SPEAKER EMBEDDING CONVERSION FOR BACKWARD AND CROSS-CHANNEL COMPATIBILITY

Tianxiang Chen, Elie Khoury

Pindrop, Atlanta, GA, USA
{tchen,ekhoury}@pindrop.com

ABSTRACT

The accuracy of automatic speaker verification (ASV) systems has shown tremendous improvements due to the recent breakthroughs in low-rank speaker representations and deep learning techniques, leading to the success of ASV in real-world applications from call centers to mobile applications and smart devices. Particularly, some ASV providers have been migrating their legacy systems from the traditional GMM based i-vector paradigm to the deep learning based x-vector paradigm. Additionally, some of them are in need of implementing simultaneously different systems for different use cases such as 8 kHz over the phone channel and 16 kHz on virtual assistants. In either cases, the speaker embeddings extracted from one ASV system are often not compatible with another ASV system. This makes the process of interchangeability between systems very cumbersome and costly. In this paper, we address this issue by proposing a highly efficient speaker embedding converter that transforms a speaker embedding extracted from system A into a speaker embedding that can be used by system B. We evaluate the performance of the embedding converter for i-vector to x-vector upgrade scenario and for cross channel compatibility scenario. In both scenarios, we show that the proposed system achieves very low and compelling equal error rates.

1. INTRODUCTION

The speaker embedding extractor is one of the key components of the ASV system. Its goal is to extract a low-rank mathematical representation of the speaker for every input speech utterance. Such vector is often called a speaker embedding. Then, multiple embeddings from the same speaker are aggregated to create a user profile, also called enrollment model. Once a user profile is created, and when a new authentication request is received, a similarity metric such as cosine similarity or probabilistic linear discriminant analysis (PLDA) [1], is used to compare the enrollment embedding to the authentication embedding. Ideally, if the authentication embedding belongs to the same speaker as the enrolled speaker, the two embedding vectors should be close to one another in the vector space, and the ASV system should output a high verification score. Contrarily, if the authentication embedding is from a different speaker, the verification score should be low.

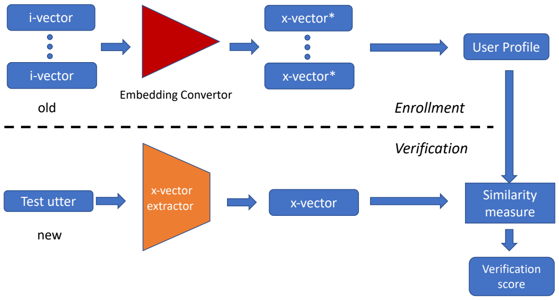


Fig. 1: *Overview of the proposed ASV system. The system uses the embedding convertor to transform the old enrolled i-vectors into the "converted" x-vectors* that could be directly compared with the newly acquired test x-vector.*

For about three decades, ASV systems were based on the GMM paradigm [2] and its most recent version uses a front end factor analysis that extracts speaker embeddings better known as *i-vectors* [3]. However, more recently, the performance of ASV systems has improved by leaps and bounds with the development of deep neural network (DNN) based speaker embeddings often called *x-vector* [4]. To take advantage of this progress, many existing real-world ASV providers have been trying to migrate from i-vectors to x-vectors. This poses practical challenges since the previously enrolled speaker models are using i-vectors and it is impossible to directly compute the similarity between i-vectors and x-vectors: typically the two sets of vectors do not live in the same vector space. In preparation for a system upgrade from i-vector to x-vector, ASV providers have historically tackled the problem in three different ways: 1) re-extract the x-vectors on the "old" enrollment audio utterances, but this solution is often difficult due to the legal and privacy concerns, 2) completely purge old enrolled models, and ask the users to repeat the enrollment process, but this solution will lead to customer dissatisfaction, 3) run old and new ASV systems in parallel as proposed in [5], but this will in-

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

ASV System	Train Data	Samp. Rate	Feat. Size
ivector-16k	VoxCeleb 1+2	16 kHz	200
xvector-16k	VoxCeleb 1+2	16 kHz	200
ivector-8k	SWB+SRE	8 kHz	200
xvector-8k	SWB+SRE	8 kHz	150

Table 1: *The four different ASV systems used in this work. ivector-16k and xvector-16k are trained using Kaldi's VoxCeleb recipe and ivector-8k and xvector-8k are trained using Kaldi's SRE 2016 recipe.*

evitably increase the operational and engineering cost of such upgrade, and lead to higher computational cost.

To the best of our knowledge, there is one single attempt [6] to tackle this migration problem using machine learning. Their work uses a simple neural network with a regression loss based on mean square error (MSE) to convert between old and new i-vectors. While their results were promising, their study was limited to only i-vectors, and telephony channel scenario.

Our work build upon this baseline system [6], and propose a new loss to train the convertor model based cosine proximity. Furthermore, we extend the studies to cover various practical scenarios. We investigates the cross paradigm scenarios of i-vector to x-vector conversion, which we hypothesized to more challenging, as well as x-vector to x-vector conversion between narrow-band and wide-band conditions.

To favor the reproducibility of our work, we use the pre-trained i-vector and x-vector ASV systems from Kaldi's[1] VoxCeleb [7] and SRE 2016 [8] recipes. The proposed convertor system is evaluated on the VoxCeleb and NIST SRE 2016 dataset. The experimental results show that the proposed system is able to bring feature compatibility in all three scenarios.

2. DATASETS

2.1. VoxCeleb Dataset

The VoxCeleb 1 and 2 datasets [9, 10] consist of over 7,000 speakers and 2,000 hours of speech. The training set is a combination of the VoxCeleb 1 and 2, except the held out VoxCeleb 1 test dataset that is used to evaluate the system. Data augmentation is described in [4] and is applied to the training dataset. The i-vector embedding extraction model is trained purely on the clean data. The PLDA model of both i-vector and x-vector systems are trained only on clean data.

2.2. NIST SRE Dataset

As described in [4], the training set combines the Switchboard (microphone) and SRE (telephone) datasets. The Switchboard data consists of Switchboard 2 Phases 1, 2, and 3 as well as Switchboard Cellular. The SRE data combines the NIST SREs from 2004 to 2010 and Mixer 6. Data augmentation is also applied to the Switchboard and SRE data. The data augmentation technique is introduced in [4]. Both the i-vector and x-vector embedding extraction models are trained on the combination of Switchboard and SRE training data. The PLDA is only trained using the SRE data. The NIST SRE 2016 evaluation dataset is used to evaluate the systems. The SRE 2016 unlabeled data is used for domain adaptation.

3. METHODOLOGY

3.1. ASV Systems

3.1.1. Wide-band ASV Systems

The wide-band 16 kHz ASV i-vector and x-vector systems are trained using VoxCeleb 1 and 2 data described in Sec 2.1. The i-vector system is based on the Kaldi's GMM-UBM recipe. The low-level features are 24-dimensional MFCCs extracted on 25 ms windows with 10 ms frame shift. The UBM is a 2,048 component GMM. The output of the i-vector extractor is a 400 dimensions feature vector.

The x-vector extraction model is a DNN based model. The model is trained on 30-dimensional MFCCs with a frame-length of 25 ms and 10 ms shift. The full DNN architecture is detailed in [4]. The output of the embedding is a 512-dimensional vector. After extracting the i-vectors and x-vectors, a LDA+PLDA backend is trained using extracted embedding vectors. The LDA model reduces the dimension of i-vector and x-vector to 200.

3.1.2. Telephony ASV Systems

The narrow-band 8 kHz ASV i-vector and x-vector systems are trained using Switchboard and SRE data described in the Section 2.2. The i-vector system is similar to wide-band i-vector system. The low-level features are 20-dimensional MFCCs extracted on 25 ms windows with 10 ms frame shift. The output of the i-vector extractor is a 600-dimensional vector. The x-vector extraction model is trained on 23-dimensional MFCCs. The output of the x-vector extractor is 512-dimensional vector. The LDA+PLDA backend is used in 8 kHz ASV systems. The LDA model reduces the dimension of i-vector and x-vector to 200 and 150, respectively.

3.2. Embedding Convertor

As shown in Table 2, the proposed embedding convertor model is a shallow DNN consists of three fully connected layers and followed by a length normalization layer. The embedding convertor is aiming to convert the input feature

[1] https://kaldi-asr.org

Layer	Input size	Output size
Fully Connected (selu)	$vect_{in}$	1024
Fully Connected (selu)	1024	512
Fully Connected (linear)	512	$vect_{out}$
Length Normalization	-	-

Table 2: *The architecture of the convertor DNN model. The input and output sizes are defined by the shape of* $vect_{in}$ *and* $vect_{out}$.

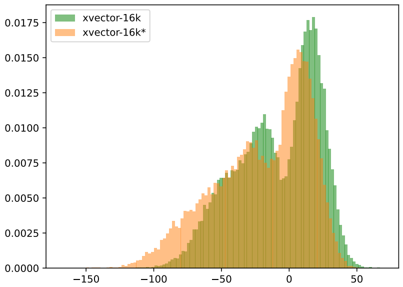


Fig. 2: *Score distribution comparison between original x-vectors converted x-vectors* on the VoxCeleb 1 test set.*

embedding to the target feature embedding vector space. Totally three embedding convertors are trained in this work, 8 kHz i-vector to x-vector convertor, 16 kHz i-vector to x-vector convertor and 8 kHz x-vector to 16 kHz x-vector cross-channel convertor.

For i-vector to x-vector convertor, the input is the 600-dimensional i-vector and the target is the x-vector after LDA transformation and length normalization. The 8 kHz convertor is trained using the i-vectors and x-vectors extracted on the Switchboard and SRE training data described in Section 2.3. The 16 kHz convertor is trained using i-vectors and x-vectors extracted on VoxCeleb 1 and 2 training data described in Section 2.1.

For the cross-channel convertor, the VoxCeleb 1 and 2 data is used to train and evaluate the system. We first downsample the VoxCeleb data to 8 kHz and extract the x-vectors using the telephony embedding extractor described in Section 3.2. The 16 kHz x-vectors are extracted using the wide-band embedding extractor introduced in Section 3.1. Then, LDA transformation and length normalization are applied to all x-vectors extracted from both systems. Finally, the cross-channel convertor is trained on the transformed x-vectors. The baseline convertors are trained using the same data with mean square error loss function. All our proposed embedding convertors are trained using mean cosine similarity loss and Adam optimizer for 30 epochs. The mini-batch size is 200.

3.3. System Framework

Figure 1 shows an overview of the ASV system with embedding convertor. Same as other common ASV systems, it consists of two phases, enrollment and verification. Suppose the "new" x-vector ASV system received a new verification request and the user has already enrolled in the "old" i-vector system. The enrolled i-vectors are first fed into the speaker embedding convertor. Then, the speaker embedding converter converts the "old" i-vectors to converted speaker embeddings. Next, the system aggregates all converted speaker embeddings to create a user profile. The user profile can be used to directly compare with the "new" x-vector. Finally, a PLDA score is computed between the user profile and the auth x-vector. The verification decision can be made based on the PLDA score.

Conversion	Enrollment	Verification	EER	minC
No	ivector-16k	ivector-16k	5.33%	0.53
No	xvector-16k	xvector-16k	3.11%	0.36
Baseline [6]	ivector-16k	xvector-16k	4.91%	0.57
Proposed	ivector-16k*	xvector-16k	4.87%	0.52
Proposed	ivector-16k*	ivector-16k*	5.10%	0.50

Table 3: *Performance of the wide-band i-vector to x-vector conversion scenario on the VoxCeleb 1 test set. Feature vectors marked with a * are converted feature vectors*

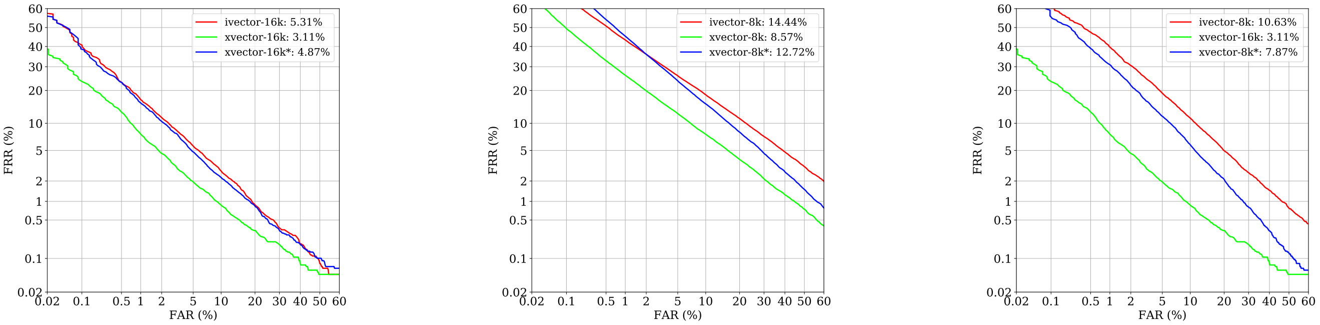
In this case, the ASV system does not need to re-extract "new" x-vector on "old" enrollments, and only one speaker embedding extractor and PLDA model is needed. The same workflow also applies to the cross-channel scenario. The user only needs to enroll in one channel such as telephony 8 kHz channel, and the enrollment vectors can be used in other channels (e.g. 16 kHz).

4. EXPERIMENTS

As described in Section 2 and 3, we construct three different evaluation benchmarks to test three different scenarios. The held out VoxCeleb 1 test set is used to evaluate the 16 kHz i-vector to x-vector conversion and cross-channel conversion. The SRE 2016 evaluation set is used to benchmark the 8 kHz i-vector to x-vector conversion scenario.

We use equal error rate (EER) and min C-Primary (minC) as the performance metrics to evaluate the systems. Both metrics are widely used in evaluating ASV systems performance. Additionally, we plot the detection error trade-off (DET) curve that shows the accuracy of the systems at different FRRs and FARs.

Table 3 shows the results of wide-band i-vector to x-vector conversion scenario. On VoxCeleb 1 test data, the i-vector based system has an EER of 5.33%, while the x-vector system achieves an EER of 3.11%. Next, we apply speaker embedding convertor to convert all i-vector enrollment embeddings. Our proposed convertor system is able to attain an EER of 4.87% and minC of 0.53 when evaluating on x-vector test samples. The performance is better than the i-vector system and the baseline convertor system. The last row in Table 3



(a)

(b)

(c)

Fig. 3: *DET curves of the three different conversion systems. (a) and (b) are for wide-band and narrow-band i-vector to x-vector conversion scenarios; (c) is for cross-channel scenario.*

Conversion	Enrollment	Verification	EER	minC
No	ivector-8k	ivector-8k	14.44%	0.78
No	xvector-8k	xvector-8k	8.57%	0.63
Baseline [6]	ivector-8k*	xvector-8k	13.08%	0.86
Proposed	ivector-8k*	xvector-8k	12.72%	0.83
Proposed	ivector-8k*	ivector-8k*	14.21%	0.86

Table 4: *Performance of the narrow-band i-vector to x-vector conversion scenario on the SRE 2016 official evaluation set. Feature vectors marked with a * are converted feature vectors*

shows the results when both enrollment and test samples are converted feature. The EER is 5.10%, which is slightly better than the i-vector system. We believe the performance gain over the i-vector system is due to the robustness of the PLDA model in the x-vector system.

The results of narrow-band i-vector to x-vector conversion scenario is presented in Table 4. The evaluation is performed on SRE 2016 eval set. The i-vector and x-vector systems achieve an EER of 14.44% and 8.57%, respectively. By using the converted embedding as enrollments, the EER on x-vector test samples is 13.02%, lower than the i-vector system.

For the cross-channel system conversion scenario, the results are detailed in Table 5. In this scenario, the proposed system converts the 8 kHz enrollment samples and tests on the 16 kHz test samples. This system achieves an EER of 7.87% and minC of 0.65. Although the proposed convertor model is slightly worse EER than the baseline, it has a lower minC. It is also worth noting that the xvector-8k system is trained on the SRE and SWBD data, thus the Voxceleb 1 test set is out of domain data. This could be the reason why the converted feature has a significant improvement over the xvector-8k.

It is important to note that although the EER of the converted feature is lower than the original i-vector systems in both narrow-band and wide-band scenarios, the DET curves (Fig 3) show that by using the convertor, the system performs slightly worse at the low FAR range. Figure 2 presents the score distributions of the x-vector and converted features on Voxceleb 1 test set. Clearly, there's a distribution shift when using the proposed feature convertor. We believe this is the reason why the system performance decreases at the low FPR range.

Conversion	Enrollment	Verification	EER	minC
No	xvector-16k	xvector-16k	3.11%	0.36
No	xvector-8k	xvector-8k	10.64%	0.71
Baseline [6]	xvector-8k*	xvector-16k	7.34%	0.72
Proposed	xvector-8k*	xvector-16k	7.87%	0.68

Table 5: *Performance of the cross-channel x-vector conversion scenario on the VoxCeleb 1 test set.*

5. CONCLUSIONS

In this paper, we propose a speaker embedding extractor along with a new paradigm for ASV system to achieve speaker embedding compatibility across different ASV systems. We totally studied 3 different scenarios, including two i-vector to x-vector system upgrade scenarios and one cross-channel conversion scenario.

The experimental results demonstrate that our proposed speaker embedding convertor is able to bring feature embedding compatibility between two different ASV systems in all scenarios, and achieve better performance than the baseline convertor system. Our proposed convertor systems can even achieve slightly better performance over the "old" ASV systems in all scenarios. However, our experiments also show that the converted feature embedding performs worse than the "old" ASV systems at the low FAR range. A useful extension of this work could be using score calibration to improve the performance at the low FAR range.

6. REFERENCES

[1] Sergey Ioffe, “Probabilistic linear discriminant analysis,” in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[2] D. Reynolds, T. Quatieri, and R. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[4] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev 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.

[5] Quan Wang and Ignacio Lopez Moreno, “Version control of speaker recognition systems,” *arXiv preprint arXiv:2007.12069*, 2020.

[6] Ondřej Glembek, Pavel Matějka, Oldřich Plchot, Jan Pešán, Lukáš Burget, and Petr Schwarz, “Migrating i-vectors between speaker recognition systems using regression neural networks,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech & Language*, vol. 60, pp. 101027, 2020.

[8] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig S Greenberg, Douglas A Reynolds, Elliot Singer, Lisa P Mason, Jaime Hernandez-Cordero, et al., “The 2016 nist speaker recognition evaluation.,” in *Interspeech*, 2017, pp. 1353–1357.

[9] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, “VoxCeleb: A Large-Scale Speaker Identification Dataset,” in *Interspeech 2017*, Hyderabad, India, 2017, pp. 2616–2620.

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, “VoxCeleb2: Deep Speaker Recognition,” in *Interspeech 2018*, Hyderabad, India, 2018, pp. 1086–1090.