Deep Asymmetric Metric Learning via Rich Relationship Mining

Xinyi Xu, Yanhua Yang, Cheng Deng, Feng Zheng

1School of Electronic Engineering, Xidian University, Xian 710071, China
2Department of Computer Science and Engineering, Southern University of Science and Technology

xyxu.xd@gmail.com, yanhyang@xidian.edu.cn, chdeng.xd@gmail.com, zhengf@sustc.edu.cn

Abstract

Learning effective distance metric between data has gained increasing popularity, for its promising performance on various tasks, such as face verification, zero-shot learning, and image retrieval. A major line of researches employs hard data mining, which makes efforts on searching a subset of significant data. However, hard data mining based approaches only rely on a small percentage of data, which is apt to overfitting. This motivates us to propose a novel framework, named deep asymmetric metric learning via rich relationship mining (DAMLRRM), to mine rich relationship under satisfying sampling size. DAMLRRM constructs two asymmetric data streams that are differently structured and of unequal length. The asymmetric structure enables the two data streams to interlace each other, which allows for the informative comparison between new data pairs over iterations. To improve the generalization ability, we further relax the constraint on the intra-class relationship. Rather than greedily connecting all possible positive pairs, DAMLRRM builds a minimum-cost spanning tree within each category to ensure the formation of a connected region. As such there exists at least one direct or indirect path between arbitrary positive pairs to bridge intra-class relevance. Extensive experimental results on three benchmark datasets including CUB-200-2011, Cars196, and Stanford Online Products show that DAMLRRM effectively boosts the performance of existing deep metric learning approaches.

1. Introduction

Metric learning aims at finding appropriate similarity measurements of data, whose major thinking is to keep the distance between similar instances close and dissimilar instances far away in an embedding space. This topic is of great practical importance due to its wide applications, including face recognition [12, 52, 45], clustering [9, 44, 53], and retrieval [57, 49, 51, 50, 22, 10]. Conventional Mahalanobis metric learning approaches learn a linear transformation of the data and measure the similarity based on Euclidean distance, which fail to capture the high-order correlation[15, 42, 47]. Riding on the development of deep neural network [21, 33, 37], deep metric learning (DML) has gained a lot of attention. Guided by a metric loss, DML projects data into an embedding space with rich semantic information through convolutional neural network. It shows potential capability even in challenging tasks, such as fine-grained classification [8, 41, 55, 25], large-category classification [2, 31, 46], and zero-shot learning [28, 56, 6, 26].

According to the types of loss, DML can be roughly divided into contrastive and triplet approaches. However, enumerating all possible pairs or triplets will arise nearly exponential sampling size, which is impractical even for a moderate number of instances. One common solution is to sample a subset of instances as a training pool. The fact is that, when the sampled training pool merely covers easy instances that contribute little to the optimization, only a weak embedding model can be obtained. Therefore, hard data mining aiming to find out confusing instances becomes an important topic, and a large number of methods are proposed [35, 34, 16, 11, 43, 54, 14]. Those methods tackle this topic to a certain extent yet are still deficient in the following three aspects. First, a complicated data preprocessing is involved to select hard data, whereas the hard level is changing with the evolution of the model [34]. Second, only a small subset of relationship is exploited. Third, the hard level is difficult to control. When the selected instances are not hard enough, the learned model is not discriminative. Conversely, when the instances are selected too hard, the overfitting problem often occurs [43].

In this work, we propose a novel framework, named deep asymmetric metric learning via rich relationship mining (DAMLRRM). DAMLRRM firstly builds two asymmetric data streams, which interlace to each other so that continues new pairs are compared during iterations. Compared with conventional one stream metric learning approaches, DAMLRRM can mine considerably richer relationship un-
under lower sampling size. Furthermore, DAMLRRM relaxes the constraint on positive pairs to extend the generalization capability. Specifically, we build positive pairs training pool by constructing a minimum connected tree for each category instead of considering all positive pairs within a mini-batch. As a result, there will exist a direct or indirect path between any positive pair, which ensures the relevance being bridged to each other. The inspiration comes from ranking on manifold [58] that spreads the relevance to their nearby neighbors one by one. The connected graph loss can help maintain the inherent distribution of the data and achieve a good generalization ability. In experiments, we empirically show the state-of-the-art results on CUB200-2011 [39], Cars196 [1], and Stanford online products [35] datasets for clustering and retrieval tasks. In a nutshell, this paper makes the following contributions:

i) We departure from the traditional hard data mining based technique and propose a novel asymmetric two streams based deep learning framework for metric learning, which also differs from conventional methods only involving one stream.

ii) We devise a relaxation technique for positive pairs constraint to improve the model generalization ability, which is verified in our empirical study.

iii) Our proposed model achieves better accuracy when using fewer than ten percents of sampling size compared with the peer methods including the lifted method [35] and N-pair [34].

2. Related Work

Siamese network [5] is the seminal work of the contrastive DML. It firstly employs twins networks to nonlinearly map two signature instances into feature space. And subsequently, a contrastive loss is employed to optimize the mapping procedure. The contrastive loss minimizes the distance between positive pairs and enlarges the distance between negative pairs if they are closer than a predetermined margin. Based on the siamese network, a collection of approaches are proposed to settle dimensionality reduction and face verification tasks [13, 7, 36, 38].

Although making great progress, contrastive metric learning approaches suffer from one drawback, that focus on absolute distance whereas relative distance matters more for most tasks [30, 31, 35]. Triplet loss, an evolution formulation of contrastive loss, has been proposed to tackle this issue. It trains a model on a triplets training pool, where each triplet consists of an anchor, a positive and a negative instance. The anchor and the positive instances share the same label, while the anchor and the negative instances have different labels. The training process encourages the network to find an embedding where the distance between positive pairs is smaller than the distance between negative pairs with some margins.

Nevertheless, contrastive and triplet losses tend to be difficult to optimize in practice, mainly influenced by the way of selecting the training pool. Confusing instances, doing a crucial contribution to optimization, should be paid huge attention to. FaceNet [31] targets on the online hard data generation, which uses large mini-batches in the order of a few thousand instances and only computes the argmin and argmax within a mini-batch. However, the batch size is 1800, which is a big memory obstacle when implementation. To take full advantage of relative relationship, Song et al. [35] allow mining the negatives from both the left and right data pair instead of negative being defined only according to anchor points. Chen et al. [16] introduce a position-dependent deep metric unit, which can be used to select hard instances to guide the deep embedding learning in an online and robust manner. Sohn et al. [34] indicate that a minority of negative instance based loss function suffers from poor local optima. As a result, they propose an $(N+1)$-tuple loss that optimizes to identify a positive instance from $N-1$ negative instances, which gains some performance improvement. More recently, Duan et al. [11] propose a deep adversarial metric learning framework to generate synthetic hard negatives from the observed negative instances.

The fundamental philosophy behind hard data mining is that for a pair of positive instances, select a significant negative sample through offline or online and penalize on the relative distance if they violate the constraint. However, both offline-based and online-based hard data mining strategies exist defects. Offline-based methods select the hard instances before training which will not be updated with the updating models. It is unreasonable as a hard relationship is dynamically decided by different models. Online-based methods decide the hard negative sample within a mini-batch along with training, which makes the comparison within a very small subset of instances. The hard quality is not guaranteed. One common drawback of these two forms is that the learned metric is insufficient because of the low utilization of pairs or triplets. Therefore, we make efforts on exploiting more pairs while controlling the sampling size in this paper.

Graph is a mathematical structure used to model pairwise relations between objects [4, 3]. A graph in the context is made up of vertices and points which are connected by edges. Graph knowledge is used to express the correlation network in many applications, such as image retrieval [32], Linguistics processing[17] and saliency detection [48]. More recently, Iscen et al. [18] utilize an undirected graph to mine an efficient training pool without label, which verifies the priority of graph in building correlation. In this paper, we take advantage of the graph to relax the constraint between positive pairs, which is quite helpful to boost the generalization ability.
3. Proposed Approach

Figure 1 illustrates the framework of our proposed method DAMLRRM. Two weight-shared networks are employed to map two asymmetric data batches, where the upper stream accepts neatly arranged data (neat stream) and the lower stream takes shuffled data as input (shuffled stream). Our model builds a minimum-cost spanning tree for each class in the neat stream which establishes a stable intra-class manifold. Furthermore, the strong discrimination capability is achieved by adopting a shuffled stream to provide various negative instances for the neat stream. We detail our proposed model in the following subsections.

3.1. Preliminaries

Let $X = \{x_i|i = 1, 2, \cdots, N^x\}$ and $S = \{s_i|i = 1, 2, \cdots, N^s\}$ be the training pools of two streams, where $N^x$ and $N^s$ are the numbers of instances in $X$ and $S$ respectively. The target of DML is to learn a nonlinear transformation to semantic embedding space $f: R^d \rightarrow R^d$, which is a differentiable deep network with parameter $\theta$. We measure the similarity of $(x_i, x_j)$ in term of the Euclidean distance in the embedding space, which is computed as $D_{ij} = \|f(x_i) - f(x_j)\|^2$. Furthermore, we construct each category as an undirected weighted subgraph $G = (V, E, D)$, where each node in $V$ corresponds to a sample, the edges in $E$ connect positive pairs, and $D$ stores the edge weights.

3.2. Rich Relationship Mining with Asymmetric Structure

To obtain rich relationship, we propose an asymmetric framework for metric learning. Asymmetry is reflected in structure and quantity, respectively. In structure, two total different structured data batches are built for two streams respectively. The data batch of the upper stream is neat while the other one is randomly shuffled. It can be clearly shown on the left side of Figure 1 and the formulations are

$$\begin{align*}
B^x &= \{x_1, x_2, \cdots, x_{s_k}, x^1, x^2, \cdots, x^k\} \\
B^s &= \{s_1, s_2, \cdots, s_{b/2}\} \\
B &= \{B^x, B^s\}
\end{align*}$$

The neat data batch $B^x$ is composed of $m$ categories where there are $k$ instances for each category. And the shuffled data batch $B^s$ contains $b/2$ randomly instances, where $b/2 = m \times k$. Hence for each iteration, the training batch $B$ consists of two parts: one neat data batch $B^x$ and one shuffled data batch $B^s$.

In quantity, the training pools’ sizes of the two streams are unequal, namely $N^x \neq N^s$. Quantity asymmetric makes it possible that the same instances in one data stream compare with different instances in another data stream at different iteration times. For example, $B^{t_1}_1$, $B^{t_2}_i$ and $B^{t_3}_n$ in Figure 2 include the same instances at different iteration times, while they compares with different instances in stream 1. Specifically, $B^{t_1}_1$ and $B^{t_2}_i$ are composed by

$$\begin{align*}
B^{t_1}_1 &= \{B^{t_1}_1, B^{t_1}_2\}, \quad B^{t_2}_i = \{B^{t_2}_1, B^{t_2}_2\} \\
B^{t_1}_1 &\neq B^{t_2}_1, \quad B^{t_1}_1 = B^{t_2}_i
\end{align*}$$

By doing so, our model can exploit abundant relationship while not increase the sampling size.

The intuitive motivations behind the asymmetric metric learning come from two aspects: 1) The neat stream mainly focus on establishing the consistent intra-class relationship by a minimum-cost spanning tree, which constrains positive
pairs to form a unified manifold. However, such an intra-
class relationship is not stable enough because it observes
limited negative instances; 2) the shuffled stream gener-
gates diverse and numerous negative instances for the neat
stream, which aims to establish a discriminative inter-class
relationship. It is worth note that, the two batches do not
cause extra memory or computation cost, because we just
split half of the batch size which previous approaches em-
ploy and the two networks share all weights.

3.3. Connected Graph based Loss Functions

Previous approaches constrain on all possible positive
pairs in a mini-batch, which is too strict and causes an
overfitting problem. Inspired by the method of ranking on
data manifold [58], that the global consistency is obtained
by spreading the relevance of source point to its nearest
neighbors one by one, we relax the constraint of positive
pairs. Rather than connecting all positive pairs, we build
a minimum-cost spanning tree for each category. By do-
ing so, a connected field within one class is obtained, which
ensure a direct or indirect path exists between arbitrary pos-
tive pairs and not too much pressure is employed on the
pairs which are not visually similar. In other words, the in-
stances that far distributed in the original visual space are
allowed indirectly associated and their distance being larger
than the threshold. The central idea is to retain the intrin-
sic distribution of data to the utmost extent while ensuring
semantic consistency.

We employ a simple minimum-spanning tree algorithm
named prim [27] to build the connected graph. Prim algo-
rithm is a greedy algorithm which finds a minimum span-
ting tree for a weighted undirected graph. It finds a sub-
set of the edges to form a tree which includes every vertex,
where the total weight of all edges in the tree is minimized.
The procedure of prim algorithm is summarized as follows:

1. Build a weighted graph $G = (V, E, D)$, where $D$ is
measured by Euclidean distance. Set $V_{visted} = \{\emptyset\}$
and $V_{unvisted} = V$. Initialize a tree with a single arbi-
trary vertex $V_{start} \in V$. Add $V_{start}$ into $V_{visted}$ and
remove it from $V_{unvisted}$.

2. Grow the tree by one edge: choosing a minimum-
weight edge $E_{minimum} \in E$ which connect $V_{visted}$
and $V_{unvisted}$, then attach it to the tree. Add the
minimum-weight-connected vertex into $V_{visted}$ and re-
move it from $V_{unvisted}$.

3. Repeat step 2 until all vertices are covered in the tree
($V_{visted} = V$).

Figure 3 gives a concrete example. Suppose the starting
vertex being the point 1 (Figure 3(a)), then the next vertex
will reach point 2 (Figure 3(b)) by choosing the minimum
weight connected to point 1. Then find out the minimum
weight of all edges connected to both point 1 and 2 and
hence reached point 4. Repeat this progress until all vertices
are included in the tree like Figure 3(c). For the situation in
Figure 3,

$$PP = \{(x_1, x_2); (x_2, x_3); (x_2, x_4); (x_4, x_5); (x_4, x_6)\},$$

(3)

where $PP$ is the connected positive pairs pool. Notably,
this minimum-cost spanning tree is quite different from
simply choosing the nearest positive pairs which does not
ensure a connected field within a category.

The objective function is defined based on the built pos-
tive pairs pool. Predefine a boundary $\alpha$ and a margin $\beta$,
the optimize goal is limiting the distance of positive pairs
smaller than $\alpha - \beta$. For the negative instances, we hope
they will not break into the tree, so the distance is forced to
be bigger than $\alpha + \beta$. The graph loss function is defined as

$$\mathcal{L}^g = \frac{1}{TP_g} \left( \sum_{i,j \in PP} |D_{i,j} - \alpha + \beta| + \sum_{i,j \in NP} [-D_{i,j} + \alpha + \beta] \right),$$

(4)
where \( P_g \) is the number of pairs that violate the constraint and \( NP \) is the negative pairs pool.

For the instances in the shuffled stream, the positive instances are expected to join into the \( prim \) tree, we force it to connect to the nearest point in the tree. And the negative instances are constrained to be far away. Hence the shuffled loss \( \mathcal{L}^s \) is defined as

\[
\mathcal{L}^s = \frac{1}{P_s} \left( \sum_{i,j \in NN} [d_{i,j} - \alpha + \beta]^2 + \sum_{i,j \in NP} [-d_{i,j} + \alpha + \beta]^2 \right),
\]

where \( P_s \) is the number of violated pairs and \( NN \) is the top 1 nearest positive pairs pool between two streams.

Combining the two loss functions, the final objective function can be formulated as:

\[
\mathcal{L} = \mathcal{L}^g + \mathcal{L}^s,
\]

where we do not employ any balance parameter when combining the two loss functions, as the motivation behind them are the same. The principle that we obey to design the objective function is respecting the data distribution most as long as the semantic consistency satisfied. The goal, accomplished by connecting connect the nearest positive pairs of the two stream and build \( prim \) tree within a neat stream, is to relax the constraint and achieve generalization ability.

4. Experiments

In this section, we evaluate the effectiveness of our proposed DAMLRRM on three public benchmark datasets for both image retrieval and clustering tasks. The Caffe package [20] is used through the experiments. All images are resized to 256-by-256 at first. For data augmentation, the training instances are performed standard random crop and horizontal mirroring, while a single center crop for testing. The embedding size is set to \( d = 512 \) for all embedding vectors [40, 11]. GoogLeNet [37] pretrained on ImageNet ILSVRC dataset [29] is used for initialization and a randomly initialized fully connected layer is added. The base learning rate is set to \( 10e^{-4} \) and 10 times faster for the newly added fully connected layer. We use SGD with 40k training iterations and 60 mini-batch size for each stream.

4.1. Benchmark Datasets

We conduct our experiments on CUB-200-2011 [39], Cars196 [1] and Stanford Online Products [35]. For all datasets, we follow the conventional protocol of splitting training and testing [35]:

**CUB-200-2011** [39] dataset covers 200 species of birds with 11,788 instances, where the first 100 species (5,864 images) are used for training and the rest of 100 species (5,924 images) are used for testing.

**Cars196** [1] 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.

**Stanford Online Products** [35] 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 remaining 11,316 classes (60,502 images) are used for testing.

When building the tree in the neat stream, we set \( k = 5 \) for CUB-200-2011 and Cars196, and \( k = 3 \) for Stanford Online Products because each product has only about 5.3 images.

4.2. Baselines

To verify the superiority of our proposed method, we compare with eight baseline deep metric learning algorithms, which are 1) DDML [23]; 2) contrastive embedding loss (Contrastive) [13]; 3) Triplet embedding loss (Triplet) [42]; 4) triplet loss with N-pair sampling, (Triplet+N-pair); 5) Lifted [35]; 6) N-pair loss (N-pair) [34]; 7) Angular loss (Angular) [40]; and 8) adversarial metric loss (AML) [11].
### 4.4. Result Analysis

#### Retrieval and clustering

Table 1, 2 and 3 report the clustering and retrieval results for CUB-200-2011, Cars196 and Stanford Online Products separately. We color the best results with red and the second best with blue. The comparison between traditional contrastive or triplet and Lifted or N-pair shows that hard data mining indeed help to boost the performance. N-pair can be cooperated with many metric learning approaches and achieve improvement mainly because of the advance in its batch construction. Among all baselines, our proposed method DAMLRRM achieves state-of-the-art performance in most cases. It worth mentioning that, DAMLRRM does not need complicated offline data preprocessing and release from hard data mining.

Figure 4 and 5 show the visualization results of CUB-200-2011 and Cars196, which implemented by dimensionality reduction algorithm t-SNE [24]. We zoom in four regions to highlight several representative classes and the various colors of the bounding box are corresponding to different categories. Two of the zoom-in regions are used for demonstrating the compact feature embedding of intra-class and the rest two for illustrating the discrimination between different classes. Despite the large pose and appearance variation, our method effectively generates a significant feature mapping that preserves semantic similarity. Figure 6 gives some instances of query and top-5 ranking images for Stanford Online Products. Despite the huge changes in the viewpoint, configuration, and illumination, our method can successfully retrieve instances from the same class.

#### Ablation study: effect of boundary $\alpha$ and margin $\beta$

There are two hyperparameters involved in our method, which are boundary $\alpha$ and margin $\beta$ respectively. Table 4 and 5 study the impact of various parameters for the retrieval task on CUB-200-2011 dataset. We set $\beta = 0.5$.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clustering(%)</th>
<th>Recall@at(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI</td>
<td>F1</td>
<td>R@1</td>
</tr>
<tr>
<td>DDML[23]</td>
<td>47.3</td>
<td>13.1</td>
</tr>
<tr>
<td>Contrastive[13]</td>
<td>47.2</td>
<td>12.5</td>
</tr>
<tr>
<td>Triplet[42]</td>
<td>49.8</td>
<td>15.0</td>
</tr>
<tr>
<td>Triplet+N-pair</td>
<td>54.1</td>
<td>20.0</td>
</tr>
<tr>
<td>Lifted[35]</td>
<td>56.4</td>
<td>22.6</td>
</tr>
<tr>
<td>N-pair[34]</td>
<td>60.2</td>
<td>28.2</td>
</tr>
<tr>
<td>Angular[40]</td>
<td>61.0</td>
<td>30.2</td>
</tr>
<tr>
<td>AML[11]</td>
<td><strong>61.3</strong></td>
<td>29.5</td>
</tr>
<tr>
<td>OURS</td>
<td><strong>61.7</strong></td>
<td><strong>31.2</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Clustering(%)</th>
<th>Recall@at(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI</td>
<td>F1</td>
<td>R@1</td>
</tr>
<tr>
<td>DDML[23]</td>
<td>41.7</td>
<td>10.9</td>
</tr>
<tr>
<td>contrastive[13]</td>
<td>42.3</td>
<td>10.5</td>
</tr>
<tr>
<td>Triplet[42]</td>
<td>52.9</td>
<td>17.9</td>
</tr>
<tr>
<td>Triplet+N-pair</td>
<td>54.3</td>
<td>19.6</td>
</tr>
<tr>
<td>Lifted[35]</td>
<td>55.1</td>
<td>25.1</td>
</tr>
<tr>
<td>N-pair[34]</td>
<td>62.7</td>
<td>31.8</td>
</tr>
<tr>
<td>Angular[40]</td>
<td>62.4</td>
<td>31.8</td>
</tr>
<tr>
<td>AML[11]</td>
<td><strong>63.1</strong></td>
<td><strong>31.9</strong></td>
</tr>
<tr>
<td>OURS</td>
<td><strong>64.2</strong></td>
<td><strong>33.5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Clustering(%)</th>
<th>Recall@at(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI</td>
<td>F1</td>
<td>R@1</td>
</tr>
<tr>
<td>DDML[23]</td>
<td>83.4</td>
<td>10.7</td>
</tr>
<tr>
<td>contrastive[13]</td>
<td>82.4</td>
<td>10.1</td>
</tr>
<tr>
<td>Triplet[42]</td>
<td>86.3</td>
<td>20.2</td>
</tr>
<tr>
<td>Triplet+N-pair</td>
<td>86.4</td>
<td>21.0</td>
</tr>
<tr>
<td>Lifted[35]</td>
<td>87.2</td>
<td>25.3</td>
</tr>
<tr>
<td>N-pair[34]</td>
<td>87.9</td>
<td>27.1</td>
</tr>
<tr>
<td>Angular[40]</td>
<td>87.8</td>
<td>26.5</td>
</tr>
<tr>
<td>AML[11]</td>
<td><strong>89.1</strong></td>
<td><strong>31.7</strong></td>
</tr>
<tr>
<td>OURS</td>
<td><strong>88.2</strong></td>
<td><strong>30.5</strong></td>
</tr>
</tbody>
</table>

**Table 4. Comparison of different boundary $\alpha$ on CUB-200-2011 [39] dataset**

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>R@1</th>
<th>R@2</th>
<th>R@4</th>
<th>R@8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 26$</td>
<td>52.9</td>
<td>65.4</td>
<td>76.1</td>
<td>85.1</td>
</tr>
<tr>
<td>$\alpha = 28$</td>
<td>53.1</td>
<td>65.3</td>
<td>76.1</td>
<td>84.7</td>
</tr>
<tr>
<td>$\alpha = 30$</td>
<td>55.1</td>
<td>66.5</td>
<td>76.8</td>
<td>85.3</td>
</tr>
<tr>
<td>$\alpha = 32$</td>
<td>54.5</td>
<td>66.0</td>
<td>76.4</td>
<td>85.3</td>
</tr>
</tbody>
</table>

**Table 5. Comparison of different margin $\beta$ on CUB-200-2011 [39] dataset**

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>R@1</th>
<th>R@2</th>
<th>R@4</th>
<th>R@8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta = 0.1$</td>
<td>51.9</td>
<td>64.5</td>
<td>75.8</td>
<td>84.9</td>
</tr>
<tr>
<td>$\beta = 0.3$</td>
<td>52.7</td>
<td>64.9</td>
<td>75.6</td>
<td>84.4</td>
</tr>
<tr>
<td>$\beta = 0.5$</td>
<td>55.1</td>
<td>66.5</td>
<td>76.8</td>
<td>85.3</td>
</tr>
<tr>
<td>$\beta = 0.7$</td>
<td>53.9</td>
<td>65.7</td>
<td>76.2</td>
<td>85.3</td>
</tr>
<tr>
<td>$\beta = 1.0$</td>
<td>53.2</td>
<td>66.3</td>
<td>76.8</td>
<td>85.4</td>
</tr>
</tbody>
</table>

### 4.3. Evaluation Metrics

Following the standard protocol used in [35, 34], we calculate the Recall@a metric [19] for retrieval task. Specifically, for each query image, top a nearest images will be returned based on Euclidean distance, then the recall score will be 1 if at least one positive image appears in the returned a images and 0 otherwise. For clustering evaluation, we adopt the k-means algorithm to cluster testing instances and the quality is reported in terms of the standard F1 and NMI metrics. Refer to [35] for detailed formulation.
Figure 4. Visualization of feature embedding computed by our method using t-SNE on CUB-200-2011 dataset.

Figure 5. Visualization of feature embedding computed by our method using t-SNE on Cars196 dataset.


<table>
<thead>
<tr>
<th>Method</th>
<th># Sampling Size</th>
<th>Recall@a(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@1</td>
</tr>
<tr>
<td>Lifted[35]</td>
<td>700K</td>
<td>46.9</td>
</tr>
<tr>
<td>N-pair[34]</td>
<td>500K</td>
<td>51.0</td>
</tr>
<tr>
<td>OURS¹</td>
<td>36K</td>
<td>52.3</td>
</tr>
</tbody>
</table>

when varying the value of $\alpha$, and set $\alpha = 0.5$ when discussing $\beta$. It can be seen that the best performance is obtained when $\alpha = 30, \beta = 0.5$. Furthermore, DAMLRRM is not sensitive to the two parameters and we set boundary to 30 and margin to 0.5 throughout our experiments.

Ablation study: effect of asymmetric batches. In order to verify the effectiveness of asymmetric structure, we

Table 7. Comparison of full combined positive pairs and prim tree connected positive pairs on CUB-200-2011 [39] dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall@a(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1</td>
</tr>
<tr>
<td>Full PP</td>
<td>48.2</td>
</tr>
<tr>
<td>OURS²</td>
<td>51.2</td>
</tr>
</tbody>
</table>
remove the graph loss and keep the shuffled loss only, which is denoted as OURS\textsuperscript{1}. We compare it with conventional one stream data batch construction methods: Lifted and N-pair algorithms. Table 6 reports the retrieval metrics of CUB-200-2011 and demonstrates the priority of two asymmetric stream batches construction. Notably, our method only samples about 36K images which are about ten percentage of Lifted and N-pair.

Ablation study: effect of graph pairs construction. To illustrate the difference between two positive training pools established by minimum-cost spanning tree and fully combination, we remove the shuffled loss from DAMLRRM and keep graph loss. We denote them as OURS\textsuperscript{2} and Full PPs respectively. Table 7 reports the retrieval result of CUB-200-2011. We can observe that minimum-cost tree based positive pairs training pool is significant for improving the performance, which is mainly because relaxing the constraint employed on positive pairs and the generalization ability is enhanced.

Algorithmic complexity analysis. Compared with Lifted[35] and N-pair[34], our proposed method builds a prim tree within each category additionally. The computational complexity of prim tree is: $O_p = \sum_{i=1}^{k-1} i \cdot (k - i)$, where $k$ is the number of instances in a tree. The comparison of training time cost is shown in Table 8, we believe that the additional offline training time is worthy given the significantly improved accuracy. For testing, all instances are mapped by one stream model, and the time cost is the same.

Table 8. Comparison of training time on CUB\textsubscript{200,2011}[39] dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Lifted[35]</th>
<th>N-pair[34]</th>
<th>OURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations/Sec</td>
<td>2.2</td>
<td>2.2</td>
<td>0.84</td>
</tr>
<tr>
<td>Training Time</td>
<td>5.1 h</td>
<td>5.1 h</td>
<td>13.2 h</td>
</tr>
</tbody>
</table>

To min the rich relationship, we construct two structured and quantified asymmetric data streams, which interlace to each other during iterations. Such an asymmetric structure enables continuous newly combined pairs to be compared when optimizing the model, and hence a rich relationship is mined under a small amount of sampling size. To enhance its generalization ability, we relax the constraint on positive pairs. Instead of connecting all possible positive pairs, we build a minimum-cost spanning tree within one category to ensure the form of connected field. Minimum-cost spanning tree based sampling algorithm obeys the inherent distribution of data, where not all positive instances are associated directly. Our proposed model releases from hard data mining and achieves higher accuracy while even at the cost of fewer than ten percents sampling images compared with the peer methods including the lifted method [35] and N-pair [34].

6. Acknowledgment

Our work was also supported by the National Natural Science Foundation of China under Grant 61572388 and 61703327, Key R&D Program-The Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-05-04-02, 2017ZDCXL-GY-05-02, and 2018ZDXM-GY-176, and the National Key R&D Program of China under Grant 2017YFE0104100.
References


