Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

Information preservation pooling for speaker embedding

Min Hyun Han, Woo Hyun Kang, Sung Hwan Mun, and Nam Soo Kim

Department of Electrical and Computer Engineering and the Institute of New Media and Communications
Seoul National University, the Republic of Korea
mhhan@hi.snu.ac.kr, whkang@hi.snu.ac.kr, shmun@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

In the task of speaker recognition, pooling plays an important role in the speaker embedding extraction process, which is essentially summarizing inputs with variable length into a fixed dimensional output. One of the most popular pooling method for text-independent speaker verification is the attention-based pooling method which utilizes an attention mechanism to give different weights to each frame. Once the attention weights are calculated, the utterance-level feature is generated by computing weighted mean and standard deviation of the frame-level features. However, due to the compressive nature of the pooling operation, useful information in frame-level features can be compromised while extracting the speaker embedding. In this paper, we propose a information preservation pooling method that exploits a mutual information neural estimator to preserve local information in frame-level features during the pooling step. We conducted a set of experiments on the VoxCeleb datasets, and showed that the proposed method improves the speaker verification performance in terms of EER.

1. Introduction

Speaker verification is the task of verifying whether the claimed identity of an unknown speaker is true or not. Speaker verification generally can be divided into two steps. In the first step, fixed-dimensional utterance-level feature vectors called speaker embedding, which represent the speaker's characteristics, are extracted from the input speech signal and the enrolled speech signal, which may have different lengths. Then the verification score is derived by computing the similarity between the feature vectors of the input and enrolled speech. Over the last decade, the combination of i-vector [1] feature extractor and probabilistic linear discriminant analysis (PLDA) [2] scoring backend has become the one of the dominant approach for text-independent speaker recognition.

Recently, with the advance of deep learning technology, many studies have been conducted on the method of applying deep learning techniques to speaker recognition. In general, one major approach in deep learning-based speaker recognition is to extract the speaker embedding directly using deep neural architecture. In this approach, usually a neural network is trained to classify a target speaker given the input acoustic features (such as mel frequency cepstral coefficients, MFCCs). Once the network is trained, the activation from one of the internal hidden layer is extracted and used as a speaker embedding instead of the conventional embedding. Then a scoring metric such as cosine similarity or PLDA is used to compute the verification score.

The main objective of speaker embedding is to map acoustic features of various duration to a single fixed dimensional vector. In [3], the last output of the long short-term memory (LSTM) is used as speaker embedding for text-dependent speaker verification. For text-independent speaker verification system, an average pooling layer has been introduced to aggregate the frame-level features of variable length to obtain the utterance-level speaker feature vector [4]. Snyder et al. [5] proposed an extension of average pooling called statistics pooling in which not only the means but also the standard deviations of the frame-level features are calculated. In [6], even higher-order statistics were used for text-independent speaker verification. Meanwhile, in [7], they proposed to use dictionary-based NetVLAD layer [8] to aggregate information into fixed-sized representation.

Other studies have focused on applying the attention mechanism to the pooling method for speaker embedding [9, 10, 11]. In attention-based pooling scheme, different weights are given to each frame. The attention weights, which denote the importance of each corresponding frame, are calculated by an additional neural network called attention network. With the calculated weights, weighted mean and variance of the frame-level features are computed. By giving different weights to each frame, the embedding system can focus on frames with more speaker-discriminative information. The attention network is trained jointly with the entire system without the need for any additional loss functions. However, weighted mean pooling can be viewed as a process of smoothing the frame-level features. In the pooling step, some useful information in the frame-level features can be lost.

Mutual information is a fundamental quantity that measures mutual dependence between random variables. Although mutual information is a pivotal quantity in data science, the exact computation is intractable especially for continuous and high-dimensional random variables. Recently, Belghazi et al. [12] proposed to estimate and maximize mutual information using deep neural network by exploiting the lower-bound of mutual information (mutual information neural estimator, MINE). Moreover, there were some remarkable strategies to learn representations by MINE in an unsupervised manner [13, 14, 15]. Especially in [13], they proposed to learn useful representation by maximizing the mutual information between local and global features.

In this paper, we propose a MINE-based approach to maximize the mutual information between the frame-level features and the utterance-level features. This can prevent information leakage, which can occur during the pooling step and potentially hinder the speaker verification performance. We experimented the proposed mutual information regularization technique with the x-vector baseline system on the VoxCeleb datasets, which is one of the most popular large-scale speaker recogntion dataset.

10.21437/Odyssey.2020-9

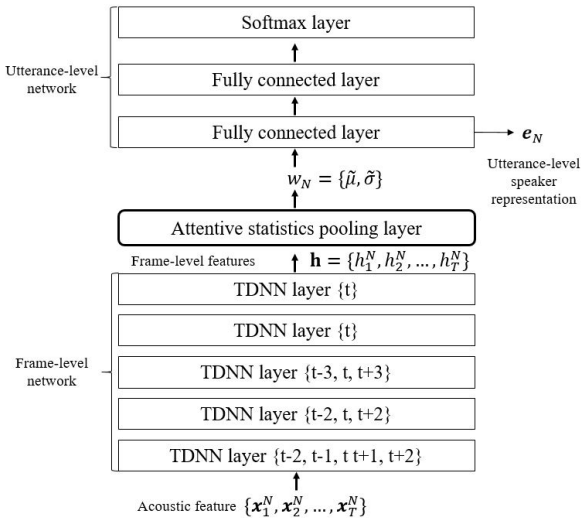

Figure 1: *X-vector baseline system*

2. Related works

2.1. X-vector baseline system

The architecture of our x-vector baseline system is based on the system described in [11]. As shown in Figure 1, the x-vector framework consists of frame-level network, pooling layer, and an utterance-level network.

The frame-level network comprises five time delay neural networks (TDNN) [16], which operates as a frame-level feature extractor. It takes a sequence of acoustic features, e.g., MFCCs and spectra, as an input and each layer considers the time context of adjacent frames.

The pooling layer aggregates the frame-level features of different length and converts them into a fixed dimensional vector. In this paper, we adopt the attentive statistics pooling layer [10] which is one of the most widely used pooling method. In the attentive pooling layer, attention score e_t is computed from m-dimensional frame-level feature h_t for each frame.

$$e_t = v^T f(Wh_t + b), \tag{1}$$

where $f(\cdot)$ indicates a non-linear activation function, such as hyperbolic tangent or ReLU function. W is $m' \times m$ matrix, v and b are m'-dimesnional vectors. A softmax function is applied to normalize the score to make the sum of the scores be unity:

$$\alpha_t = \frac{exp(e_t)}{\sum_\tau^T exp(e_\tau)}. \tag{2}$$

In the pooling layer, the weighted mean and the weighted standard deviation vector is calculated using the normalized score α_t as the attention weight for corresponding frame t.

$$\tilde{\mu} = \frac{1}{T}\sum_t^T \alpha_t h_t, \tag{3}$$

$$\tilde{\sigma} = \sqrt{\frac{1}{T}\sum_t^T h_t \odot h_t - \tilde{\mu} \odot \tilde{\mu}}, \tag{4}$$

where $\odot$ represents the Hadamard product.

These segment-level statistics are concatenated and passed to the utterance-level network, which consists of two additional hidden layers operating as an utterance-level feature extractor. The last layer is a softmax layer where each output node corresponds to a single speaker within the training dataset. The x-vector system is trained with cross entropy loss and back-propagation. Once the system is trained, the softmax layer is removed, and we use the activation from one of the hidden layers in the utterance-level network as a speaker embedding.

2.2. Mutual Information Estimation and Maximization

The success of deep learning technology can be attributed to its ability to learn useful representations from the input data which can be used for various downstream task. One intuitive way to extract information-rich representation using a neural network is to maximize the mutual information between its inputs and outputs. However, mutual information is hard to compute, especially in continuous and high-dimensional domain. Fortunately, recent studies showed that effective estimation and maximization of mutual information between high-dimensional input and output pair is possible using a deep learning-based technique called mutual information neural estimator.

2.2.1. Mutual Information

In probability theory, the mutual information between two random variables is a measure of the interdependency between two variables. The mutual information between two continuous random variables z_1 and z_2 is defined as below:

$$\begin{aligned} I(z_1; z_2) &= \int_{z_1}\int_{z_2} p(z_1, z_2)\log(\frac{p(z_1, z_2)}{p(z_1)p(z_2)})dz_1 dz_2 \\ &= D_{KL}(p(z_1, z_2)||p(z_1)p(z_2)), \end{aligned} \tag{5}$$

where D_{KL} is the Kullback-Leibler (KL) divergence. From equation 5, mutual information can be viewed as the KL-divergence between the joint distribution and the product of marginals. Mutual information is zero if and only if z_1 and z_2 are independent random variables.

2.2.2. Dual Representations of KL-divergence

Dual representations of the KL-divergence are the key element of MINE. The following theorem gives a useful representation of the KL-divergence, namely Donsker-Varadhan representation, which gives the lower bound to the mutual information. The Donsker-Varadhan representation of KL-divergence is

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^T] \tag{6}$$

where the supremum is taken over all functions T such that the two expectations are finite.

2.2.3. Mutual Information Neural Estimator (MINE)

The idea of mutual information neural estimator is to model a function T with a function parameterized by a deep neural network with parameters ω. In [12], it is proven that the mutual information can be estimated and maximized by maximizing its Donsker Varadhan bound.

$$I(z_1; z_2) := D_{KL}(\mathbb{J}||\mathbb{M}) \geq \hat{I}_\omega^{(DV)}(z_1; z_2), \tag{7}$$

$$\hat{I}_\omega^{(DV)}(z_1; z_2) := \mathbb{E}_{\mathbb{J}}[T_\omega(z_1, z_2)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_\omega(z_1, z_2)}], \tag{8}$$

where $T_\omega : z_1 \times z_2 \to \mathbb{R}$ is a discriminator function modeled by a neural network with parameters ω.

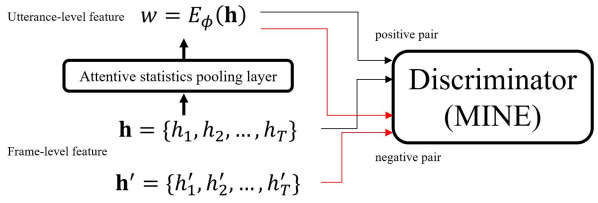

Figure 2: *Information preservation pooling - maximizing mutual information between the input/output pair of the pooling layer*

2.2.4. Mutual Information Maximization

The main goal of MINE is to maximize the mutual information between the input and output pairs of the encoder E_ϕ which is a neural network with parameters ϕ [13]. Mutual information maximization mostly relies on a sampling strategy, where the positive and negative samples are drawn from the joint and the product of marginal distributions, respectively [13, 14]. In general, the positive samples are collected from the same image or utterance, while the negative samples are obtained by randomly sampling from another image or utterance. Hence, We can optimize the mutual information with the MINE objective [12]:

$$L(\omega, \phi) = \mathbb{E}_{X_p}[(T_\omega(z, E_\phi(z)))] - log(\mathbb{E}_{X_n}[e^{T_\omega(z, E_\phi(z'))}]). \tag{9}$$

Since our main goal is to maximize the mutual information and isn't to concern its precise value, we can exploit different objective functions, like in [13, 14]. Binary cross-entropy (BCE) loss is one of the simplest solution adopted in [13, 14, 17, 18]:

$$L(\omega, \phi) = \mathbb{E}_{X_p}[log(T_\omega(z, E_\phi(z)))] + \mathbb{E}_{X_n}[log(1 - T_\omega(z, E_\phi(z')))], \tag{10}$$

$\mathbb{E}_{X_p}$ and $\mathbb{E}_{X_n}$ indicate the expectation over positive and negative samples, respectively. BCE estimates the Jensen-Shannon divergence between two random variables rather than the KL divergence. It assumes that the loss does not optimize the exact KL-based definition of mutual information but optimizes a similar divergence. Unlike the standard mutual information, because BCE is bounded, it makes the convergence of the architecture more numerically stable.

3. Proposed Method

In this section, we introduce a novel method of regularizing the embedding vector to have high mutual information with the frame-level features, to preserve the useful information in frame-level features. As shown in Figure 2, by adding an additional discriminator which takes a pair of frame-level and utterance-level features as an input, the mutual information between the two features are estimated and maximized with the MINE objective which can be viewed as a regularization term for the pooling layer.

3.1. Global mutual information maximization

For speaker verification, it is important to model an embedding with sufficient information on the input speech signal. One way to achieve this is to apply MINE to maximize the mutual information between the frame-level features and the utterance-level feature. In here, the two random variables will be the sequence of frame-level features $\mathbf{h} = \{h_1, h_2, ..., h_T\}$ and utterance-level feature w. The attentive statistics pooling layer E_ϕ is optimized jointly with the embedding system by estimating and maximizing $I(\mathbf{h}, E_\phi(\mathbf{h}))$. This framework is often called the Global Info Max (GIM) because it takes all frame-level features as input, and maximizes global information of all frame-level features [13].

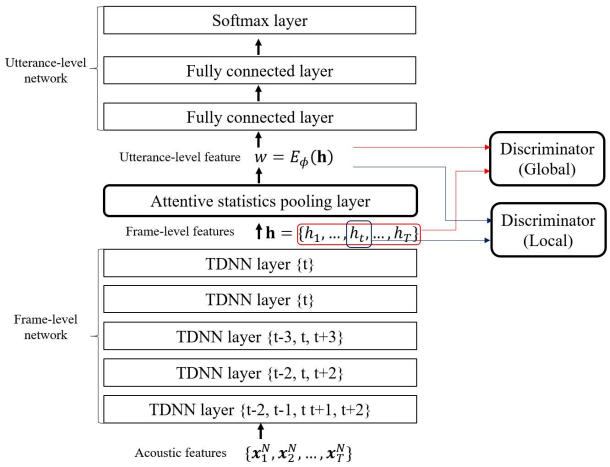

Figure 3: *Overall architecture of the proposed method*

$$(\hat{\omega}_G, \hat{\phi}) = \arg\max_{\omega_G, \phi} \hat{I}_{\omega_G}(\mathbf{h}; E_\phi(\mathbf{h})), \tag{11}$$

where subscript G denotes "global" and E_ϕ represents the pooling layer with the parameters ϕ.

3.2. Local mutual information maximization

The GIM aims to maximize the mutual information between the embedding and a group of multiple frame-level features. At the point of the discriminator, it is trained to classify whether the frame-level/utterance-level feature pair comes from the same utterance or not. In this way, only few frame-level features would be enough to take the right decision. Therefore, some useful information in the individual frame-level features can be ignored. To prevent this problem, we suggest an another objective to maximize the mutual information between the individual frame-level feature and the utterance-level feature. To achieve this, the discriminator is trained to classify with the single frame-level feature and the utterance-level feature pair. Using a single frame-level feature and utterance-level feature pair is similar to the Local Info Max (LIM) introduced in [13], but we utilize this with the opposite intent. In [13], they proposed the LIM to make not to focus on trivial pixel-level noise, optimizing the average mutual information between low-level features and high-level features.

However, since our aim is to focus on every frame-level features, we propose to optimize the mutual information between one frame-level feature and utterance-level feature, instead of optimizing an average of local mutual information. While training, we randomly sample a single frame-level feature to estimate and maximize the mutual information.

$$(\hat{\omega}_L, \hat{\phi}) = \arg\max_{\omega_L, \phi} \hat{I}_{\omega_L}(h_t; E_\phi(\mathbf{h})), \tag{12}$$

Table 1: *Model configuration*

Module		Configuration	Input size	Output size
Frame-level network		conv1d, kernel size=5, dilation rate=1, output nodes=512	T × 30	T-6 × 512
		conv1d, kernel size=3, dilation rate=2, output nodes=512	T-6 × 512	T-10 × 512
		conv1d, kernel size=3, dilation rate=3, output nodes=512	T-10 × 512	T-14 × 512
		conv1d, kernel size=1, dilation rate=1, output nodes=512	T-14 × 512	T-14 × 512
		conv1d, kernel size=1, dilation rate=1, output nodes=1536	T-14 × 512	T-14 × 1536
Attention network		fully connected, output nodes=512	T-14 × 1536	T-14 × 512
		fully connected, output nodes=1	T-14 × 512	T-14 × 1
Pooling layer		weighted mean, variance pooling	T-14 × 1536	1 × 1536*2
Utterance-level network		fully connected, output nodes=512	1 × 1536*2	1 × 512
		fully connected, output nodes=512	1 × 512	1 × 512
Discriminator (Global)	Dimension reduction network	fully connected, output nodes=128	T-12 × 1536	T-12 × 128
		fully connected, output nodes=64	T-12 × 128	T-12 × 64
	Discriminator	fully connected, output nodes=256	1 × (T-12)*64	1 × 256
		fully connected, output nodes=1	1 × 256	1 × 1
Discriminator (Local)		fully connected, output nodes=64	1 × (1536+3072)	1 × 64
		fully connected, output nodes=1	1 × 64	1 × 1

where subscript L denotes "local" and h_t is a single frame-level feature randomly sampled.

3.3. Information preservation pooling

The GIM and LIM can be applied together when training the embedding system. We call it information preservation pooling, whose objectives are optimized jointly with the speaker classification loss during training. The entire objective function can be described:

$$\begin{aligned} L_{total} &= L_S + \alpha L_G(\omega_G, \phi) + \beta L_L(\omega_L, \phi) \\ &= \sum_{i=1}^{M} \log \frac{e^{W_{y_i}^T e_i + b_{y_i}}}{\sum_{j=1}^{N} e^{W_j^T e_i + b_j}} + \alpha \hat{I}_{\omega_G}(\mathbf{h}; E_\phi(\mathbf{h})) \\ &\qquad + \beta \hat{I}_{\omega_L}(h_t; E_\phi(\mathbf{h})). \end{aligned} \tag{13}$$

The first term L_S is the speaker classification loss (softmax cross entropy). Second and third term $L_G(\omega_G, \phi), L_L(\omega_L)$ are the global and local MINE objectives (ω_G and ω_L are the parameters for the global and local discriminators). These MINE objective operates as a regularization term with weight α and β while training the entire system.

4. Experiments

4.1. Experimental settings

4.1.1. Datasets

The experiments are conducted on the VoxCeleb 1 and 2 dataset [19]. We followed the training and evaluation protocols provided in [19]. The development set of the VoxCeleb1 and VoxCeleb2 datasets, which comprises 1,251 and 5,994 speakers respectively. For evaluation, we used the original trial list of VoxCeleb1 which consists of 40 speakers.

4.1.2. Input features

The acoustic features used in the experients were 30-dimensional MFCC feature extracted using a 25-ms hamming window with a 10-ms shift. During training, each utterance was truncated to a 2.5 seconds segment, resulting in 30-dimensional vector sequence of 250 frames. Mean and variance normalization was applied over each segment. No voice activity detection(VAD), automatic silence removal, or any kind of data augmentation was applied.

4.1.3. Model configuration

The details of the model structure is described in Table 1. We constructed the embedding network based on [10], namely the attentive statistics pooling (ASP) x-vector system. Five TDNNs were used for frame-level feature extractor. The number of output nodes in each TDNN layer was 512, except for the last TDNN layer having 1536 output nodes. The kernel size of each layer was set to be {5, 3, 3, 1, 1}, and the dilation rate was {1, 2, 3, 1, 1}.

For the attention model, the number of hidden nodes was 512 and the output was a 1-dimensional scalar. The number of hidden nodes for the fully connected layers constructing the utterance-level network was 512.

For the GIM network, the dimensionality of the frame-level features and utterance-level mean/variance vector were reduced to 64 dimension. The dimension-reduced vectors were concatenated and passed into the GIM discriminator network, which consists of one hidden layer with 512 output nodes. The LIM network is composed of one hidden layer with 64 nodes and the single output layer.

For the activation functions in all hidden layers, leakyReLU function was used, and batch normalization was applied after passing the activation. L2-regularization was applied to every weight and bias.

4.1.4. Training details

At each training step, 128 utterances were sampled randomly. To make input for MINE network, we concatenated the segment-level feature to its frame-level features to make positive pair, while concatenating segment-level feature to other utterance's frame-level features for making negative pair.

We used the Adam optimizer with an initial learning rate of 10^{-3}, and exponentially decreased learning rate every epoch until final rate of 10^{-8} for 128 epochs. The neural network implementation was done using Tensorflow [20].

4.1.5. Backend scoring metric

After training the speaker embedding network, cosine similarity and PLDA were used as a scoring backend. When using

cosine similarity, the activation of the last hidden layer was used as speaker embedding, which showed better performance. When using PLDA, the output of the second-to-the-last layer in utterance-level network was extracted as speaker embedding. Linear Discriminant Analysis (LDA) was applied to reduce the embedding dimension to 200 followed by length normalization and whitening. Then, PLDA was used to compute the verification scores. The results were shown in terms of equal error rate (EER).

4.2. Results

4.2.1. Global mutual information maximization

Table 2 shows the performance of the x-vector systems trained with GIM on VoxCeleb1. The proposed method outperformed the baseline. When the coefficient for GIM is too small ($\alpha = 0.001$), the GIM loss had little effect on the system performance. However, large GIM loss ($\alpha = 1$) rather degraded the system. Using two kinds of scoring backends, cosine similarity and PLDA, the proposed method improves the system performance from the baseline by 3.6% and 5.7% in the best case, respectively

Table 2: *Performance on VoxCeleb1 test set, applied global information maximization regularization (trained with VoxCeleb1 dev set)*

system	backend	EER (%)
x-vector baseline (ASP) (our implementation)	cosine sim.	6.74
	PLDA	5.66
ASP with GIM ($\alpha = 1$)	cosine sim.	7.24
	PLDA	5.55
ASP with GIM ($\alpha = 0.1$)	cosine sim.	6.82
	PLDA	**5.34**
ASP with GIM ($\alpha = 0.01$)	cosine sim.	**6.50**
	PLDA	5.39
ASP with GIM ($\alpha = 0.001$)	cosine sim.	6.57
	PLDA	5.57

4.2.2. Local mutual information maximization

From Table 3, it can be clearly see that LIM loss can improve the verification performance. The LIM loss showed similar tendency with the GIM loss, but we can see that overall performance is slightly better than the performance of GIM loss. In the best case, the system performance is improved from the baseline by 7.6% and 9.7%, using cosine similarity and PLDA, respectively.

4.2.3. Information preservation pooling

In this section, we implemented the proposed information preservation pooling in equation 13, in various hyperparameter settings. Table 4 shows the results on the VoxCeleb1 dataset. For the scoring backend, cosine similarity was used. In the best case, the proposed method reduces the error from the baseline by 8.9%.

In order to verify the proposed method in larger data set,

Table 3: *Performance on VoxCeleb1 test set, applied local information maximization regularization (trained with VoxCeleb1 dev set)*

system	backend	EER (%)
ASP with LIM ($\beta = 1$)	cosine sim.	6.31
	PLDA	5.18
ASP with LIM ($\beta = 0.1$)	cosine sim.	6.44
	PLDA	**5.11**
ASP with LIM ($\beta = 0.01$)	cosine sim.	**6.23**
	PLDA	5.29
ASP with LIM ($\beta = 0.001$)	cosine sim.	6.71
	PLDA	5.34

Table 4: *EERs (%) on VoxCeleb1 test set, for various hyperparameter case using information preservation pooling (trained with VoxCeleb1 dev set)*

	α = 0.001	α = 0.005	α = 0.01
β = 0.01	6.37	6.54	6.42
β = 0.05	6.64	6.37	6.31
β = 0.1	6.26	6.32	**6.14**

we conducted an experiment using the VoxCeleb2 dataset. In this configuration, we used VoxCeleb2 development set, which contains 5994 speakers, as a training set, and the system was evaluated on the same VoxCeleb1 trial set.

From Table 5, we can see that the proposed method can further improve the system performance in large dataset. In both cases of using cosine similarity and PLDA, our proposed method reduces the error by 23.2% and 14.6%, respectively. Figure 3 depicts the detection error tradeoff (DET) curves for the baseline and the proposed method. In almost all thresholds, our proposed method outperformed the baseline x-vector system.

Table 5: *Performance on VoxCeleb1 test set using information preservation pooling (trained with VoxCeleb2 dev set)*

system	backend	EER (%)
x-vector baseline (ASP) (our implementation)	cosine sim.	6.50
	PLDA	3.62
Information preservation pooling ($\alpha = 0.01, \beta = 0.1$)	cosine sim.	**4.99**
	PLDA	**3.09**

5. Conclusion

In this paper, we proposed a novel information preservation regularization technique to improve the speaker embedding performance. The proposed method utilizes a MINE to maximize the mutual information between the frame-level features and the utterance-level feature. This enables the utterance-level feature to preserve useful information within the frame-level features.

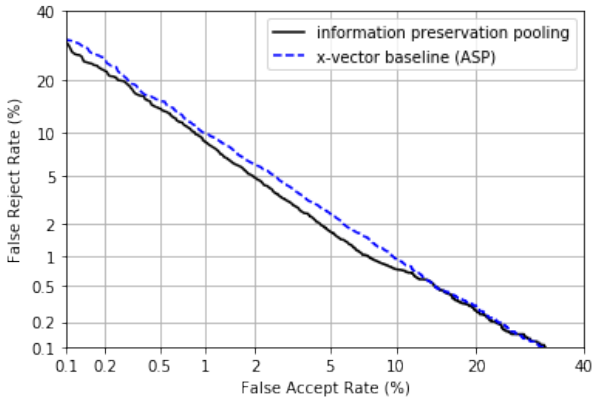

Figure 4: *DET curve (VoxCeleb2)*

The new methods were evaluated on the VoxCeleb dataset, which is one of the most commonly used open datasets. In the experiment, we investigated the proposed losses on various settings. Results from the text-independent speaker verification experiments showed that the proposed method can improve the verification performance, achieving a relative improvement of 14.6%, in terms of EER. This shows that the proposed regularization technique can improve the performance, by selecting appropriate hyperparameters.

In our future research, we will apply the proposed method to other pooling methods, such as NetVLAD and spatial pyramid pooling. More efforts will be made to combine the proposed method with other losses like margin softmax loss. Furthermore, we will also modify the training strategy to obviate the need of adjusting the hyperparameters.

6. Acknowledgement

This research was supported by Projects for Research and Development of Police science and Technology under Center for Research and Development of Police science and Technology and Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (PA-J000001-2017-101).

7. References

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

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

[3] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam 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.

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier 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.

[5] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017, pp. 999–1003.

[6] Lanhua You, Wu Guo, Lirong Dai, and Jun Du, "Multi-task learning with high-order statistics for x-vector based text-independent speaker verification," in *Proc. Interspeech*, 2019, pp. 1158–1162.

[7] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5791–5795.

[8] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[9] FA Rezaur rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[10] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[11] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification.," in *Interspeech*, 2018, pp. 3573–3577.

[12] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm, "Mutual information neural estimation," in *International Conference on Machine Learning*, 2018, pp. 531–540.

[13] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[14] Mirco Ravanelli and Yoshua Bengio, "Learning speaker representations with mutual information," *arXiv preprint arXiv:1812.00271*, 2018.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[16] Vijayaditya Peddinti, Daniel Povey, and Sanjeev 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.

[17] Philemon Brakel and Yoshua Bengio, "Learning independent features with adversarial nets for non-linear ica," *arXiv preprint arXiv:1710.05050*, 2017.

[18] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[19] 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.

[20] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.