Team 27 system description for SdSV 2020 challenge task2

Mari Ganesh Kumar[1], *Ishika Gupta*[1]*, *Sudhanshu Srivastava*[1]*, *Saish Jaiswal*[1], *B. Bharathi*[2], *and Hema A. Murthy*[1]

[1]Indian Institute of Technology Madras
[2]SSN college of Engineering
{mari,hema}@cse.iitm.ac.in, bharathib@ssn.edu.in

Abstract

In this report, we present the details of the systems submitted to task 2 Short Duration Speaker Verification (SdSV) 2020 challenge. In this report, the standard MFCC features are replaced with the slope of the spectrum along the frequency (mel-filter slope or MFS) in the x-vector-based speaker verification system. This report also explores the use of i-vector and x-vector embeddings jointly by performing early fusion. The results indicate while MFS improves the baseline x-vector performance, the early fusion of i-vector and x-vector embeddings further improves the performance, which suggests that the x-vector and i-vector provide the required complementary information. Using combined representation and various features, the final system submitted to the challenge achieves an EER of 3.01% on the SdSV 2020 challenge evaluation set.

Index Terms: speaker recognition, MFS, LFS, MFCC, i-vector, x-vector, combined representation

1. Introduction

Deep neural network (DNN) based systems have recently shown to improve speaker recognition performance. However, recognizing speakers from short utterances is still a challenging problem. Task 2 of the short-duration speaker verification (SdSV) 2020 challenge provides a standard benchmark for evaluating text-independent speaker verification systems on short utterances (ranging from 1 to 8 seconds) [1]. This report discusses the systems developed by Team 27 for the SdSV 2020 challenge.

Speaker recognition systems using neural embeddings (x-vectors) obtained from deep neural networks (DNN) [2] are the current state-of-the-art. To date, most of the x-vector systems have been studied using mel frequency cepstral coefficients (MFCC) as features [3–7]. In this report, we revisit the feature extraction part of the x-vector system using the MFS [8, 9] features. On the evaluation set of the SdSV 2020 challenge, we show that the x-vector system trained using MFS features performs better than traditional MFCC features.

Further in this report, we combinedly use DNN based x-vector [2] and E-M based i-vector [10] to recognize speakers from short utterances. On the SdSV 2020 challenge evaluation, we observe that a combined representation in which both i-vectors and x-vectors are concatenated results in significantly improved performance.

The remainder of the report is organized as follows. Section 2 discusses the systems developed for the challenge. Section 3 outlines the experimental setup as per the SdSV 2020 challenge evaluation plan. The results of the proposed systems are given in Section 4 followed by the conclusion in Section 5.

*-authors contributed equally

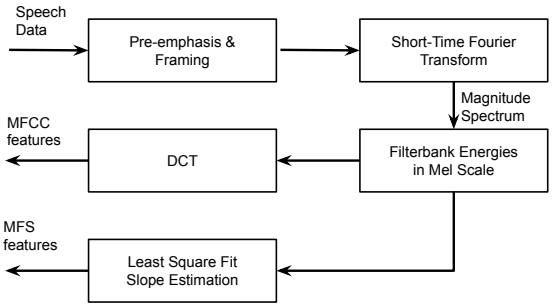

Figure 1: *Block diagram for MFS and MFCC feature extraction*

2. Speaker Recognition Systems

This section gives the details of features and speaker recognition systems developed for the challenge, which consists of the i-vector, x-vector baseline, and their combined system.

2.1. Features

Mel-Frequency Cepstral Coefficients (MFCC) is the most common feature used in speech processing. MFCC features are computed by applying DCT on the equally spaced filter bank energies in the mel scale. Alternatively, in [9], Mel Filter-bank Slope (MFS) features, which emphasize higher-order formants better are shown to be a better feature for speaker recognition. This is mainly because the higher-order formants are shown to carry speaker information [9, 11]. Instead of applying DCT on filterbank energies, the MFS features capture the variance across the filterbank energies by estimating the slope. A detailed algorithm for extracting MFS features is given in [9]. Figure 1 highlights the difference in the extraction of MFS and MFCC features. In addition to the MFS features, Linear Filter-bank Slope (LFS) features were also used. LFS computes the slope on the linear filter-bank energies.

All three features were extracted using a frame-length of 25ms and a frame-shift of 10ms. MFCC features were extracted using 30 coefficients derived from 30 filterbank energies. For MFS features, as shown in [9], the slope was estimated using 4 consecutive filterbank energies. 40 and 80 dimensional MFS features were extracted for the i-vector and x-vector system, respectively. For LFS features only 40 features were extracted for both the i-vector and x-vector system, owing to the time limitation.

2.2. i-vector systems

i-vectors are generative systems that model speaker variabilities in a low dimensional domain keeping the relevant information. In i-vector, a single total variability space (defined by **T**-matrix) is used to model the statistics obtained from the universal background model (UBM) [10]. As mentioned in Section 1, both the UBM and the **T**-matrix are estimated using E-M.

For building the i-vector systems, the delta and acceleration were also computed for the features mentioned in Section 2.1. A 2048 mixture UBM was used, and the dimension of i-vector was empirically set to 400. The i-vector systems were developed using the Kaldi toolkit [12].

2.3. x-vector systems

Contrasting i-vector, x-vectors are discriminative-DNN based systems that represent the relevant speaker-information in a low dimensional space [2]. The x-vector DNN consists of few TDNN layers that work at frame level, a statistical pooling layer that accumulates statistics of the frame-level outputs from TDNN layers, embedding layers that function on the segment-level and at a final softmax output layer. After training the DNN, x-vectors are extracted as embeddings from one of the layers operating on segment level [2]. When trained with a large number of speakers, the x-vectors have been shown to generalize better than the i-vectors[2]. The configuration of the DNN is the same as given in [2]. x-vectors of 512 dimensions are extracted as embeddings from the 6th layer, similar to [2]. The x-vector system was also developed using Kaldi toolkit [12].

2.4. Combined representation

The i-vector system is trained using a generative model, while the x-vector based system is trained using a discriminative model. Many previous works have stated that these systems capture complementary information. In [13, 14], the late fusion of scores from the i-vector and x-vector technique was shown to improve the performance of speaker diarization. In [6], a transformation by canonical correlation analysis of i-vector and x-vector is used to make the later generative. In [15], the same is done using a parallel factor analysis. In this report, we show that a simple concatenation of the i-vector and x-vectors, henceforth referred as c-vector, gives a better result than the individual and the late fusion system.

2.5. Back-ends

After extracting the embeddings (i-vector, x-vector, or c-vector), they are projected to a lower-dimensional space using LDA after mean centering. For all the embeddings, the dimension of the LDA projection was set as 200. After projecting to the LDA subspace, embeddings are length normalized and scored using both the PLDA [16] and cosine similarity (CS) back-end.

3. Experimental Setup

3.1. Training data

3.1.1. Training data for i-vector and x-vector

The following corpora were used to train the i-vector (UBM and T-matrix) and the x-vector (DNN) system:

- Development subset of the VoxCeleb1 dataset [17], which contains over 100k utterances extracted from YouTube videos of 1.2k speakers
- VoxCeleb2 dataset [18] which contains about 6.1k speakers and 1 million utterances again from the YouTube dataset
- LibriSpeech dataset [19] with 2.4k speakers and a total duration of 1000 hours
- In-domain Deepmine dataset [20] (for task 2 [1]) with 588 speakers about 85k utterance

All the corpora, as mentioned above, were allowed to be used as training data for the SdSV challenge 2020 [1].

For training the DNN for x-vector systems, additional data were synthetically generated by augmenting the above data given data with noise. The procedure and non-speech datasets that were used to augment the training data is the same as [2]. A total of 1 million augmented utterances were randomly added to the training data [2].

3.1.2. Training data for LDA and PLDA

Only the in-domain Deepmine dataset and the Voxceleb1 dataset were used to train the LDA and the PLDA models. But including more data, we did not observe any significant gain in the performance on the evaluation data set.

4. Results

The baseline system for the SdSV 2020 challenge is an x-vector system with PLDA back-end trained only on the VoxCeleb1 and the VoxCeleb2 datasets using MFCC features. The systems were developed following the experimental setup given in Section 3. The results of the primary and single system, along with the baseline, are given in Table 1.

It was observed that cosine similarity gave consistently better results than the PLDA back-end. Hence, for the final systems, we used only the cosine similarity back-end. Further, the scores of the final systems were normalized using T-norm. For every speaker, 200 imposters were chosen from the in-domain training data consisting of 588 speakers. The results of the final system are given in Table 1.

The x-vector system with MFS features and cosine similarity back-end was the best performing single system on the SdSVC 2020 evaluation set. After normalizing the scores the system gave an EER of 3.52% (Table 1). The c-vector system, for the MFS, LFS and MFCC features, were observed to give an EER of 3.40%, 4.34% and 3.63%, respectively. For the primary system, we apply a late fusion on the scores of these three systems, which resulted in an EER of 3.01% and a min-DCF value of 0.1371.

5. Conclusion

In this report, for short utterances, we show that MFS features, which emphasize higher-order formants, improve the performance of x-vector systems. The late fusion over c-vectors of different features were submitted as the primary system for the challenge. This final system scored an EER of 3.01% and a min-DCF of 0.1371 for the SdSV challenge.

6. Acknowledgment

The authors would like to thank the Department of Science and Technology for funding under the scheme Teachers Associationship for Research Excellence (TARE). Ref.No. TAR/2018/000052.

Table 1: *Performance of the baseline and the final systems submitted to SDSVC 2020 challenge.*

System No	System Description	Back-end	EER	min-DCF
a) - Challenge Baseline				
1	MFCC-x-vector	PLDA	10.67	0.4324
b) - Submitted Single System				
2	MFS-x-vector	CS*	**3.52**	**0.1654**
c) - Concatenated systems (c-vector)				
3	MFCC-c-vector	CS*	3.63	0.1643
4	MFS-c-vector	CS*	3.40	0.1595
5	LFS-c-vector	CS*	4.34	0.1968
d) - Submitted Primary System				
6	Score fusion of system 3, 4 & 5		**3.01**	**0.1371**

*score normalization was done using T-norm

7. References

[1] "Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan."

[2] 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.

[3] S. Ramoji, P. Krishnan V, P. Singh, and S. Ganapathy, "Pairwise discriminative neural plda for speaker verification," *arXiv preprint arXiv:2001.07034*, 2020.

[4] O. Novotnỳ, O. Plchot, P. Matejka, L. Mosner, and O. Glembek, "On the use of x-vectors for robust speaker recognition." in *Odyssey*, 2018, pp. 168–175.

[5] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," *arXiv preprint arXiv:1909.06351*, 2019.

[6] L. Xu, R. K. Das, E. Yılmaz, J. Yang, and H. Li, "Generative x-vectors for text-independent speaker verification," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1014–1020.

[7] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Interspeech*, 2018, pp. 3573–3577.

[8] H. A. Murthy, F. Beaufays, L. P. Heck, and M. Weintraub, "Robust text-independent speaker identification over telephone channels," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 554–568, 1999.

[9] S. R. Madikeri and H. A. Murthy, "Mel filter bank energy-based slope feature and its application to speaker recognition," in *2011 National Conference on Communications (NCC)*. IEEE, 2011, pp. 1–4.

[10] 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.

[11] P. Rose, *Forensic speaker identification*. cRc Press, 2002.

[12] 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*, no. CONF. IEEE Signal Processing Society, 2011.

[13] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge." in *Interspeech*, vol. 2018, 2018, pp. 2808–2812.

[14] Q. Lin, R. Yin, M. Li, H. Bredin, and C. Barras, "Lstm based similarity measurement with spectral clustering for speaker diarization," *arXiv preprint arXiv:1907.10393*, 2019.

[15] L. Xu, B. Ren, G. Zhang, and J. Yang, "Linear transformation on x-vector for text-independent speaker verification," *Electronics Letters*, vol. 55, no. 15, pp. 864–866, 2019.

[16] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[17] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1929

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[20] H. Zeinali, L. Burget, and J. Cernocky, "A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: the DeepMine database," in *Proc. ASRU 2019 The 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.