Large-scale Distance Metric Learning with Uncertainty

Qi Qian Jiasheng Tang Hao Li Shenghuo Zhu Rong Jin
Alibaba Group, Bellevue, WA, 98004, USA
{qi.qian, jiasheng.tjs, lihao.lh, shenghuo.zhu, jinrong.jr}@alibaba-inc.com

Abstract

Distance metric learning (DML) has been studied extensively in the past decades for its superior performance with distance-based algorithms. Most of the existing methods propose to learn a distance metric with pairwise or triplet constraints. However, the number of constraints is quadratic or even cubic in the number of the original examples, which makes it challenging for DML to handle the large-scale data set. Besides, the real-world data may contain various uncertainty, especially for the image data. The uncertainty can mislead the learning procedure and cause the performance degradation. By investigating the image data, we find that the original data can be observed from a small set of clean latent examples with different distortions. In this work, we propose the margin preserving metric learning framework to learn the distance metric and latent examples simultaneously. By leveraging the ideal properties of latent examples, the training efficiency can be improved significantly while the learned metric also becomes robust to the uncertainty in the original data. Furthermore, we can show that the metric is learned from latent examples only, but it can preserve the large margin property even for the original data. The empirical study on the benchmark image data sets demonstrates the efficacy and efficiency of the proposed method.

1. Introduction

Distance metric learning (DML) aims to learn a distance metric where examples from the same class are well separated from examples of different classes. It is an essential task for distance-based algorithms, such as k-means clustering [18], k-nearest neighbor classification [17] and information retrieval [2]. Given a distance metric M, the squared Mahalanobis distance between examples $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed as

$$\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)$$

Most of existing DML methods propose to learn the metric by minimizing the number of violations in the set of pairwise or triplet constraints. Given a set of pairwise constraints, DML tries to learn a metric such that the distances between examples from the same class are sufficiently small (e.g., smaller than a predefined threshold) while those between different ones are large enough [3, 18]. Different from pairwise constraints, each triplet constraint consists of three examples $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label and $\mathbf{x}_k$ is from a different class. An ideal metric can push away $\mathbf{x}_k$ from $\mathbf{x}_i$ and $\mathbf{x}_j$ by a large margin [17]. Learning with triplet constraints optimizes the local positions of examples and is more flexible for real-world applications, where defining the appropriate thresholds is hard for pairwise constraints. In this work, we will focus on DML with triplet constraints.

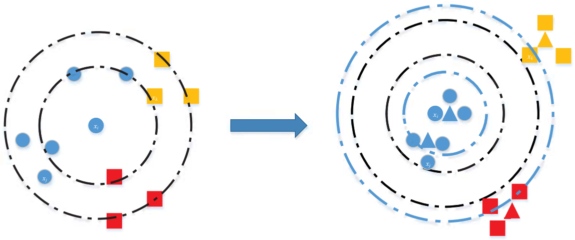


Figure 1. Illustration of the proposed method. Let round and square points denote the target data and impostors, respectively. Let triangle points denote the corresponding latent examples. Data points with the same color are from the same class. It demonstrates that the metric learned with latent examples not only separates the dissimilar latent data with a large margin but also preserves the large margin for the original data.

Optimizing the metric with a set of triplet constraints is challenging since the number of triplet constraints can be up to $\mathcal{O}(n^3)$, where n is the number of the original training examples. It makes DML computationally intractable for the large-scale problems. Many strategies have been developed to deal with this challenge and most of them fall into two categories, learning by stochastic gradient descent (SGD) and learning with the active set. With the strategy of SGD, DML methods can sample just one constraint or a mini-batch of constraints at each iteration to observe an unbiased estimation of the full gradient and avoid computing the gradient from the whole set [2, 10]. Other methods learn the metric with a set of active constraints (i.e., violated by the current metric), where the size can be significantly smaller than the original set [17]. It is a conventional strategy applied by cutting plane methods [1]. Both of these strategies can alleviate the large-scale challenge but have inherent drawbacks. Approaches based on SGD have to search through the whole set of triplet constraints, which results in the slow convergence, especially when the number of active constraints is small. On the other hand, the methods relying on the active set have to identify the set at each iteration. Unfortunately, this operation requires computing pairwise distances with the current metric, where the cost is $\mathcal{O}(n^2)$ and is too expensive for large-scale problems.

Besides the challenge from the size of data set, the uncertainty in the data is also an issue, especially for the image data, where the uncertainty can come from the differences between individual examples and distortions, e.g., pose, illumination and noise. Directly learning with the original data will lead to a poor generalization performance since the metric tends to overfit the uncertainty in the data. By further investigating the image data, we find that most of original images can be observed from a much smaller set of clean latent examples with different distortions. The phenomenon is illustrated in Fig. 5. This observation inspires us to learn the metric with latent examples in lieu of the original data. The challenge is that latent examples are unknown and only images with uncertainties are available.

In this work, we propose a framework to learn the distance metric and latent examples simultaneously. It sufficiently explores the properties of latent examples to address the mentioned challenges. First, due to the small size of latent examples, the strategy of identifying the active set becomes affordable when learning the metric. We adopt it to accelerate the learning procedure via avoiding the attempts on inactive constraints. Additionally, compared with the original data, the uncertainty in latent examples decreases significantly. Consequently, the metric directly learned from latent examples can focus on the nature of the data rather than the uncertainty in the data. To further improve the robustness, we adopt the large margin property that latent examples from different classes should be pushed away with a data dependent margin. Fig. 1 illustrates that an appropriate margin for latent examples can also preserve the large margin for the original data. We conduct the empirical study on benchmark image data sets, including the challenging ImageNet data set, to demonstrate the efficacy and efficiency of the proposed method.

The rest of the paper is organized as follows: Section 2 summarizes the related work of DML. Section 3 describes the details of the proposed method and Section 4 summarizes the theoretical analysis. Section 5 compares the proposed method to the conventional DML methods on the benchmark image data sets. Finally, Section 6 concludes this work with future directions.

2. Related Work

Many DML methods have been proposed in the past decades [3, 17, 18] and comprehensive surveys can be found in [7, 19]. The representative methods include Xing's method [18], ITML [3] and LMNN [17]. ITML learns a metric according to pairwise constraints, where the distances between pairs from the same class should be smaller than a predefined threshold and the distances between pairs from different classes should be larger than another predefined threshold. LMNN is developed with triplet constraints and a metric is learned to make sure that pairs from the same class are separated from the examples of different classes with a large margin. Compared with pairwise constraints, triplet constraints are more flexible to depict the local geometry.

To handle the large number of constraints, some methods adopt SGD or online learning to sample one constraint or a mini-batch of constraints at each iteration [2, 10]. OASIS [2] randomly samples one triplet constraint at each iteration and computes the unbiased gradient accordingly. When the size of the active set is small, these methods require extremely large number of iterations to improve the model. Other methods try to explore the concept of the active set. LMNN [17] proposes to learn the metric effectively at each iteration by collecting an active set that consists of constraints violated by the current metric within the k-nearest neighbors for each example. However, it requires $\mathcal{O}(n^2)$ to obtain the appropriate active set.

Besides the research about conventional DML, deep metric learning has attracted much attention recently [9, 13, 15, 16]. These studies also indicate that sampling active triplets is essential for accelerating the convergence. FaceNet [15] keeps a large size of mini-batch and searches hard constraints within a mini-batch. LeftedStruct [16] generates the mini-batch with the randomly selected positive examples and the corresponding hard negative examples. Proxy-NCA [9] adopts proxy examples to reduce the size of triplet constraints. Once an anchor example is given, the similar and dissimilar examples will be searched within the set of proxies. In this work we propose to learn the metric only with latent examples which can dramatically reduce the computational cost of obtaining the active set. Besides, the triangle inequality dose not hold for the squared distance, which makes our analysis significantly different from the existing work.

3. Margin Preserving Metric Learning

Given a training set $\{(\mathbf{x}_i, y_i)|i = 1, \cdots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is an example and y_i is the corresponding label, DML aims to learn a good distance metric such that

$$\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \quad \mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_k) - \mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_j) \geq 1$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same class and $\mathbf{x}_k$ is different. Given the distance metric $M \in \mathcal{S}_+^{d\times d}$, the squared distance is defined as

$$\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)$$

where $\mathcal{S}_+^{d\times d}$ denotes the set of $d \times d$ positive semi-definite (PSD) matrices.

For the large-scale image data set, we assume that each observed example is from a latent example with certain zero mean distortions, i.e.,

$$\forall i, \quad E[\mathbf{x}_i] = \mathbf{z}_{o:f(i)=o}$$

where $f(\cdot)$ projects the original data to its corresponding latent example.

Then, we consider the expected distance [20] between observed data and the objective is to learn a metric such that

$$\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \quad E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_k)] - E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_j)] \geq 1 \quad (1)$$

Let $\mathbf{z}_o$, $\mathbf{z}_p$ and $\mathbf{z}_q$ denote latent examples of $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ respectively. For the distance between examples from the same class, we have

$$\begin{aligned} &E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_j)] = E[(\mathbf{x}_i - \mathbf{z}_o + \mathbf{z}_o)^\top M(\mathbf{x}_i - \mathbf{z}_o + \mathbf{z}_o)] \\ &+ E[(\mathbf{x}_j - \mathbf{z}_p + \mathbf{z}_p)^\top M(\mathbf{x}_j - \mathbf{z}_p + \mathbf{z}_p)] - E[2\mathbf{x}_i^\top M\mathbf{x}_j] \\ &= \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_p) + E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)] + E[\mathcal{D}^2_M(\mathbf{x}_j, \mathbf{z}_p)] \\ &= \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_p) + 2E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)] \qquad (2) \end{aligned}$$

The last equation is due to the fact that $\mathbf{x}_i$ and $\mathbf{x}_j$ are i.i.d, since they are from the same class.

By applying the same analysis for the dissimilar pair, we have

$$\begin{aligned} &E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_k)] = \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_q) + E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)] \\ &+ E[\mathcal{D}^2_M(\mathbf{x}_k, \mathbf{z}_q)] \geq \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_q) + E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)] \quad (3) \end{aligned}$$

The inequality is because that M is a PSD matrix.

Combining Eqns. 2 and 3, we find that the difference between the distances in the original triplet can be lower bounded by those in the triplet consisting of latent examples

$$\begin{aligned} &E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_k)] - E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{x}_j)] \\ &\geq \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_q) - \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_p) - E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)] \end{aligned}$$

Therefore, the metric can be learned with the constraints defined on latent examples such that

$$\forall \mathbf{z}_o, \mathbf{z}_p, \mathbf{z}_q \ \ \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_q) - \mathcal{D}^2_M(\mathbf{z}_o, \mathbf{z}_p) \geq 1 + E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)]$$

Once the metric is observed, the margin for the expected distances between original data (i.e., as in Eqn. 1) is also guaranteed. Compared with the original constraints, the margin between latent examples is increased by the factor of $E[\mathcal{D}^2_M(\mathbf{x}_i, \mathbf{z}_o)]$. This term indicates the expected distance between the original data and its corresponding latent example. It means that the tighter a local cluster is, the less a margin should be increased. Furthermore, each class takes a different margin, which depends on the distribution of the original data and makes it more flexible than a global margin.

With the set of triplets $\{\mathbf{z}_o^t, \mathbf{z}_p^t, \mathbf{z}_q^t\}$, the optimization problem can be written as

$$\min_{M\in\mathcal{S}_+^{d\times d}, \|M\|_F\leq\delta, \mathbf{z}\in\mathbb{R}^{d\times m}} \mathcal{L}(M, \mathbf{z}) = \sum_t \ell(\mathbf{z}_o^t, \mathbf{z}_p^t, \mathbf{z}_q^t; M)$$

where $m \ll n$ is the number of latent examples. We add a constraint for the Frobenius norm of the learned metric to prevent it from overfitting. $\ell(\cdot)$ is the loss function and the hinge loss is applied in this work.

$$\begin{aligned} &\ell(\mathbf{z}_o^t, \mathbf{z}_p^t, \mathbf{z}_q^t; M) \\ &= [1 + E[\mathcal{D}^2_M(\mathbf{x}_i^t, \mathbf{z}_o^t)] - (\mathcal{D}^2_M(\mathbf{z}_o^t, \mathbf{z}_q^t) - \mathcal{D}^2_M(\mathbf{z}_o^t, \mathbf{z}_p^t))]_+ \end{aligned}$$

This problem is hard to solve since both the metric and latent examples are the variables to be optimized. Therefore, we propose to solve it in an alternating way and the detailed steps are demonstrated below.

3.1. Update z with Upper Bound

When fixing M_{k-1}, the subproblem at the k-th iteration becomes

$$\begin{aligned} \min_{\mathbf{z}} \mathcal{L}(M_{k-1}, \mathbf{z}) = \sum_t \Big[1 + \underbrace{E[\mathcal{D}^2_{M_{k-1}}(\mathbf{x}_i^t, \mathbf{z}_o^t)]}_{a} \\ - \underbrace{(\mathcal{D}^2_{M_{k-1}}(\mathbf{z}_o^t, \mathbf{z}_q^t) - \mathcal{D}^2_{M_{k-1}}(\mathbf{z}_o^t, \mathbf{z}_p^t))}_{b}\Big]_+ \qquad (4) \end{aligned}$$

The variable $\mathbf{z}$ appears in both the term of margin a and the term of the triplet difference b, which makes it hard to optimize directly. Our strategy is to find an appropriate upper bound for the original problem and solve the simple problem instead.

Theorem 1. *The function $\mathcal{L}(M_{k-1}, \mathbf{z})$ can be upper bounded by the series of functions $\sum_r \mathcal{F}_r(\mathbf{z})$. For the r-th class, we have*

$$\mathcal{F}_r(\mathbf{z}) = c_1 E[\mathcal{D}^2_{M_{k-1}}(\mathbf{x}_i, \mathbf{z}_o)] + c_2 + c_3 \sum_o \mathcal{D}^2_{M_{k-1}}(\mathbf{z}_o, \mathbf{z}_o^{k-1})$$

where c_1, c_2 and c_3 are constants and $\sum_r \mathcal{F}_r(\mathbf{z}^{k-1}) = \mathcal{L}(M_{k-1}, \mathbf{z}^{k-1})$.

The detailed proof can be found in Section 4.

After removing the constant terms and rearrange the coefficients, optimizing $\mathcal{F}_r(\mathbf{z})$ is equivalent to optimizing the following problem

$$\min_{\mathbf{z}\in\mathbb{R}^{d\times m_r},\boldsymbol{\mu}:\mu_{i,o}\in\{0,1\},\sum_o \mu_{i,o}=1} \tilde{\mathcal{F}}_r(\mathbf{z}) = \sum_{i:y(i)=r}\sum_o \mu_{i,o}\mathcal{D}^2_{M_{k-1}}(\mathbf{x}_i,\mathbf{z}_o) + \gamma\sum_o \mathcal{D}^2_{M_{k-1}}(\mathbf{z}_o,\mathbf{z}_o^{k-1}) \tag{5}$$

where $\boldsymbol{\mu}$ denotes the membership that assigns a latent example for each original example.

Till now, it shows that the original objective $\mathcal{L}(M_{k-1},\mathbf{z})$ can be upper bounded by $\sum_r \mathcal{F}_r(\mathbf{z})$. Minimizing the upper bound is similar to k-means but with the distance defined on the metric M_{k-1}. So we can solve it by the standard EM algorithm.

When fixing $\boldsymbol{\mu}$, latent examples can be updated by the closed-form solution

$$\forall o,\quad \mathbf{z}_o = \frac{1}{\sum_i \mu_{i,o}+\gamma}\Big(\sum_i \mu_{i,o}\mathbf{x}_i + \gamma\mathbf{z}_o^{k-1}\Big) \tag{6}$$

When fixing $\mathbf{z}$, $\boldsymbol{\mu}$ just assigns each original example to its nearest latent example with the distance defined on the metric M_{k-1}

$$\forall i,\quad \mu_{i,o} = \begin{cases} 1 & o = \arg\min_o D^2_{M_{k-1}}(\mathbf{x}_i,\mathbf{z}_o) \\ 0 & o.w. \end{cases} \tag{7}$$

Alg. 1 summarizes the method for solving $\tilde{\mathcal{F}}_r(\mathbf{z})$.

Algorithm 1 Algorithm of Updating $\mathbf{z}$

```
Input: data set $\{X, Y\}$, $\mathbf{z}^{k-1}$, $M_{k-1}$, $\gamma$ and $S$
Initialize $\mathbf{z} = \mathbf{z}^{k-1}$
for $s = 1$ to $S$ do
    Fix $\mathbf{z}$ and obtain the assignment $\boldsymbol{\mu}$ as in Eqn. 7
    Fix $\boldsymbol{\mu}$ and update $\mathbf{z}$ as in Eqn. 6
end for
return $\mathbf{z}^k = \mathbf{z}$
```

3.2. Update M with Upper Bound

When fixing $\mathbf{z}^k$ at the k-th iteration, the subproblem becomes

$$\min_{M\in\mathcal{S}_+^{d\times d}} \mathcal{L}(M,\mathbf{z}^k) = \sum_t [1 + \underbrace{E[\mathcal{D}^2_M(\mathbf{x}_i^t,\mathbf{z}_o^t)]}_{a} - \underbrace{(\mathcal{D}^2_M(\mathbf{z}_o^t,\mathbf{z}_q^t) - \mathcal{D}^2_M(\mathbf{z}_o^t,\mathbf{z}_p^t))}_{b}]_+ \tag{8}$$

where M also appears in multiple terms. With the similar procedure, an upper bound can be found to make the optimization simpler.

Theorem 2. *The function $\mathcal{L}(M,\mathbf{z}^k)$ can be upper bounded by the function $\mathcal{H}(M)$ which is*

$$\mathcal{H}(M) = \frac{\lambda}{2}\|M - M_{k-1}\|_F^2 + \sum_t \Big[1 + E[\mathcal{D}^2_{M_{k-1}}(\mathbf{x}_i^t,\mathbf{z}_o^t)] - (\mathcal{D}^2_M(\mathbf{z}_o^t,\mathbf{z}_q^t) - \mathcal{D}^2_M(\mathbf{z}_o^t,\mathbf{z}_p^t))\Big]_+$$

where λ is a constant and $\mathcal{H}(M_{k-1}) = \mathcal{L}(M_{k-1},\mathbf{z}^k)$.

Minimizing $\mathcal{H}(M)$ is a standard DML problem. Since the number of latent examples $\mathbf{z}^k$ is small, many existing DML methods can handle the problem well. In this work we solve the problem by SGD but sample one epoch active constraints at each stage. The active constraints contain the triplets of $\mathbf{z}^k$ that incur the hinge loss with the distance defined on M_{k-1}. This strategy enjoys the efficiency of SGD and the efficacy of learning with the active set. To further improve the efficiency, one projection paradigm is adopted to avoid the expensive PSD projection which costs $\mathcal{O}(d^3)$. It performs the PSD projection once at the end of the learning algorithm and shows to be effective in many applications [2, 11]. Finally, since the problem is strongly convex, we apply the α-suffix averaging strategy, which averages the solutions over the last several iterations, to obtain the optimal convergence rate [12]. The complete approach for obtaining M_k is shown in Alg. 2.

Algorithm 2 Algorithm of Updating M

```
Input: data set $\{X, Y\}$, $\mathbf{z}^k$, $M_{k-1}$, $\delta$, $\lambda$ and $S$
Initialize $M_0 = M_{k-1}$
Sample one epoch active constraints $\mathcal{A}$ according to $\mathbf{z}^k$ and $M_{k-1}$
for $s = 1$ to $S$ do
    Randomly sample one constraint from $\mathcal{A}$
    Compute the stochastic gradient $g = \nabla\mathcal{H}(M)$
    Update the metric as $M_s' = M_{s-1} - \frac{1}{\lambda s}g$
    Check the Frobenius norm $M_s = \Pi_\delta(M_s')$
end for
Project the learned matrix onto the PSD cone
$M_k = \Pi_{PSD}(\frac{2}{S}\sum_{s=S/2+1}^{S} M_s)$
return $M_k$
```

Alg. 3 summarizes the proposed margin preserving metric learning framework. Different from the standard alternating method, we only optimize the upper bound for each subproblem. However, the method converges as shown in the following theorem.

Theorem 3. *Let $(\mathbf{z}^{k-1}, M_{k-1})$ and $(\mathbf{z}^k, M_k)$ denote the results obtained by applying the algorithm in Alg. 3 at $(k-1)$-th and k-th iterations respectively. Then, we have*

$$\mathcal{L}(\mathbf{z}^k, M_k) \le \mathcal{L}(\mathbf{z}^{k-1}, M_{k-1})$$

which means the proposed method can converge.

Algorithm 3 Margin Preserving Metric Learning (MaPML)

```
Input: data set $\{X, Y\}$, $\delta$, $m$, $\gamma$, $\lambda$ and $K$
Initialize $M_0 = I$
for $k = 1$ to $K$ do
    Fix $M_{k-1}$ and obtain latent examples $\mathbf{z}^k$ by Alg. 1
    Fix $\mathbf{z}^k$ and update the metric $M_k$ by Alg. 2
end for
return $M_K$ and $\mathbf{z}^K$
```

Computational Complexity The proposed method consists of two parts: obtaining latent examples and metric learning. For the former one, the cost is linear in the number of latent examples and original examples as $\mathcal{O}(mn)$. For the latter one, the cost of sampling an active set dominates the learning procedure. Since the number of iterations is fixed, the complexity of sampling becomes $\min\{\mathcal{O}(Sm), \mathcal{O}(m^2)\}$. Therefore, the whole algorithm can be linear in the number of latent examples. Note that the efficiency can be further improved with distributed computing since many components of MaPML can be implemented in parallel. For example, when updating $\mathbf{z}$, each class is independent and all subproblems can be solved simultaneously.

4. Theoretical Analysis

4.1. Proof of Theorem 1

Proof. First, for the distance of the dissimilar pair in term b of Eqn. 4, we have

$$\begin{aligned}
&\mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_q) = \mathcal{D}_M^2(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1})\\
&+ \mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1}) + 2(\mathbf{z}_o - \mathbf{z}_o^{k-1})^\top M(\mathbf{z}_o^{k-1} - \mathbf{z}_q^{k-1})\\
&+ \mathcal{D}_M^2(\mathbf{z}_q, \mathbf{z}_q^{k-1}) - 2(\mathbf{z}_q - \mathbf{z}_q^{k-1})^\top M(\mathbf{z}_o^{k-1} - \mathbf{z}_q^{k-1})\\
&- 2(\mathbf{z}_o - \mathbf{z}_o^{k-1})^\top M(\mathbf{z}_q - \mathbf{z}_q^{k-1})\\
&\geq \mathcal{D}_M^2(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1}) - 2\mathcal{D}_M(\mathbf{z}_o, \mathbf{z}_o^{k-1})\mathcal{D}_M(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1})\\
&- 2\mathcal{D}_M(\mathbf{z}_q, \mathbf{z}_q^{k-1})\mathcal{D}_M(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1})
\end{aligned}$$

where $\mathbf{z}^{k-1}$ are latent examples from the last iteration. We let M denote M_{k-1} in this proof for simplicity. The inequality is from that M is a PSD matrix and can be decomposed as $M = LL^\top$. Then it is obtained by applying the Cauchy-Schwarz inequality. With the assumptions that $\forall o, \mathcal{D}_M(\mathbf{z}_o, \mathbf{z}_o^{k-1})$ is sufficiently large and $\mathcal{D}_M(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1})$ is bounded by a constant $\frac{c}{2}$, the inequality can be simplified as

$$\begin{aligned}
&\mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_q) \geq \qquad (9)\\
&\mathcal{D}_M^2(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1}) - c\mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1}) - c\mathcal{D}_M^2(\mathbf{z}_q, \mathbf{z}_q^{k-1})
\end{aligned}$$

The assumption is easy to verify since

$$\mathcal{D}_M(\mathbf{z}_o^{k-1}, \mathbf{z}_q^{k-1}) \leq \|\mathbf{z}_o^{k-1} - \mathbf{z}_q^{k-1}\|_2^2 \|M_{k-1}\|_2$$

Note that $\|M_{k-1}\|_2 \leq \|M_{k-1}\|_F \leq \delta$ and $\mathbf{z}$ is in the convex hull of the original data, and the constant c can be set as $c = 8\delta \max_i \|\mathbf{x}_i\|_2^2$.

With the similar procedure, we have the bound for the distance of the similar pair as

$$\begin{aligned}
&\mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_p) \leq \mathcal{D}_M^2(\mathbf{z}_o^{k-1}, \mathbf{z}_p^{k-1}) \qquad (10)\\
&+(c+2)\mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1}) + (c+2)\mathcal{D}_M^2(\mathbf{z}_p, \mathbf{z}_p^{k-1})
\end{aligned}$$

Taking Eqns. 9 and 10 back to the original function $\mathcal{L}(M_{k-1}, \mathbf{z})$ and using the property of the hinge loss, the original one can be upper bounded by

$$\begin{aligned}
\mathcal{G}(\mathbf{z}) = &\sum_t [1 + E[\mathcal{D}_M^2(\mathbf{x}_i^t, \mathbf{z}_o^t)] - (\mathcal{D}_M^2(\mathbf{z}_o^{t:k-1}, \mathbf{z}_q^{t:k-1})\\
&- \mathcal{D}_M^2(\mathbf{z}_o^{t:k-1}, \mathbf{z}_p^{t:k-1}))]_+ + c_3 \sum_o^m \mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1})
\end{aligned}$$

where $c_3 = \mathcal{O}(Tc)$ is a constant. By investigating the structure of this problem, we find that each class is independent in the optimization problem and the subproblem for the r-th class can be written as

$$\begin{aligned}
\min_{\mathbf{z} \in \mathbb{R}^{d \times m_r}} \mathcal{G}_r(\mathbf{z}) = &\sum_{t: y(\mathbf{z}_o^t) = r} [E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] + c_t]_+\\
+ c_3 &\sum_{o: y(\mathbf{z}_o) = r} \mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1})
\end{aligned}$$

where m_r is the number of latent examples for the r-th class and c_t is a constant as

$$c_t = 1 - (\mathcal{D}_M^2(\mathbf{z}_o^{t:k-1}, \mathbf{z}_q^{t:k-1}) - \mathcal{D}_M^2(\mathbf{z}_o^{t:k-1}, \mathbf{z}_p^{t:k-1}))$$

Next we try to upper bound the hinge loss in $\mathcal{G}_r(\mathbf{z})$ with a linear function in the interval of $[c_t, E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o^{k-1})] + c_t]$, where the hinge loss incurred by the optimal solution $\mathbf{z}^k$ is guaranteed to be in it.

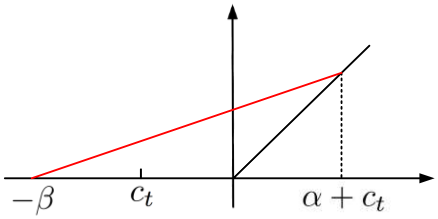


Figure 2. Illustration of bounding the hinge loss. The hinge loss between $[c_t, \alpha + c_t]$ is upper bounded by the linear function denoted by the red line.

Let $\alpha = E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o^{k-1})]$, which is the expected distance between the original data of the r-th class and the corresponding latent examples from the last iteration, and β be a constant sufficiently large as

$$\beta \geq -\min_t c_t$$

Then, for each active hinge loss (i.e., $\alpha + c_t > 0$), if

$$E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] \leq \alpha \tag{11}$$

we have

$$\begin{aligned}&[E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] + c_t]_+ \\ &\leq \frac{\alpha + c_t}{\alpha + c_t + \beta}(E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] + c_t + \beta)\end{aligned}$$

Fig. 2 illustrates the linear function that can bound the hinge loss and the proof is straightforward. We will show that the condition in Eqn. 11 can be satisfied throughout the algorithm later.

With the upper bound of the hinge loss, $\mathcal{G}_r(\mathbf{z})$ can be bounded by

$$\mathcal{F}_r(\mathbf{z}) = c_1 E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] + c_2 + c_3 \sum_o \mathcal{D}_M^2(\mathbf{z}_o, \mathbf{z}_o^{k-1})$$

where

$$c_1 = \sum_{t:y(\mathbf{z}_o^t)=r} \frac{\alpha + c_t}{\alpha_t + c_t + \beta} \mathbb{I}(\alpha + c_t)$$

and

$$c_2 = \sum_{t:y(\mathbf{z}_o^t)=r} \frac{\alpha + c_t}{\alpha_t + c_t + \beta}(c_t + \beta)\mathbb{I}(\alpha + c_t)$$

$\mathbb{I}(\cdot)$ is an indicator function as $\mathbb{I}(\nu) = \begin{cases} 1 & \nu > 0 \\ 0 & o.w. \end{cases}$

Finally, we check the condition in Eqn. 11. Let $\mathbf{z}^k$ denote latent examples obtained by optimizing $\tilde{\mathcal{F}}(\mathbf{z})$ with Alg. 1. Since we use $\mathbf{z}^{k-1}$ as the starting point to optimize $\tilde{\mathcal{F}}_r(\mathbf{z})$, it is obvious that

$$\tilde{\mathcal{F}}_r(\mathbf{z}^k) \leq \tilde{\mathcal{F}}_r(\mathbf{z}^{k-1})$$

At the same time, we have

$$\sum_o \mathcal{D}_M^2(\mathbf{z}_o^k, \mathbf{z}_o^{k-1}) \geq \sum_o \mathcal{D}_M^2(\mathbf{z}_o^{k-1}, \mathbf{z}_o^{k-1}) = 0$$

It is observed that Eqn. 11 is satisfied by combining these inequalities.

□

4.2. Proof of Theorem 2

Proof. For the term a in Eqn. 8, we have

$$\begin{aligned}&E[\mathcal{D}_M^2(\mathbf{x}_i, \mathbf{z}_o)] \\ &= E[\mathcal{D}_{M_{k-1}}^2(\mathbf{x}_i, \mathbf{z}_o) + (\mathbf{x}_i - \mathbf{z}_o)^\top (M - M_{k-1})(\mathbf{x}_i - \mathbf{z}_o)] \\ &\leq E[\mathcal{D}_{M_{k-1}}^2(\mathbf{x}_i, \mathbf{z}_o)] + \max_i \|\mathbf{x}_i - \mathbf{z}_o\|_2^2 \|M - M_{k-1}\|_F \\ &\leq E[\mathcal{D}_{M_{k-1}}^2(\mathbf{x}_i, \mathbf{z}_o)] + \tilde{c}\|M - M_{k-1}\|_F^2\end{aligned}$$

where we assume that $\|M - M_{k-1}\|_F$ is sufficiently large and $\tilde{c}$ is a constant which has $\max_i \|\mathbf{x}_i - \mathbf{z}_o\|_2^2 \leq \tilde{c}$ and can be set as $\tilde{c} = 4\max_i \|x_i\|_2^2$.

Therefore, the original function $\mathcal{L}(M, \mathbf{z}^k)$ can be upper bounded by

$$\begin{aligned}\mathcal{H}(M) = \frac{\lambda}{2}\|M - M_{k-1}\|_F^2 + \sum_t \Big[1 + E[\mathcal{D}_{M_{k-1}}^2(\mathbf{x}_i^t, \mathbf{z}_o^t)] \\ - (\mathcal{D}_M^2(\mathbf{z}_o^t, \mathbf{z}_q^t) - \mathcal{D}_M^2(\mathbf{z}_o^t, \mathbf{z}_p^t))\Big]_+\end{aligned}$$

where $\lambda = \mathcal{O}(T\tilde{c})$. □

4.3. Proof of Theorem 3

Proof. When fixing M_{k-1} at the k-th iteration, we have

$$\begin{aligned}\mathcal{L}(M_{k-1}, \mathbf{z}^k) &\leq \sum_r \mathcal{G}_r(\mathbf{z}^k) \leq \sum_r \mathcal{F}_r(\mathbf{z}^k) \\ &\leq \sum_r \mathcal{F}_r(\mathbf{z}^{k-1}) = \mathcal{L}(M_{k-1}, \mathbf{z}^{k-1})\end{aligned}$$

When fixing $\mathbf{z}^k$, we have

$$\mathcal{L}(M_k, \mathbf{z}^k) \leq \mathcal{H}(M_k) \leq \mathcal{H}(M_{k-1}) = \mathcal{L}(M_{k-1}, \mathbf{z}^k)$$

Therefore, after each iteration, we have

$$\mathcal{L}(M_k, \mathbf{z}^k) \leq \mathcal{L}(M_{k-1}, \mathbf{z}^{k-1})$$

Since the value of $\mathcal{L}(\cdot)$ is bounded, the sequence will converge after a finite number of iterations. □

5. Experiments

We conduct the empirical study on four benchmark image data sets. 3-nearest neighbor classifier is applied to verify the efficacy of the learned metrics from different methods. The methods in the comparison are summarized as follows.

- **Euclid**: 3-NN with Euclidean distance.
- **LMNN** [17]: the state-of-the-art DML method that identifies a set of active triplets with the current metric at each iteration. The active triplets are searched within 3-nearest neighbors for each example.
- **OASIS** [2]: an online DML method that receives one random triplet at each iteration. It only updates the metric when the triplet constraint is active.
- **HR-SGD** [10]: one of the most efficient DML methods with SGD. We adopt the version that randomly samples a mini-batch of triplets at each iteration in the comparison. After sampling, a Bernoulli random variable is generated to decide if updating the current metric or not. With the PSD projection, it guarantees that the learned metric is in the PSD cone at each iteration.

- **MaPML$_\tau$**: the proposed method that learns the metric and latent examples simultaneously, where τ denotes the ratio between the number of latent examples and the number of original ones

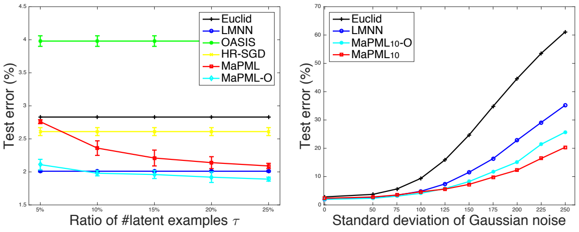
  $$\tau\% = \frac{m}{n}$$

  Different from other methods, 3-NN is implemented with latent examples as reference points. The method that takes 3-NN with original data is referred as **MaPML$_\tau$-O**.

The parameters of OASIS, HR-SGD and MaPML are searched in $\{10^i : i = -3, \cdots, 3\}$. The size of mini-batch in HR-SGD is set to be 10 as suggested [10]. To train the model sufficiently, the number of iterations for LMNN is set to be 10^3 while the number of randomly sampled triplets is 10^5 for OASIS and HR-SGD. The number of iterations for MaPML is set as $K = 10$ while the number of maximal iterations for solving M_k in the subproblem is set as $S = 10^4$, which roughly has the same number of triplets as OASIS and HR-SGD. All experiments are implemented on a server with 96 GB memory and 2 Intel Xeon E5-2630 CPUs. Average results with standard deviation over 5 trails are reported.

5.1. MNIST

First, we evaluate the performance of different algorithms on MNIST [8]. It consists of $60,000$ handwritten digit images for training and $10,000$ images for test. There are 10 classes in the data set, which are corresponding to the digits 0 - 9. Each example is a 28×28 grayscale image which leads to the 784-dimensional features and they are normalized to the range of $[0, 1]$.

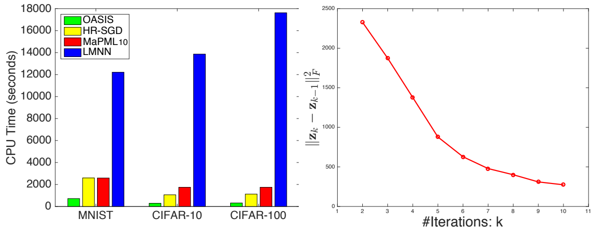


(a) Comparison of error rate (b) Comparison with Gaussian noise

Figure 3. Comparisons on MNIST.

Fig. 3 (a) compares the performance of different metrics on the test set. For MaPML, we vary the ratio of latent examples from 5% to 25%. First of all, It is obvious that the metrics learned with the active set outperform those from random triplets. It confirms that the strategy of sampling triplets randomly can not explore the data set sufficiently due to the extremely large number of triplets. Secondly, the performance of MaPML$_{10}$-O is comparable with LMNN, which shows that the proposed method can learn a good metric with only a small amount of latent examples (i.e., 10%). Finally, both MaPML and MaPML-O work well with the metric obtained by MaPML, which verifies that the learned metric can preserve the large margin property for both the original and latent data. Note that when the number of latent examples is small, the performance of k-NN with latent examples is slightly worse than that with the whole training set. However, k-NN with latent examples can be more robust in real-world applications.

To demonstrate the robustness, we conduct another experiment that randomly introduces the zero mean Gaussian noise (i.e., $\mathcal{N}(0, \sigma^2)$) to each pixel of the original training images. The standard deviation of the Gaussian noise is varied in the range of $[50/255, 250/255]$ and τ is fixed as 10. Fig. 3 (b) summarizes the results. It shows that MaPML$_{10}$ has the comparable performance as MaPML$_{10}$-O and LMNN when the noise level is low. However, with the increasing of the noise, the performance of LMNN drops dramatically. This can be interpreted by the fact that the metric learned with the original data has been misled by the noisy information. In contrast, the errors made by MaPML and MaPML-O increase mildly and it demonstrates that the learned metric is more robust than the one learned from the original data. MaPML performs best among all methods and it is due to the reason that the uncertainty in latent examples are much less than that in the original ones. It implies that k-NN with latent examples is more appropriate for real-world applications with large uncertainty.



(a) CPU time for training (b) Convergence curve of MaPML

Figure 4. Illustration of the efficiency of the proposed method.

Then, we compare the CPU time cost by different algorithms to evaluate the efficiency. The results can be found in Fig. 4 (a). First, as expected, all algorithms with SGD are more efficient than LMNN, which has to compute the full gradient from the redefined active set at each iteration. Moreover, the running time of MaPML$_{10}$ is comparable to that of HR-SGD, which shows the efficiency of MaPML with the small set of latent examples. Note that OASIS has the extremely low cost, since it allows the internal metric to be out of the PSD cone. Fig. 4 (b) illustrates the convergence curve of MaPML and shows that the proposed method converges fast in practice.

Finally, since we apply the proposed method to the original pixel features directly, the learned latent examples can be recovered as images. Fig. 5 illustrates the learned latent examples and the corresponding examples in the original training set. It is obvious that the original examples are from latent examples with different distortions as claimed.

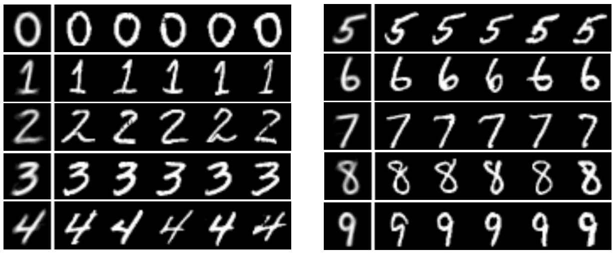


Figure 5. Illustration of the learned latent examples and corresponding original examples from MNIST. The left column indicates latent examples while five original images from each corresponding cluster are on the right.

5.2. CIFAR-10 & CIFAR-100

CIFAR-10 contains 10 classes with $50,000$ color images of size 32×32 for training and $10,000$ images for test. CIFAR-100 has the same number of images in training and test but for 100 classes [6]. Since deep learning algorithms show the overwhelming performance on these data sets, we adopt ResNet18 [4] in Caffe [5], which is pre-trained on ImageNet ILSVRC 2012 data set [14], as the feature extractor and each image is represented by a 512-dimensional feature vector.

Table 1. Comparison of error rate (%) on CIFAR-10 and CIFAR-100.

Methods	CIFAR-10	CIFAR-100
Euclid	16.81	42.57
OASIS	15.22 ± 0.18	42.46 ± 0.21
HR-SGD	15.16 ± 0.22	42.53 ± 0.19
LMNN	13.62 ± 0.12	40.05 ± 0.13
MaPML$_{10}$-O	13.59 ± 0.14	40.49 ± 0.15
MaPML$_{10}$	$\mathbf{12.64 \pm 0.16}$	$\mathbf{34.70 \pm 0.16}$

Table 1 summarizes error rates of methods in the comparison. First, we have the same observation as on MNIST, where the performance of methods adopting active triplets is much better than that of the methods with randomly sampled triplets. Different from MNIST, MaPML$_{10}$ outperforms LMNN on both of the data sets. It is because that images in these data sets describe natural objects which contain much more uncertainty than digits in MNIST. Finally, the performance of MaPML$_{10}$-O is superior over OASIS and HR-SGD, which shows the learned metric can work well with the original data represented by deep features. It confirms that the large margin property is preserved even for the original data.

5.3. ImageNet

Finally, we demonstrate that the proposed method can handle the large-scale data set with ImageNet. ImageNet ILSVRC 2012 consists of $1,281,167$ training images and $50,000$ validation data. The same feature extraction procedure as above is applied for each image. Given the large number of training data, we increase the number of triplets for OASIS and HR-SGD to 10^6. Correspondingly, the number of maximal iterations for solving the subproblem in MaPML is also raised to 10^5.

Table 2. Comparison of error rate (%) on ImageNet.

Methods	Test error (%)
Euclid	35.65
OASIS	36.51 ± 0.08
HR-SGD	36.15 ± 0.08
MaPML$_5$-O	35.59 ± 0.03
MaPML$_5$	$\mathbf{33.92 \pm 0.09}$

LMNN does not finish the training after 24 hours so the result is not reported for it. In contrast, MaPML obtains the metric within about one hour. The performance of available methods can be found in Table 2. Since ResNet18 is trained on ImageNet, the extracted features are optimized for this data set and it is hard to further improve the performance. However, with latent examples, MaPML can further reduce the error rate by 1.7%. It indicates that latent examples with low uncertainty are more appropriate for the large-scale data set as the reference points. Note that the small number of reference points will also accelerate the test phase. For example, it costs 0.15s to predict the label of an image with the original set while the cost is only 0.007s if evaluating with latent examples. It makes MaPML with latent examples a potential method for real-time applications.

6. Conclusion

In this work, we propose a framework to learn the distance metric and latent examples simultaneously. By learning from a small set of clean latent examples, MaPML can sample the active triplets efficiently and the learning procedure is robust to the uncertainty in the real-world data. Moreover, MaPML can preserve the large margin property for the original data when learning merely with latent examples. The empirical study confirms the efficacy and efficiency of MaPML. In the future, we plan to evaluate MaPML on different tasks (e.g., information retrieval) and different types of data. Besides, incorporating the proposed strategy to deep metric learning is also an attractive direction. It can accelerate the learning for deep embedding and the resulting latent examples may further improve the performance.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 2

[2] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 11:1109–1135, 2010. 1, 2, 4, 6

[3] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007. 1, 2

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 8

[5] 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 *ACM MM*, pages 675–678, 2014. 8

[6] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. 8

[7] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013. 2

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[9] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017. 2

[10] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu. Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD). *ML*, 99(3):353–372, 2015. 2, 6, 7

[11] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, pages 3716–3724, 2015. 4

[12] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012. 4

[13] O. Rippel, M. Paluri, P. Dollár, and L. D. Bourdev. Metric learning with adaptive density discrimination. In *ICLR*, 2016. 2

[14] 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. *IJCV*, 115(3):211–252, 2015. 8

[15] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 2

[16] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 2

[17] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009. 1, 2, 6

[18] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512, 2002. 1, 2

[19] L. Yang and R. Jin. Distance metric learning: a comprehensive survery. 2006. 2

[20] H. Ye, D. Zhan, X. Si, and Y. Jiang. Learning mahalanobis distance metric: Considering instance disturbance helps. In *IJCAI*, pages 3315–3321, 2017. 3