ENSEMBLE ADDITIVE MARGIN SOFTMAX FOR SPEAKER VERIFICATION

Ya-Qi Yu, Lei Fan, Wu-Jun Li

National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
{yuyq,fanl}@lamda.nju.edu.cn, liwujun@nju.edu.cn

ABSTRACT

End-to-end speaker embedding systems have shown promising performance on speaker verification tasks. Traditional end-to-end systems typically adopt softmax loss as training criterion, which is not strong enough for training discriminative models. In this paper, we adapt the additive margin softmax (AM-Softmax) loss, which is originally proposed for face verification, to speaker embedding systems. Furthermore, we propose a novel ensemble loss, called ensemble additive margin softmax (EAM-Softmax) loss, for speaker embedding by integrating Hilbert-Schmidt independence criterion (HSIC) into the speaker embedding system with the AM-Softmax loss. Experiments on a large-scale dataset VoxCeleb show that AM-Softmax loss is better than traditional loss functions, and approaches using EAM-Softmax loss can outperform existing speaker verification methods to achieve state-of-the-art performance.

Index Terms— Speaker verification, additive margin softmax, ensemble, Hilbert-Schmidt independence criterion

1. INTRODUCTION

Recently, demands for high-precision speaker verification (SV) technology increase quickly in security domain, because SV has great potential with a low requirement for collecting devices and operating environment. The task of SV systems is to verify whether a given utterance matches a specific speaker, whose characteristic can be extracted from enrollment utterances recorded in advance. The characteristic of an utterance is typically represented as an embedding vector, which is calculated by speaker embedding systems.

For the last decade, approaches based on i-vectors [1], which represent speaker and channel variability in a low dimensional space called total variability space, have dominated the field of speaker embedding. Nevertheless, there is a paradigm shift in recent speaker embedding studies, from i-vector to deep neural networks (DNN) [2, 3, 4], mostly with end-to-end training. The difference between i-vector and end-to-end systems is that i-vector adopts generative models for embedding but end-to-end systems adopt DNN for embedding. In end-to-end systems, we generally use an intermediate layer of neural networks as the embedding layer instead of the last layer or 'classification' layer, because the intermediate layer appears to be more robust in open-set tasks. To complete speaker verification, the speaker embeddings, either learned by end-to-end embedding systems or by i-vector, can be followed by back-ends like probabilistic linear discriminant analysis (PLDA) [5]. In addition, cosine similarity based back-end can also be used for speaker verification, which is much simpler than PLDA. Although i-vector based systems are still effective if the utterances have sufficient length [1], end-to-end systems appear to outperform i-vector based methods for short utterances which are more common in real applications.

In end-to-end systems, an appropriate training criterion (loss function) is important for exploiting the power of neural networks. Most traditional systems adopt a softmax loss function to supervise the training of the neural networks. However, in speaker verification tasks, the embeddings learned by the softmax loss based systems cannot achieve satisfactory performance on minimizing intra-class divergence [6, 7].

To improve the performance of end-to-end systems, researchers have recently proposed several new loss functions for SV which can be divided into two major categories. The first category is classification loss, such as center loss and angular softmax (A-Softmax) loss [6, 7]. Center loss [6], which tries to reduce the intra-class distance, is typically used in a combination with softmax loss to train an embedding system. A-Softmax loss [7] tries to incorporate the angular margin into the softmax loss function, which has achieved promising performance. However, the margin in A-Softmax loss is constrained by a positive integer, which is not flexible enough.

The second category is metric learning loss, in which triplet loss [8] and pairwise loss [9, 10] are widely used ones. Triplet loss is defined on a set of triplets, each of which consists of an anchor sample, a positive sample and a negative sample. Triplet loss based systems try to maximize the distance between anchor sample and negative sample as well as minimize the distance between anchor sample and positive sample at the same time. Pairwise loss, such as contrastive loss [9, 10], is defined on a set of pairs. Pairwise loss tries to maximize the distance between two samples if they have different class labels, otherwise minimize it. For models supervised by metric learning loss, the target of training and the requirement of inference are consistent, which should have promising performance as long as the training is sufficient. Nevertheless, metric learning loss based systems have a shortcoming that the size of dataset and the strategies for sampling and composing triplets or pairs significantly affect the performance, bringing obstacle to training. Thus, they are usually used in combination with classification loss.

Very recently, a novel loss function, called additive margin softmax (AM-Softmax) loss [11], is proposed for face verification. AM-Softmax loss has achieved better performance than other loss functions in face verification. In this paper, we adapt the AM-Softmax loss to speaker embedding systems. Furthermore, we propose a novel ensemble loss, called ensemble additive margin softmax (EAM-Softmax) loss, for SV by integrating Hilbert-Schmidt independence criterion (HSIC) [12] into the speaker embedding system with the AM-Softmax loss. Experiments on a large-scale dataset VoxCeleb show that AM-Softmax loss is better than traditional loss functions, and approaches using EAM-Softmax loss can outperform existing speaker verification methods to achieve state-of-the-art performance.

2. PRELIMINARIES

In this section, we introduce some loss functions which have been used in SV tasks, including softmax loss, A-Softmax loss and contrastive loss.

2.1. Softmax Loss

The softmax loss is defined as follows:

$$\mathcal{L}_S = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\mathbf{w}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^{c} e^{\mathbf{w}_j^T \mathbf{x}_i + b_j}}, \tag{1}$$

where c is the number of classes, $\mathbf{x}_i$ is the input of the last fully connected layer corresponding to sample i, $y_i \in \{1, 2, \ldots, c\}$ is the class label of sample i, N is the number of samples, $\mathbf{w}_j$ and b_j are respectively the weight vector and bias of the last fully connected layer corresponding to class j.

2.2. A-Softmax Loss

Note that $\mathbf{w}^T\mathbf{x}$ in softmax loss can be rewritten as $\|\mathbf{w}\|\|\mathbf{x}\|\cos(\theta)$, where θ is the angle between $\mathbf{w}$ and $\mathbf{x}$. Hence, the softmax loss can be rewritten as follows:

$$\mathcal{L}_S = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\|\mathbf{w}_{y_i}\|\|\mathbf{x}_i\|\cos(\theta_{y_i,i})+b_{y_i}}}{\sum_{j=1}^{c} e^{\|\mathbf{w}_j\|\|\mathbf{x}_i\|\cos(\theta_{j,i})+b_j}}, \tag{2}$$

where $\theta_{j,i}$ denotes the angle between $\mathbf{w}_j$ and $\mathbf{x}_i$.

By normalizing weight $\mathbf{w}$, zeroing bias b and replacing cosine with a tighter function $\psi(\theta) < \cos(\theta)$ for the intra-class part, the formulation of a generalized margin softmax loss is given by:

$$\mathcal{L}_{MS} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j=1;j\neq y_i}^{c} e^{\|\mathbf{x}_i\|\cos(\theta_{j,i})}}.$$

A-Softmax loss [13] adopts $\psi(\theta) = cos(m\theta)$, where m is the hyperparameter related to the margin. This makes sense in intuition but $m\theta$ should not be larger than π to preserve monotonicity. To avoid this problem, A-Softmax uses the following monotone decreasing function:

$$\psi(\theta) = (-1)^k \cos(m\theta) - 2k, \tag{3}$$

where $\theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ and $k \in \{0, \ldots, m-1\}$. Usually, m is a positive integer in this function, and hence the margin in A-Softmax loss is not flexible enough.

2.3. Contrastive Loss

Contrastive loss [14, 15] is a kind of pairwise loss in which the samples are organized into pairs with a label $z \in \{0, 1\}$ indicating whether the two elements of the corresponding pair belong to the same class or not. The formulation of contrastive loss is as follows:

$$\mathcal{L}_C = \frac{1}{2M}\sum_{i=1}^{M} \Big(z_i \cdot d_i^2 + (1 - z_i)\max(\rho - d_i, 0)^2\Big), \tag{4}$$

where M is the number of pairs. d_i is the Euclidean distance between the two embeddings of the elements in the i-th pair

$$d_i = \|f(\mathbf{p}_{i,1}; \omega) - f(\mathbf{p}_{i,2}; \omega)\|_2,$$

where $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$ are the two elements from the i-th pair, $f(\cdot; \omega)$ is a non-linear function which represents the embedding system and ω represents the model parameters. The distance between embeddings with different class labels is expected to be larger than a margin ρ.

3. ENSEMBLE ADDITIVE MARGIN SOFTMAX LOSS

This section presents the details of our proposed loss function, called EAM-Softmax loss.

3.1. Additive Margin Softmax Loss

As stated in Section 2.2, A-Softmax is not flexible enough. To overcome this shortcoming of A-Softmax and explore more possible margins, additive margin softmax (AM-Softmax) loss [11] adopts a simpler function $\psi(\theta) = \cos(\theta) - m$ and further normalizes $\mathbf{x}$. The AM-Softmax loss is defined as follows:

$$\mathcal{L}_{AMS} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{\cos(\theta_{y_i,i})-m}}{e^{\cos(\theta_{y_i,i})-m} + \sum_{j=1;j\neq y_i}^{c} e^{\cos(\theta_{j,i})}}.$$

The traditional softmax loss aims to learn a weight vector set and bias set for different classes that satisfy $\mathbf{w}_{y_i}^T\mathbf{x}_i + b_{y_i} > \mathbf{w}_j^T\mathbf{x}_i + b_j$ ($j \in \{1, \ldots, c\}, j \neq y_i$). And the decision boundary satisfies $\min\{(\mathbf{w}_{y_i}^T\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j^T\mathbf{x}_i + b_j)\} = 0$. But AM-Softmax loss obtains a boundary satisfying $\min\{\cos(\theta_{y_i,i}) - \cos(\theta_{j,i})\} = m$, which forces the embeddings to be more discriminative and makes the verification to be more robust.

Since a large margin m might push the decision boundary too hard and make the training difficult to converge, a hyperparameter s is introduced to scale the cosine value and the actual AM-Softmax loss function is given by:

$$\mathcal{L}_{AMS} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j=1;j\neq y_i}^{c} e^{s\cdot\cos(\theta_{j,i})}}.$$

3.2. Hilbert-Schmidt Independence Criterion

The Hilbert-Schmidt independence criterion (HSIC) [12] indicates the independence of two random variables $\mathcal{A}$ and $\mathcal{B}$, and the empirical HSIC is an estimator of HSIC given a finite number of observations.

Definition 1 (Empirical HSIC) *Consider a series of n independent observations $\mathcal{Z} = \{(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_n, \mathbf{b}_n)\} \subseteq \mathcal{A} \times \mathcal{B}$ drawn from $p_{\mathbf{ab}}$. The empirical HSIC is given by*

$$\mathrm{HSIC}(\mathcal{Z}, \mathcal{F}, \mathcal{G}) = (n-1)^{-2}\mathrm{tr}(\mathbf{KHLH}), \tag{5}$$

where $\mathbf{K}$ and $\mathbf{L}$ are Gram matrices with $\mathbf{K}_{ij} = \kappa(\mathbf{a}_i, \mathbf{a}_j)$, $\mathbf{L}_{ij} = \ell(\mathbf{b}_i, \mathbf{b}_j)$. Here, $\kappa(\mathbf{a}_i, \mathbf{a}_j)$ and $\ell(\mathbf{b}_i, \mathbf{b}_j)$ are the kernel functions defined in space $\mathcal{F}$ and $\mathcal{G}$ respectively. $\mathbf{H} = \mathbf{I}_n - n^{-1}\mathbf{J}_n$, where $\mathbf{I}_n$ and $\mathbf{J}_n \in \mathbb{R}^{n\times n}$ are an identity matrix and a matrix of all ones respectively.

3.3. Ensemble Additive Margin Softmax Loss

Diversity in weak learners proves to be critical for the performance of ensemble models. Inspired by the work in [16], we exploit parallel fully connected layers to encourage diversity in homogenous learners. Weights in these layers are highly pairwise independent with the constraint of HSIC. Moreover, the kernel functions in HSIC which map variances into reproducing kernel Hilbert spaces (RKHS) give it the ability to measure nonlinear dependence.

Unlike classification tasks in [16], the classification layer in embedding systems is not suitable for exploiting different models since it will no longer be used once the training is finished. Hence, we add

the HSIC constraint to the embedding layer rather than the classification layer.

Assume that there are V parallel fully connected layers for embedding in the ensemble systems, and each fully connected layer contains a weight matrix $\mathbf{W} \in \mathbb{R}^{l\times n}$ where l and n are the input size and output size of the embedding layer respectively. The formulation of HSIC constraint for the v-th weight matrix $\mathbf{W}^{(v)}$ ($v \in \{1, \ldots, V\}$) is as follows:

$$\text{HSIC}(\mathbf{W}^{(v)}) = \sum_{u=1;u\neq v}^{V} (n-1)^{-2}\text{tr}(\mathbf{K}^{(v)}\mathbf{H}\mathbf{K}^{(u)}\mathbf{H}), \quad (6)$$

where $\mathbf{K}_{ij}^{(v)} = k(\mathbf{W}_i^{(v)}, \mathbf{W}_j^{(v)})$ and $\mathbf{K}_{ij}^{(u)} = k(\mathbf{W}_i^{(u)}, \mathbf{W}_j^{(u)})$, with $\mathbf{W}_i^{(v)}$ being the i-th column of $\mathbf{W}^{(v)}$. Although more complex kernels can be expected to achieve better performance, inner product kernel $\mathbf{K} = \mathbf{W}^T\mathbf{W}$ is just adopted for illustration in this paper. Since weight matrix with small magnitude will have small HSIC constraint, $\{\mathbf{W}^{(v)}\}$ are normalized along vertical axis.

Note that the time and space complexity of the HSIC constraint computation mainly depends on the number of columns in matrix $\mathbf{W}$, which equals to the dimensionality of embedding vectors in our network architectures. Hence, with a low dimensionality of embedding vectors which is typically adopted in practice, we can easily handle several models in the ensemble without worrying about the rapidly increasing memory usage and computational cost faced by [16].

There are two ways to construct the final ensemble model. The first one is to average the outputs of the fully connected layers, and this is equivalent to averaging the weights of the fully connected layers since

$$\frac{1}{V}\sum_{v=1}^{V}\left(\left[\mathbf{W}^{(v)}\right]^T\mathbf{x}\right) = \left(\frac{1}{V}\sum_{v=1}^{V}\mathbf{W}^{(v)}\right)^T\mathbf{x}.$$

The second way is to concatenate the outputs of the fully connected layers. One shortcoming of this way is that the embedding size and the number of parameters in the classification layer are proportional to the number of models in the ensemble, leading to higher storage and computational burden. Hence, we adopt the first way to construct the final ensemble model in this paper.

Rather than optimizing the ensemble model by multiple standalone softmax loss functions, which is adopted in [16] and may lead to inconsistency between training and inference, we directly average the outputs of embedding layers before they are forwarded to the classification layer and optimize the ensemble model by a single softmax loss function.

Finally, by combining the AM-Softmax loss and the HSIC constraint for the embedding layers, we can get the formulation of ensemble additive margin softmax (EAM-Softmax) loss for speaker embedding systems:

$$\mathcal{L}_{EAMS} = -\frac{V}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j=1;j\neq y_i}^{c} e^{s\cdot\cos(\theta_{j,i})}} + \lambda\sum_{v=1}^{V}\sum_{u=1;u\neq v}^{V}(n-1)^{-2}\text{tr}(\mathbf{K}^{(v)}\mathbf{H}\mathbf{K}^{(u)}\mathbf{H}),$$

where λ is a hyperparameter denoting the tradeoff between the AM-Softmax loss and the HSIC constraint.

Table 1. Dataset for evaluation. POI denotes Person of Interest.

Dataset	#	Dev	Test	Total
VoxCeleb1	POIs	1,211	40	1,251
	utterances	148,642	4,874	153,516
	hours	-	-	352
VoxCeleb2	POIs	5,994	118	6,112
	utterances	1,092,009	36,237	1,128,246
	hours	-	-	2,442

4. EXPERIMENTS

We compare our method with other baselines in real dataset.

4.1. Dataset

In the experiments, we use two datasets including VoxCeleb1 [4] and VoxCeleb2 [9]. Both datasets are gender balanced and contain a large number of utterances from thousands of speakers. The utterances are collected from YouTube videos in which the speakers belong to different races and have a wide range of accents. The datasets contain background noise from a large number of environments, e.g., overlapping speech, which makes the audio segments challenging for speaker verification.

Both datasets are split into development set and test set. We adopt the same strategy as that in [9] for evaluation. In particular, the development set of VoxCeleb2 is used for training and the test set of VoxCeleb1 is used for testing. Details of VoxCeleb1 and VoxCeleb2 are described in Table 1. There are no overlapping identities between these two datasets.

4.2. Implementation Details

In order to facilitate fair comparison of experimental results, we try to make our experimental settings consistent with those of baselines [4, 9], except for the loss functions and ensemble strategy. Thus we adopt similar network architectures, data processing, training and testing strategies in our experiments.

Networks. Network architectures are modified from the original residual networks (ResNet) [17] to take spectrograms as input features. In particular, ResNet-34 and ResNet-50 are used in our experiments. The details of network architectures are described in Table 2. With an input feature length of 512, the output size of *conv5_x* will be $9 \times h$, where h is determined by the audio segment length. The *conv6* layer is employed to combine information from different frequency domains, where the filter size is 9×1 and the output size is $1 \times h$. The adaptive average pool *avgpool*, which supports different input sizes, calculates a temporal mean of size 1×1. These modifications make the network architectures sensitive to frequency variance rather than temporal position, which is desired in text independent SV.

Features. Spectrograms computed through a sliding hamming window are used as input features. Window width and window step are 25ms and 10ms respectively. Feature length is set to 512. Normalization is performed along axis of frequency.

Hyperparameter. Margin m and scale factor s for AM-Softmax loss are set to 0.35 and 30.0 respectively. Ensemble number $V = 4$. Hyperparameter λ for balancing AM-Softmax loss and HSIC constraint in the EAM-Softmax loss is set to 0.1.

Training. 3-second utterances are randomly sampled from each audio file in training, each producing a spectrogram of size $512 \times$

Table 2. Network architectures modified from ResNet-34 and ResNet-50 for spectrogram inputs. The *conv6* layers are implemented with 2d convolutional layers, where the number of groups equals to the number of channels.

layer name	34-layer	50-layer
conv1	7×7, 64, stride 2	
maxpool	3×3 max pool, stride 2	
conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
conv6	9×1, 512, stride 1	9×1, 2048, stride 1
avgpool	1×1, adaptive average pool, stride 1	
embedding	512×512	2048×512
classification	512×5994	

300. Models are optimized by momentum stochastic gradient descent (SGD), in which the momentum is 0.9 and the weight decay is $5 \times e^{-4}$. Mini-batch size is 64. Learning rate is initialized as 0.1. For the 34-layer network, the learning rate is divided by 10 after the 6-th and the 12-th epochs. For the 50-layer network, the learning rate is divided by 10 after the 10-th and the 20-th epochs. The training terminates earlier to avoid overfitting if the performance on a validation set, which is randomly sampled from VoxCeleb1 development set, stops improving after 12 epochs for the 34-layer network and 20 epochs for the 50-layer network.

Testing. Full length utterances are used in testing, so the generated spectrograms are in different sizes. Adaptive average pooling is employed to output embeddings of the same size.

4.3. Metric

Two metrics are used for performance evaluation:

(1) Equal error rate (EER): the error rate when false rejection probability P_{fr} equals false acceptance probability P_{fa};

(2) Minimum detection cost function (minimum DCF): similar to EER, but takes different costs of misclassification and uneven target/nontarget probability into account. The formulation of minimum DCF is given by

$$C_{det}^{min} = \min\{C_{fr} * P_{fr} * P_{tar} + C_{fa} * P_{fa} * (1 - P_{tar})\},$$

where C_{fr} and C_{fa} indicate the cost of false rejection and false acceptance respectively, and P_{tar} is the target probability. All of the three parameters are application dependent. In our experiments, we adopt the same values as those in [9] for C_{fr} (1.0), C_{fa} (1.0) and P_{tar} (0.01).

Table 3. Experimental results. Here, * denotes that the results are from [9]. The letters in the brackets are the initials of loss functions, where S, C, AMS and EAMS denote softmax, contrastive, AM-Softmax and EAM-Softmax respectively.

Model	**Trained on**	C_{det}^{min}	**EER** (%)
i-vector + PLDA	VoxCeleb1	0.73*	8.8*
VGG-M (S)	VoxCeleb1	0.75*	10.2*
VGG-M (C)	VoxCeleb1	0.71*	7.8*
VGG-M (C)	VoxCeleb2	0.609*	5.94*
ResNet-34 (C)	VoxCeleb2	0.543*	5.04*
ResNet-50 (C)	VoxCeleb2	0.449*	4.19*
ResNet-34 (AMS)	VoxCeleb2	0.304	3.35
ResNet-34 (EAMS)	VoxCeleb2	0.305	3.14
ResNet-50 (AMS)	VoxCeleb2	0.303	3.10
ResNet-50 (EAMS)	VoxCeleb2	**0.278**	**2.94**

4.4. Baseline

Methods and results explored in the experiments of [4, 9] are used as baselines, including:

(1) I-vector based embedding system with a PLDA back-end;

(2) End-to-end embedding systems with a cosine similarity based back-end, in which the architectures are modified from networks introduced by visual geometry group (VGG-M) [18] or ResNet [17]. Those networks modified from ResNet are exactly the same as the networks employed in the experiments of AM-softmax loss and EAM-Softmax loss, except for the extra *embedding* layer.

For the end-to-end embedding systems, softmax loss and contrastive loss are employed. Nevertheless, standalone contrastive loss is hard to learn. Baseline models supervised by contrastive loss are obtained in two stages. First, softmax loss is used to initialize the weights of networks. Then the classification layer is replaced by a fully connected layer with a smaller output size. This fully connected layer is treated as the embedding layer and the contrastive loss is used for tuning its parameters.

4.5. Results

Results on VoxCeleb1 test set are listed in Table 3. Our method achieves state-of-the-art performance, decreasing EER to 2.94% and minimum DCF to 0.278, which are 29.8% and 38.1% relatively lower than the best results in [9].

VGG-M architecture trained with softmax loss is slightly weaker than the traditional i-vector based approach, but VGG-M architecture trained with contrastive loss surpasses i-vector based approach. Furthermore, end-to-end systems using AM-Softmax loss outperform all of the baselines, and approaches using EAM-Softmax loss achieve the best results.

5. CONCLUSION

This paper first adapts the AM-Softmax loss to speaker verification, and then proposes a novel EAM-Softmax loss for speaker verification. Experiments on real datasets show that the proposed methods can achieve state-of-the-art performance.

Acknowledgement. This work has been supported by the NSFC-NRF Joint Research Project (No. 61861146001).

6. 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, 2011.

[2] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[3] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[4] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[5] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[6] Na Li, Deyi Tuo, Dan Su, Zhifeng Li, and Dong Yu, "Deep discriminative embeddings for duration robust speaker verification," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2262–2266.

[7] Zili Huang, Shuai Wang, and Kai Yu, "Angular softmax for short-duration text-independent speaker verification," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3623–3627.

[8] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 2017.

[9] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: deep speaker recognition," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086–1090.

[10] Gautam Bhattacharya, Md Jahangir Alam, Vishwa Gupta, and Patrick Kenny, "Deeply fused speaker embeddings for text-independent speaker verification," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 3588–3592.

[11] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[12] Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *International Conference on Algorithmic Learning Theory (ALT)*, 2005, pp. 63–77.

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: deep hypersphere embedding for face recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6738–6746.

[14] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 539–546.

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1735–1742.

[16] Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z Li, "Ensemble soft-margin softmax loss for image classification," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 992–998.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: delving deep into convolutional nets," *British Machine Vision Conference (BMVC)*, 2014.