic features. In this paper, we chose MFCC and FBank feature as the combination of multi-feature due to they are different in processing completeness. The presented multi-feature integration strctures could be used in other complementary feature combinations such as MFCC and PLP.

The most straightforward approach to apply multi-feature is concatenating features directly as the input data of the x-vector system which is shown in Fig 1. We name this structure as the direct integration structure. The appending process could be written as:

$$x_{new} \leftarrow Append(x_{MFCC}, x_{FBank}) \tag{1}$$

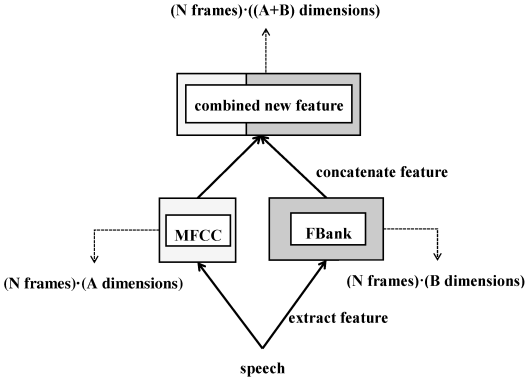

Fig. 1. The processing of concatenating multi-feature

where x_{MFCC},x_{FBank} represent the feature matrix of MFCC and FBank respectively, and x_{new} represents the concatenated feature's matrix.

Considering the data distribution of different features is comparatively dissimilar and the information between features is complementary, we proposed a multi-feature integration structure as shown in Fig. 2. In this structure, two different acoustic features extracted from the same speech segment are sent into two independent neural network branches. We use $T_1(\cdot)$, $T_2(\cdot)$ to represent the computaion in the TDNN block for MFCC and for FBank respectively. The initial configuration of two branches for two acoustic features are strictly the same except the input dimensions, which is named as the symmetric branches. So after training, $T_1(\cdot)$, $T_2(\cdot)$ still remain high similarity in activation logic. Then $T_1(\cdot)$ and $T_2(\cdot)$ are stitched in a fully connected layer $S(\cdot)$, named stitching layer, before being sent into the statistics pooling layer. Considering that there may be several hidden layers after the stitching layer, we use the stitching block to represent the stitching layer and the hidden layers between the stitching layer and the statistics pooling layer. The mathematical relation in stitching block could be written as:

$$y_{new} = S(T_1(x_{MFCC}) + T_2(x_{FBank})) \tag{2}$$

By doing so, two neural network blocks will separately learn unique information from various acoustic features. Furthermore, assigning different specific hidden layer as the stitching layer could lead to various performance in specific databases. So we experimented different hidden layer to be the stitching layer.

On the other hand, each kind of features may be fitted with its own best-matched neural network structure due to distinctive information. So we proposed another more distinguishable architecture in which two features are processed by two different kinds of network branches to learn complementary information. In this paper, the structure consists of TDNN and ResNet neural networks was explored, which was name as the asymmetric branches. In Figure 3, $T(\cdot)$, $R(\cdot)$ represent the computation of TDNN block for MFCC and ResNet block for FBank, respectively. We used the FBank feature which contains more original knowledge as ResNet block's

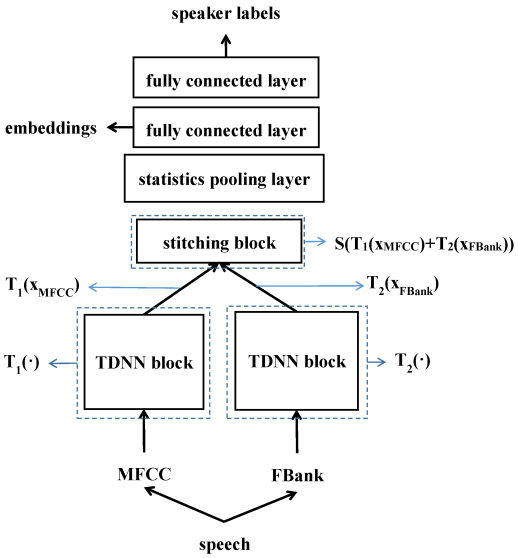

Fig. 2. The multi-feature integration with the symmetric branches

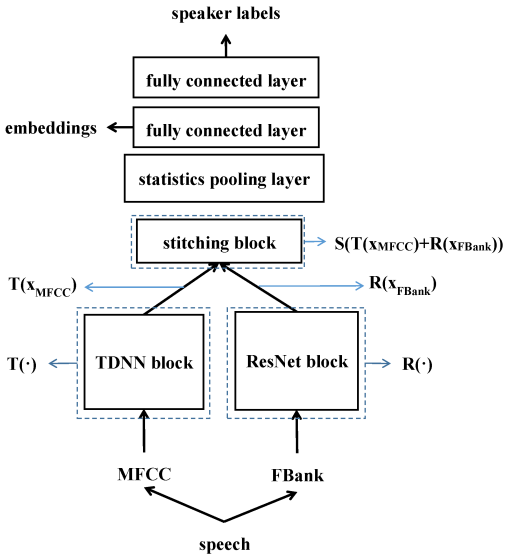

Fig. 3. The multi-feature integration with the asymmetric branches

input, because the ResNet is expert at handling original and complex information. The mathematical relation in stitching block could be rewritten as:

$$y_{new} = S(T(x_{MFCC}) + R(x_{FBank})) \tag{3}$$

After training, the output of the penultimate hidden layer is extracted as speaker embedding (x-vector). Mean subtraction, length normalization, LDA and PLDA would be applied to speaker embeddings in sequence.

IV. EXPIMENTAL SETTINGS

For the purpose of exploratory experiment, all the conceptions aforementioned were only implemented with MFCC and FBank, but it is also the same for the other different features or even with more than two kinds of features. All the experiments were executed on Kaldi toolkit [24].

A. Database

The VoxCeleb 1 [25] training data set was utilized as our training set and the VoxCeleb 1 test data set was used to evaluate the models. Before training, we used the same data augmentation algorithm as Kaldi's recipe[1] to expand the training data. We randomly chosen 140,000 noisy utterances and mixed them with the original training data.

B. Baseline systems

The baseline x-vector systems based on TDNN were identical with Kaldi's official recipe, using one kind of acoustic features, respectively. The baseline x-vector systems based on ResNet used the typical ResNet configuration and the number of ResNet block's hidden layers were the same as the systems based on TDNN. The 30-dimensional MFCC and 40-dimensional FBank were used as input acoustic features with a frame shift of 10ms and a frame length of 25ms. CMVN with a 3-second window and energy-based VAD were applied. The backend was also the identical as what's in the Kaldi's recipe. Mean subtraction, length normalization were applied to x-vectors and then the vectors dimensions were reduced to 200 with LDA. The 200-dimensional vectors were used to train the PLDA model, on which the final verification scores were obtained.

[1]egs/voxceleb/v2

C. Proposed systems

The settings in the direct integration structure was the same as Kaldi's official recipe except for the input feature was an new feature concatenated by MFCC and FBank.

The configuration in the multi-feature integration with symmetric branches could be regarded as the bifurcated structure of the baseline x-vector systems before the stitching layer.

Meanwhile, the configuration before the stitching layer in the multi-feature integration with asymmetric branches could be considered as the bifurcated structure with TDNN and ResNet. Moreover, the settings before the statistics pooling layer were the same as the parameters of TDNN for MFCC or those of ResNet for FBank.

The backend process in experiments were the same as the baseline systems.

V. RESULTS

The results of the baseline systems and the proposed systems are reported in Table 1, including the evaluation metrics of minDCF08 (p-target=0.01), minDCF10 (p-target=0.001) and equal error rate (EER).

Given the single kind of acoustic feature, TDNN with MFCC outperformed the same network structure with FBank. In contrast, ResNet with FBank obtained better performance than ResNet with MFCC. It is obvious to find out that TDNN with MFCC and ResNet with FBank would be the appropriate configurations, of which the former one is abbreviated as the baseline 1 and the latter one as the baseline 2.

TABLE I
THE RESULTS OF SYSTEMS ON VOXCELEB 1

No.	System	Details	EER(%)	minDCF08	minDCF10
01	x-vector	only MFCC (baseline 1)	4.67	0.4479	0.5496
02	x-vector	only FBank	5.35	0.4693	0.6402
03	ResNet	only MFCC	6.46	0.5584	0.7554
04	ResNet	only FBank (baseline 2)	4.51	0.4418	0.4956
05	direct integration structure	-	4.67	0.4256	0.5304
06	symmetric branches	stitching layer @ 5th	4.40	0.4554	0.5706
07	symmetric branches	stitching layer @ 4th	4.22	0.4241	0.4933
08	symmetric branches	stitching layer @ 4th (fine-tuning)	3.70	**0.3582**	**0.4256**
09	symmetric branches	stitching layer @ 3th	4.31	0.4082	0.5704
10	symmetric branches	stitching layer @ 2th	4.39	0.4044	0.5081
11	asymmetric branches	fine-tuning	**3.48**	0.3681	0.4566
12	fusion 1+2	Equal weight fusion	4.19	0.3990	0.5706
13	fusion 1+4	Equal weight fusion	3.81	0.3776	0.5003
14	fusion 1+2+8	Equal weight fusion	3.48	0.3365	**0.4817**
15	fusion 1+4+11	Equal weight fusion	**3.38**	**0.3331**	0.4939

We also implemented the direct integration system, and the result of the direct integration structure showed that merely combining various features as an new feature for x-vector can scarcely improve the performance.

For the symmetric branches based systems, we analyzed that how the stitching layer affected the performance. In our experiments on the VoxCeleb 1 dataset, the fourth hidden layer before the statistics pooling layer would be the best choice as the stitching layer, while with a relative reduction of 6.3% in EER compared to the baseline 2. With fine tuning, the symmetric branches based system obtained an optimized performance of 3.70% in EER value, which was better than that of score fusion between the baseline 1 and the baseline 2.

For the asymmetric branches based system, the best performance was achieved with 22.8% relative improvement in comparison with the baseline 2. This result indicated that the proposed asymmetric branches would learn more beneficial details in raw acoustic features and magnify the differences between speakers, and each branch network could expolit its own relevant feature more efficiently. Furthermore, we compared the score fusion strategies with different systems. With the equal weighted fusion parameters, the score fusion of system 1, system 4 and system 11 gained the promising performance of 3.38% in EER value, which was nearly 25% relative improvement in contrast with the baseline 2.

VI. CONCLUSIONS

In this paper, we present two speaker embedding extractors with multi-feature integration structure. The multi-feature integration models could be jointly trained with different features to learn complementary and auxiliary information. The best result is 22.8% relative better in EER than the best baseline on VoxCeleb 1. Our experiments illustrated the effectiveness of the proposed multi-feature integration method and achieved the promising improvement from score-level fusion at last.

In the future, we will try to optimize the network learning strategies by modifying different loss functions for each feature.

VII. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (Grant No.61876160).

REFERENCES

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[4] 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, 2010.

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

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[7] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[8] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[9] S. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.
[10] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.
[11] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
[12] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The nist speaker recognition evaluation–overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, 2000.
[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.
[14] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.
[15] R. Li, M. Zhao, Z. Li, L. Li, and Q. Hong, "Anti-spoofing speaker verification system with multi-feature integration and mutli-task learning," in *Interspeech*, 2019, p. (accepted).
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[17] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and mfcc features for speaker recognition," *IEEE signal processing letters*, vol. 13, no. 1, pp. 52–55, 2005.
[18] Z. Li, L. He, W. Zhang, and J. Liu, "Multi-feature combination for speaker recognition," in *2010 7th International Symposium on Chinese Spoken Language Processing*. IEEE, 2010, pp. 318–321.
[19] S. Yaman, J. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.
[20] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4814–4818.
[21] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.
[22] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," *the International Speech Communication Association (ISCA)*, 2015.
[23] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.
[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011.
[25] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.