Deep Metric Learning with Angular Loss

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu and Yuanqing Lin
Baidu Research
{wangjian33,zhoufeng09,wenshilei,liuxiao12,linyuanqing}@baidu.com

Abstract

The modern image search system requires semantic understanding of image, and a key yet under-addressed problem is to learn a good metric for measuring the similarity between images. While deep metric learning has yielded impressive performance gains by extracting high level abstractions from image data, a proper objective loss function becomes the central issue to boost the performance. In this paper, we propose a novel angular loss, which takes angle relationship into account, for learning better similarity metric. Whereas previous metric learning methods focus on optimizing the similarity (contrastive loss) or relative similarity (triplet loss) of image pairs, our proposed method aims at constraining the angle at the negative point of triplet triangles. Several favorable properties are observed when compared with conventional methods. First, scale invariance is introduced, improving the robustness of objective against feature variance. Second, a third-order geometric constraint is inherently imposed, capturing additional local structure of triplet triangles than contrastive loss or triplet loss. Third, better convergence has been demonstrated by experiments on three publicly available datasets.

1. Introduction

Metric learning for computer vision aims at finding appropriate similarity measurements between pairs of images that preserve desired distance structure. A good similarity can improve the performance of image search, particularly when the number of categories is very large [2] or unknown. Classical metric learning methods studied the case of finding a better Mahalanobis distance in linear space. However, linear transformation has a limited number of parameters and cannot model high-order correlations between the original data dimensions. With the ability of directly learning non-linear feature representation, deep metric learning has achieved promising results on various tasks, such as visual product search [1, 20, 17], face recognition [6, 30, 24], feature matching [7], fine-grained image classification [33, 38], zero-shot learning [11, 35] and collaborative filtering [13].

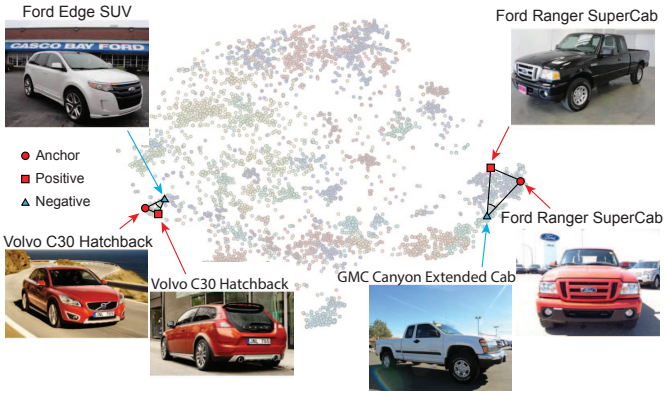

Figure 1. Example of feature embedding computed by t-SNE [32] for the Stanford car dataset [18], where the images of Ford Ranger SuperCab (right) have a more diverse distribution than Volvo C30 Hatchback (left). Conventional triplet loss has difficulty in dealing with such unbalanced intra-class variation. The proposed angular loss addresses this issue by minimizing the scale-invariant angle at the negative point.

Despite the various forms, the major work of deep metric learning can be categorized as minimizing either the contrastive loss (*a.k.a.*, Siamese network) [6] or the triplet loss [34, 5]. However, it has been widely noticed that directly optimizing distance-based objectives in deep learning framework is difficult, requiring many practical tricks, such as multi-task learning [1, 38] or hard negative mining [33, 8]. Recent work including the lifted structure [26] and the N-pair loss [25] proposed to more effectively mine relations among samples within a mini-batch. Nevertheless, all of these works rely on certain distance measurement between pairs of similar and dis-similar images. We hypothesize that the difficulty of training deep metric learning also comes from the limitation by defining the objective only in distance. First, distance metric is sensitive to scale change. Traditional triplet loss constrains the distance gap between dis-similar clusters. However, it is inappropriate to choose the same absolute margin for clusters in different scales of intra-class variation. For instance, Fig. 1 shows the t-SNE [32] feature embedding of Stanford car dataset [18], where the sample distribution of Ford Ranger SuperCabs

is much more diverse than Volvo C30 Hatchback. Second, distance only considers second-order information between samples. Optimizing distance-based objectives in stochastic training leads to sub-optimal convergence in high-order solution space.

To circumvent these issues, we propose a novel angular loss to augment conventional distance metric learning. The main idea is to encode the third-order relation inside triplet in terms of the angle at the negative point. By constraining the upper bound of the angle, our method pushes the negative point away from the center of positive cluster, and drags the positive points closer to each other. Our idea is analogous to the usage of high-order information for augmenting pair-wise constraints in the domain of graph matching [9] and Markov random fields [10]. To the best of our knowledge, this is the first work to explore angular constraints in deep metric learning. In particular, the proposed angular loss improves traditional distance-based loss in two aspects. First, compared to distance-based metric, angle is not only rotation-invariant but also scale-invariant by nature. This renders the objective more robust against the variation of local feature map. For instance, the two triplets shown in Fig. 1 are quite different in their scales. It is more reasonable to constrain the angle that is proportional to the relative ratio between Euclidean distances. Second, angle defines the third-order triangulation among three points. Given the same triplet, angular loss describes its local structure more precisely than distance-based triplet loss. Our idea is general and can be potentially combined with existing metric learning frameworks. The experimental study shows it achieves substantial improvement over state-of-the-arts methods on several benchmark datasets.

2. Related work

Metric learning has been a long-standing problem in machine learning and computer vision. The simplest form of metric learning may be considered as learning the Mahalanobis distance between pairs of points. It has a deep connection with classical dimension reduction methods such as PCA, LLE and clustering problems but in a discriminative setting. An exhaustive review of previous work is beyond the scope of this paper. We refer to the survey of Kulis *et al*. [19] on early works of metric learning. Here we focus on the two main streams in deep metric learning, contrastive embedding and triplet embedding, and their recent variants used in computer vision.

The seminal work of Siamese network [4] consists of two identical sub-networks that learn contrastive embedding from a pair of samples. The distance between a positive pair is minimized and small distance between a negative pair is penalized, such that the derived distance metric should be smaller for pairs from the same class, and larger for pairs from different classes. It was originally designed for signature verification [4], but gained a lot of attention recently due to its superior performance in face verification [6, 30, 28, 36].

Despite its great success, contrastive embedding requires that training data contains real-valued precise pair-wise similarities or distances, which is usually not available in practice. To address this issue, triplet embedding [23] is proposed to explore the relative similarity of different pairs and it has been widely used in image retrieval [33, 5] and face recognition [24]. A triplet is made up of three samples from two different classes, that jointly constitute a positive pair and a negative pair. The positive pair distance is encouraged to be smaller than the negative pair distance, and a soft nearest neighbor classification margin is maximized by optimizing a hinge loss.

Compared to softmax loss, it has been shown that Siamese network or triplet loss is much more difficult to train in practice. To make learning more effective and efficient, hard sample mining which only focuses on a subset of samples that are considered hard is usually employed. For instance, FaceNet [24] suggested an online strategy by associating each positive pair in the minibatch with a semi-hard negative example. Wang *et al*. [33] designed a more effective sampling strategy to draw out-class and in-class negative images to avoid overfitting for training triplet loss. To more effectively bootstrap a large flower dataset, Cui *et al*. [8] utilized the hard negative images labeled by humans, which are often neglected in traditional dataset construction. Huang *et al*. [14] introduced a position-dependent deep metric unit, which can be used to select hard samples to guide the deep embedding learning in an online and robust manner. More recently, Yuan *et al*. [37] proposed a cascade framework that can mine hard examples with increasing complexities.

Recently, there are also some works on designing new loss functions for deep metric embedding. A simple yet effective way is to jointly train embedding loss with classification loss. With additional supervision, the improvement of triplet loss has been evidenced in face verification [28], fine-grained object recognition [38] and product search problems [1]. However, these methods still suffer from the limitation of the conventional sampling that focuses only on the relation within each triplet. To fix this issue, Song *et al*. [26] proposed the lifted structure to enable updating dense pair combinations in the mini-batch. Sohn [25] further extended the triplet loss into N-pair loss, which significantly improves upon the triplet loss by pushing away multiple negative examples jointly at each update. In addition to these efforts that only explore local relation inside each mini-batch, another direction of work is designed to optimize clustering-like metric that is aware of the global structure of all training data. Early methods such as neighborhood components analysis (NCA) [12, 23] can directly

optimize leave-one-out nearest-neighbor classification loss. When applied to mini-batch training, however, NCA is limited as it requires to see the entire training data in each iteration. Rippel *et al*. [21] improved NCA by maintaining an model of the distributions of the different classes in feature space. The class distribution overlap is then penalized to achieve discrimination. More recently, Song *et al*. [27] proposed a new metric learning framework which encourages the network to learn an embedding function that directly optimizes a clustering quality metric. Nevertheless, all above-mentioned losses are defined in term of distances of points, and very few [31] has considered other possible forms of loss. Our work re-defines the core component of metric learning loss using angle instead of distance, and we show it can be easily adapted into existing architectures such as N-pair loss to further improve their performance.

3. Proposed method

In this section, we present a novel angular loss to augment conventional deep metric learning. We first review the conventional triplet loss in its mathematical form. We then derive the angular loss by constructing a stable triplet triangle. Finally, we detail the optimization of the angular loss on a mini-batch.

3.1. Review of triplet loss

Suppose that we are given a set of training images $\{(\mathbf{x}, y), \cdots\}$ of K classes, where $\mathbf{x} \in \mathbb{R}^D$ denotes the feature embedding of each sample extracted by CNN and $y \in \{1, \cdots, K\}$ its label. At each training iteration, we sample a mini-batch of triplets, each of which $\mathcal{T} = (\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$ consists of an anchor point $\mathbf{x}_a$, associated with a pair of positive $\mathbf{x}_p$ and negative $\mathbf{x}_n$ samples, whose labels satisfy $y_a = y_p \neq y_n$. The goal of triplet loss is to push away the negative point $\mathbf{x}_n$ from the anchor $\mathbf{x}_a$ by a distance margin $m > 0$ compared to the positive $\mathbf{x}_p$:

$$\|\mathbf{x}_a - \mathbf{x}_p\|^2 + m \leq \|\mathbf{x}_a - \mathbf{x}_n\|^2. \tag{1}$$

For instance, as shown in Fig. 2, we expect the anchor $\mathbf{x}_a$ to stay closer to the positive $\mathbf{x}_p$ compared to the negative $\mathbf{x}_n$. To enforce this constraint, a common relaxation of Eq. 1 is the minimization of the following hinge loss,

$$l_{tri}(\mathcal{T}) = \Big[\|\mathbf{x}_a - \mathbf{x}_p\|^2 - \|\mathbf{x}_a - \mathbf{x}_n\|^2 + m\Big]_+, \tag{2}$$

where the operator $[\cdot]_+ = \max(0, \cdot)$ denotes the hinge function. It is worth mentioning that the feature map often needs to be normalized to have unit length, *i.e*., $\|\mathbf{x}\| = 1$, in order to be robust to the variation in image illumination and contrast.

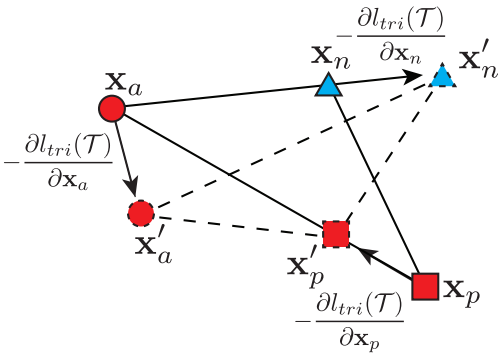

Figure 2. Illustration of the triplet loss and its gradient on a synthetic example.

To optimize Eq. 2, we can calculate its gradient with respect to the three samples of triplet respectively as

$$\begin{aligned}
\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_n} &= 2(\mathbf{x}_a - \mathbf{x}_n), \\
\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_p} &= 2(\mathbf{x}_p - \mathbf{x}_a), \\
\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_a} &= 2(\mathbf{x}_n - \mathbf{x}_p),
\end{aligned} \tag{3}$$

if the constraint (Eq. 1) is violated, or zero otherwise.

It is widely observed that stochastic gradient descent converges poorly on optimizing the triplet loss. There are a few reasons contributing to this difficulty: First, it is impractical to enumerate all possible triplets due to the cubic sampling size. Therefore, it calls for an effective sampling strategy to ensure the triplet quality and learning efficiency. Second, the goal of the objective (Eq. 2) is to separate clusters by a distance margin m. However, it is inappropriate to apply the single global margin m on the inter-class gap as the intra-class distance can vary dramatically in real-world tasks. Third, the gradient (Eq. 3) derived for each point only takes its pair-wise relation with the second point, but fails to consider the interaction with the third point. Consider the negative point $\mathbf{x}_n$ in Fig. 2 for an example. Its gradient $2(\mathbf{x}_a - \mathbf{x}_n)$ may not be optimal without the guarantee of moving away from the class which both the anchor $\mathbf{x}_a$ and positive sample $\mathbf{x}_p$ belong to.

3.2. Angular loss

To alleviate the problems elaborated above, a variety of techniques [1, 38, 33, 8, 26, 25] have been proposed in the last few years. However, the fundamental component in the loss definition, *i.e*., the pair-wise distance between points, has rarely been changed. Instead, this section introduces an angular loss that leads to a novel solution to improve deep metric learning.

Let's first consider the triplet example shown in Fig. 3a, where the triplet $\mathcal{T} = (\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$ forms the triangle

$\triangle apn$, whose edges are denoted as $\mathbf{e}_{an} = \mathbf{x}_a - \mathbf{x}_n$, $\mathbf{e}_{pn} = \mathbf{x}_p - \mathbf{x}_n$ and $\mathbf{e}_{ap} = \mathbf{x}_a - \mathbf{x}_p$ respectively. The original triplet constraint (Eq. 1) penalizes a longer edge $\mathbf{e}_{an}$ compared to the one $\mathbf{e}_{ap}$ on the bottom. Because the anchor and positive samples share the same label, we can derive a symmetrical triplet constraint that enforces $\|\mathbf{e}_{ap}\| + m \leq \|\mathbf{e}_{pn}\|$. According to the cosine rule, it can be proved that the angle $\angle n$ surrounded by the longer edges $\mathbf{e}_{an}$ and $\mathbf{e}_{pn}$ has to be the smallest one, *i.e.*, $\angle n \leq \min(\angle a, \angle p)$. Furthermore, because $\angle n + \angle a + \angle p = 180°$, $\angle n$ has to be less than $60°$. This fact motivates us to constrain the upper bound of $\angle n$ for each triplet triangle,

$$\angle n \leq \alpha, \tag{4}$$

where $\alpha > 0$ is a pre-defined parameter. Intuitively, this constraint selects the triplet that forms a skinny triangle whose shortest edge $\mathbf{e}_{ap}$ connects nodes of the same class. Compared to the traditional constraint (Eq. 1) that is defined on the absolute distance between points, the proposed angular constraint offers three advantages: 1) Angle is a similarity-transform-invariant metric, proportional to the relative comparison of triangle edges. With a fixed margin α, Eq. 4 always holds for any re-scaling of the local feature map. 2) The cosine rule determines the calculation of $\angle n$ involves all the three edges of the triangle. In contrast, the original triplet only takes two edges into account. The additional constraint improves the robustness and effectiveness of the optimization. 3) In the original triplet constraint (Eq. 1), it is difficult to choose a proper distance margin m without meaningful reference. By comparison, setting α in the angular constraint is an easier task because it has concrete and interpretable meaning in geometry.

However, a straightforward implementation of Eq. 4 becomes unstable in some special case. Consider the triangle shown in Fig. 3a, where $\angle a > 90°$. By enforcing Eq. 4 to reduce $\angle n$, the negative point $\mathbf{x}_n$ would be potentially dragged towards $\mathbf{x}'_n$, which is closer to the anchor point $\mathbf{x}_a$. This result contradicts our original goal of enlarging the distance between points of different classes. To fix this issue, we re-construct the triplet triangle to make Eq. 4 more stable. Our intuition is to model the relation between the negative $\mathbf{x}_n$ with the local sample distribution defined by the anchor $\mathbf{x}_a$ and the positive $\mathbf{x}_p$, shown in Fig. 3b. A natural approximation to this distribution is the circumcircle $\mathcal{C}$ passing through $\mathbf{x}_a$ and $\mathbf{x}_p$, centered at the middle $\mathbf{x}_c = (\mathbf{x}_a + \mathbf{x}_p)/2$. We then introduce a hyper-plane $\mathcal{P}$, which is perpendicular to the edge $\mathbf{e}_{nc} = \mathbf{x}_n - \mathbf{x}_c$ at $\mathbf{x}_c$. The hyper-plane $\mathcal{P}$ intersects the circumcircle $\mathcal{C}$ at two nodes, one of which is denoted as $\mathbf{x}_m$. Based on these auxiliary structures, we define the new triangle $\triangle_{mcn}$ by shifting the anchor $\mathbf{x}_a$ and positive $\mathbf{x}_p$ to $\mathbf{x}_c$ and $\mathbf{x}_m$ respectively. Given the new triangle, we re-formulate Eq. 4 to constrain the angle $\angle n'$ closed by the edge of $\mathbf{e}_{nc}$ and $\mathbf{e}_{nm}$ to be less than a

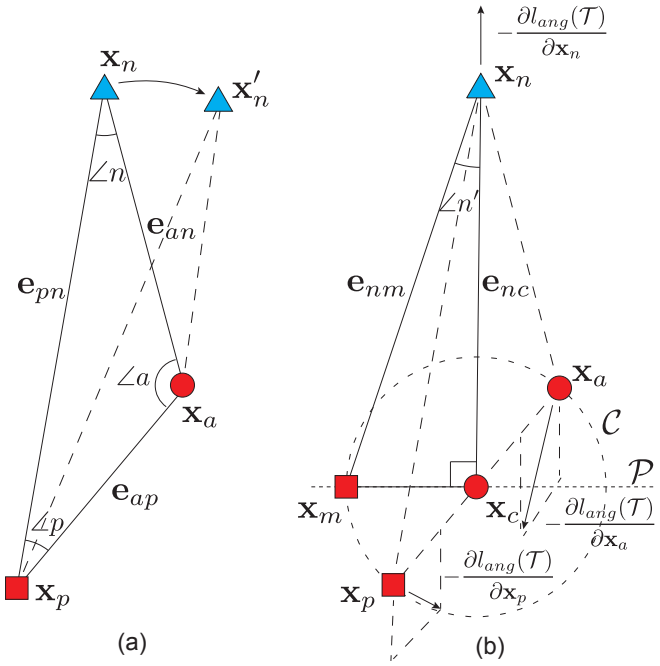

Figure 3. Illustration of the angular constraint on a synthetic triplet where $\angle a > 90°$. (a) Directly minimizing $\angle n$ is unstable as it would drag $\mathbf{x}_n$ closer to $\mathbf{x}_a$. (b) The more stable $\angle n'$ defined by re-constructing the triangle $\triangle_{mcn}$.

pre-define upper bound α, *i.e.*,

$$\tan \angle n' = \frac{\|\mathbf{x}_m - \mathbf{x}_c\|}{\|\mathbf{x}_n - \mathbf{x}_c\|} = \frac{\|\mathbf{x}_a - \mathbf{x}_p\|}{2\|\mathbf{x}_n - \mathbf{x}_c\|} \leq \tan \alpha, \tag{5}$$

where $\|\mathbf{x}_m - \mathbf{x}_c\|$ is the radius of the circumcircle $\mathcal{C}$, which equals to $\|\mathbf{x}_a - \mathbf{x}_p\|/2$.

Inspired by the triplet loss (Eq. 2), we seek for the optimum embedding such that the samples of different classes can be separated well as the angular constraint (Eq. 5) describes. In a nutshell, our angular loss consists of minimizing the following hinge loss,

$$l_{ang}(\mathcal{T}) = \Big[\|\mathbf{x}_a - \mathbf{x}_p\|^2 - 4\tan^2\alpha \|\mathbf{x}_n - \mathbf{x}_c\|^2\Big]_+. \tag{6}$$

To better understand the effect of optimizing the angular loss, we can investigate the gradient of l_{ang} with respect to $\mathbf{x}_a$, $\mathbf{x}_p$ and $\mathbf{x}_n$, which are

$$\begin{aligned}
\frac{\partial l_{ang}(\mathcal{T})}{\partial \mathbf{x}_a} &= 2(\mathbf{x}_a - \mathbf{x}_p) - 2\tan^2\alpha(\mathbf{x}_a + \mathbf{x}_p - 2\mathbf{x}_n), \\
\frac{\partial l_{ang}(\mathcal{T})}{\partial \mathbf{x}_p} &= 2(\mathbf{x}_p - \mathbf{x}_a) - 2\tan^2\alpha(\mathbf{x}_a + \mathbf{x}_p - 2\mathbf{x}_n), \\
\frac{\partial l_{ang}(\mathcal{T})}{\partial \mathbf{x}_n} &= 4\tan^2\alpha\Big[(\mathbf{x}_a + \mathbf{x}_p) - 2\mathbf{x}_n\Big],
\end{aligned} \tag{7}$$

if $\angle n'$ is larger than α, or zero otherwise. As illustrated in Fig. 3b, the gradient pushes the negative point $\mathbf{x}_n$ away

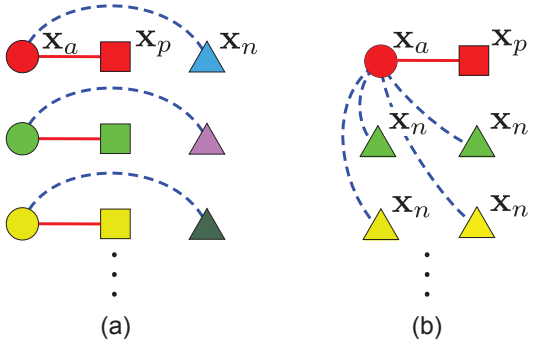

Figure 4. Comparison between different sampling methods. For each node, we use color to indicate the class label and shape for its role (*i.e.*, anchor, positive or negative) in triplet. (a) Traditional triplet sampling. (b) N-pair sampling. To keep plot clean, we only show the connection inside one tuplet.

from $\mathbf{x}_c$, the center of local cluster defined by $\mathbf{x}_a$ and $\mathbf{x}_p$. In addition, the anchor $\mathbf{x}_a$ and the positive $\mathbf{x}_p$ are dragged towards each other. Compared to the original triplet loss whose gradients (Eq. 3) only depend on two points, the gradients in Eq. 7 are much more robust as they consider all the three points simultaneously.

3.3. Implementation details

Eq. 6 defines the angular loss on a triplet. When optimizing a mini-batch containing multiple triplets, we found our method can be further improved in two ways.

First, we enhance the mini-batch optimization by making the full use of the batch. As illustrated in Fig. 4a, the conventional sample strategy constructs a mini-batch as multiple disjoint triplets without interaction among them. This poses a large bottleneck in optimization as it can only encode a limited amount of information. To allow joint comparison among all samples in the batch, we follow the sampling strategy proposed in N-pair loss [25] to construct tuplets with multiple negative points. More concretely, we first draw $N/2$ different classes, from each of which we then randomly sample two training images. The main benefit behind N-pair sampling is that it can avoid the quadratic possible combinations of tuplets. For instance, as shown in Fig. 4b, given a batch with N samples $\mathcal{B} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, there are in total N tuplets, each of which is composed by a pair of anchor $\mathbf{x}_a \in \mathcal{B}$ and positive $\mathbf{x}_p \in \mathcal{B}$ of the same class, and $N-2$ negative from other classes.

Second, a direct extension of Eq. 6 to consider more than one negative point would result in a very non-smooth objective function. Inspired by recent work [26, 25, 27], we replace the original hinge loss with its smooth upper bound, *i.e.*, $\log(\exp(y_1)+\exp(y_2)) \geq \max(y_1, y_2)$. By assuming feature is of unit length (*i.e.*, $\|\mathbf{x}\| = 1$) in Eq. 6, we derive the angular loss for a batch $\mathcal{B}$ using the following *log-sum-exp* formulation:

$$l_{ang}(\mathcal{B}) = \frac{1}{N}\sum_{\mathbf{x}_a\in\mathcal{B}}\left\{\log\left[1+\sum_{\substack{\mathbf{x}_n\in\mathcal{B}\\ y_n\neq y_a, y_p}}\exp\left(f_{a,p,n}\right)\right]\right\}, \tag{8}$$

where in $f_{a,p,n}$, we drop the constant terms depending on the value of $\|\mathbf{x}\|$ in a similar spirit to N-pair loss [25], *i.e.*,

$$f_{a,p,n} = 4\tan^2\alpha(\mathbf{x}_a+\mathbf{x}_p)^T\mathbf{x}_n - 2(1+\tan^2\alpha)\mathbf{x}_a^T\mathbf{x}_p.$$

Our work on angular loss explores the third-order relations beyond the scope of the well-studied pair-wise distance. Due to its flexibility and generality, we can easily combine the angular constraint with traditional distance metric loss to boost the overall performance. As an example, we mainly investigate the combination with the N-pair loss [25], one of the latest work for deep metric learning,

$$l_{npair\&ang}(\mathcal{B}) = l_{npair}(\mathcal{B}) + \lambda l_{ang}(\mathcal{B}), \tag{9}$$

where $l_{npair}(\mathcal{B})$ denotes the original N-pair loss as,

$$l_{npair}(\mathcal{B}) = \frac{1}{N}\sum_{\mathbf{x}_a\in\mathcal{B}}\left\{\log\left[1+\sum_{\substack{\mathbf{x}_n\in\mathcal{B}\\ y_n\neq y_a, y_p}}\exp\left(\mathbf{x}_a^T\mathbf{x}_n - \mathbf{x}_a^T\mathbf{x}_p\right)\right]\right\}, \tag{10}$$

and λ is a trade-off weight between N-pair and the angular loss. In all experiments, we always set $\lambda = 2$ as it consistently yields promising result.

4. Experiments

In this section, we evaluate deep metric learning algorithms on both image retrieval and clustering tasks. Our method has been shown to achieve state-of-the-art performance on three public benchmark datasets.

4.1. Benchmark datasets

We conduct our experiments on three public benchmark datasets. For all datasets, we follow the conventional protocol of splitting training and testing:

CUB-200-2011 [3] dataset has 200 species of birds with 11,788 images included, where the first 100 species (5,864 images) are used for training and the remaining 100 species (5,924 images) are used for testing.

Stanford Car [18] dataset is composed by 16,185 cars images of 196 classes. We use the first 98 classes (8,054 images) for training and the other 98 classes (8,131 images) for testing.

Online Products [26] dataset contains 22,634 classes with 120,053 product images in total, where the first 11,318 classes (59,551 images) are used for training and the rest classes (60,502 images) are used for testing.

4.2. Baselines

In order to evaluate the superiority of the proposed method, we compare with three baselines:

Triplet Loss: We implement the standard triplet embedding by optimizing Eq. 2. To be fair in comparison, we apply triplet loss embedding with two sampling strategies. Following the most standard setting, the mini-batch of **Triplet-I (T-I)** was constructed by sampling disjoint triplets as illustrated in Fig. 4a. In the second case of **Triplet-II (T-II)**, we optimize Eq. 2 using the N-pair sampling as shown in Fig. 4b to keep consistent with the angular loss.

Lifted Structure (LS) [26]: We adopt the open-source code from the authors' website with the default parameters used in the paper.

N-pair Loss (NL) [25]: We implement N-pair loss (Eq. 10) closely following the illustration of the paper. We found our implementation achieved similar results as reported in the paper.

For our method, we implement two versions, **Angular Loss (AL)** and **N-pair & Angular Loss (NL&AL)**, that optimize Eq. 8 and Eq. 9 respectively. To be comparable with prior work, we employ the N-pair sampling (Fig. 4b) shared by the baselines of **Triplet-II** and **N-pair Loss**.

As the focus of this work is the similarity measure, we did not employ any hard negative mining strategies to complicate the comparison. But it is worth mentioning that our work can be easily combined with any hard negative mining method.

4.3. Evaluation metrics

Following the standard protocol used in [26, 25], we evaluate the performance of different methods in both retrieval and clustering tasks. We split each dataset into two sets of disjoint classes, one for training and the other for testing the retrieval and clustering performance of the unseen classes. For retrieval task, we calculate the percentage of the testing examples whose R nearest neighbors contain at least one example of the same class. This quantity is also known as Recall@R, the defacto metric [15] for image retrieving evaluation. For clustering evaluation, we adopt the code from [26] by clustering testing examples using the k-means algorithm. The quality of clustering is reported in terms of the standard F_1 and NMI metrics. See [26] for their detailed definition.

4.4. Training setup

The Caffe package [16] is used throughout the experiments. All images are normalized to 256-by-256 before further processing. The embedding size is set to $D = 512$ for all embedding vectors, and no normalization is conducted before computing loss. We omit the comparison on different embedding sizes as the performance change is minor. This fact is also evidenced in [26]. GoogLeNet [29] pretrained on ImageNet ILSVRC dataset [22] is used for initialization and a randomly initialized fully connected layer is added. The new layer is optimized with 10 times larger learning rate than the other layers. We fix the base learning rate to 10^{-4} for all datasets except for the CUB-200-2011 dataset, for which we use a smaller rate 10^{-5} as it has fewer images and is more likely to meet the overfitting problem. We use SGD with 20k training iterations and 128 mini-batch size. Standard random crop and random horizontal mirroring are used for data augmentation. Notice that our method incurs negligible computational cost compared to traditional triplet loss. Therefore, the training time is almost same as other baselines.

4.5. Result analysis

Tables 1, 2 and 3 compare our method with all baselines in both clustering and retrieval tasks. These tables show that the two recent baselines, lifted structure (LS) [26] and N-pair loss (NL) [25], can always improve the standard triplet loss (T-I and T-II). In particular, N-pair achieves a larger margin in improvement because of the advance in its loss design and batch construction. Compared to previous work, the proposed angular loss (AL) consistently achieves better results on all three benchmark datasets. It is important to notice that the proposed angular loss (AL) employs the same sampling strategies as triplet loss (T-II) and N-pair loss (NL). This clearly indicates the superiority of the new loss for solving deep metric learning problem. By integrating with the original N-pair loss, the joint optimization of angular loss in NL&AL can lead to the best performance among all the methods in all metrics.

Fig. 5 compares NL&AL with N-pair loss on the task of image retrieval. As it can be observed, the proposed NL&AL learns a more discriminative feature that helps in identifying the correct images especially when the intra-class variance is large. For example, given a query image of FIAT 500 Convertible 2012 at the fourth row of Fig. 5 on the right side, the top-5 images retrieved by NL&AL contain four successful matches that belong to the same class as the query, while N-pair method fails to identify them. In addition, Fig. 6 visualizes the feature embedding computed by our method (NL&AL) in 2-D using t-SNE [32]. We highlight several representative classes by enlarging the corresponding regions in the corners. Despite the large pose and appearance variation, our method effectively generates a compact feature mapping that preserves semantic similarity.

A key parameter of our method is the margin α, that determines to what degree the constraint (Eq. 5) would be activated. Table 4 and Table 5 study the impact of choosing different α for the retrieval task on the Stanford car and online product datasets, respectively. Choosing $\alpha = 45^\circ$ for Stanford car and $\alpha = 36^\circ$ for online product lead to the

Method	Clustering (%)		Recall@R (%)			
	NMI	F_1	R=1	R=2	R=4	R=8
T-I	53.7	19.7	42.2	54.4	66.2	76.7
T-II	54.1	20.0	42.8	54.9	66.2	77.6
LS	56.2	22.7	46.5	58.1	69.8	80.2
NL	60.2	28.2	51.9	64.3	74.9	83.2
AL	61.0	**30.2**	53.6	65.0	75.3	83.7
NL&AL	**61.1**	29.4	**54.7**	**66.3**	**76.0**	**83.9**

Table 1. Comparison of clustering and retrieval on the CUB-200-2011 [3] dataset.

Method	Clustering (%)		Recall@R (%)			
	NMI	F_1	R=1	R=2	R=4	R=8
T-I	53.8	18.7	45.5	59.0	71.0	80.8
T-II	54.3	19.6	46.3	59.9	71.4	81.3
LS	55.1	21.5	48.3	61.1	71.8	81.1
NL	62.7	31.8	68.9	78.9	85.8	90.9
AL	62.4	31.8	71.3	80.7	87.0	91.8
NL&AL	**63.2**	**32.2**	**71.4**	**81.4**	**87.5**	**92.1**

Table 2. Comparison of clustering and retrieval on the Stanford car [18] dataset.

Method	Clustering (%)		Recall@R (%)			
	NMI	F_1	R=1	R=10	R=100	R=1000
T-I	86.2	19.9	56.5	74.7	88.3	96.2
T-II	86.4	21.0	58.1	76.0	89.1	96.4
LS	87.4	24.7	63.0	80.5	91.7	97.5
NL	87.7	26.3	66.9	83.0	92.3	97.7
AL	87.8	26.5	67.9	83.2	92.2	97.7
NL&AL	**88.6**	**29.9**	**70.9**	**85.0**	**93.5**	**98.0**

Table 3. Comparison of clustering and retrieval on the online products [26] dataset.

best performance for the method of NL&AL. We found that our method performs consistently well in all three dataset for $36^\circ \leq \alpha \leq 55^\circ$. It deserves to be mentioned that, without integrating with NL, AL preforms comparably with NL, and even better when mining a proper value of α, which is shown in Table 5.

5. Conclusion

In this paper, we propose a novel angular loss for deep metric learning. Unlike most methods that formulate objective based on distance, we resort to constrain the angle of the triplet triangle in the loss. Compared to pair-wise distance,

NL&AL(α)	Recall@R (%)			
	R=1	R=2	R=4	R=8
$\alpha = 36^\circ$	69.9	79.7	86.8	91.8
$\alpha = 42^\circ$	70.7	80.5	87.2	91.9
$\alpha = 45^\circ$	**71.4**	**81.4**	**87.5**	**92.1**
$\alpha = 48^\circ$	71.3	80.4	87.0	91.9
$\alpha = 55^\circ$	69.0	78.1	85.3	90.8

Table 4. Comparison of different values for α for our method on Stanford car dataset.

Method	NL	NL&AL($\alpha = 45^\circ$)	NL&AL($\alpha = 36^\circ$)
Recall@1 (%)	66.9	69.2	**70.9**
Method	NL	AL($\alpha = 45^\circ$)	AL($\alpha = 36^\circ$)
Recall@1 (%)	66.9	66.4	**67.9**

Table 5. Comparison of different values for α for our method on the online product dataset.

angle is a rotation and scale invariant metric, rendering the objective more robust against the large variation of feature map in real data. In addition, the value of angle encodes the triangular geometry of three points simultaneously. Given the same triplet, it offers additional source of constraints to ensure that dis-similar points can be separated. Furthermore, we show how the angular loss can be easily integrated into other frameworks such as N-pair loss [25]. The superiority of our method over existing state-of-the-art work is verified on several benchmark datasets.

In the future, we hope to extend our work in two directions. First, our method origins from the triplet loss and leverages the third-order relation among three points. It is interesting to consider more general case with four or more samples. Previous work [38, 14] studied the case of quadruplet but still employed certain distance-based objectives. One possible extension of our idea on quadruplet is to construct a triangular pyramid and constrain the angle between the side edge and the plane on the bottom. Second, it is beneficial to combine our method with other practical tricks such as hard negative mining [37] or new clustering-like frameworks [21, 27].

References

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph.*, 34(4):98:1–98:10, 2015.

[2] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, pages 730–738, 2015.

[3] S. Branson, G. V. Horn, C. Wah, P. Perona, and S. Belongie. The ignorant led by the blind: A hybrid human-machine vi-

Figure 5. Comparison of queries and top-5 retrievals between N-pair (NP) and our method (NP&AL). From top to bottom, we plot two examples for the CUB-200-2011, Stanford car and online products dataset respectively. The retrieved images pointed by an arrow are the ones that belong to the same class as the query.

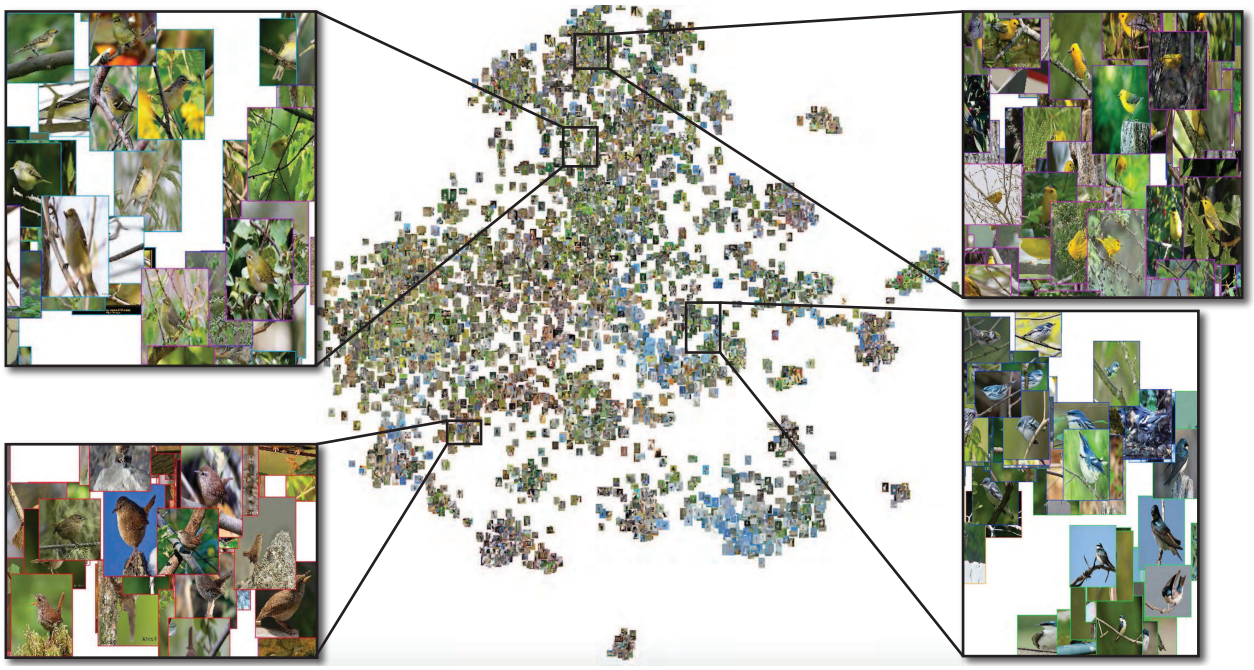

Figure 6. Visualization of feature embedding computed by our method (NP&AL) using t-SNE on the CUB-200-2011 dataset.

sion system for fine-grained categorization. *Int. J. Comput. Vis.*, 108(1-2):3–29, 2014.

[4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a Siamese time delay neural network. In *NIPS*, 1993.

[5] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[7] C. B. Choy, J. Gwak, S. Savarese, and M. K. Chandraker. Universal correspondence network. In *NIPS*, 2016.

[8] Y. Cui, F. Zhou, Y. Lin, and S. J. Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *CVPR*, 2016.

[9] O. Duchenne, F. R. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2383–2395, 2011.

[10] A. Fix, A. Gruber, E. Boros, and R. Zabih. A graph cut algorithm for higher-order markov random fields. In *ICCV*, pages 1020–1027, 2011.

[11] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. DeViSE: A deep visual-semantic embedding model. In *NIPS*, 2013.

[12] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighborhood components analysis. In *NIPS*, 2004.

[13] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin. Collaborative metric learning. In *WWW*, 2017.

[14] C. Huang, C. C. Loy, and X. Tang. Local similarity-aware deep feature embedding. In *NIPS*, pages 1262–1270, 2016.

[15] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014.

[17] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, 2015.

[18] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013.

[19] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[20] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via CNN image purification. *ACM Trans. Graph.*, 34(6):234:1–234:12, 2015.

[21] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. In *CVPR*, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[23] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

[24] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[25] K. Sohn. Improved deep metric learning with multi-class N-pair loss objective. In *NIPS*, 2016.

[26] H. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[27] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Learnable structured clustering framework for deep metric learning. *CoRR*, abs/1612.01213, 2016.

[28] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[30] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[31] E. Ustinova and V. S. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, pages 4170–4178, 2016.

[32] L. van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[33] 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 *CVPR*, 2014.

[34] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[35] J. Weston, S. Bengio, and N. Usunier. WSABIE: scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011.

[36] D. Yi, Z. Lei, and S. Z. Li. Deep metric learning for practical person re-identification. *CoRR*, abs/1407.4979, 2014.

[37] Y. Yuan, K. Yang, and C. Zhang. Hard-aware deeply cascaded embedding. *CoRR*, abs/1611.05720, 2016.

[38] X. Zhang, F. Zhou, Y. Lin, and S. Zhang. Embedding label structures for fine-grained feature representation. In *CVPR*, 2016.