IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 24, 2019, accepted January 14, 2020, date of publication January 17, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967348

A Local-to-Global Metric Learning Framework From the Geometric Insight

YAXIN PENG[1], **(Member, IEEE), NIJING ZHANG**[1], **YING LI**[2], **AND SHIHUI YING**[1], **(Member, IEEE)**

[1]Department of Mathematics, School of Science, Shanghai University, Shanghai 200444, China
[2]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

Corresponding author: Ying Li (yinglotus@t.shu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 11771276, Grant 61573274, and Grant 11971296, and in part by the Capacity Construction Project of Local Universities in Shanghai under Grant 18010500600.

ABSTRACT Metric plays a key role in the description of similarity between samples. An appropriate metric for data can well represent their distribution and further promote the performance of learning tasks. In this paper, to better describe the heterogeneous distributions of data, we propose a semi-supervised local-to-global metric learning framework from the geometric insight. Our contributions can be summarized as: Firstly, to enlarge the application scope of local metric learning, we introduce the unsupervised information as the regularization term into our smoothly glued nonlinear metric model. Secondly, we propose two different nonlinear semi-supervised metric learning models with two different loss terms, and find that the smooth loss performs better than the hinge loss by comparison results. Thirdly, we have established not only two metric learning models, but also a nonlinear metric learning framework based on local metrics, which includes supervised and semi-supervised as well as linear and nonlinear metric learning. Moreover, we present an intrinsic steepest descent algorithm on the positive definite manifold for implementation of our semi-supervised nonlinear metric learning models with smooth triplet constrain loss. Finally, we compare our approaches with several state-of-the-art methods on a variety of datasets. The results validate that the robustness and accuracy of classification are both improved under our metrics.

INDEX TERMS Local metric learning, semi-supervised method, partition of unity, intrinsic steep descent method.

I. INTRODUCTION

In artificial intelligence tasks, it is often required to judge whether a sample is similar or dissimilar to others. Thus, we need a measurement to evaluate the similarity of samples. Metric is often used as a tool for this purpose. Therefore, learning an appropriate distance metric to measure the distance or similarity between samples, i.e., metric learning, becomes one of the key topics in machine learning [1]–[4]. The applications of metric learning include re-identification, medical image analysis and place recognition [5]–[10].

From the geometric viewpoint, metric learning can be categorized as linear and nonlinear metric learnings [11], [12]. For example, the linear metric learning can be formulated by a globally linear mapping from the original domain to the new data space.

To improve the separability in classification and retrieval tasks, metric learning often aims to find an appropriate distance metric which keeps similar samples closer while dissimilar ones farther. A variety of linear metric learning methods are developed, such as Distance Metric Learning (DML) [13], Neighborhood Component Analysis (NCA) [14], Large Margin Nearest Neighbors (LMNN) [15], and Mirror Descent Metric Learning (MDML) [16]. Among them, LMNN is the most representative method which separates samples from different classes with a large margin while keeping k-nearest neighbors similar and tight [15]. Both empirically and theoretically, these metric learning methods can significantly improve the performance in a variety of machine learning problems via a linear global metric. In many real world applications, however, such methods cannot fit the complicated data distribution well and lead to unsatisfactory performance.

Therefore, many methods on nonlinear metric learning, including kernel and multi-metric methods, have been

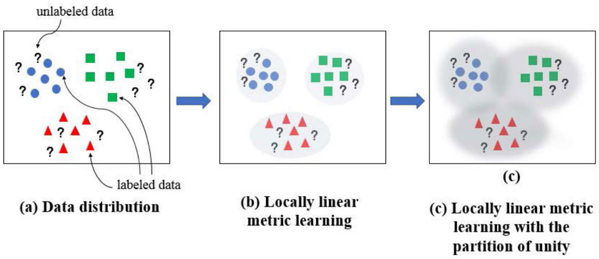
The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

proposed and performed well on many real datasets. In particular, kernel-basic metric learning is to map the original low-dimensional data into a high-dimensional feature space for a better separation. Davis *et al.* [17] kernelize the model by a low-rank kernel learning method in Information-Theoretic Metric Learning (ITML). Jain *et al.* [18] propose a general kernel-based framework for learning metrics via linear transformations, which bridges the metric learning and kernel learning. Later, Li *et al.* [19] develop a multiple kernel metric learning method to choose an appropriate kernel for classification. However, the computational burden limits their application to large scale datasets.

Meanwhile, multi-metric learning is to learn multiple local metrics on local regions [20], [21]. A usual kind of local metric learning method constructs either instance-specific metrics [22] or cluster-specific metrics based on prior knowledge such as label information [23]. For instance, Zhan *et al.* propose an Instance Specific Distances learning by metric propagation under a convex optimization framework, which propagates and adapts metrics of individual labeled instances to individual unlabeled ones. Wang *et al.* [24] propose Parametric Local Metric Learning (PLML) based on LMNN, which defines several anchor points with different basic Mahalanobis metrics. Later, Peng *et al.* [25] develop Global Nonlinear Smooth Metric Learning (GNSML) by gluing the local linear metrics, which constructs a smooth nonlinear metric for every sample. Then, Nguyen *et al.* [26] propose the Clustered Multi-Metric Learning (CMML) by learning multiple distance metrics jointly with triplet constraints constructed in clusters.

Nevertheless, these methods are supervised and do not sufficiently consider those unlabeled data. Thus, making good use of those large amounts of unlabeled data is a challenging problem. Semi-supervised methods can exploit the information of labeled and unlabeled data to learn an appropriate metric, which combines the advantages of supervised and unsupervised metric learning methods. With semi-supervised clustering, Bilenko *et al.* [27] conduct distance metric learning by using the pairwise constraints to learn an appropriate metric. Later, on the basis of such model, Hoi *et al.* [28] propose a semi-supervised metric learning method, which encodes the similarity between point pairs with a weight matrix and a graph Laplacian regularity. Moreover, Li and Fu [29] introduce the low-rank constraint into the semi-supervised metric learning. Liu *et al.* [30] construct the weight matrix in a refined way. Baghshah and Shouraki [31], Zhong *et al.* [32] and Wang *et al.* [33] use different manifold regularizers respectively, in which Wang *et al*. define three semi-supervised assumptions, i.e., smoothness, manifold and cluster, via density and similarity [33].

However, most of these semi-supervised metric learning methods simply learn a global linear metric, which may fail to deal with heterogeneously distributed data. Furthermore, we need a smooth metric defined on every point in the entire space, not only on the training set.



(a) Data distribution

(b) Locally linear metric learning

(c) Locally linear metric learning with the partition of unity

FIGURE 1. Flowchart of the proposed local-to-global metric learning framework.

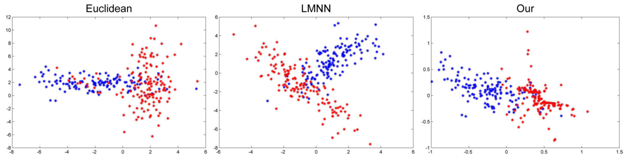


FIGURE 2. Visualization on a toy example with heterogeneous distribution. Different colors represent different classes.

In this paper, we propose a semi-supervised local-to-global metric learning framework with multi-local metrics for heterogeneously distributed data, which introduces the manifold regularization to preserve the data structure. In particular, we conduct metric learning by smoothly gluing locally linear metrics, whose basic framework is illustrated in Fig. 1. It is worth to mention that, as illustrated in Fig. 2, our framework can handle the heterogeneously distributed data well, while the global metric learning method LMNN fails.

The remainder of this paper is organized as follows. In Section II, we introduce a local-to-global metric construction theory and strategy with the partition of unity. Under this metric construction, in Section III we propose a semi-supervised multi-metric learning framework, and then discuss two models for the semi-supervised local metric learning problem with different triplet constraint losses and the associated optimization algorithms. In Section IV, we conduct extensive experiments to validate the effectiveness of the proposed methods. Section V concludes the paper.

II. THE PROPOSED METHODOLOGY

The goal of metric learning is to learn an appropriate metric for describing data. Since the data are often nonlinear and heterogeneously distributed and can be viewed lying on a latent manifold, a global linear metric is no longer proper. As we know, manifold can be viewed as Euclidean locally. Intuitively, to fit the data manifold locally, we consider learning multiple local linear metrics since similar samples share the similar distribution within a local area. Then we glue these local metrics smoothly to have a smooth nonlinear metric for every point in the entire data space.

A. THE LOCAL-TO-GLOBAL METRIC CONSTRUCTION

Therefore, a smooth way to glue the locally linear metric on the data manifold is needed. Fortunately, the theorem of Partition of Unity [34], an important theorem in differentiable manifold, establishes a bridge from local to global. Intuitively, we can use the theorem of Partition of Unity for gluing the local structures to obtain a global smooth metric and to fit the data manifold. Here we state one version of the theorem.

Theorem 1 (Partition of Unity): Let $A \subseteq \mathbb{R}^n$ and $\mathcal{O}$ be an open cover of A. Then there exists a collection Φ of C^∞ functions ϕ defined on an open set containing A, with the properties below:

1) $\forall \boldsymbol{x} \in A,\ 0 \leq \phi(\boldsymbol{x}) \leq 1$.

2) $\forall \boldsymbol{x} \in A$, there is an open set V containing $\boldsymbol{x}$ such that all but finitely many $\phi \in \Phi$ are 0 on V.

3) $\forall \boldsymbol{x} \in A$, $\sum_{\phi \in \Phi} \phi(\boldsymbol{x}) = 1$.

4) $\forall \phi \in \Phi$, $\exists U \in \mathcal{O}$, s.t. $\phi = 0$ holds outside of some closed set contained in open set U.

A collection Φ that satisfies 1) to 3) is called a C^∞ partition of unity for A. Moreover, if Φ also satisfies 4), it is said to be subordinate to the cover $\mathcal{O}$. We introduce the strategy to construct a novel local-to-global distance metric via the Partition of Unity in the following.

Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, without loss of generality, we assume that the first l samples in $\mathcal{D}$ are the labeled data $\mathcal{L} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l\}$ with the label $\mathcal{Y} = \{y_1, \cdots, y_l\}$.

Definition 1 (Smoothly Glued Metric): Suppose we construct a partition of unity $\{\phi_c \in C^\infty(\mathcal{M}) \mid 1 \leq c \leq N\}$ on the data manifold $\mathcal{M}$ (samples are finite so the manifold is compact) such that the global metric M at any point $\boldsymbol{x}$ could be defined as:

$$M(\boldsymbol{x}) = \sum_{c=1}^{N} \phi_c(\boldsymbol{x}) M_c, \quad \forall \boldsymbol{x} \in \mathcal{M},$$

where $\phi_c(\boldsymbol{x})$ is a truncated function for $0 \leq \phi_c(\boldsymbol{x}) \leq 1$, $\sum_{c=1}^{N} \phi_c(\boldsymbol{x}) = 1$, and $M_1, M_2, \cdots, M_N$ are N basic metrics defined on N local regions.

These local regions could 1) be clusters acquired via some structure unsupervised methods, such as k-means; and 2) depend on the label information of classes. For simplicity, we allow different classes be viewed as different regions in our experiments. So, N could be the number of different classes. Then the squared distance between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be defined as:

$$D_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T m(M(\boldsymbol{x}_i), M(\boldsymbol{x}_j))(\boldsymbol{x}_i - \boldsymbol{x}_j),$$

where $m(M(\boldsymbol{x}_i), M(\boldsymbol{x}_j))$ is set as $M(\boldsymbol{x}_j)$ by default in some literature. Obviously, the symmetry $D_{ij}^2 = D_{ji}^2$ cannot be satisfied when the metrics defined on $M(\boldsymbol{x}_i)$ and $M(\boldsymbol{x}_j)$ are different. We then introduce a symmetric metric as

$$\begin{aligned} m(M(\boldsymbol{x}_i), M(\boldsymbol{x}_j)) &= (M(\boldsymbol{x}_i) + M(\boldsymbol{x}_j))/2 \\ &= \sum_{c=1}^{N} \frac{\phi_c(\boldsymbol{x}_i) + \phi_c(\boldsymbol{x}_j)}{2} M_c. \end{aligned}$$

Therefore, the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be rewritten with the symmetrical metric

$$D_{ij}^2 = \sum_{c=1}^{N} \frac{\phi_c(\boldsymbol{x}_i) + \phi_c(\boldsymbol{x}_j)}{2} (\boldsymbol{x}_i - \boldsymbol{x}_j)^T M_c (\boldsymbol{x}_i - \boldsymbol{x}_j).$$

The key of this work is the construction of truncated functions $\{\phi_c\}_{c=1}^N$.

Unlike most existing metrics defined only on the training data, we try to define a metric on every point in the entire data space by establishing the truncated function with the heat-diffusion:

$$\phi_c(\boldsymbol{x}) = f_c(\boldsymbol{x}) / \sum_i f_i, \quad f_c(\boldsymbol{x}) = e^{-\frac{d_c^2(\boldsymbol{x}, \boldsymbol{x}_c)}{\sigma_0^2}},$$

where $\boldsymbol{x}_c$ is the center point of the c-th class, and σ_0 is a hyperparameter to describe the covering of samples. By definition, the value of $f_c(\boldsymbol{x})$ is inversely proportional to the distance between sample $\boldsymbol{x}$ and $\boldsymbol{x}_c$, which means that the basic metric M_c would have less effect on the samples farther away from the class center. As long as the basic metrics are given, the global metric value at any sample $\boldsymbol{x}$ is determined. Thus, we obtain the smooth metric function defined on every point in the entire manifold.

B. THE PROPOSED SEMI-SUPERVISED METRIC LEARNING FRAMEWORK

In this section, we firstly propose a semi-supervised metric learning framework by defining an objective function as

$$E = \lambda Reg + (1 - \lambda) L, \tag{1}$$

where L is the loss of data fitting term, and Reg is the regularization term. Here $0 \leq \lambda \leq 1$ is a trade-off parameter. Usually, the loss term keeps the inner- and inter- class data balanced, and the regularization term controls the smoothness and structure.

C. THE TRIPLET CONSTRAINT LOSS

For well fitting the data and quantifying the similarity of the samples, we introduce a triplet constraint in an intuitive way, i.e.,

$$\mathcal{T} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) : D_{ij}^2 < D_{ik}^2\},$$

where $\boldsymbol{x}_i$ is similar to $\boldsymbol{x}_j$, and dissimilar to $\boldsymbol{x}_k$. It means that the distance of samples in different classes should be as large as possible, while as small as possible in the same class.

Hinge loss function [25] [15] is often used for the triplet constraint $D_{ij}^2 + 1 < D_{ik}^2$ as

$$L_H = \sum_{\mathcal{T}} [1 + D_{ij}^2 - D_{ik}^2]_+, \tag{2}$$

where $[m]_+ = m$ if $m \geq 0$, and 0 otherwise. It is clear that the hinge loss function is non-smooth and time-consuming for computation. Furthermore, to balance the scale between the similar and dissimilar data, we rewrite the triplet constraint in a smooth version as

$$\mathcal{T}' = \{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) : D_{ij}^2 < \gamma \ D_{ik}^2\}$$

by introducing an adaptive parameter $\gamma = 1/(1 + \tilde{D}^{-1}) \in (0, 1]$, where $\tilde{D}$ is the mean distance between samples. The parameter γ fully depends on the distribution of dataset, and can well balance the difference of inner- and inter-class.

Then, we obtain the corresponding smooth loss function for this smooth triplet constraint as

$$L_S = \sum_{\mathcal{T}'} (D_{ij}^2 - \gamma D_{ik}^2). \tag{3}$$

With this loss function, we can well describe the dataset structure and reduce the computation complexity. Nevertheless, the corresponding loss term is supervised and prone to over-fitting due to the absence of regularization, especially in high dimensions.

D. REGULARIZATION TERM

Inspired by the regularization method in [33], with the help of some unlabeled data, we introduce a regularizer

$$Reg = \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} D_{ij}^2 \tag{4}$$

to better describe the sample structure and distribution.

Here $\beta_i \in \mathbb{R}^+$ is a parameter related to density for the sample $\boldsymbol{x}_i$, $N(i)$ is the set of the neighbor samples of $\boldsymbol{x}_i$, and S_{ij} is the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. This term can better preserve the topology structure by penalizing large distance between inputs and their neighbors. Moreover, an adaptive weight depending on density is used to penalize the cluster information. The regularization term is unsupervised since the label information are unknown.

E. SEMI-SUPERVISED METRIC LEARNING MODELS

Based on the analysis of the triplet constraint loss (2) (3) and regularizer (4), the optimization problem for semi-supervised metric learning framework can be established.

We consider two semi-supervised metric learning models based on different triplet constraints respectively, namely Partition of Unity based Local Learning Metric with Hinge loss (PULLMH) and Partition of Unity based Local Learning Metric with Smooth constraint (PULLMS). The two models can be presented as

$$\begin{aligned} \min_{M_1, \cdots, M_N} \ & E_H = \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} D_{ij}^2 \\ & \quad + (1-\lambda) \sum_{\mathcal{T}} [1 + D_{ij}^2 - D_{ik}^2]_+ \\ \text{s.t.} \ & M_1, \cdots, M_N \succeq \mathbf{0}, \end{aligned} \tag{5}$$

and

$$\begin{aligned} \min_{M_1, \cdots, M_N} \ & E_S = \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} Dn_{ij}^2 \\ & \quad + (1-\lambda) \sum_{\mathcal{T}'} (D_{ij}^2 - \gamma D_{ik}^2) \\ \text{s.t.} \ & M_1, \cdots, M_N \succ \mathbf{0}, \end{aligned} \tag{6}$$

respectively.

E_H is the objective function of PULLMH, while E_S is that of PULLMS. In fact, the previous work [35] [25] can be viewed as two special cases of our metric learning framework when the local metrics $M_1, \cdots, M_N$ are the same or the trade-off parameter $\lambda = 0$ respectively.

III. OPTIMIZATION

We will optimize these two models with different optimization methods in this section.

A. OPTIMIZATION MODEL WITH HINGE LOSS - PULLMH

The optimization problem for PULLMH can be rewritten as:

$$\begin{aligned} \min_{M_1, \cdots, M_N} \ & E_H = \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} Sn_{ij} Dnn_{ij}^2 \\ & \quad + (1-\lambda) \sum_{i,j,k=1}^{l} h_{ij}(1 - y_{ik})[1 + D_{ij}^2 - D_{ik}^2]_+ \\ \text{s.t.} \ & M_1, \cdots, M_N \succeq \mathbf{0}, \end{aligned} \tag{7}$$

and

$$h_{ij} = \begin{cases} 1 & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are neighbors in the same class;} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_k \text{ are in the same class;} \\ 0 & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_k \text{ are in different classes} \end{cases}$$

are indicator functions respectively.

The second term in (7) is the loss, penalizing small distances between samples with different labels. We hope that there exists a finite margin (default 1) between samples with different labels. Like [23], we implement an iterative sub-gradient projection method to optimize the model (7) in terms of the positive semi-definite metrics $M_1, \cdots, M_N$.

Specifically, to simplify the notation, let $Z_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T$ and $\Phi_{ij}^c = \frac{\phi_c(\boldsymbol{x}_i) + \phi_c(\boldsymbol{x}_j)}{2}$. Then at the t-th iteration, the squared distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $D_{ij}^2(t) = \sum_{c=1}^{N} \Phi_{ij}^c \operatorname{tr}(M_c(t) Z_{ij})$. Consequently, (7) can be rewritten as:

$$\begin{aligned} E_H &= \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} \sum_{c=1}^{N} \Phi_{ij}^c \operatorname{tr}(M_c(t) Z_{ij}) \\ & \quad + (1-\lambda) \sum_{i,j,k=1}^{l} h_{ij}(1 - y_{ik}) \\ & \quad [1 + \sum_{c=1}^{N} \Phi_{ij}^c \operatorname{tr}(M_c(t) Z_{ij}) - \sum_{c=1}^{N} \Phi_{ik}^c \operatorname{tr}(M_c(t) Z_{ik})]_+ \end{aligned} \tag{8}$$

Note that (8) is piecewise linear with respect to the basic metrics $M_1, \cdots, M_N$. Without the loss of generality, we define a set of triplets N_t, such that $(i, j, k) \in N_t$ if and only if the indices (i, j, k) trigger the hinge loss in (8). Then we can get the gradient $G_c(t)$ of $E(M_c)$ as:

$$\begin{aligned} G_c(t) = \frac{\partial E}{\partial M_c(t)} &= \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} \Phi^c_{ij} Z_{ij} \\ &+ (1-\lambda) \sum_{i,j,k \in N_t} h_{ij}(1-y_{ik})(\Phi^c_{ij} Z_{ij} - \Phi^c_{ik} Z_{ik}) \end{aligned} \tag{9}$$

Thus, at the next $(t+1)$-th iteration,

$$\hat{M}_c(t+1) = M_c(t) - \alpha(t) G_c(t),$$

where $\alpha(t)$ is the optimal step-size at t-th iteration. Since (8) requires the positive semi-definiteness of the basic metric M_c, we project $\hat{M}_c(t+1)$ onto the positive semi-definite cone:

$$M_c(t+1) = \mathcal{P}_{\mathcal{S}}(\hat{M}_c(t+1)) = V \max(\Delta, \mathbf{0}) V^T,$$

where V and Δ are the matrices of eigenvectors and eigenvalues of $\hat{M}_c(t+1)$. Therefore, we have the iterative formula of metrics.

However, the sub-gradient projection method can only find a semi-definite positive matrix, which in fact is a pseudo-metric since it cannot satisfy the distinguishability. Therefore, we use the intrinsic steepest descent (ISD) algorithm to solve the optimization problem (6) on positive definite matrix group.

B. OPTIMIZATION MODEL WITH SMOOTH LOSS - PULLMS

The optimization problem for PULLMS can be rewritten as:

$$\begin{aligned} \min_{M_1,\cdots,M_N} \ E_S &= \lambda \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} D^2_{ij} \\ &+ (1-\lambda) \sum_{i,j,k=1}^{l} h_{ij}(1-y_{ik})(D^2_{ij} - \gamma D^2_{ik}) \\ \text{s.t. } & M_1, \cdots, M_N \succ \mathbf{0}, \end{aligned} \tag{10}$$

It is remarkable that the basic metrics $M_1, \cdots, M_N$ in (10) are positive definite and can be solved in the positive definite matrix group with intrinsic matrix iteration. To achieve that, we rewrite the model as a minimization problem with the variable of matrix.

First, we translate the loss term in (10) as

$$\begin{aligned} &\sum_{i,j,k=1}^{l} h_{ij}(1-y_{ik})(D^2_{ij} - \gamma D^2_{ik}) \\ &\quad = \sum_{i,j=1}^{l} [(\boldsymbol{e}_i^T(\mathbf{1}\mathbf{1}^T - Y)\mathbf{1}) h_{ij} D^2_{ij} - \gamma \boldsymbol{e}_i^T H \mathbf{1}(1-y_{ij}) D^2_{ij}], \end{aligned}$$

where $Y = (y_{ij})_{l \times l}$ and $H = (h_{ij})_{l \times l}$ are indicator matrices, $\mathbf{1}$ is a column vector with all elements 1, and $\boldsymbol{e}_i$ is a column vector with all elements 0 but the ith element 1.

If let

$$P^1_{ij} = (\boldsymbol{e}_i^T(\mathbf{1}\mathbf{1}^T - Y)\mathbf{1}) h_{ij},$$

$$P^0_{ij} = \gamma \boldsymbol{e}_i^T H \mathbf{1}(1-y_{ij}),$$

then

$$\sum_{i,j,k=1}^{l} h_{ij}(1-y_{ik})(D^2_{ij} - \gamma D^2_{ik}) = \sum_{i,j=1}^{l} (P^1_{ij} D^2_{ij} - P^0_{ij} D^2_{ij}).$$

Also, let $P_{ij} = P^1_{ij} - P^0_{ij}$, then

$$\sum_{i,j,k=1}^{l} h_{ij}(1-y_{ik})(D^2_{ij} - \gamma D^2_{ik}) = \sum_{i,j}^{l} P_{ij} D^2_{ij}. \tag{11}$$

Because

$$\begin{aligned} D^2_{ij} &= \sum_{c=1}^{N} \frac{\phi_c(\boldsymbol{x}_i) + \phi_c(\boldsymbol{x}_j)}{2} (\boldsymbol{x}_i - \boldsymbol{x}_j)^T M_c (\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= \sum_{c=1}^{N} \Phi^c_{ij} (\boldsymbol{x}_i^T M_c \boldsymbol{x}_i + \boldsymbol{x}_j^T M_c \boldsymbol{x}_j - 2\boldsymbol{x}_i^T M_c \boldsymbol{x}_j), \end{aligned}$$

Eq. (11) can be rewritten by

$$\begin{aligned} &\sum_{i,j=1}^{l} P_{ij} D^2_{ij} \\ &= \sum_{i,j=1}^{l} \sum_{c=1}^{N} (P_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M_c \boldsymbol{x}_i \\ &\quad + P_{ij} \Phi^c_{ij} \boldsymbol{x}_j^T M_c \boldsymbol{x}_j - 2 P_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M_c \boldsymbol{x}_j) \\ &= \sum_{i,j=1}^{l} \sum_{c=1}^{N} (\mathrm{tr}(P_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M n_c \boldsymbol{x}_i) + \mathrm{tr}(P_{ij} \Phi^c_{ij} \boldsymbol{x}_j^T M_c \boldsymbol{x}_j) \\ &\quad - 2\,\mathrm{tr}(P_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M_c \boldsymbol{x}_j)) \\ &= \sum_{i,j=1}^{l} \sum_{c=1}^{N} (\mathrm{tr}(\boldsymbol{x}_i P n n_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M_c) + \mathrm{tr}(\boldsymbol{x}_j P_{ij} \Phi^c_{ij} \boldsymbol{x}_j^T M_c) \\ &\quad - 2\,\mathrm{tr}(\boldsymbol{x}_j P n_{ij} \Phi^c_{ij} \boldsymbol{x}_i^T M n_c)) \\ &= \sum_{c=1}^{N} \mathrm{tr}(X_L (D^c_P - 2Q^c) X_L^T M_c), \end{aligned} \tag{12}$$

where $X_L = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l]$ is the labeled data matrix, $D^c_P = \mathrm{diag}(\mathrm{diag}(P(\Phi^c)^T + P^T \Phi^c))$, $Q^c = P \odot \Phi^c$, and $\odot$ is the multiplication between corresponding elements in two matrices.

Similarly, we can calculate the regularization term:

$$\begin{aligned} Reg(M_c) &= \sum_{i=1}^{n} \beta_i \sum_{j \in N(i)} S_{ij} D^2_{ij} \\ &= \sum_{i=1}^{n} \beta_i \sum_{j=1}^{n} N_{ij} S_{ij} D^2_{ij} \\ &= \sum_{i,j=1}^{n} W_{ij} D^2_{ij}, \end{aligned} \tag{13}$$

where $N_{ij} = \begin{cases} 1, & \text{if } j \in N(i), \\ 0, & \text{if } j \notin N(i), \end{cases}$ and $W_{ij} = \beta_i N_{ij} S_{ij}$.

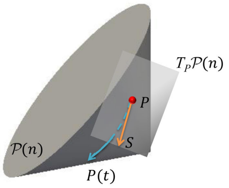


FIGURE 3. Geometry of the positive definite matrix group.

Clearly, similar to the loss term, the regularization term of (10) can be rewritten by

$$Reg(M_c) = \sum_{c=1}^{N} \mathrm{tr}(X(D_W^c - 2R^c)X^T M_c),$$

where $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$ is the data matrix consisting of all samples, $D_W^c = \mathrm{diag}(\mathrm{diag}(W(\Phi^c)^T + W^T\Phi^c))$, $W = (\mathrm{diag}(\beta)S) \odot N$, and $R^c = W \odot \Phi^c$.

Consequently, the model (10) is rewritten in the matrix form:

$$\min_{M_1,\cdots,M_N} E_S = \sum_{c=1}^{N}[\lambda\,\mathrm{tr}(A_c M_c) + (1-\lambda)\,\mathrm{tr}(B_c M_c)]$$
$$\text{s.t. } M_1, \cdots, M_N \succ 0. \tag{14}$$

where $A_c = X(D_W^c - 2R^c)X^T$, and $B_c = X_L(D_P^c - 2Q^c)X_L^T$.

In order to avoid solving (14) on a positive definite matrix group and projecting onto the PSD cone, we use the ISD algorithm [36] [35].

1) GEODESIC STRUCTURE ON SYMMETRIC POSITIVE DEFINITE GROUP

Let $\mathcal{P}(n)$ be the set of all n-th order symmetric positive definite matrices. That is,

$$\mathcal{P}(n) := \{P \in \mathbb{R}^{n\times n} | P = P^T, P \succ 0\}.$$

Note that $\mathcal{P}(n)$ is a smooth Riemannian manifold with dimension $n(n+1)/2$. Its tangent space $T_P\mathcal{P}(n)$ at the point $P \in \mathcal{P}(n)$ represents the set of all tangent vectors at point P, which is a local linearization of $\mathcal{P}(n)$ at the point P. Then, from the Riemannian manifold structure, a geodesic starting from the identity I along the direction of $S \in T_I\mathcal{P}(n)$ is given explicitly by the exponential map $\exp(tS)$ in the neighborhood of I.

As shown in Fig. 3, using the invariance under congruent transformations, the blue geodesic $P(t)$ with $P(0) = P \in \mathcal{P}(n)$ and $\dot{P}(0) = S \in T_P\mathcal{P}(n)$ is therefore given by

$$P(t) = P^{\frac{1}{2}} \exp(tP^{-\frac{1}{2}}SP^{-\frac{1}{2}})P^{\frac{1}{2}}.$$

2) INTRINSIC ITERATIVE ALGORITHM FOR PULLMS MODEL

Inspired by the geodesic structure on symmetric positive definite group, on the positive definite matrix group, the iterative formula from the current step $M_c(t)$ to the next step $M_c(t+1)$ turns to be

$$M_c(t+1)=M_c(t)^{\frac{1}{2}} \exp[\alpha(t)\cdot M_c(t)^{-\frac{1}{2}}S(t)M_c(t)^{-\frac{1}{2}}]M_c(t)^{\frac{1}{2}}, \tag{15}$$

where $M_c(t)^{-\frac{1}{2}}S(t)M_c(t)^{-\frac{1}{2}}$ is a descent direction, and $\alpha(t)$ is the optimal step-size at time t. The minor gradient, i.e., the steepest descent direction, is always selected as the descent direction. Then, the iterative formula is rewritten by

$$M_c(t+1) = M_c(t)^{\frac{1}{2}} \exp(-\alpha(t)\cdot G_c(t))M_c(t)^{\frac{1}{2}}, \tag{16}$$

where $-G_c(t)$ is the steepest descent direction at point $M_c(t)$. As the exponential $\exp(M)$ of a matrix M in Lie groups is defined by $\exp(M) = \sum_{k=0}^{\infty} \frac{1}{k!}M^k$, the exponential of any symmetric matrix is a positive-definite symmetric matrix. So $M_c(t+1)$ is still symmetric positive definite.

According to the objective function $E_S(M_1, \cdots, M_N)$ of (14), we gain a steepest descent flow (16) on the positive definite matrix group. Generally, the gradient of the objective function should be symmetrized with a symmetric operator

$$\mathrm{Sym}[\nabla_{M_c(t)}E_S] := \frac{1}{2}[\nabla_{M_c(t)}E_S + \nabla_{M_c(t)}E_S^T],$$

before it is selected as the gradient direction.

It is remarkable that, in the geometric viewpoint, the symmetrization of the gradient is a projection of the gradient vector to the Lie algebra of the positive definite matrix group.

The symmetrization step is omitted here, since the gradient of the objective function E_S in our model (10) is symmetric. That is, we have the descent direction as:

$$G_c(t) = [M_c(t)]^{-\frac{1}{2}}[\nabla_{M_c(t)}E_S][M_c(t)]^{-\frac{1}{2}}.$$

The rest is calculating the gradient of the objective function by

$$\nabla_{M_c(t)}E_S = \lambda A_c^T + (1-\lambda)B_c^T.$$

After obtaining the partition of unity $\phi_c(\boldsymbol{x})$ and the metric M_c, we input the labeled samples and use the learned squared distance D_{ij}^2 for classification via the kNN method.

3) THE COMPUTATIONAL COMPLEXITY OF THE PROPOSED OPTIMIZATION METHOD

Note that the ISD algorithm on matrix manifolds is at least linear convergence. Denoting the precision of the objective function by $\epsilon > 0$, the iteration will be terminated at $O(\log \frac{1}{\epsilon})$. In each iteration, the distance with the k-nearest samples should be calculated for n samples, and hence the complexity is $O(n^2)$. Therefore, the total complexity is $O(n^2 \log \frac{1}{\epsilon})$.

For the local metrics version, in each iteration, n samples should calculate the distance with the k-nearest samples under N basic metrics, and hence the complexity is $O(n^2N)$. Therefore, the total complexity is $O(n^2N \log \frac{1}{\epsilon})$.

IV. EXPERIMENTS

To demonstrate the effectiveness of the proposed metric learning methods, we implement experiments on five datasets, i.e., UCI dataset[1], ORL face dataset[2], USPS digit

[1] http://archive.ics.uci.edu/ml/datasets.php

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

TABLE 1. Details of the datasets.

Datasets	Samples	Attributes	Classes	$\|\mathcal{L}\|$	$\|\mathcal{U}\|$	$\|\mathcal{L}\|/\|\mathcal{D}\|$ (%)
wine	178	13	3	15	163	8.43
iris	150	4	3	15	135	10
balance	625	4	3	15	610	2.4
dermatology	358	34	6	30	328	8.38
ionosphere	351	34	2	20	331	5.70
heart	270	13	2	20	250	7.41
ORL	400	1024	40	120	200	30
USPS	1000	256	10	300	500	30
COIL-20	1440	1439	20	432	720	30
ADNI	103	93	2	10	93	9.7

image dataset[3], COIL-20 object dataset[4], and Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset[5]. Table 1 provides the details of these datasets, where $|\mathcal{L}|$, $|\mathcal{U}|$, $|\mathcal{D}|$ are the sizes of the labeled data, unlabeled data and the whole dataset, respectively, and $|\mathcal{L}|/|\mathcal{D}|$ is the labeled sample ratio.

We evaluate the 1NN classification performance after PULLMH and PULLMS, respectively. Then, we compare them with

1) supervised single metric learning method: LMNN [23];

2) semi-supervised single metric learning method: ISSML [35];

3) supervised multiple metric learning methods: multiple metrics LMNN (mmLMNN) [23], PLML [24], GNSML [25] and CMML [26];

4) a baseline experiment, i.e., the Euclidean method, is conducted by using kNN classifier with the Euclidean distance. The metric matrix M is initialized by the identity matrix I.

A. CLASSIFICATION ON UCI DATASETS

For UCI, the experiments of classification are conducted on six datasets, i.e., wine, iris, balance, heart, dermatology and ionosphere. We randomly separate all datasets into two subsets: the labeled dataset $\mathcal{L}$, and the unlabeled dataset $\mathcal{U}$. Specifically, we select the same number of samples from each class as the labeled dataset, then the rest samples as the unlabeled dataset. The unlabeled dataset also acts as the testing dataset due to data inadequacy. Training samples are randomly chosen in every experiment. The average results of 30 times are presented.

1) EXPERIMENTAL SETTING

We set different values of the trade-off parameter λ for different datasets, by tuning on the training data. In order to calculate $\beta_i = f[p(\boldsymbol{x}_i)]$, a simple linear mapping $f[p(\boldsymbol{x}_i)] = p(\boldsymbol{x}_i)$ is adopted. We use the Parzen window to estimate the density

$$p(\boldsymbol{x}_i) = \frac{1}{|\mathcal{N}(i)|h^d} \sum_{j \in \mathcal{N}(i)} K_h\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{h}\right),$$

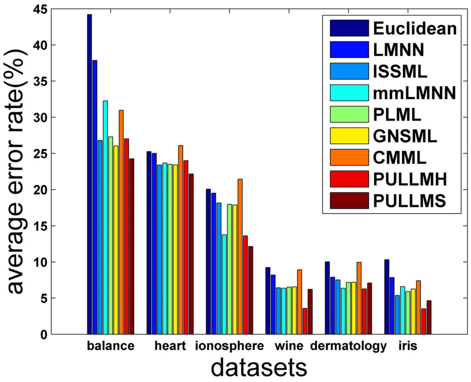


FIGURE 4. Test error rates for 1NN classification using different metrics on UCI.

where $\mathcal{N}(i)$ is the neighbor list of $\boldsymbol{x}_i$, its size $|\mathcal{N}(i)|$ is set to 10, d is the dimension of the sample $\boldsymbol{x}_i$, h is the bandwidth, and $K_h : \mathbb{R}^d \rightarrow \mathbb{R}$ is a Gaussian-kernel. Then we normalize the estimated density by $p(\boldsymbol{x}_i) := p(\boldsymbol{x}_i)/\max\{p(\boldsymbol{x})\}$.

Similarity can be calculated as

$$S_{ij} = \exp(-d_{ij}^2/2\sigma^2),$$

where d_{ij} is the Euclidean distance between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and

$$\sigma = \min D + (1/v)(\max D - \min D), \tag{17}$$

where $\max D$ and $\min D$ are the maximum and minimum Euclidean distances between samples in the dataset, respectively. In our models, the value of v depends on the dataset, varying from 1 to 15.

2) EXPERIMENTAL RESULTS

By using the 1 nearest neighborhood (1NN) classification, the results under the learned metrics via LMNN, ISSML, mmLMNN, PLML, GNSML, CMML and our methods PULLMH, PULLMS are shown in Fig. 4.

Table 2 and Fig. 4 provide the classification results (average error rate) based on a 1NN classifier using different distance metrics. For each dataset, we assign rank 1 to the method with the lowest error, rank 2 to the one with the second-lowest error, and so on. The average rank for each method over all datasets is reported in the last row of Table 2.

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[4] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[5] http://www.loni.ucla.edu/ADNI

TABLE 2. Comparison of average error (standard deviation) rates on UCI (%).

method	Euclidean	LMNN	ISSML	mmLMNN	PLML	GNSML	CMML	PULLMH	PULLMS
wine	9.24(3.37)	8.18(2.94)	6.42(2.66)	6.36(2.53)	6.52(2.18)	6.56(2.26)	8.92(3.07)	**3.58(1.82)**	6.22(2.67)
iris	10.32(3.80)	7.85(3.38)	5.36(3.21)	6.59(3.46)	5.90(3.31)	6.25(3.40)	7.41(2.98)	**3.51(2.02)**	4.64(2.42)
dermatology	10.02(2.26)	7.89(1.96)	7.51(1.92)	6.37(1.77)	7.17(1.93)	7.19(1.94)	9.96(3.26)	**6.28(2.02)**	7.08(1.88)
ionosphere	20.06(5.36)	19.52(5.50)	18.15(5.19)	13.75(3.49)	17.97(4.43)	17.87(4.48)	21.42(4.40)	13.60(5.65)	**12.15(3.15)**
balance	44.21(5.49)	37.86(6.03)	26.79(9.55)	32.27(7.74)	27.30(8.19)	26.01(9.93)	30.96(9.73)	27.01(9.00)	**24.26(11.97)**
heart	25.25(4.64)	24.99(4.77)	23.39(4.03)	23.65(4.51)	23.48(4.53)	23.41(4.49)	26.08(5.17)	23.99(4.39)	**22.15(3.55)**
Rank	8.83	7.33	4.00	4.33	4.50	4.17	7.67	2.50	**1.67**

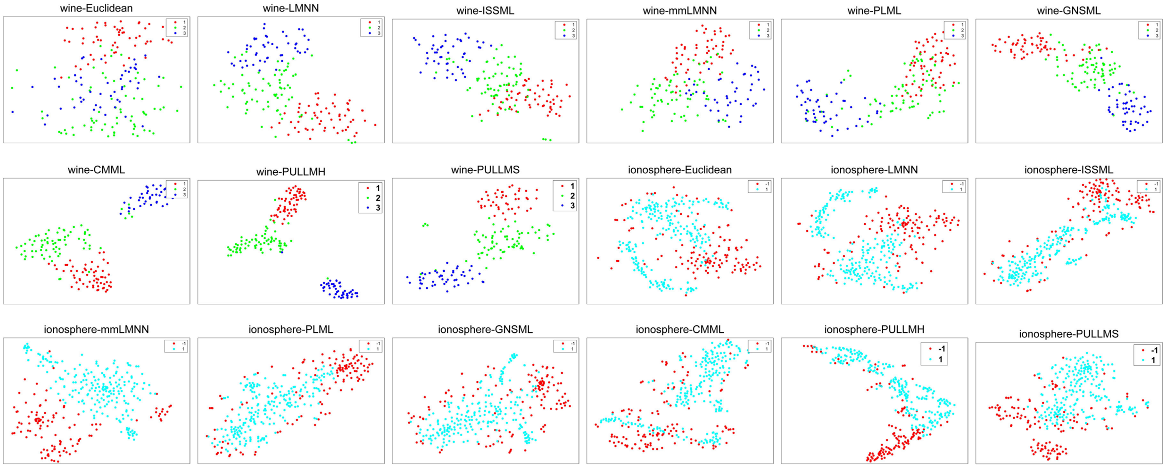


FIGURE 5. Visualization results of 'wine' and 'ionosphere'. Different colors represent different classes.

We have four observations from Table 2 and Fig. 4.

First, almost all metric learning methods improve the recognition results of the Euclidean distance (without metric learning) on all datasets.

Second, according to the average rank, PULLMS performs the best among all methods, followed by PULLMH. It is interesting that CMML performs even worse than some single metric learning methods. This is mainly due to the lack of labeled data in the training process, since CMML needs to use sufficient labeled data to implement the clustered multi-metric learning. ISSML works well overall with unlabeled data information.

Third, methods using the regularizer, i.e., ISSML, PULLMH and PULLMS, are better than LMNN which lacks the regularizer.

Fourth, multiple-metric learning methods are more suitable for nonlinear datasets, while ISSML achieves satisfactory performance on the highly linear dataset iris.

To better illustrate the classification effect of PULLMH and PULLMS, we use tSNE [37] to visualize the results on wine and ionosphere, as plotted in Fig. 5. The local multiple metrics provided by PULLMH and PULLMS with regularization show high discriminative capability, compared with those learned by other methods.

Fig. 6 shows the box plot of recognition error rate along with the iteration with 30 tests. Fig. 7 presents the variation of objective function along with the iteration. The recognition error rate and the value of objective function decrease along with the iterations in the statistical sense, and they also converge finally. On the other hand, PULLMS with ISD decreases faster and converges earlier than PULLMH with PSD.

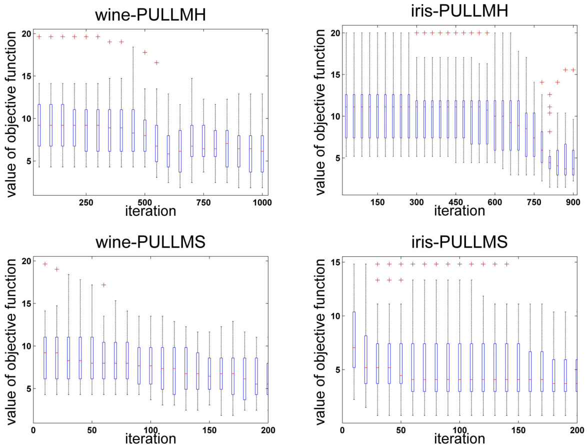


FIGURE 6. The box plot of the recognition error rate along with the iteration of the algorithms.

3) PARAMETER ANALYSIS

Moreover, we test the sensitivity of the parameter λ in (10) and v in (17) to compute σ in Gaussian kernel by fixing other settings.

Fig. 8(a) and Fig. 8(c) show the recognition error rate against the parameter λ from 0.1 to 0.9 on UCI.

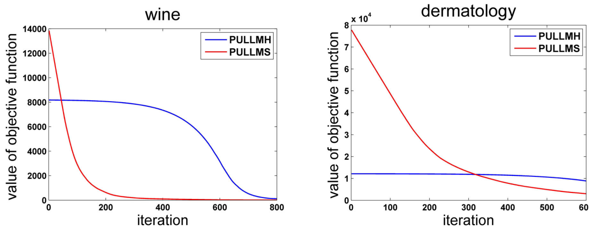
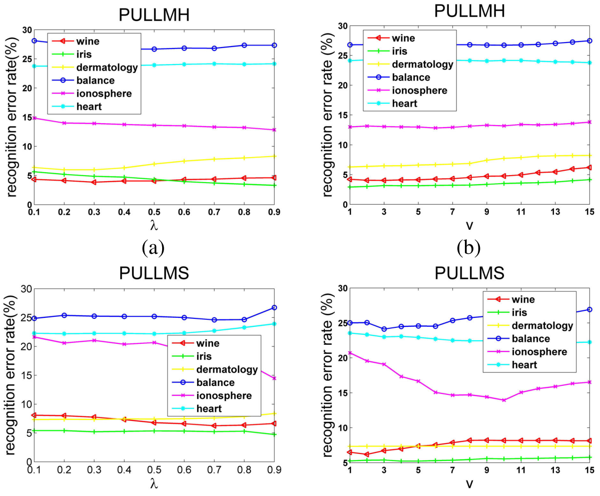


FIGURE 7. The variation of objective function along with the iteration of the algorithms.



(a) (b) (c) (d)

FIGURE 8. (a), (c) The variation of the recognition error rate along with turning λ in [0.1,0.9] and the step size 0.1 on UCI with other parameters fixed. (b), (d) The variation of the recognition error rate along with turning *v* in [1,15] and the step size 1.

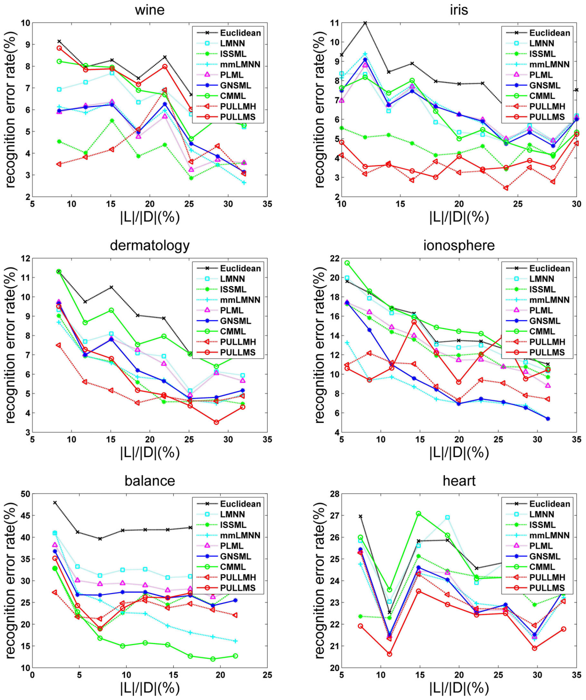


FIGURE 9. The curves of the mean recognition error rates w.r.t the $|\mathcal{L}|/|\mathcal{D}|$ on UCI datasets.

FIGURE 10. Examples of images from ORL dataset.

The recognition rates on most datasets are stable when the parameter λ changes. The ionosphere dataset is relatively sensitive to λ in PULLMS. In general, our proposed methods are not very sensitive to the parameter λ.

With regard to the parameter v used to calculate σ in similarity $S_{ij} = \exp(-d_{ij}^2/2\sigma^2)$, where $\sigma = \min D + (1/v)(\max D - \min D)$, we set different values of v for different datasets. When other parameters are fixed, the effect of v to the recognition error rate is presented in Fig. 8(b) and Fig. 8(d). Clearly, the recognitions in PULLMH are stable to v, while those in PULLMS are a little sensitive.

To further test the performance of all algorithms with different magnitudes of labeled data, we select different ratios of labeled data. Fig. 9 shows the changes of the mean recognition error rates with respect to the ratio $|\mathcal{L}|/|\mathcal{D}|$. The curves of all methods tend to decrease when the ratio $|\mathcal{L}|/|\mathcal{D}|$ increases, and our proposed methods have the lowest mean recognition error rates.

In general, our proposed methods are more accurate for classification than several conventional methods without losing computational efficiency. At the same time, they are stable to the parameters. For further testing our proposed methods, we apply them to four real datasets for classification and image retrieval below.

B. CLASSIFICATION ON ORL DATASET

The ORL facial dataset contains 400 images with 40 classes, where the variability between images of the same person is mainly due to different lighting conditions. The images are automatically centered and then converted to vectors. All images are cropped and resized to the size of 32×32. Examples of images from ORL are shown in Fig. 10.

We randomly select 30% of the data as the labeled samples $\mathcal{L}$, 50% as the unlabeled samples $\mathcal{U}$ and the rest 20% as the testing samples. Same as the UCI dataset, we compare our performance with the Euclidean, LMNN, ISSML, mmLMNN, PLML, GNSML and CMML. All the algorithm parameters are tuned for the best performance. The test process is repeated 20 times, and the average error rate is used for comparison. The results are shown in Table 3. Similar experimental setting and result illustration are also implemented on USPS and COIL-20 datasets in the following.

Table 3 verifies the efficiency of PULLMS from a numerical point of view, and also in USPS and COIL-20. PULLMH performs unsatisfactory, which may due to the heavy overfitting for large-scale problems. Meanwhile, this demonstrates that the hinge loss function is not suitable for smooth gluing metrics. We can find that LMNN is effective and stable for this kind of problems.

TABLE 3. Comparison of average error (standard deviation) rates on real datasets (%).

Dataset	USPS	ORL	COIL
Euclidean	11.35(1.79)	23.31(3.75)	4.21(1.20)
LMNN	8.68(1.63)	14.38(3.66)	2.50(1.21)
ISSML	9.25(1.81)	13.81(3.23)	2.36(0.78)
mmLMNN	7.40(1.67)	20.81(3.57)	1.64(0.54)
PLML	15.88(2.80)	17.38(3.67)	3.14(1.27)
GNSML	8.13(1.69)	17.63(3.60)	1.86(0.64)
CMML	11.10(1.96)	31.63(4.92)	4.14(1.09)
PULLMH	8.88(1.64)	20.00(3.51)	3.50(1.02)
PULLMS	**6.20(1.69)**	**9.38(2.94)**	**1.50(0.39)**

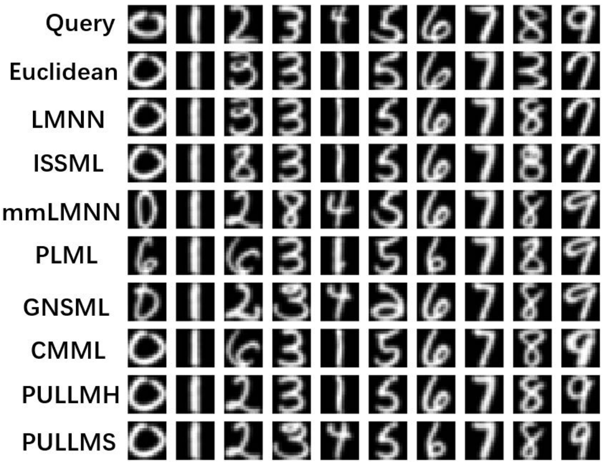


FIGURE 11. Retrieval result of 1NN classification on USPS.

C. RETRIEVAL ON USPS DATASET

USPS is a handwritten digits dataset with grayscale images of ‘0’ through ‘9’. The digits have been normalized to size 16×16 gray-level images. Thus the dimension of input space is 256. It is used to validate algorithms for image retrieval in [38]. The results are shown in Table 3 and Fig. 11.

Fig. 11 shows the nearest sample to the query image under the metrics of Euclid, LMNN, ISSML, mmLMNN, PLML, GNSML, CMML and our proposed methods PULLMH and PULLMS, respectively. The first row shows the queries. The rest rows correspond to the nearest neighbors of the queries obtained under metric learning methods. It indicates that PULLMS finds much better results than the methods using other learned metrics.

D. RETRIEVAL ON COIL-20 DATASET

COIL-20 is a 3D object dataset containing 1440 images with 20 different objects. Each object contains 72 images with size 128×128, where each image corresponds to a projection angle ranging from 0° to 355° with an interval of 5 degrees. Before training, we conduct principal component analysis (PCA) on the whole COIL-20 to reduce the dimension and relieve the computation burden, and it only contains 1439 principal components. Thus we reduce all samples to 1439 dimension with PCA and then train the metric learning model. The experiment results are shown in Table 3 and Fig. 12.

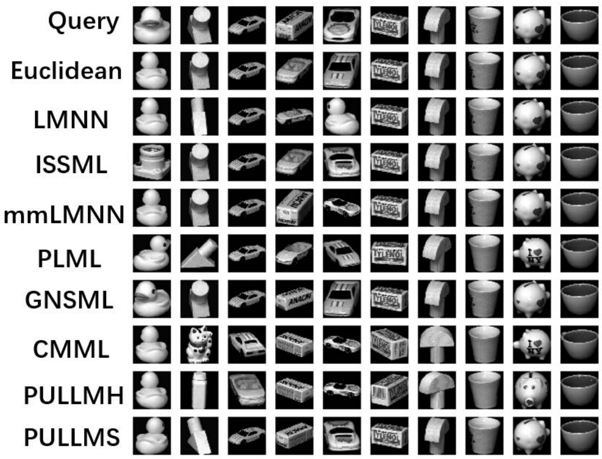


FIGURE 12. Retrieval result of 1NN classification on COIL-20.

TABLE 4. Experimental results on ADNI data.

Dataset	Err (std)(%)	AUC	F1
Euclidean	29.72(6.53)	0.7015	0.6936
LMNN	21.97(6.14)	0.7795	0.7770
ISSML	19.40(5.19)	0.8055	0.8039
mmLMNN	21.12(5.35)	0.7326	0.7196
PLML	21.41(6.18)	0.7253	0.7101
GNSML	21.53(6.40)	0.7303	0.7092
CMML	20.88(4.99)	0.7906	0.7879
PULLMH	23.33(4.89)	0.7244	0.7087
PULLMS	**16.91(4.75)**	**0.8304**	**0.8293**

Fig. 12 shows some results about retrieved samples. The first row is the retrieval samples, and the rest are the retrieved images obtained with the Euclidean distance and other metrics. It can be seen that PULLMS improves the retrieval performance effectively.

E. ANALYSIS ON ADNI DATASET

ADNI is a medical dataset consisting of 51 Alzheimer’s Disease (AD) patients and 52 normal controls (NC) with 93 features, extracting from MRI images. We select 10 AD patients and 10 NC as the labeled dataset $\mathcal{L}$ randomly and the rest as the unlabeled dataset $\mathcal{U}$, which also as the test set due to the lack of samples. Every experiment selects the training samples randomly and the average results of repeated 30 times are presented in Table 4.

To evaluate the performance of those metric learning methods, in ADNI dataset, besides the average recognition error rate, we adopt the area under ROC curve (AUC) and F1-score. We can find that PULLMS outperforms other state-of-the-art methods under different evaluation indexes.

V. CONCLUSION

In this paper, we have extended the smoothly glued local metric learning method to a nonlinear semi-supervised metric learning framework via introducing a manifold regularization to preserve the data structure. We have proposed two different nonlinear semi-supervised metric learning models with two different loss terms, showing that the smooth loss performs better than the hinge loss. Comparison with several state-of-the-art methods on standard datasets shows that our proposed method improves the accuracy of classification and the robustness.

REFERENCES

[1] B. Kulis, "Metric learning: A survey," *FNT Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.

[2] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 615–623.

[3] B. Nguyen, C. Morell, and B. De Baets, "Supervised distance metric learning through maximization of the Jeffrey divergence," *Pattern Recognit.*, vol. 64, pp. 215–225, Apr. 2017.

[4] B. Nguyen, C. Morell, and B. De Baets, "Distance metric learning for ordinal classification based on triplet constraints," *Knowl.-Based Syst.*, vol. 142, pp. 17–28, Feb. 2018.

[5] Z. Ding and Y. Fu, "Robust Multiview data analysis through collective low-rank subspace," *IEEE Trans. Neural Netw. Learn.Syst.*, vol. 29, no. 5, pp. 1986–1997, May 2018.

[6] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on grassmann manifold with application to video based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 140–149.

[7] Z. Huang, R. Wang, S. Shan, and X. Chen, "Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning," *Pattern Recognit.*, vol. 48, no. 10, pp. 3113–3124, Oct. 2015.

[8] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 363–376, Jan. 2013.

[9] Y. Gao, R. Ji, P. Cui, Q. Dai, and G. Hua, "Hyperspectral image classification through bilayer graph-based learning," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2769–2778, Jul. 2014.

[10] S. Du, Y. Guo, G. Sanroma, D. Ni, G. Wu, and D. Shen, "Building dynamic population graph for accurate correspondence detection," *Med. Image Anal.*, vol. 26, no. 1, pp. 256–267, Dec. 2015.

[11] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 174–184, Jan. 2010.

[12] M. R. Min, H. Guo, and D. Song, "Exemplar-centered supervised shallow parametric data embedding," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2479–2485.

[13] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. NIPS*, 2002, pp. 521–528.

[14] J. Goldberger, G. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. NIPS*, 2005, pp. 513–520.

[15] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. NIPS*, 2005, pp. 1473–1480.

[16] G. Kunapuli and J. Shavlik, "Mirror descent for metric learning: A unified approach," in *Machine Learning and Knowledge Discovery in Databases*, 2012, pp. 859–874.

[17] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 209–216.

[18] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and Kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 519–547, Jan. 2012.

[19] X. Li, Y. Bai, Y. Peng, S. Du, and S. Ying, "Nonlinear semi-supervised metric learning via multiple kernels and local topology," *Int. J. Neural Syst.*, vol. 28, no. 02, Mar. 2018, Art. no. 1750040.

[20] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou, "What Makes objects similar: A unified multi-metric learning approach," in *Proc. NIPS*, 2016, pp. 1235–1243.

[21] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou, "What makes objects similar: A Unified Multi-Metric Learning Approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1257–1270, May 2019.

[22] D.-C. Zhan, M. Li, Y.-F. Li, and Z.-H. Zhou, "Learning instance specific distances using metric propagation," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1225–1232.

[23] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, no. 2, pp. 207–244, 2009.

[24] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Proc. NIPS*, 2012, pp. 1601–1609.

[25] Y. Peng, L. Hu, S. Ying, and C. Shen, "Global nonlinear metric learning by gluing local linear metrics," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 423–431.

[26] B. Nguyen, F. J. Ferri, C. Morell, and B. De Baets, "An efficient method for clustered multi-metric learning," *Inf. Sci.*, vol. 471, pp. 149–163, Jan. 2019.

[27] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 81–88.

[28] S. C. H. Hoi, W. Liu, and S. F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 6, no. 3, p. 18, 2010.

[29] S. Li and Y. Fu, "Low-rank coding with B-matching constraint for semi-supervised classification," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1472–1478.

[30] W. Liu, S. Ma, D. Tao, J. Liu, and P. Liu, "Semi-supervised sparse metric learning using alternating linearization optimization," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 1139–1148.

[31] M. Baghshah and S. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1217–1222.

[32] G. Zhong, K. Huang, and C.-L. Liu, "Low rank metric learning with manifold regularization," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 1266–1271.

[33] Q. Wang, P. C. Yuen, and G. Feng, "Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions," *Pattern Recognit.*, vol. 46, no. 9, pp. 2576–2587, Sep. 2013.

[34] G. de Rham, *Differentiable Manifolds*. Berlin, Germany: Springer-Verlag, 1984.

[35] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao, "Manifold preserving: An intrinsic approach for semisupervised distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2731–2742, Jul. 2018.

[36] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, 2007.

[37] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[38] K. Zhao, W. Liu, and J. Liu, "Optimal semi-supervised metric learning for image retrieval," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 893–896.

YAXIN PENG (Member, IEEE) received the B.Sc. degree in mathematics from Anhui Normal University, Wuhu, China, in 2002, the M.Sc. degree in mathematics from East China Normal University (ECNU), Shanghai, China, in 2005, and the Ph.D. degree in mathematics from the École Normale Supérieure de Lyon, Lyon, France, and ECNU, in 2008. She is currently an Associate Professor with the Department of Mathematics, School of Science, Shanghai University, Shanghai, China. Her research interests include geometric variation, metric learning, point cloud, and image processing.

NIJING ZHANG is currently pursuing the M.S. degree with the Department of Mathematics, School of Science, Shanghai University. Her research interests include machine learning, metric learning, transfer learning, and domain adaptation.

YING LI received the Ph.D. degree in computational mathematics from Xi'an Jiaotong University, in 2014. She is currently a Lecturer with the School of Computer Engineering and Science, Shanghai University. Her main research interests include machine learning and numerical simulation.

SHIHUI YING (Member, IEEE) received the B.Eng. degree in mechanical engineering and the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2008, respectively. He held a postdoctoral position at the Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, from 2012 to 2013. He is currently a Professor with the Department of Mathematics, School of Science, Shanghai University, Shanghai, China. His current research interest includes geometric theory and methods for medical image processing and machine learning.