AN ITERATIVE FRAMEWORK FOR SELF-SUPERVISED DEEP SPEAKER REPRESENTATION LEARNING

Danwei Cai[⋆]*, Weiqing Wang*[⋆]*, Ming Li*[⋆†]

[⋆]Department of Electrical and Computer Engineering, Duke University, Durham, USA
[†]Data Science Research Center, Duke Kunshan University, Kunshan, China
ming.li369@duke.edu

ABSTRACT

In this paper, we propose an iterative framework for self-supervised speaker representation learning based on a deep neural network (DNN). The framework starts with training a self-supervision speaker embedding network by maximizing agreement between different segments within an utterance via a contrastive loss. Taking advantage of DNN's ability to learn from data with label noise, we propose to cluster the speaker embedding obtained from the previous speaker network and use the subsequent class assignments as pseudo labels to train a new DNN. Moreover, we iteratively train the speaker network with pseudo labels generated from the previous step to bootstrap the discriminative power of a DNN. Speaker verification experiments are conducted on the VoxCeleb dataset. The results show that our proposed iterative self-supervised learning framework outperformed previous works using self-supervision. The speaker network after 5 iterations obtains a 61% performance gain over the speaker embedding model trained with contrastive loss.

Index Terms— speaker recognition, speaker embedding, self-supervised learning, contrastive learning, clustering

1. INTRODUCTION

Speaker recognition refers to identify or verify a claimed speaker by analyzing the given speech from that speaker. Over the past few years, supervised deep learning methods greatly improve the performance of speaker recognition system [1, 2, 3]. These methods require large-scale datasets to learn discriminative speaker representations. However, manually annotating speaker labels for a large scale dataset may sometimes be expensive and problematic. On the other hand, there are vast numbers of unlabeled speech data that can be used for training DNNs. With self-supervision methods, deep learning can automate the labeling process and benefit from massive amounts of data.

In visual representation learning, most self-supervised methods fall into two classes: generative or discriminative. The generative approaches directly model the pixels of input images and do reconstruction [4, 5]. However, learning a full generative model in pixel-level may be computationally expensive for representation learning. Recently, discriminative approaches based on contrastive learning emerge and show promising results in visual representation learning [6, 7, 8].

In self-supervised speaker representation learning, Stafylakis *et al.* [9] propose a generative method to learn speaker embedding with the help of a phone decoder network. Ravanelli *et al.* [10] propose to learn speaker representation by maximizing the mutual information between two speech segments within the same utterance. Discriminative approaches based on contrastive learning have recently shown promising results [11, 12]. In particular, Huh *et al.* [12] propose to use a strong data augmentation strategy to improve the generalizability of the learned speaker representation and apply augmentation adversarial training to remove the channel noise caused by the augmentation applied. Audio-visual application is also becoming increasingly popular in self-supervised representation learning. It take advantage of this multi-modal information, i.e., images and sound, of the video data to learn multi-modality based identity representation [13, 14].

When learning speaker representation without supervision, contrastive learning based methods assume that speech segments from different utterances belong to different speakers. This assumption naturally introduces label error when training the network. Even if the network learns some discriminative information about speaker identities, this label error might drive the speaker network to learn the opposite. In this paper, we propose an iterative, self-evolving framework that learns to avoid this kind of label error. It starts with training a speaker embedding network using contrastive self-supervised learning and generates pseudo labels of the training data with this network. The framework then iteratively trains the speaker network with the pseudo labels and generates new labels using the new converged network. The idea behind the proposed framework is simple: to take advantage of the DNN's ability to learn from data with label noise and bootstrap its discriminative power. This idea is similar to the deep clustering method [15] in unsupervised visual features learning, which performs clustering every few epochs. In this paper, we perform clustering whenever the network is

converged.

This work is partly motivated by our previous study on unsupervised speaker recognition [16], which is based on the traditional i-vector/PLDA pipeline. It iteratively optimizes the probabilistic linear discriminant analysis (PLDA) scoring backend with the fixed speaker representation of i-vectors. In this paper, the speaker representation is learned discriminatively using the generated pseudo labels at each iteration.

2. METHODS

This section describes the proposed iterative framework for self-supervised speaker embedding learning.

- Step 1 (initial round): Train a speaker embedding network with contrastive self-supervised learning.
- Step 2: With the previous speaker embedding network, extract speaker embeddings for the whole training data. Perform a clustering algorithm on the embeddings to generate pseudo labels.
- Step 3: Train the speaker embedding network with a classification layer and cross-entropy loss using the generated pseudo labels.
- Repeat step 2 and step 3 with limited rounds. Use the last speaker embedding network as the final model.

2.1. Contrastive Self-supervised Learning

We employ the contrastive self-supervised learning (CSL) framework similar to the framework in [8, 17]. Let $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ be an unlabeled dataset with N utterances, CSL assumes that each data sample defines its own class. During training, we randomly sample a mini-batch $\mathcal{B} = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ of M utterances from $\mathcal{D}$. For each utterance $\mathbf{x}_i$, two segments $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}$ are randomly taken and considered as a positive pair. This produces $2M$ data samples in total. Stochastic data augmentation is then performed on these data samples and the speaker embedding $\mathbf{z}_{i,j}$ are extracted as

$$\mathbf{z}_{i,j} = f_\theta(\mathrm{aug}(\mathbf{x}_{i,j})),\ j \in \{1, 2\} \tag{1}$$

where $f_\theta(\cdot)$ is the speaker embedding network with parameters θ, and aug represents the stochastic data augmentation process: a reverberation noise and (or) additive background noise are (is) added to the clean segment. We adapt the contrastive loss from SimCLR [8] as:

$$\mathcal{L}_{\mathrm{CSL}} = \frac{1}{2M}\sum_{i=1}^{M}(l_{i,1} + l_{i,2}) \tag{2}$$

$$l_{i,j} = -\log \frac{\exp(\cos(\mathbf{z}_{i,1}, \mathbf{z}_{i,2})/\tau)}{\sum_{k=1}^{M}\sum_{l=1}^{2} \mathbb{1}_{\substack{k\neq i \\ l\neq j}} \exp(\cos(\mathbf{z}_{i,j}, \mathbf{z}_{k,l})/\tau)} \tag{3}$$

where $\mathbb{1}$ is an indicator function evaluating 1 when $k \neq i$ and $l \neq j$, $\cos$ denotes the cosine similarity and τ is a temperature parameter to scale the similarity scores. $l_{i,j}$ can be interpreted as the loss for ***anchor*** feature $\mathbf{z}_{i,j}$. It computes positive score for ***positive*** feature $\mathbf{z}_{i,(j+1)\mathrm{mod}2}$ and negative scores across all $2(M-1)$ ***negative*** pairs $\{\mathbf{z}_{k,j} | k \neq i, j = 1, 2\}$.

2.2. Generating Pseudo Labels by Clustering

2.2.1. k-means clustering

Given the speaker embeddings of the training data, we employ a clustering algorithm to generate cluster assignments. In this paper, we use the well-know *k*-means because of its simplicity, fast-speed and capability with large dataset. Let the speaker embedding in d-dimensional feature space $\mathbf{z} \in \mathbb{R}^d$, *k*-means learns a centroid matrix $\mathbf{C} \in \mathbb{R}^{d\times k}$ and the cluster assignment $y_i \in \{1, \cdots, k\}$ for each speaker embedding $\mathbf{z}_i$ with the following learning objectives

$$\min_{\mathbf{C}} \frac{1}{N}\sum_{i=1}^{N}\min_{y_i} \|\mathbf{z}_i - \mathbf{C}_{y_i}\|_2^2 \tag{4}$$

where $\mathbf{C}_{y_i}$ is the y_i^{th} column of the centroid matrix $\mathbf{C}$. The optimal assignments $\{y_1, \cdots, y_N\}$ are used as pseudo labels.

2.2.2. Purifying pseudo labels

One problem with the generated pseudo labels is the massive label noise. To mitigate this problem, we apply the following simple steps to purify the generated pseudo labels. (*a*) By defining the clustering confidence as $-\|\mathbf{z}_i - \mathbf{C}_{y_i}\|_2^2$ for each speaker embedding $\mathbf{z}_i$, we filter out p portion of the remaining data with least clustering confidence. (*b*) To further reduce the possibility that one actual speaker appears in several pseudo clusters, we only keep the pseudo clusters with at least S samples.

The level of label noise is a trade-off between the remaining data's size and the purifying process's intensity. With a more aggressive purifying process, the remaining training data's size becomes smaller, and the level of label noise is reduced. In the first few iterations, we apply an aggressive purifying process to the pseudo labels, which keeps 30% to 50% of training data. As the speaker model becomes more discriminative, we relax the purifying process and allow more data to train the network.

2.3. Learning with Pseudo Labels

After the pseudo labels purifying process, the remaining training dataset $\mathcal{D}' = \{\mathbf{x}_1, \cdots, \mathbf{x}_{N'}\}$ contains N' utterances. With the generated pseudo labels $\{y_1, \cdots, y_{N'}\}$, the speaker embedding network can be discriminatively trained with a parametrized classifier $g_W(\cdot)$ which predicts the labels for

the speaker embedding $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$. The parameters $\{\theta, W\}$ are jointly trained with the cross-entropy loss:

$$\mathcal{L}_{\text{spk}} = -\sum_{i=1}^{N'} \log \frac{\exp(g_{W\,y_i}(\mathbf{z}_i))}{\sum_{j=1}^{K} \exp(g_{W\,j}(\mathbf{z}_i))} \tag{5}$$

where $g_{W\,j}(\mathbf{z}_i)$ is the j^{th} element ($j \in [1, K]$) of the class score vector $g_W(\mathbf{z}_i)$, K is the number of the pseudo class.

2.4. Network Architecture

We opt for a residual convolutional network (ResNet) to learn speaker representation from the spectral feature sequence of varying length [2, 18]. The ResNet's output feature maps are aggregated with a global statistics pooling layer, which calculates means and standard deviations for each feature map. A fully connected layer is employed afterward to extract the 128-dimensional speaker embedding.

3. EXPERIMENTS

3.1. Dataset

The experiments are conducted on the development set of Voxceleb 2, which contains 1,092,009 utterances from 5,994 speakers [19]. Speaker labels are not used in the proposed method.

For evaluation, the development set and test set of Voxceleb 1 are used [20]. We report the speaker verification results on 3 trial sets as defined in [19]: the *original test set* of Voxceleb 1 containing 37,720 trials from 40 speakers, the *Voxceleb 1-E* test set (using the entire dataset) containing 581,480 trials from 1251 speakers, the *Voxceleb 1-H* test set (within the same nationality and gender) containing 552,536 trials from 1190 speakers.

3.2. Data Augmentation

Data augmentation is proven to be an effective strategy for both conventional learning with supervision [21] and contrastive self-supervision learning [11, 12, 8] in the context of deep learning. We perform data augmentation with MUSAN dataset [22]. The noise type includes ambient noise, music, television, and babble noise for the background additive noise. The television noise is generated with one music file and one speech file. The babble noise is constructed by mixing three to eight speech files into one. For the reverberation, the convolution operation is performed with 40,000 simulated room impulse responses (RIR) in MUSAN. We only use RIRs from small and medium rooms.

With contrastive self-supervised learning, three augmentation types are randomly applied to each training utterance: applying only noise addition, applying only reverberation, and applying both noise and reverberation. The signal-to-noise ratios (SNR) are set between 5 to 20 dB.

Table 1. Clustering and label purifying results. $\{p, S\}$ are hyper-parameters defined in section 2.2.2. The number of remaining utterance after the label purifying process is presented. The normalized mutual information (NMI) values before and after the purifying process are also reported.

Model	p	S	#utterances	NMI
CSL	0.6	8	347,625	0.8162 $\rightarrow$ 0.9381
Round 1	0.4	10	631,408	0.8669 $\rightarrow$ 0.9404
Round 2	0.4	10	644,692	0.8940 $\rightarrow$ 0.9603
Round 3	0.4	6	733,865	0.9114 $\rightarrow$ 0.9638
Round 4	0.3	6	843,770	0.9231 $\rightarrow$ 0.9618

When training with pseudo labels, either background noise or reverberation noise is added to the clean utterances with a probability of 0.6. The SNR is randomly set between 0 to 20 dB.

3.3. Implementation Details

3.3.1. Contrastive self-supervised learning setup

For feature extraction, we choose a 40-dimensional log Mel-spectrogram with a 25ms Hamming window and 10ms shifts. The duration between 2 to 4 seconds is randomly generated for each data batch.

We use the same network architecture as in [21]. ReLU non-linear activation and batch normalization are applied to each convolutional layer in ResNet. Network parameters are updated using Adam optimizer [23] with an initial learning rate of 0.001 and a batch size of 256. The temperature τ in equation (3) is set as 0.1.

3.3.2. Clustering setup

The cluster number is set to 6,000 for *k*-means. In Voxceleb 2, the audio segments are obtained with the self-supervised SyncNet [19] from videos. We take advantage of this segment information and average the speaker embeddings from the same video. The *k*-means clustering is performed on the averaged embeddings for the sake of clustering efficiency.

3.3.3. Setup for models learned with pseudo labels

For the network learned with pseudo labels, we use an 80-dimensional log Mel-spectrogram as input features. A duration between 3 to 4 seconds is randomly generated for each data batch. The network architecture is the same as the one used in CSL but with double feature map channels. Dropout is added before the speaker classification layer to prevent overfitting [24]. Network parameters are updated using stochastic gradient descent (SGD) algorithm. The learning rate is initially set to 0.1 and is divided by 10 whenever the training loss reaches a plateau.

Table 2. Speaker verification performance (minDCF and EER[%]). The utterance and speaker number of the training data are presented. The NMIs of the pseudo labels for each iteration are also reported.

Model	#Utterances	#Clusters	NMI	Voxceleb 1 test		Voxceleb 1-E		Voxceleb 1-H	
Fully Supervised	1,092,009	5,994	1	0.097	1.51	0.102	1.59	0.178	3.00
Nagrani *et al.* [13]	1,092,009	-	-	-	22.09	-	-	-	-
Chung *et al.* [14]	1,092,009	-	-	-	17.52	-	-	-	-
Inoue *et al.* [11]	1,092,009	-	-	-	15.26	-	-	-	-
Huh *et al.* [12]	1,092,009	-	-	0.454	8.65	-	-	-	-
Initial round (CSL)	1,092,009	-	-	0.508	8.86	0.570	10.15	0.710	16.20
Round 1	347,625	2,839	0.9381	0.429	6.96	0.433	7.91	0.561	11.73
Round 2	631,408	4,776	0.9404	0.341	5.42	0.358	6.22	0.479	9.60
Round 3	644,692	4,708	0.9603	0.300	4.73	0.316	5.29	0.433	8.17
Round 4	733,865	5,018	0.9638	0.263	4.16	0.278	4.55	0.391	7.39
Round 5	843,770	5,407	0.9618	0.241	3.45	0.246	4.02	0.363	6.57

3.4. Experimental Results

We use normalized mutual information (NMI) as a measurement of clustering quality. NMI measures the information shared between the true speaker labels U and the predictive pseudo labels V. It is defined as:

$$\mathrm{NMI}(U, V) = \frac{2 \times I(U; V)}{H(U) + H(V)} \tag{6}$$

where $I(U; V)$ is the mutual information between U and V, and $H(\cdot)$ denotes entropy. When two label assignments that are largely independent, NMI becomes 0. When they are in significant agreement, NMI equals to 1. Table 1 shows the NMI before and after the label purifying process for each round. We apply an aggressive purifying process in the first round and relax the process to allow more data to train the network in the following rounds. The original NMIs before the process increase as the iteration increase, which indicates the speaker embedding network becomes discriminative. Also, the increased NMI after the label purifying process demonstrates the effectiveness of the process.

For the speaker verification experiments, cosine similarity is used for scoring at the test stage. We use equal error rate (EER) and minimum detection cost function (minDCF) as the performance metric. The reported minDCF is evaluated with $P_{\mathrm{target}} = 0.05, C_{\mathrm{miss}} = C_{\mathrm{fa}} = 1$. Table 2 reports the experimental results. In round 1, with only 32% of the training data and noisy pseudo labels, the speaker model outperforms the one trained with CSL by 21.4% in terms of EER. Results in table 2 also show that the performance of the speaker verification system keeps improving with the increase of round number, which demonstrates the effectiveness of the proposed iterative self-supervised learning framework. By comparing round 4 and round 5 to their previous rounds, we observed that the enlarged training data contributes to the performance improvements. We also see that although round 3 and round 2 has similar training data size (645k and 631k utterances), round 3 achieves a 12.8% EER reduction comparing to round 2. This performance improvement comes from the improved clustering quality: the NMI is 0.94 for round 2's training data and increases to 0.96 for round 3.

3.5. Discussion

Since the pseudo labels generated in each round do not share cluster indexes, the speaker embedding network is trained from scratch in each round, and it takes longer time to train the model as training data increase. We argue that our proposed framework could benefit from the metric learning objectives, which assumes an open-set setting with unseen speakers [3]. With metric learning objectives, the speaker model at each round could continue learning from the last round with the newly generated pseudo labels.

To mitigate the adverse effects caused by the pseudo label noise, we use a simple label purifying process to exclude the data sample with little clustering confidence in this work. The more sophisticated solution may come from the deep learning solution overcoming label noise, such as curriculum loss [25], label smoothing [26], and so on.

4. CONCLUSION

In this paper, we proposed an iterative framework for self-supervision speaker recognition. It consists of an initial round of contrastive self-supervised learning and several rounds of discriminative training with pseudo labels obtained by a clustering algorithm. The framework exploits DNN's ability to learn from data with noisy label and improves the model's discriminative power iteratively. Experimental results on Voxceleb show that our proposed framework improves speaker verification systems' performance compared to purely contrastive self-supervision learning.

5. REFERENCES

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "x-vectors: Robust DNN Embeddings for Speaker Recognition," in *ICASSP*, 2018, pp. 5329–5333.

[2] W. Cai, J. Chen, and M. Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Speaker Odyssey*, 2018, pp. 74–81.

[3] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[4] R. Zhang, P. Isola, and A. Efros, "Colorful Image Colorization," in *ECCV*, 2016, pp. 649–666.

[5] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434*, 2015.

[6] A. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv:1807.03748*, 2018.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *CVPR*, 2020, pp. 9729–9738.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," *arXiv:2002.05709*, 2020.

[9] T. Stafylakis, J. Rohdin, O. Plchot, P. Mizera, and L. Burget, "Self-Supervised Speaker Embeddings," in *Interspeech*, 2019, pp. 2863–2867.

[10] M. Ravanelli and Y. Bengio, "Learning Speaker Representations with Mutual Information," in *Interspeech*, 2019, pp. 1153–1157.

[11] N. Inoue and K. Goto, "Semi-Supervised Contrastive Learning with Generalized Contrastive Loss and Its Application to Speaker Recognition," *arXiv:2006.04326*, 2020.

[12] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and Joon S. Chung, "Augmentation Adversarial Training for Unsupervised Speaker Recognition," *arXiv:2007.12085*, 2020.

[13] A. Nagrani, J. S. Chung, S. Albanie, and A. Zisserman, "Disentangled Speech Embeddings Using Cross-Modal Self-Supervision," in *ICASSP*, 2020, pp. 6829–6833.

[14] S. W. Chung, H. G. Kang, and J. S. Chung, "Seeing Voices and Hearing Voices: Learning Discriminative Embeddings Using Cross-Modal Self-Supervision," *arXiv:2004.14326*, 2020.

[15] H. Song, M. Kim, D. Park, and J. Lee, "Learning from Noisy Labels with Deep Neural Networks: A Survey," in *ECCV*, 2020.

[16] W. Liu, Z. Yu, and M. Li, "An Iterative Framework for Unsupervised Learning in the PLDA based Speaker Verification," in *ISCSLP*, 2014, pp. 78–82.

[17] W. Falcon and K. Cho, "A Framework For Contrastive Self-Supervised Learning And Designing A New Approach," *arXiv:2009.00104*, 2020.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016, pp. 770–778.

[19] J. Son Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep Speaker Recognition," in *Interspeech*, 2018, pp. 1086–1090.

[20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A Large-Scale Speaker Identification Dataset," in *Interspeech*, 2017, pp. 2616–2620.

[21] D. Cai, W. Cai, and M. Li, "Within-Sample Variability-Invariant Loss for Robust Speaker Recognition Under Noisy Environments," in *ICASSP*, 2020, pp. 6469–6473.

[22] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484*, 2015.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] Y. Lyu and I. W. Tsang, "Curriculum Loss: Robust Learning and Generalization Against Label Corruption," in *ICLR*, 2020.

[26] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *ICLR*, 2017.