CosFace: Large Margin Cosine Loss for Deep Face Recognition

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou,
Zhifeng Li[*], and Wei Liu[*]

Tencent AI Lab

{hawelwang,yitongwang,encorezhou,denisji,sagazhou,michaelzfli}@tencent.com
gongdihong@gmail.com wliu@ee.columbia.edu

Abstract

Face recognition has made extraordinary progress owing to the advancement of deep convolutional neural networks (CNNs). The central task of face recognition, including face verification and identification, involves face feature discrimination. However, the traditional softmax loss of deep CNNs usually lacks the power of discrimination. To address this problem, recently several loss functions such as center loss, large margin softmax loss, and angular softmax loss have been proposed. All these improved losses share the same idea: maximizing inter-class variance and minimizing intra-class variance. In this paper, we propose a novel loss function, namely large margin cosine loss (LMCL), to realize this idea from a different perspective. More specifically, we reformulate the softmax loss as a cosine loss by L_2 normalizing both features and weight vectors to remove radial variations, based on which a cosine margin term is introduced to further maximize the decision margin in the angular space. As a result, minimum intra-class variance and maximum inter-class variance are achieved by virtue of normalization and cosine decision margin maximization. We refer to our model trained with LMCL as CosFace. Extensive experimental evaluations are conducted on the most popular public-domain face recognition datasets such as MegaFace Challenge, Youtube Faces (YTF) and Labeled Face in the Wild (LFW). We achieve the state-of-the-art performance on these benchmarks, which confirms the effectiveness of our proposed approach.

1. Introduction

Recently progress on the development of deep convolutional neural networks (CNNs) [15, 18, 12, 9, 44] has significantly advanced the state-of-the-art performance on a wide variety of computer vision tasks, which makes deep CNN a dominant machine learning approach for computer vision. Face recognition, as one of the most common computer vision tasks, has been extensively studied for decades [37, 45, 22, 19, 20, 40, 2]. Early studies build shallow models with low-level face features, while modern face recognition techniques are greatly advanced driven by deep CNNs. Face recognition usually includes two sub-tasks: face verification and face identification. Both of these two tasks involve three stages: face detection, feature extraction, and classification. A deep CNN is able to extract clean high-level features, making itself possible to achieve superior performance with a relatively simple classification architecture: usually, a multilayer perceptron networks followed by

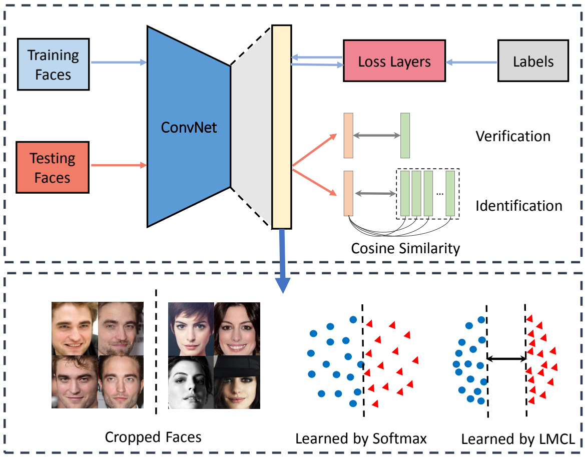

Figure 1. An overview of the proposed CosFace framework. In the training phase, the discriminative face features are learned with a large margin between different classes. In the testing phase, the testing data is fed into CosFace to extract face features which are later used to compute the cosine similarity score to perform face verification and identification.

[*] Corresponding authors

a softmax loss [35, 32]. However, recent studies [42, 24, 23] found that the traditional softmax loss is insufficient to acquire the discriminating power for classification.

To encourage better discriminating performance, many research studies have been carried out [42, 5, 7, 10, 39, 23]. All these studies share the same idea for maximum discrimination capability: maximizing inter-class variance and minimizing intra-class variance. For example, [42, 5, 7, 10, 39] propose to adopt multi-loss learning in order to increase the feature discriminating power. While these methods improve classification performance over the traditional softmax loss, they usually come with some extra limitations. For [42], it only explicitly minimizes the intra-class variance while ignoring the inter-class variances, which may result in suboptimal solutions. [5, 7, 10, 39] require thoroughly scheming the mining of pair or triplet samples, which is an extremely time-consuming procedure. Very recently, [23] proposed to address this problem from a different perspective. More specifically, [23] (A-softmax) projects the original Euclidean space of features to an angular space, and introduces an angular margin for larger inter-class variance.

Compared to the Euclidean margin suggested by [42, 5, 10], the angular margin is preferred because the cosine of the angle has intrinsic consistency with softmax. The formulation of cosine matches the similarity measurement that is frequently applied to face recognition. From this perspective, it is more reasonable to directly introduce cosine margin between different classes to improve the cosine-related discriminative information.

In this paper, we reformulate the softmax loss as a cosine loss by L_2 normalizing both features and weight vectors to remove radial variations, based on which a cosine margin term m is introduced to further maximize the decision margin in the angular space. Specifically, we propose a novel algorithm, dubbed Large Margin Cosine Loss (LMCL), which takes the normalized features as input to learn highly discriminative features by maximizing the inter-class cosine margin. Formally, we define a hyper-parameter m such that the decision boundary is given by $cos(\theta_1) - m = cos(\theta_2)$, where θ_i is the angle between the feature and weight of class i.

For comparison, the decision boundary of the A-Softmax is defined over the angular space by $\cos(m\theta_1) = \cos(\theta_2)$, which has a difficulty in optimization due to the nonmonotonicity of the cosine function. To overcome such a difficulty, one has to employ an extra trick with an ad-hoc piecewise function for A-Softmax. More importantly, the decision margin of A-softmax depends on θ, which leads to different margins for different classes. As a result, in the decision space, some inter-class features have a larger margin while others have a smaller margin, which reduces the discriminating power. Unlike A-Softmax, our approach defines the decision margin in the cosine space, thus avoiding the aforementioned shortcomings.

Based on the LMCL, we build a sophisticated deep model called CosFace, as shown in Figure 1. In the training phase, LMCL guides the ConvNet to learn features with a large cosine margin. In the testing phase, the face features are extracted from the ConvNet to perform either face verification or face identification. We summarize the contributions of this work as follows:

(1) We embrace the idea of maximizing inter-class variance and minimizing intra-class variance and propose a novel loss function, called LMCL, to learn highly discriminative deep features for face recognition.

(2) We provide reasonable theoretical analysis based on the hyperspherical feature distribution encouraged by LMCL.

(3) The proposed approach advances the state-of-the-art performance over most of the benchmarks on popular face databases including LFW[13], YTF[43] and Megaface [17, 25].

2. Related Work

Deep Face Recognition. Recently, face recognition has achieved significant progress thanks to the great success of deep CNN models [18, 15, 34, 9]. In DeepFace [35] and DeepID [32], face recognition is treated as a multi-class classification problem and deep CNN models are first introduced to learn features on large multi-identities datasets. DeepID2 [30] employs identification and verification signals to achieve better feature embedding. Recent works DeepID2+ [33] and DeepID3 [31] further explore the advanced network structures to boost recognition performance. FaceNet [29] uses triplet loss to learn an Euclidean space embedding and a deep CNN is then trained on nearly 200 million face images, leading to the state-of-the-art performance. Other approaches [41, 11] also prove the effectiveness of deep CNNs on face recognition.

Loss Functions. Loss function plays an important role in deep feature learning. Contrastive loss [5, 7] and triplet loss [10, 39] are usually used to increase the Euclidean margin for better feature embedding. Wen *et al.* [42] proposed a center loss to learn centers for deep features of each identity and used the centers to reduce intra-class variance. Liu *et al.* [24] proposed a large margin softmax (L-Softmax) by adding angular constraints to each identity to improve feature discrimination. Angular softmax (A-Softmax) [23] improves L-Softmax by normalizing the weights, which achieves better performance on a series of open-set face recognition benchmarks [13, 43, 17]. Other loss functions [47, 6, 4, 3] based on contrastive loss or center loss also demonstrate the performance on enhancing discrimination.

Normalization Approaches. Normalization has been studied in recent deep face recognition studies. [38] normalizes the weights which replace the inner product with cosine similarity within the softmax loss. [28] applies the L_2 constraint on features to embed faces in the normalized space. Note that normalization on feature vectors or weight vectors achieves much lower intra-class angular variability by concentrating more on the angle during training. Hence the angles between identities can be well optimized. The von Mises-Fisher (vMF) based methods [48, 8] and A-Softmax [23] also adopt normalization in feature learning.

3. Proposed Approach

In this section, we firstly introduce the proposed LMCL in detail (Sec. 3.1). And a comparison with other loss functions is given to show the superiority of the LMCL (Sec. 3.2). The feature normalization technique adopted by the LMCL is further described to clarify its effectiveness (Sec. 3.3). Lastly, we present a theoretical analysis for the proposed LMCL (Sec. 3.4).

3.1. Large Margin Cosine Loss

We start by rethinking the softmax loss from a cosine perspective. The softmax loss separates features from different classes by maximizing the posterior probability of the ground-truth class. Given an input feature vector x_i with its corresponding label y_i, the softmax loss can be formulated as:

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -\log p_i = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{f_{y_i}}}{\sum_{j=1}^{C} e^{f_j}}, \tag{1}$$

where p_i denotes the posterior probability of x_i being correctly classified. N is the number of training samples and C is the number of classes. f_j is usually denoted as activation of a fully-connected layer with weight vector W_j and bias B_j. We fix the bias $B_j = 0$ for simplicity, and as a result f_j is given by:

$$f_j = W_j^T x = \|W_j\|\|x\| \cos\theta_j, \tag{2}$$

where θ_j is the angle between W_j and x. This formula suggests that both norm and angle of vectors contribute to the posterior probability.

To develop effective feature learning, the norm of W should be necessarily invariable. To this end, We fix $\|W_j\| = 1$ by L_2 normalization. In the testing stage, the face recognition score of a testing face pair is usually calculated according to cosine similarity between the two feature vectors. This suggests that the norm of feature vector x is not contributing to the scoring function. Thus, in the training stage, we fix $\|x\| = s$. Consequently, the posterior probability merely relies on cosine of angle. The modified loss can be formulated as

$$L_{ns} = \frac{1}{N}\sum_{i} -\log \frac{e^{s\cos(\theta_{y_i},i)}}{\sum_j e^{s\cos(\theta_j,i)}}. \tag{3}$$

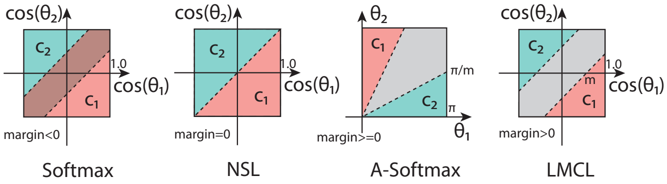

Figure 2. The comparison of decision margins for different loss functions the binary-classes scenarios. Dashed line represents decision boundary, and gray areas are decision margins.

Because we remove variations in radial directions by fixing $\|x\| = s$, the resulting model learns features that are separable in the angular space. We refer to this loss as the Normalized version of Softmax Loss (NSL) in this paper.

However, features learned by the NSL are not sufficiently discriminative because the NSL only emphasizes correct classification. To address this issue, we introduce the cosine margin to the classification boundary, which is naturally incorporated into the cosine formulation of Softmax.

Considering a scenario of binary-classes for example, let θ_i denote the angle between the learned feature vector and the weight vector of Class C_i ($i = 1, 2$). The NSL forces $\cos(\theta_1) > \cos(\theta_2)$ for C_1, and similarly for C_2, so that features from different classes are correctly classified. To develop a large margin classifier, we further require $\cos(\theta_1) - m > \cos(\theta_2)$ and $\cos(\theta_2) - m > \cos(\theta_1)$, where $m \geq 0$ is a fixed parameter introduced to control the magnitude of the cosine margin. Since $\cos(\theta_i) - m$ is lower than $\cos(\theta_i)$, the constraint is more stringent for classification. The above analysis can be well generalized to the scenario of multi-classes. Therefore, the altered loss reinforces the discrimination of learned features by encouraging an extra margin in the cosine space.

Formally, we define the Large Margin Cosine Loss (LMCL) as:

$$L_{lmc} = \frac{1}{N}\sum_{i} -\log \frac{e^{s(\cos(\theta_{y_i},i)-m)}}{e^{s(\cos(\theta_{y_i},i)-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_j,i)}}, \tag{4}$$

subject to

$$\begin{aligned} W &= \frac{W^*}{\|W^*\|}, \\ x &= \frac{x^*}{\|x^*\|}, \\ cos(\theta_j, i) &= {W_j}^T x_i, \end{aligned} \tag{5}$$

where N is the numer of training samples, x_i is the i-th feature vector corresponding to the ground-truth class of y_i, the W_j is the weight vector of the j-th class, and θ_j is the angle between W_j and x_i.

3.2. Comparison on Different Loss Functions

In this subsection, we compare the decision margin of our method (LMCL) to: Softmax, NSL, and A-Softmax, as illustrated in Figure 2. For simplicity of analysis, we consider the binary-classes scenarios with classes C_1 and C_2. Let W_1 and W_2 denote weight vectors for C_1 and C_2, respectively.

Softmax loss defines a decision boundary by:

$$\|W_1\|\cos(\theta_1) = \|W_2\|\cos(\theta_2).$$

Thus, its boundary depends on both magnitudes of weight vectors and cosine of angles, which results in an overlapping decision area (margin < 0) in the cosine space. This is illustrated in the first subplot of Figure 2. As noted before, in the testing stage it is a common strategy to only consider cosine similarity between testing feature vectors of faces. Consequently, the trained classifier with the Softmax loss is unable to perfectly classify testing samples in the cosine space.

NSL normalizes weight vectors W_1 and W_2 such that they have constant magnitude 1, which results in a decision boundary given by:

$$\cos(\theta_1) = \cos(\theta_2).$$

The decision boundary of NSL is illustrated in the second subplot of Figure 2. We can see that by removing radial variations, the NSL is able to perfectly classify testing samples in the cosine space, with margin = 0. However, it is not quite robust to noise because there is no decision margin: any small perturbation around the decision boundary can change the decision.

A-Softmax improves the softmax loss by introducing an extra margin, such that its decision boundary is given by:

$$C_1 : \cos(m\theta_1) \geq \cos(\theta_2),$$
$$C_2 : \cos(m\theta_2) \geq \cos(\theta_1).$$

Thus, for C_1 it requires $\theta_1 \leq \frac{\theta_2}{m}$, and similarly for C_2. The third subplot of Figure 2 depicts this decision area, where gray area denotes decision margin. However, the margin of A-Softmax is not consistent over all θ values: the margin becomes smaller as θ reduces, and vanishes completely when $\theta = 0$. This results in two potential issues. First, for difficult classes C_1 and C_2 which are visually similar and thus have a smaller angle between W_1 and W_2, the margin is consequently smaller. Second, technically speaking one has to employ an extra trick with an ad-hoc piecewise function to overcome the nonmonotonicity difficulty of the cosine function.

LMCL (our proposed) defines a decision margin in cosine space rather than the angle space (like A-Softmax) by:

$$C_1 : \cos(\theta_1) \geq \cos(\theta_2) + m,$$
$$C_2 : \cos(\theta_2) \geq \cos(\theta_1) + m.$$

Therefore, $\cos(\theta_1)$ is maximized while $\cos(\theta_2)$ being minimized for C_1 (similarly for C_2) to perform the large-margin classification. The last subplot in Figure 2 illustrates the decision boundary of LMCL in the cosine space, where we can see a clear margin($\sqrt{2}m$) in the produced distribution of the cosine of angle. This suggests that the LMCL is more robust than the NSL, because a small perturbation around the decision boundary (dashed line) less likely leads to an incorrect decision. The cosine margin is applied consistently to all samples, regardless of the angles of their weight vectors.

3.3. Normalization on Features

In the proposed LMCL, a normalization scheme is involved on purpose to derive the formulation of the cosine loss and remove variations in radial directions. Unlike [23] that only normalizes the weight vectors, our approach simultaneously normalizes both weight vectors and feature vectors. As a result, the feature vectors distribute on a hypersphere, where the scaling parameter s controls the magnitude of radius. In this subsection, we discuss why feature normalization is necessary and how feature normalization encourages better feature learning in the proposed LMCL approach.

The necessity of feature normalization is presented in two respects: First, the original softmax loss without feature normalization implicitly learns both the Euclidean norm (L_2-norm) of feature vectors and the cosine value of the angle. The L_2-norm is adaptively learned for minimizing the overall loss, resulting in the relatively weak cosine constraint. Particularly, the adaptive L_2-norm of easy samples becomes much larger than hard samples to remedy the inferior performance of cosine metric. On the contrary, our approach requires the entire set of feature vectors to have the same L_2-norm such that the learning only depends on cosine values to develop the discriminative power. Feature vectors from the same classes are clustered together and those from different classes are pulled apart on the surface of the hypersphere. Additionally, we consider the situation when the model initially starts to minimize the LMCL. Given a feature vector x, let $\cos(\theta_i)$ and $\cos(\theta_j)$ denote cosine scores of the two classes, respectively. Without normalization on features, the LMCL forces $\|x\|(\cos(\theta_i) - m) > \|x\|\cos(\theta_j)$. Note that $\cos(\theta_i)$ and $\cos(\theta_j)$ can be initially comparable with each other. Thus, as long as $(\cos(\theta_i) - m)$ is smaller than $\cos(\theta_j)$, $\|x\|$ is required to decrease for minimizing the loss, which degenerates the optimization. Therefore, feature normalization is critical under the supervision of LMCL, especially when the networks are trained from scratch. Likewise, it is more favorable to fix the scaling parameter s instead of adaptively learning.

Furthermore, the scaling parameter s should be set to a properly large value to yield better-performing features with lower training loss. For NSL, the loss continuously goes

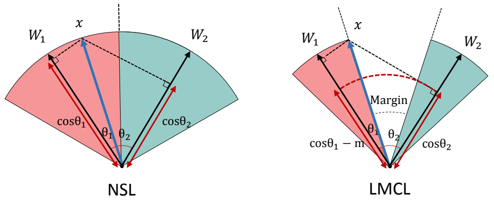

Figure 3. A geometrical interpretation of LMCL from feature perspective. Different color areas represent feature space from distinct classes. LMCL has a relatively compact feature region compared with NSL.

down with higher s, while too small s leads to an insufficient convergence even no convergence. For LMCL, we also need adequately large s to ensure a sufficient hyperspace for feature learning with an expected large margin.

In the following, we show the parameter s should have a lower bound to obtain expected classification performance. Given the normalized learned feature vector x and unit weight vector W, we denote the total number of classes as C. Suppose that the learned feature vectors separately lie on the surface of the hypersphere and center around the corresponding weight vector. Let P_W denote the expected minimum posterior probability of class center (*i.e.*, W), the lower bound of s is given by [1]:

$$s \geq \frac{C-1}{C} \log \frac{(C-1)P_W}{1-P_W}. \tag{6}$$

Based on this bound, we can infer that s should be enlarged consistently if we expect an optimal P_w for classification with a certain number of classes. Besides, by keeping a fixed P_w, the desired s should be larger to deal with more classes since the growing number of classes increase the difficulty for classification in the relatively compact space. A hypersphere with large radius s is therefore required for embedding features with small intra-class distance and large inter-class distance.

3.4. Theoretical Analysis for LMCL

The preceding subsections essentially discuss the LMCL from the classification point of view. In terms of learning the discriminative features on the hypersphere, the cosine margin servers as momentous part to strengthen the discriminating power of features. Detailed analysis about the quantitative feasible choice of the cosine margin (*i.e.*, the bound of hyper-parameter m) is necessary. The optimal choice of m potentially leads to more promising learning of highly discriminative face features. In the following, we delve into the decision boundary and angular margin in the feature space to derive the theoretical bound for hyper-parameter m.

First, considering the binary-classes case with classes C_1 and C_2 as before, suppose that the normalized feature vector x is given. Let W_i denote the normalized weight vector, and θ_i denote the angle between x and W_i. For NSL, the decision boundary defines as $\cos\theta_1 - \cos\theta_2 = 0$, which is equivalent to the angular bisector of W_1 and W_2 as shown in the left of Figure 3. This addresses that the model supervised by NSL partitions the underlying feature space to two close regions, where the features near the boundary are extremely ambiguous (*i.e.*, belonging to either class is acceptable). In contrast, LMCL drives the decision boundary formulated by $\cos\theta_1 - \cos\theta_2 = m$ for C_1, in which θ_1 should be much smaller than θ_2 (similarly for C_2). Consequently, the inter-class variance is enlarged while the intra-class variance shrinks.

Back to Figure 3, one can observe that the maximum angular margin is subject to the angle between W_1 and W_2. Accordingly, the cosine margin should have the limited variable scope when W_1 and W_2 are given. Specifically, suppose a scenario that all the feature vectors belonging to class i exactly overlap with the corresponding weight vector W_i of class i. In other words, every feature vector is identical to the weight vector for class i, and apparently the feature space is in an extreme situation, where all the feature vectors lie at their class center. In that case, the margin of decision boundaries has been maximized (*i.e.*, the strict upper bound of the cosine margin).

To extend in general, we suppose that all the features are well-separated and we have a total number of C classes. The theoretical variable scope of m is supposed to be: $0 \leq m \leq (1 - \max(W_i^T W_j))$, where $i, j \leq n, i \neq j$. The softmax loss tries to maximize the angle between any of the two weight vectors from two different classes in order to perform perfect classification. Hence, it is clear that the optimal solution for the softmax loss should uniformly distribute the weight vectors on a unit hypersphere. Based on this assumption, the variable scope of the introduced cosine margin m can be inferred as follows [2]:

$$\begin{aligned}
&0 \leq m \leq 1 - \cos\frac{2\pi}{C}, && (K = 2)\\
&0 \leq m \leq \frac{C}{C-1}, && (C \leq K+1)\\
&0 \leq m \ll \frac{C}{C-1}, && (C > K+1)
\end{aligned} \tag{7}$$

where C is the number of training classes and K is the dimension of learned features. The inequalities indicate that as the number of classes increases, the upper bound of the cosine margin between classes are decreased correspondingly. Especially, if the number of classes is much larger than the feature dimension, the upper bound of the cosine margin will get even smaller.

[1] Proof is attached in the supplemental material.

[2] Proof is attached in the supplemental material.

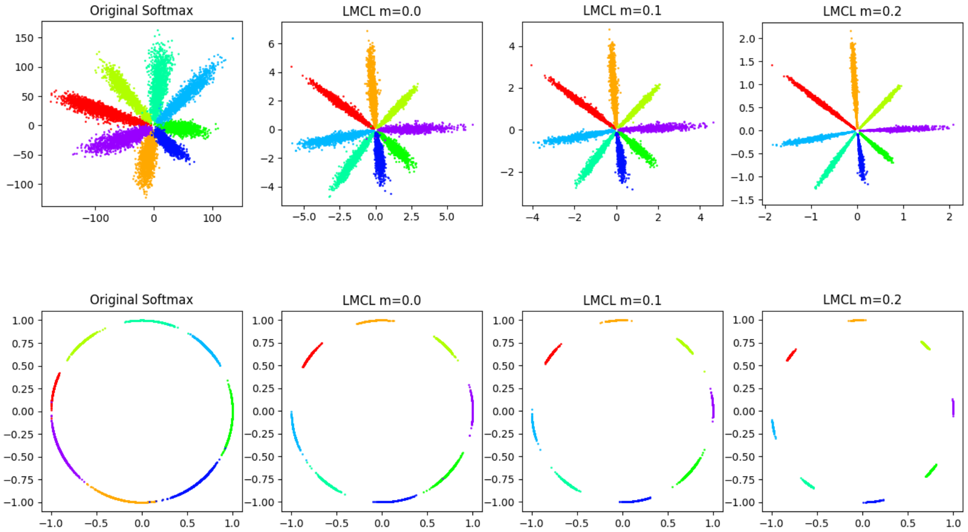

Figure 4. A toy experiment of different loss functions on 8 identities with 2D features. The first row maps the 2D features onto the Euclidean space, while the second row projects the 2D features onto the angular space. The gap becomes evident as the margin term m increases.

A reasonable choice of larger $m \in [0, \frac{C}{C-1})$ should effectively boost the learning of highly discriminative features. Nevertheless, parameter m usually could not reach the theoretical upper bound in practice due to the vanishing of the feature space. That is, all the feature vectors are centered together according to the weight vector of the corresponding class. In fact, the model fails to converge when m is too large, because the cosine constraint (*i.e.*, $\cos\theta_1 - m > \cos\theta_2$ or $\cos\theta_2 - m > \cos\theta_1$ for two classes) becomes stricter and is hard to be satisfied. Besides, the cosine constraint with overlarge m forces the training process to be more sensitive to noisy data. The ever-increasing m starts to degrade the overall performance at some point because of failing to converge.

We perform a toy experiment for better visualizing on features and validating our approach. We select face images from 8 distinct identities containing enough samples to clearly show the feature points on the plot. Several models are trained using the original softmax loss and the proposed LMCL with different settings of m. We extract 2-D features of face images for simplicity. As discussed above, m should be no larger than $1 - \cos\frac{\pi}{4}$ (about 0.29), so we set up three choices of m for comparison, which are $m = 0$, $m = 0.1$, and $m = 0.2$. As shown in Figure 4, the first row and second row present the feature distributions in Euclidean space and angular space, respectively. We can observe that the original softmax loss produces ambiguity in decision boundaries while the proposed LMCL performs much better. As m increases, the angular margin between different classes has been amplified.

4. Experiments

4.1. Implementation Details

Preprocessing. Firstly, face area and landmarks are detected by MTCNN [16] for the entire set of training and testing images. Then, the 5 facial points (two eyes, nose and two mouth corners) are adopted to perform similarity transformation. After that we obtain the cropped faces which are then resized to be 112 × 96. Following [42, 23], each pixel (in [0, 255]) in RGB images is normalized by subtracting 127.5 then dividing by 128.

Training. For a direct and fair comparison to the existing results that use small training datasets (less than 0.5M images and 20K subjects) [17], we train our models on a small training dataset, which is the publicly available CASIA-WebFace [46] dataset containing 0.49M face images from 10,575 subjects. We also use a large training dataset to evaluate the performance of our approach for benchmark comparison with the state-of-the-art results (using large training dataset) on the benchmark face dataset. The large training dataset that we use in this study is composed of several public datasets and a private face dataset, containing about 5M images from more than 90K identities. The training faces are horizontally flipped for data augmentation. In our experiments we remove face images belong to identities that appear in the testing datasets.

For the fair comparison, the CNN architecture used in our work is similar to [23], which has 64 convolutional layers and is based on residual units[9]. The scaling parameter s in Equation (4) is set to 64 empirically. We use Caffe[14] to implement the modifications of the loss layer and run the

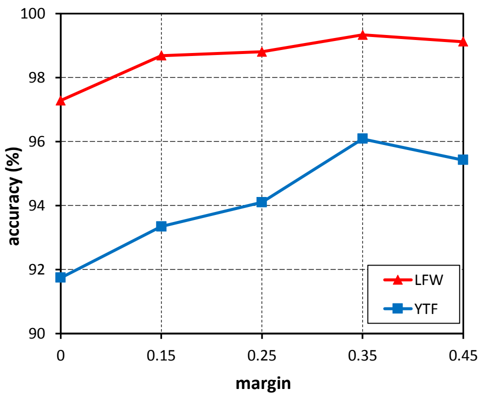

Figure 5. Accuracy (%) of CosFace with different margin parameters m on LFW[13] and YTF [43].

models. The CNN models are trained with SGD algorithm, with the batch size of 64 on 8 GPUs. The weight decay is set to 0.0005. For the case of training on the small dataset, the learning rate is initially 0.1 and divided by 10 at the 16K, 24K, 28k iterations, and we finish the training process at 30k iterations. While the training on the large dataset terminates at 240k iterations, with the initial learning rate 0.05 dropped at 80K, 140K, 200K iterations.

Testing. At testing stage, features of original image and the flipped image are concatenated together to compose the final face representation. The cosine distance of features is computed as the similarity score. Finally, face verification and identification are conducted by thresholding and ranking the scores. We test our models on several popular public face datasets, including LFW[13], YTF[43], and MegaFace[17, 25].

4.2. Exploratory Experiments

Effect of m. The margin parameter m plays a key role in LMCL. In this part we conduct an experiment to investigate the effect of m. By varying m from 0 to 0.45 (If m is larger than 0.45, the model will fail to converge), we use the small training data (CASIA-WebFace [46]) to train our CosFace model and evaluate its performance on the LFW[13] and YTF[43] datasets, as illustrated in Figure 5. We can see that the model without the margin (in this case m=0) leads to the worst performance. As m being increased, the accuracies are improved consistently on both datasets, and get saturated at $m = 0.35$. This demonstrates the effectiveness of the margin m. By increasing the margin m, the discriminative power of the learned features can be significantly improved. In this study, m is set to fixed 0.35 in the subsequent experiments.

Effect of Feature Normalization. To investigate the effect of the feature normalization scheme in our approach, we train our CosFace models on the CASIA-WebFace with and without the feature normalization scheme by fixing m to 0.35, and compare their performance on LFW[13], YTF[43], and the Megaface Challenge 1(MF1)[17]. Note that the model trained without normalization is initialized by softmax loss and then supervised by the proposed LMCL. The comparative results are reported in Table 1. It is very clear that the model using the feature normalization scheme consistently outperforms the model without the feature normalization scheme across the three datasets. As discussed above, feature normalization removes radical variance, and the learned features can be more discriminative in angular space. This experiment verifies this point.

Normalization	LFW	YTF	MF1 Rank 1	MF1 Veri.
No	99.10	93.1	75.10	88.65
Yes	99.33	96.1	77.11	89.88

Table 1. Comparison of our models with and without feature normalization on Megaface Challenge 1 (MF1). “Rank 1” refers to rank-1 face identification accuracy and “Veri.” refers to face verification TAR (True Accepted Rate) under 10^{-6} FAR (False Accepted Rate).

4.3. Comparison with state-of-the-art loss functions

In this part, we compare the performance of the proposed LMCL with the state-of-the-art loss functions. Following the experimental setting in [23], we train a model with the guidance of the proposed LMCL on the CAISA-WebFace[46] using the same 64-layer CNN architecture described in [23]. The experimental comparison on LFW, YTF and MF1 are reported in Table 2. For fair comparison, we are strictly following the model structure (a 64-layers ResNet-Like CNNs) and the detailed experimental settings of SphereFace [23]. As can be seen in Table 2, LMCL consistently achieves competitive results compared to the other losses across the three datasets. Especially, our method not only surpasses the performance of A-Softmax with feature normalization (named as A-Softmax-NormFea in Table 2), but also significantly outperforms the other loss functions on YTF and MF1, which demonstrates the effectiveness of LMCL.

4.4. Overall Benchmark Comparison

4.4.1 Evaluation on LFW and YTF

LFW [13] is a standard face verification testing dataset in unconstrained conditions. It includes 13,233 face images from 5749 identities collected from the website. We evaluate our model strictly following the standard protocol of unrestricted with labeled outside data [13], and report the result on the 6,000 pair testing images. YTF [43] contains 3,425 videos of 1,595 different people. The average length of a video clip is 181.3 frames. All the video sequences were downloaded from YouTube. We follow the

Method	LFW	YTF	MF1 Rank1	MF1 Veri.
Softmax Loss [23]	97.88	93.1	54.85	65.92
Softmax+Contrastive [30]	98.78	93.5	65.21	78.86
Triplet Loss [29]	98.70	93.4	64.79	78.32
L-Softmax Loss [24]	99.10	94.0	67.12	80.42
Softmax+Center Loss [42]	99.05	94.4	65.49	80.14
A-Softmax [23]	**99.42**	95.0	72.72	85.56
A-Softmax-NormFea	99.32	95.4	75.42	88.82
LMCL	99.33	**96.1**	**77.11**	**89.88**

Table 2. Comparison of the proposed LMCL with state-of-the-art loss functions in face recognition community. All the methods in this table are using the same training data and the same 64-layer CNN architecture.

Method	Training Data	#Models	LFW	YTF
Deep Face[35]	4M	3	97.35	91.4
FaceNet[29]	200M	1	99.63	95.1
DeepFR [27]	2.6M	1	98.95	97.3
DeepID2+[33]	300K	25	99.47	93.2
Center Face[42]	0.7M	1	99.28	94.9
Baidu[21]	1.3M	1	99.13	-
SphereFace[23]	0.49M	1	99.42	95.0
CosFace	5M	1	**99.73**	**97.6**

Table 3. Face verification (%) on the LFW and YTF datasets. "#Models" indicates the number of models that have been used in the method for evaluation.

Method	Protocol	MF1 Rank1	MF1 Veri.
SIAT MMLAB[42]	Small	65.23	76.72
DeepSense - Small	Small	70.98	82.85
SphereFace - Small[23]	Small	75.76	90.04
Beijing FaceAll V2	Small	76.66	77.60
GRCCV	Small	77.67	74.88
FUDAN-CS SDS[41]	Small	77.98	79.19
CosFace(Single-patch)	Small	77.11	89.88
CosFace(3-patch ensemble)	Small	**79.54**	**92.22**
Beijing FaceAll Norm 1600	Large	64.80	67.11
Google - FaceNet v8[29]	Large	70.49	86.47
NTechLAB - facenx large	Large	73.30	85.08
SIATMMLAB TencentVision	Large	74.20	87.27
DeepSense V2	Large	81.29	95.99
YouTu Lab	Large	83.29	91.34
Vocord - deepVo V3	Large	**91.76**	94.96
CosFace(Single-patch)	Large	82.72	96.65
CosFace(3-patch ensemble)	Large	84.26	**97.96**

Table 4. Face identification and verification evaluation on MF1. "Rank 1" refers to rank-1 face identification accuracy and "Veri." refers to face verification TAR under 10^{-6} FAR.

Method	Protocol	MF2 Rank1	MF2 Veri.
3DiVi	Large	57.04	66.45
Team 2009	Large	58.93	71.12
NEC	Large	62.12	66.84
GRCCV	Large	75.77	74.84
SphereFace	Large	71.17	84.22
CosFace (Single-patch)	Large	74.11	86.77
CosFace(3-patch ensemble)	Large	**77.06**	**90.30**

Table 5. Face identification and verification evaluation on MF2. "Rank 1" refers to rank-1 face identification accuracy and "Veri." refers to face verification TAR under 10^{-6} FAR .

unrestricted with labeled outside data protocol and report the result on 5,000 video pairs.

As shown in Table 3, the proposed CosFace achieves state-of-the-art results of 99.73% on LFW and 97.6% on YTF. FaceNet achieves the runner-up performance on LFW with the large scale of the image dataset, which has approximately 200 million face images. In terms of YTF, our model reaches the first place over all other methods.

4.4.2 Evaluation on MegaFace

MegaFace [17, 25] is a very challenging testing benchmark recently released for large-scale face identification and verification, which contains a gallery set and a probe set. The gallery set in Megaface is composed of more than 1 million face images. The probe set has two existing databases: Facescrub [26] and FGNET [1]. In this study, we use the Facescrub dataset (containing 106,863 face images of 530 celebrities) as the probe set to evaluate the performance of our approach on both Megaface Challenge 1 and Challenge 2.

MegaFace Challenge 1 (MF1). On the MegaFace Challenge 1 [17], The gallery set incorporates more than 1 million images from 690K individuals collected from Flickr photos [36]. Table 4 summarizes the results of our models trained on two protocols of MegaFace where the training dataset is regarded as small if it has less than 0.5 million images, large otherwise. The CosFace approach shows its superiority for both the identification and verification tasks on both the protocols.

MegaFace Challenge 2 (MF2). In terms of MegaFace Challenge 2 [25], all the algorithms need to use the training data provided by MegaFace. The training data for Megaface Challenge 2 contains 4.7 million faces and 672K identities, which corresponds to the large protocol. The gallery set has 1 million images that are different from the challenge 1 gallery set. Not surprisingly, Our method wins the first place of challenge 2 in table 5, setting a new state-of-the-art with a large margin (1.39% on rank-1 identification accuracy and 5.46% on verification performance).

5. Conclusion

In this paper, we proposed an innovative approach named LMCL to guide deep CNNs to learn highly discriminative face features. We provided a well-formed geometrical and theoretical interpretation to verify the effectiveness of the proposed LMCL. Our approach consistently achieves the state-of-the-art results on several face benchmarks. We wish that our substantial explorations on learning discriminative features via LMCL will benefit the face recognition community.

References

[1] *FG-NET Aging Database,http://www.fgnet.rsunit.com/*. 8

[2] P. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997. 1

[3] J. Cai, Z. Meng, A. S. Khan, Z. Li, and Y. Tong. Island Loss for Learning Discriminative Features in Facial Expression Recognition. *arXiv preprint arXiv:1710.03144*, 2017. 2

[4] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *arXiv preprint arXiv:1704.01719*, 2017. 2

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[6] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 2

[7] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 2

[8] M. A. Hasnat, J. Bohne, J. Milgram, S. Gentric, and L. Chen. von Mises-Fisher Mixture Model-based Deep learning: Application to Face Verification. *arXiv preprint arXiv:1706.04264*, 2017. 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6

[10] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2015. 2

[11] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *International Conference on Computer Vision Workshops (ICCVW)*, 2015. 2

[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. *arXiv preprint arXiv:1709.01507*, 2017. 1

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Technical Report 07-49, University of Massachusetts, Amherst*, 2007. 2, 7

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 2016 ACM on Multimedia Conference (ACM MM)*, 2014. 6

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2

[16] K. Zhang, Z. Zhang, Z. Li and Y. Qiao. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *Signal Processing Letters*, 23(10):1499–1503, 2016. 6

[17] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6, 7, 8

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1, 2

[19] Z. Li, D. Lin, and X. Tang. Nonparametric discriminant analysis for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:755–761, 2009. 1

[20] Z. Li, W. Liu, D. Lin, and X. Tang. Nonparametric subspace analysis for face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1

[21] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015. 8

[22] W. Liu, Z. Li, and X. Tang. Spatio-temporal embedding for statistical face recognition from video. In *European Conference on Computer Vision (ECCV)*, 2006. 1

[23] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4, 6, 7, 8

[24] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-Margin Softmax Loss for Convolutional Neural Networks. In *International Conference on Machine Learning (ICML)*, 2016. 2, 8

[25] A. Nech and I. Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 7, 8

[26] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347. IEEE, 2014. 8

[27] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 8

[28] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained Softmax Loss for Discriminative Face Verification. *arXiv preprint arXiv:1703.09507*, 2017. 2

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 8

[30] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 8

[31] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015. 2

[32] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[33] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 8

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 8

[36] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016. 8

[37] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991. 1

[38] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. NormFace: L_2 Hypersphere Embedding for Face Verification. In *Proceedings of the 2017 ACM on Multimedia Conference (ACM MM)*, 2017. 2

[39] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[40] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, Sept. 2004. 1

[41] Z. Wang, K. He, Y. Fu, R. Feng, Y.-G. Jiang, and X. Xue. Multi-task Deep Neural Network for Joint Face Recognition and Facial Attribute Prediction. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (ICMR)*, 2017. 2, 8

[42] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision (ECCV)*, pages 499–515, 2016. 2, 6, 8

[43] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2, 7

[44] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016. 1

[45] Y. Xiong, W. Liu, D. Zhao, and X. Tang. Face recognition via archetype hull ranking. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

[46] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 6, 7

[47] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range Loss for Deep Face Recognition with Long-tail. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[48] X. Zhe, S. Chen, and H. Yan. Directional Statistics-based Deep Metric Learning for Image Classification and Retrieval. *arXiv preprint arXiv:1802.09662*, 2018. 3

A. Supplementary Material

This supplementary document provides mathematical details for the derivation of the lower bound of the scaling parameter s (Equation 6 in the main paper), and the variable scope of the cosine margin m (Equation 7 in the main paper).

Proposition of the Scaling Parameter s

Given the normalized learned features x and unit weight vectors W, we denote the total number of classes as C where $C > 1$. Suppose that the learned features separately lie on the surface of a hypersphere and center around the corresponding weight vector. Let P_w denote the expected minimum posterior probability of the class center (*i.e.*, W). The lower bound of s is formulated as follows:

$$s \geq \frac{C-1}{C} \ln \frac{(C-1)P_W}{1-P_W}$$

Proof:

Let W_i denote the i-th unit weight vector. $\forall i$, we have:

$$\frac{e^s}{e^s + \sum_{j,j\neq i} e^{s(W_i^T W_j)}} \geq P_W, \tag{8}$$

$$1 + e^{-s} \sum_{j,j\neq i} e^{s(W_i^T W_j)} \leq \frac{1}{P_W}, \tag{9}$$

$$\sum_{i=1}^{C} (1 + e^{-s} \sum_{j,j\neq i} e^{s(W_i^T W_j)}) \leq \frac{C}{P_W}, \tag{10}$$

$$1 + \frac{e^{-s}}{C} \sum_{i,j,i\neq j} e^{s(W_i^T W_j)} \leq \frac{1}{P_W}. \tag{11}$$

Because $f(x) = e^{s \cdot x}$ is a convex function, according to Jensen's inequality, we obtain:

$$\frac{1}{C(C-1)} \sum_{i,j,i\neq j} e^{s(W_i^T W_j)} \geq e^{\frac{s}{C(C-1)} \sum_{i,j,i\neq j} W_i^T W_j}. \tag{12}$$

Besides, it is known that

$$\sum_{i,j,i\neq j} W_i^T W_j = (\sum_i W_i)^2 - (\sum_i W_i^2) \geq -C. \tag{13}$$

Thus, we have:

$$1 + (C-1) e^{-\frac{sC}{C-1}} \leq \frac{1}{P_W}. \tag{14}$$

Further simplification yields:

$$s \geq \frac{C-1}{C} \ln \frac{(C-1)P_W}{1-P_W}. \tag{15}$$

The equality holds if and only if every $W_i^T W_j$ is equal ($i \neq j$), and $\sum_i W_i = 0$. Because at most $K+1$ unit vectors are able to satisfy this condition in the K-dimension hyper-space, the equality holds only when $C \leq K+1$, where K is the dimension of the learned features.

Proposition of the Cosine Margin m

Suppose that the weight vectors are uniformly distributed on a unit hypersphere. The variable scope of the introduced cosine margin m is formulated as follows :

$$\begin{aligned} &0 \leq m \leq 1 - \cos \frac{2\pi}{C}, \quad (K=2) \\ &0 \leq m \leq \frac{C}{C-1}, \quad (K>2, C \leq K+1) \\ &0 \leq m \ll \frac{C}{C-1}, \quad (K>2, C > K+1) \end{aligned}$$

where C is the total number of training classes and K is the dimension of the learned features.

Proof:

For $K=2$, the weight vectors uniformly spread on a unit circle. Hence, $\max(W_i^T W_j) = \cos \frac{2\pi}{C}$. It follows $0 \leq m \leq (1 - \max(W_i^T W_j)) = 1 - \cos \frac{2\pi}{C}$.

For $K > 2$, the inequality below holds:

$$\begin{aligned} C(C-1)\max(W_i^T W_j) &\geq \sum_{i,j,i\neq j} W_i^T W_j \\ &= (\sum_i W_i)^2 - (\sum_i W_i^2) \\ &\geq -C. \end{aligned} \tag{16}$$

Therefore, $\max(W_i^T W_j) \geq \frac{-1}{C-1}$, and we have $0 \leq m \leq (1 - \max(W_i^T W_j)) \leq \frac{C}{C-1}$.

Similarly, the equality holds if and only if every $W_i^T W_j$ is equal ($i \neq j$), and $\sum_i W_i = 0$. As discussed above, this is satisfied only if $C \leq K+1$. On this condition, the distance between the vertexes of two arbitrary W should be the same. In other words, they form a regular simplex such as an equilateral triangle if $C = 3$, or a regular tetrahedron if $C = 4$.

For the case of $C > K+1$, the equality cannot be satisfied. In fact, it is unable to formulate the strict upper bound. Hence, we obtain $0 \leq m \ll \frac{C}{C-1}$. Because the number of classes can be much larger than the feature dimension, the equality cannot hold in practice.