Machine Vision and Applications (2024) 35:105
https://doi.org/10.1007/s00138-024-01581-9

RESEARCH

Semi-supervised metric learning incorporating weighted triplet constraint and Riemannian manifold optimization for classification

Yizhe Xia[1] · **Hongjuan Zhang**[2]

Received: 22 February 2024 / Revised: 17 May 2024 / Accepted: 28 June 2024 / Published online: 26 July 2024

Abstract
Metric learning focuses on finding similarities between data and aims to enlarge the distance between the samples with different labels. This work proposes a semi-supervised metric learning method based on the point-to-class structure of the labeled data, which is computationally less expensive, especially than using point-to-point structure. Specifically, the point-to-class structure is formulated into a new triplet constraint, which could narrow the distance of inner-class data and enlarge the distance of inter-class data simultaneously. Moreover, for measuring dissimilarity between different classes, weights are introduced into the triplet constraint and forms the weighted triplet constraint. Then, two kinds of regularizers such as spatial regularizer are rationally incorporated respectively in this model to mitigate the overfitting phenomenon and preserve the topological structure of the data. Furthermore, Riemannian gradient descent algorithm is adopted to solve the proposed model, since it can fully exploit the geometric structure of Riemannian manifolds and the proposed model can be regarded as a generalization of the unconstrained optimization problem in Euclidean space on Riemannian manifold. By introducing such solution strategy, the variables are constrained to a specific Riemannian manifold in each step of the iterative solution process, thereby enabling efficient and accurate model resolution. Finally, we conduct classification experiments on various data sets and compare the classification performance to state-of-the-art methods. The experimental results demonstrate that our proposed method has better performance in classification, especially for hyperspectral image data.

Keywords Metric learning · Manifold optimization · Semi-supervised learning · Classification

1 Introduction

Metric learning, one of the core problems in pattern recognition, aims to measure the similarity between data when the classification algorithms such as k-nearest neighbors (kNN) could not find its meaningful results. Therefore, many metric learning methods have been proposed, for example the Mahalanobis metric learning, whose goal is to narrow the distance between the samples with the same labels, and to enlarge the distance between the samples with different labels. In fact, for some traditional classification methods, such as Support Vector Machine (SVM) [1], and k-nearest neighbors (kNN) [2], they also rely on a better metric to measure the dissimilarities between data, and their results will be greatly improved by combining metric learning. On this basis, more and more improved metric learning based models and algorithms have been proposed in recent years. Generally speaking, they can be divided into three categories by training samples unsupervised metric learning [3–10], supervised metric learning [11–18], and semi-supervised metric learning [19–22].

Unsupervised metric learning methods mainly deal with the data without label information, which just aim at finding a latent data manifold embedded in the higher dimensional space, such that local or global structure between data could be preserved. For example, Principal Component Analysis (PCA) [3], as a well-known classical method, is usually regarded as a dimensional reduction method, but in essence, it can be viewed as an unsupervised metric learning method. Similarly, Multidimensional scaling (MDS) [4] and Nonnegative Matrix Factorization (NMF) [5] can both be regarded as the unsupervised metric learning methods. However, these

✉ Hongjuan Zhang
zhanghongjuan@shu.edu.cn

Yizhe Xia
xia_yizhe@126.com

[1] Department of Mathematics, Shanghai University, Shanghai 200444, People's Republic of China

[2] Newtouch Center for Mathematics, Shanghai University, Shanghai 200444, People's Republic of China

above methods do not work when dealing with nonlinear data. Therefore, the amount of nonlinear methods is proposed. For example, Laplacian Eigenamp (LE) [7] is a nonlinear method by the eigenfunctions of the graphic LaplaceCBeltrami operator and the Hessian operator, respectively to preserve the local neighbor of every single data. Subsequently, on this basis, Locality Preserving Projections (LPP) [8], a linear approximation version of LE was proposed. In contrast, the required nonlinear map in LE is replaced by a linear map, which simplifies the model but can also have a good performance. Another classical nonlinear method is locally linear embedding (LLE) [9], which preserves the local order relation of data in both the embedding space and the intrinsic space. Moreover, LLE also has its linear approximation version called neighborhood preserving embedding (NPE) [10]. Similarly, linear projection obtained by NPE replaces the desired map in LLE, whose performance is still comparable to that of the former LLE. However, these unsupervised methods do not fully take into account the label information of the data that have a very positive impact in classification. Therefore, label-information-based metric learning method has been proposed in recent years.

The main idea of supervised metric learning is to use the label information to shorten the distance between similar samples while enlarge the distance between samples in different classes. Xing et al. [11] first proposed the Mahalanobis metric learning model by introducing the "similar pairs" and "dissimilar pairs" constraints, which can separate similar and dissimilar data clearly. Based on it, the amount of Mahalanobis metric learning methods was investigated successively. For example, Weinberger et al. [13] presented the large margin nearest neighbors (LMNN), in which a hinge loss function is constructed with the triplet constraints, such that a large margin of distance can be held between the inter-class data points. From the information theory, Davis et al. [14] developed information-theoretic metric learning (ITML), involving a natural entropy-based objective function under the pairwise distance constraints, which can be solved as a low-rank kernel learning problem. In addition, Zuo et al. [18] developed Positive-semidefinite constrained metric learning (PCML) and Nonnegative-coefficient constrained metric learning (NCML), which are both formulated as the kernel classification problem with the positive semidefinite constraint, such that they can be solved by iterated training of support vector machines (SVMs). Based on the Riemannian geometry framework, an early approach put forward geometric mean metric learning (GMML) [17], which efficiently solves the metric learning problem through the Riemannian geometry of positive definite matrices. And Li et al. introduced a Kullback–Leibler Divergence (KLD) based metric learning model called Kullback–Leibler Divergence Metric Learning [23]. The model is based on the KLD which is extended by the introduction of a linear mapping and can well express the data distribution similarity. Then, an intrinsic steepest descent method is introduced to solve such optimization problem. Another Riemannian based method called graph embedding multi-kernel metric learning (GEMKML) algorithm was proposed [24], in which the Grassmannian manifold-valued feature representations are produced as the feature in a metric learning process. Furthermore, the Grassmannian conjugate gradient method is used during the optimization process. However, the above-mentioned supervised metric learning method is too idealistic in the label information of data, since in practical applications, it may happen that only partial information is available. Therefore, semi-supervised metric learning methods only using partial labeled information have been developed subsequently.

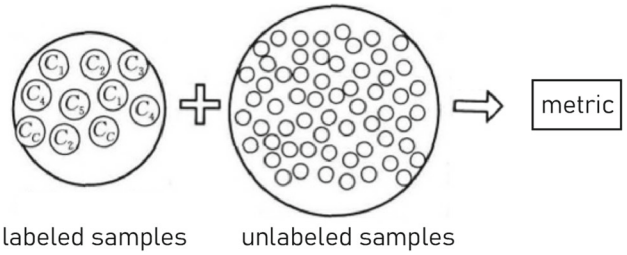


Fig. 1 Illustration of semi-supervised metric learning

Semi-supervised learning integrates the advantages of supervised and unsupervised learning, aiming to overcome the limitations inherent in both approaches. On the one hand, for unsupervised learning, it lacks real sample information to support the clustering results, thereby compromising its accuracy and stability. On the other hand, supervised learning requires a large number of training samples, while labeling samples requires a significant amount of manpower and time resources, which often leads to the lack of training samples in practical applications. Consequently, semi-supervised learning has attracted increasing attention in recent years and demonstrated its remarkable application value and potential in various fields such as medical/healthcare [25], transportation [26], manufacturing [27], etc. The goal of semi-supervised metric learning is to utilize unlabeled data with the help of labeled data to make supervised metric learning to learn an appropriate metric such that it agrees with pairwise or triplet constraint. The main concept of semi-supervised metric learning is shown in Fig. 1. For example, Hoi et al. [19] first introduced the semi-supervised learning into metric learning and put forward the Laplacian Regularized Metric Learning (LRML), which combines both labeled and unlabeled data information through an effective graph regularization framework. Another semi-supervised metric learning was developed called Regularized semi-supervised metric learning (RSSML) [20], in which the local topology and triplet constraints are considered in the model combin-

ing with the regularization by the unlabeled samples, which satisfies three basic assumptions for semi-supervised learning namely, smoothness, cluster and manifold assumptions. Based on these three assumptions, semi-supervised metric learning for stratified spaces (S2MLS2) [28] was developed, in which unsuitable local constraints is eliminated for adapting to the multi-manifolds smoothness assumption, and some non-local constraints is introduced to detect the shared structures at different positions for lack of supervised information. Moreover, Li et al. [29] put forward a semi-supervised metric learning method called regularized large margin distance metric learning (RLMM), which considers the triplet constraint and pairwise constraint at the same time in the metric learning model by using two hinge loss function. Wang et al. [30] proposed a coefficient-based semi-supervised metric learning model, in which a linear combination of a set of base vectors is learned as a new metric instead of the traditional metric matrix, combining with the pairwise constraint and sparse regularization. In addition, Ying et al. [22] proposed a metric learning algorithm based on Riemannian manifold optimization [31], in which a semi-supervised metric learning model is specially formulated by considering the metric information of inner classes and interclasses, and an adaptive parameter is designed into the triplet constraint to balance the inner metrics and intermetrics by using data point-to-point structure. Furthermore, Semi-supervised Subspace Metric Learning [32] is presented, in which a low-dimensional subspace is learned by an inverse problem on Grassmannian manifold, then some local positive definite metrics are learned on this subspace.

Different from above works, we will design a semi-supervised metric learning method based on the point-to-class structure of the labeled data. Intuitively, each point should be closer to its own class center than to any other class under a better metric. By this assumption, the point-to-class structure is formulated into a new triplet constraint, which aims to narrow the distance of inner-class data and meanwhile to enlarge the distance of inter-class data. In fact, in traditional point-to-point triplet constraint, the number of the triples between all the labeled samples is quite large. Usually, an attempt is to use the distances from one point to its neighbors rather than to all other points [22], however, which will cause the information to be lost in practice. Therefore, it is highly desired to explore the effective point-to-class based metric learning method in many fields such as computer vision tasks, which will reduce computational cost accordingly. Meanwhile, different weights are introduced into the triplet constraint to measure the inter-classes dissimilarity. In fact, its smaller value corresponds to the larger distance of data in different classes, while its larger value indicates that the points in different classes are very close, which will be prone to be misclassified without metric learning generally. This work aims to enlarge the distances between the points and their different classes, especially for the points corresponding to the large weight values. Note that, the point-to-class distance is defined as the distance between point to the class-center. Moreover, two kinds of regularization terms are incorporated respectively in this work to mitigate the overfitting phenomenon, which occasionally occurs in the LMNN due to its absence of regularization, especially in high dimension. One is about the spatial regularizer [33] for hyperspectral image data in order to capture its spatial information. For other data sets, we use a more common regularizer which aims to preserve the topological structure of the data [20]. In addition, we adopted Riemannian gradient descent algorithm to solve the proposed model, which makes full use of the geometric structure of Riemannian manifolds and can be regarded as a generalization of the unconstrained optimization problem in Euclidean space on Riemannian manifold. In order to verify its precision, we test its classification accuracy on various data sets, including four UCI data sets, USPS handwriting data set, and two face data sets yaleB and ORL. In addition, hyperspectral data classification problems is attracting wide public attention in recent years [34–37]. Therefore, two hyperspectral image data sets(Indian pines and KSC) are also used to test our method.

The rest of this work is organized as follows. Section 2 briefly reviews some related works, including metric learning and Riemannian manifold optimization. In Sect. 3, the improved metric learning with weighted triplet constraint is given and its corresponding Riemannian manifold algorithm is demonstrated. All results of the numerical experiments will be shown to prove the precision in Sect. 4. Section 5 is the conclusion of this work.

2 Related works

2.1 Metric learning

For some classification or clustering problems, the traditional Euclidean distance metric sometimes can not well capture the distance or similarity between two samples. The concept of metric learning was first introduced by Xing et al. [11] in 2002, aiming to accurately describe the relationship between training sample points based on distance in classification problems. By employing metric learning, more suitable metrics can be discovered to better capture the intrinsic characteristics of data distribution and consequently enhance the accuracy of classification tasks.

The goal of metric learning is to learn an effective distance metric, such that samples from the same class are brought closer together while samples from different classes are maximally separated under the new metric. Let $A \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix, and we define A as our distance

metric. So, under the metric A, for any two points x and y, the distance between them can be expressed by

$$d_A^2(x, y) = \|x - y\|_A^2 = (x - y)^T A(x - y) \tag{1}$$

Given data set $X = \{x_1, x_2, \ldots, x_n\}$, with $x_i \in \mathbb{R}^n$. Let S and D represent the sets of similar and dissimilar pairs, respectively, which are used to characterize the relationships of similarity or dissimilarity among sample points, and can be defined by

$$S = \{(x_i, x_j) | \ x_i \ and \ x_j \ is \ similar\} \tag{2}$$

$$D = \{(x_i, x_j) | \ x_i \ and \ x_j \ is \ dissimilar\} \tag{3}$$

The pairwise constraints in metric learning models are commonly denoted as S and D. Therefore, based on the aforementioned pairwise constraints, our metric learning model can be formulated as follows

$$\begin{aligned} &\min_A g(A) = \sum_{(x_i,x_j)\in S} \|x_i - x_j\|_A^2 \\ &s.t. f(A) = \sum_{(x_i,x_j)\in D} \|x_i - x_j\|_A^2 \geqslant 1 \\ &\quad A \succeq 0. \end{aligned} \tag{4}$$

In this metric learning model, $g(A)$ describes the distance between similar samples under pairwise constraints, while $f(A)$ describes the distance between dissimilar samples. The purpose of this model is to ensure that the distance between similar samples is minimized, while ensuring that the distance between dissimilar samples falls within a specific range. According to [11], the optimization problem 4 can be solved by a projected gradient method.

2.2 Riemannian manifold optimization

Riemannian manifold optimization is a special kind of optimization problem, which has a broad application in machine learning [38–41].

Refer to [42], the Riemannian optimization problem can be written as follows

$$\min_{x\in\mathcal{M}} f(x) \tag{5}$$

where $\mathcal{M}$ is a Riemannian manifold, and $f : \mathcal{M} \to \mathbb{R}$ is a smooth cost function or objective function. When $\mathcal{M} = \mathbb{R}^n$ for some n, the optimization problem reduces to an unconstrained optimization problem in Euclidean space. The standard gradient descent algorithm in Euclidean space $\mathbb{R}^n$ iterates

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{6}$$

where $\alpha_k > 0$ is called the step-sizes or learning rate, $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is the gradient of f, and can be given by

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right)^T \tag{7}$$

However, when the $\mathcal{M}$ is not a linear space but a general Riemannian manifold, the optimization problem may become more difficult to solve by using the gradient descent algorithm. Therefore, we need to define the Riemannian gradient and the way of moving on a Riemannian manifold.

For a Riemannian manifold $\mathcal{M}$, the inner product $< \cdot, \cdot >_x$ is equipped in the tangent space denoted by $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$. Let f be a smooth function of the Riemannian manifold $\mathcal{M}$, the differential of a smooth function on a Riemannian manifold can be defined by

$$Df(x)[v] = \frac{d}{dt} f(\gamma(t))\Big|_{t=0} \tag{8}$$

where $\gamma(t)$ is a smooth curve on $\mathcal{M}$ passing through the point x, and satisfies $\gamma'(0) = v$, $v \in T_x\mathcal{M}$. The Riemannian gradient $\mathrm{grad} f(x)$ is the tangent vector in $T_x\mathcal{M}$ satisfies

$$Df(x)[v] =< v, \mathrm{grad} f(x) >_x, \forall v \in T_x\mathcal{M}. \tag{9}$$

According to [42], when $\mathcal{M}$ is a Riemannian submanifold of Euclidean space $\mathbb{R}^n$ equipped with the metric $< \cdot, \cdot >$, and $f : \mathcal{M} \to \mathbb{R}$ is a smooth function on $\mathcal{M}$, there must be $\bar{f}(x) : \mathbb{R}^n \to \mathbb{R}$, satisfing $\bar{f}(x) = f(x), x \in \mathcal{M}$. Then the Riemannian gradient can be calculate by

$$\mathrm{grad} f(x) = Proj_x(\nabla \bar{f}(x)) \tag{10}$$

where $Proj_x : \mathbb{R}^n \to T_x\mathcal{M}$ the projector from $\mathbb{R}^n$ to $T_x\mathcal{M}$, orthogonal with respect to $< \cdot, \cdot >$.

In addition, to update the sequence like (6) on Riemannian manifold, a retraction plays an inportant role in the iteration process. A retraction R at $x \in \mathcal{M}$ denoted by R_x is a smooth map from $T_x\mathcal{M}$ to $\mathcal{M}$,which satisfies

1. $R_x(0_x) = x$ where 0_x is the zero element in $T_x\mathcal{M}$
2. $DR_x(0_x) = id_{T_x\mathcal{M}}$ where $DR_x(0_x)$ is the differential map of R_x at 0_x, $id_{T_x\mathcal{M}}$ is the identity map in $T_x\mathcal{M}$.

By introducing the retraction, we can restrict the variables to the same Riemannian manifold during each iteration step. Therefore, the Riemannian gradient descent algorithm can be written as follows.

Given $x_0 \in \mathcal{M}$ and retraction R,x_k can be iterated through

$$x_{k+1} = R_{x_k}(-\alpha_k \mathrm{grad} f(x_k)) \tag{11}$$

where α_k is the step-size in kth iteration.

3 The proposed method

3.1 Metric learning model

Given data set $X = \{x_1, x_2, \ldots, x_n\}$, with $x_i \in \mathbb{R}^n$. For semi-supervised learning, the data set is divided into two parts, X_L and X_U. The samples in $X_L = \{x_1, x_2, \ldots, x_l\}$ are all labeled, with the label set $Y = \{y_1, y_2, \ldots, y_l\}$. The other part X_U contains the rest of the samples without labels. Our goal is to find a best metric mentioned in (1) that makes samples can be classified accurately.

Generally speaking, the goal of metric learning is to narrow the distance of inner-class data and meanwhile to enlarge the distance of inter-class data. To better illustrate our model, we rewrite the notation of the samples. Suppose that the samples in X_L can be divided into c subsets with the same labels denoted by $X_1, X_2, \ldots, X_c$, where $X_i = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}\}$, n_i is the sample numbers of X_i. For each class X_i, we denote $\mu_i = \sum_{k=1}^{n_i} \frac{1}{n_i} x_k^{(i)}$ as the center of all samples in X_i. Like (1), we define the distance between a single samples x and the class set X_i by

$$d_A^2(x, X_i) = (x - \mu_i)^T A (x - \mu_i) \tag{12}$$

Inspired by the triplet constraint in [22], we define the point-to-class triplet constraint denoted by

$$\mathcal{T} = \left\{ \left(x_i^{(k)}, X_k, X_l \right) : d_A\left(x_i^{(k)}, X_k \right) < d_A\left(x_i^{(k)}, X_l \right) \right\} \tag{13}$$

Then our metric learning can be modeled as follows

$$\begin{aligned} &\min_A \sum_{\mathcal{T}} d_A\left(x_i^{(k)}, X_k \right) - d_A\left(x_i^{(k)}, X_l \right) \\ &s.t. A \succ 0 \end{aligned} \tag{14}$$

Then for each labeled sample, by the definition in (12) we rewrite the objective function as follows

$$\sum_{l \neq k} \left(d_A\left(x_i^{(k)}, \mu_k \right) - d_A\left(x_i^{(k)}, \mu_l \right) \right) \tag{15}$$

In (15), since the distance between the same class does not need to be calculated, therefore, some terms are deleted, and (15) turns into

$$d_A\left(x_i^{(k)}, \mu_k \right) - \sum_{l \neq k} d_A\left(x_i^{(k)}, \mu_l \right) \tag{16}$$

In order to balance the terms that have been deleted, and to make our model more accurately to capture those distance that attempts to become larger during metric learning, weight is added into the second terms, like many local-preserved methods [16],

$$d_A\left(x_i^{(k)}, \mu_k \right) - \sum_{l \neq k} w_{i,k,l} d_A\left(x_i^{(k)}, \mu_l \right) \tag{17}$$

where the specific weights are assigned as follows

$$w_{i,j,k} = \begin{cases} \exp\left(\frac{-\|x_i^{(j)} - \mu_k\|^2}{\sigma_{i,j}} \right) & j \neq k \\ 0 & j = k \end{cases} \tag{18}$$

In order to merge the two terms in Eq. (17), we redefine the weights as follows

$$W_{i,j,k} = \begin{cases} -\exp\left(\frac{-\|x_i^{(j)} - \mu_k\|^2}{\sigma_{i,j}} \right) & j \neq k \\ 1 & j = k \end{cases} \tag{19}$$

Then

$$\sum_{l=1}^{c} W_{i,k,l} d_A(x_i^{(k)}, \mu_l) \tag{20}$$

Summing over each $x_i^{(k)}$, we have

$$\sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} d_A\left(x_i^{(k)}, \mu_l \right) \tag{21}$$

To sum up, the proposed Riemannian-based graph-regularized metric learning (RGML) with weighted triplet constraint model finally can be expressed as the following

$$\begin{aligned} &\min_A \sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} d_A\left(x_i^{(k)}, \mu_l \right) \\ &s.t. A \succ 0 \end{aligned} \tag{22}$$

Similar with [22], in order to prevent overfitting, a graph regularizer is added to our model, which could preserve three basic assumptions of semi-supervised learning and take full advantage of the topology of the data. The semi-supervised metric learning model can be written as follows

$$\begin{aligned} &\min_A \sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} d_A\left(x_i^{(k)}, \mu_l \right) + \lambda Reg(A) \\ &s.t. A \succ 0 \end{aligned} \tag{23}$$

where the $Reg(A)$ is the graph regularizer, and λ is a parameter which controls the balance between the two terms.

Since the regularization term uses the unlabeled data, we do not consider the distance between points and classes in

this term. Note that for different types of data, we will choose different graph regularization terms according to the characteristics of different types of data. The first regularization term is

$$\sum_{i=1}^{n} \beta_i \sum_{j \in \mathcal{N}(i)} S_{ij} D_{ij}^2 \tag{24}$$

where $\beta_i = f(p(x_i)) \in \mathbb{R}^+$,and $p(x_i)$ is the density of x_i, $\mathcal{N}(i)$ is the nearest neighbor of x_i, S_{ij} is the similarity between x_i and x_j, which we want to preserve in the new metric. D_{ij} is the distance between x_i and x_j under the metric A formulated by (39). For the sake of calculation, we define $\eta_{i,j}$ as follows

$$\eta_{i,j} = \begin{cases} 1 & j \in \mathcal{N}(i) \\ 0 & others \end{cases} \tag{25}$$

Then, (24) turns into

$$\sum_{i=1}^{n} \beta_i \sum_{j=1}^{n} \eta_{ij} S_{ij} D_{ij}^2 = \sum_{i,j=1}^{n} W_{i,j}^{(s)} D_{ij}^2 \tag{26}$$

where $W_{i,j}^{(s)} = \beta_i \eta_{ij} S_{ij}$.
In addition, for the hyperspectral image, spatial regularizer [33] might be a better choice. To make it suit our metric learning model, the spatial regularizer can be reformed as

$$\sum_{i,j=1}^{n} W_{i,j}^{(sp)} D_{ij}^2 \tag{27}$$

where $W_{i,j}^{(sp)}$ is the spatial weight, which reflects the spatial similarity, and can be expressed in the following

$$W_{i,j}^{(sp)} = \begin{cases} \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma}\right) & x_i \text{ and } x_j \text{ are spatial neighbors} \\ 0 & others \end{cases} \tag{28}$$

Finally, two types of graph regularizer could be reduced to

$$\sum_{i,j=1}^{n} W_{i,j} D_{ij}^2 \tag{29}$$

Note that different $W_{i,j}$ determine which regularizer is used in our method.

Therefore, the distance metric model eventually becomes

$$\begin{aligned} &\min_A \sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} d_A\left(x_i^{(k)}, \mu_l\right) + \lambda \sum_{i,j=1}^{n} W_{i,j} D_{ij}^2 \\ &s.t. A \succ 0 \end{aligned} \tag{30}$$

To solve the it, we first simplify the loss term in (30) by

$$\begin{aligned} &\sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} \left(x_i^{(k)} - \mu_l\right)^T A \left(x_i^{(k)} - \mu_l\right) \\ &= Tr\left(\sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} \left(x_i^{(k)} - \mu_l\right) \left(x_i^{(k)} - \mu_l\right)^T A\right) \end{aligned} \tag{31}$$

Let

$$M = \sum_{k=1}^{c} \sum_{i=1}^{n_k} \sum_{l=1}^{c} W_{i,k,l} \left(x_i^{(k)} - \mu_l\right) \left(x_i^{(k)} - \mu_l\right)^T \tag{32}$$

Then (31) turns into

$$Tr(MA) \tag{33}$$

For regularizer in (30), we have

$$\begin{aligned} \sum_{i,j=1}^{n} W_{i,j} D_{ij}^2 &= \sum_{i,j=1}^{n} W_{i,j} \left(x_i - x_j\right)^T A \left(x_i - x_j\right) \\ &= Tr\left(\sum_{i,j=1}^{n} W_{i,j} \left(x_i - x_j\right) \left(x_i - x_j\right)^T A\right) \\ &= Tr\left(XLX^T A\right) \end{aligned} \tag{34}$$

where $L = D - W$ is the Laplace matrix, $W = (W_{i,j})_{n \times n}$ is the weight matrix and D is a diagonal matrix defined by $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$. Let

$$R = XLX^T \tag{35}$$

Then the regularizer turns into

$$Tr(RA) \tag{36}$$

Therefore, (30) can be rewritten as

$$\begin{aligned} &\min_A f(A) = Tr\left(MA\right) + \lambda Tr\left(RA\right) \\ &s.t. A \succ 0 \end{aligned} \tag{37}$$

3.2 Riemannian gradient descent algorithm

In our semi-metric learning model, the objective function is a smooth function defined on the Symmetric Positive Definite (SPD) manifold. According to the theory of Riemannian manifold optimization, similar to many methods based on Riemannian optimization [22, 43–47], the problem can be solved using the Riemannian Gradient Descent (RGD) method [42]. The solution strategy of Riemannian

manifold optimization is introduced for optimization problems with manifold constraints. In each iteration step, the variables are constrained to the same Riemannian manifold to achieve efficient and accurate model solutions.

Recall the following optimization problem

$$\min_{x\in\mathcal{M}} f(x) \tag{38}$$

where $\mathcal{M}$ is a Riemannian manifold, and f is a smooth function. Given $x_0 \in \mathcal{M}$, x_k can be iterated by RGD through

$$x_{k+1} = R_{x_k}(-\alpha_k \operatorname{grad} f(x_k)) \tag{39}$$

For the proposed model, the SPD manifold is the set of all n-order symmetric positive definite matrices denoted by S_n^{++}, which can be defined as follows

$$S_n^{++} = \left\{ S \in \mathbb{R}^{n\times n} \mid S^T = S, S \succ 0 \right\}. \tag{40}$$

Equipped with the affine-invariance metric mentioned in [48], the S_n^{++} becomes an $n(n+1)/2$ dimension Riemannian manifold. Refer to [22], the Riemannian gradient of the smooth function $f(A)$ in (37) can be calculate by

$$\operatorname{grad} f(A) = sym\left(\nabla f(A)\right) = sym\left(M^T + \lambda R^T\right) \tag{41}$$

where $sym(X) = \frac{X^T+X}{2}$ is the symmetrization operator, and ∇f is the Euclidean gradient of f, since f can also be regarded as a smooth function in Euclidean space $\mathbb{R}^{n\times n}$.

Moreover, the exponential map $\exp_S(L)$ is defined by

$$\exp_S(L) = S^{1/2} \exp\left(S^{-1/2} L S^{-1/2}\right) S^{1/2} \tag{42}$$

where $S \in S_n^{++}$, $L \in T_S S_n^{++}$ is a tangent vector at S, and $\exp(X) = \sum_{n=0}^{\infty} \frac{X^n}{n!}$ is the matrix exponential function. Obviously, $\exp_S(L)$ satisfies the condition in 2.2, which can be used as a retraction in our algorithm.

Therefore, the iterative scheme for Riemannian gradient descent is given by

$$A_{k+1} = A_k^{1/2} \exp\left(-\alpha A_k^{-1/2} \operatorname{grad} f(A_k) A_k^{-1/2}\right) A_k^{1/2} \tag{43}$$

To sum up, the Riemannian gradient descent algorithm for the RGML model has been shown in Algorithm 1.

Algorithm 1 Riemannian Gradient Descent Algorithm for Metric Learning

Input: Sample set X (including X_L and X_U); label set Y;
Output: Metric A
1: Initializing A_0, α (step-size), m (maximum number of iterations)
2: For $k = 1, 2, \ldots, m$ do
3: Computing the gradient $\operatorname{grad} f(A_k)$ by (41)
4: Setting $A_{k+1} = A_k^{1/2} \exp(-\alpha A_k^{-1/2} \operatorname{grad} f(A_k) A_k^{-1/2}) A_k^{1/2}$
5: End for

4 Experimental results

In order to verify the performance of the proposed metric learning method, we will conduct some classification experiments on several data sets, including four UCI data sets (wine, iris, dermatology and balance), USPS digit image data set, two face data sets (YaleB and ORL) and two hyperspectral image (HSI) data sets (Indian Pines and KSC). Then, we will compare our method with other algorithms, for example, large margin nearest neighbors (LMNN) [13], information-theoretic metric learning(ITML) [14], Laplacian Regularized Metric Learning (LRML) [19], geometric mean metric learning (GMML) [17], positive-semidefinite constrained metric learning [18], and kNN without any metric learning method namely under the Euclidean metric.

4.1 UCI data sets

The UCI machine learning repository [49] is a collection of databases, used by the machine learning community for the empirical analysis of machine learning algorithms. In the following experiments, four data sets are selected and their details are shown in Table 1.

Note that Each dataset is split into two parts, labeled data set X_L and unlabeled data set X_L. $\mid X_L \mid$ and $\mid X_U \mid$ are the number of the labeled samples and unlabeled samples. Then $\frac{|X_L|}{|X_U|}$ is the labeled sample ratio for the semi-supervised metric learning. Due to the small amount of data, a portion of the unlabeled data would be also used for testing. All the data used in the experiment are randomly selected. Firstly, we select part of the data as labeled data, and then take the rest of the data as unlabeled data, and use part of the unlabeled data for testing. For each UCI dataset, the test is repeated 30 times. The parameters of all algorithms are selected carefully to make sure they're on their best behavior, and all parameters in our method are as follows. $\sigma_{i,j} \in W_{i,j,k}$ in (18) can be calculated by $\sigma_{i,j} = \|x_i^{(j)} - x\|$, where x is the kth nearest point in X_L. The value of λ will be varied according to the data set, and we will determine its value experimentally. The regularizer used for UCI data is the similarity regularizer in (24). β_i can be calculated by $\beta_i = f(p(x_i))$, where f a linear map, and $p(x_i)$ can be given by $p(x_i) = \frac{1}{|\mathcal{N}(i)h^n|} \sum_{j\in\mathcal{N}(i)} K_h\left(\frac{x_i - x_j}{h}\right)$, where $\mathcal{N}(i)$ is the set of all neighbors of x_i, K_h is the a Gaussian kernel, h is the bandwidth. Then we normalize the $p(x_i)$ by $\frac{p(x_i)}{\max\{p(x)\}}$. S_{ij} can be obtained by a Gaussian kernel $S_{ij} = \exp\left(-d_{ij}^2/2\sigma\right)$, where $d_{ij}^2 = \|x_i - x_j\|^2$ is the Euclidean distance, and

Table 1 Details for the UCI datasets

Dataset	Labeled samples	Unlabeled samples	Testing samples	Class
Wine	30	148	30	3
Iris	30	120	30	3
Dermatology	30	328	30	6
Balance	30	595	30	3

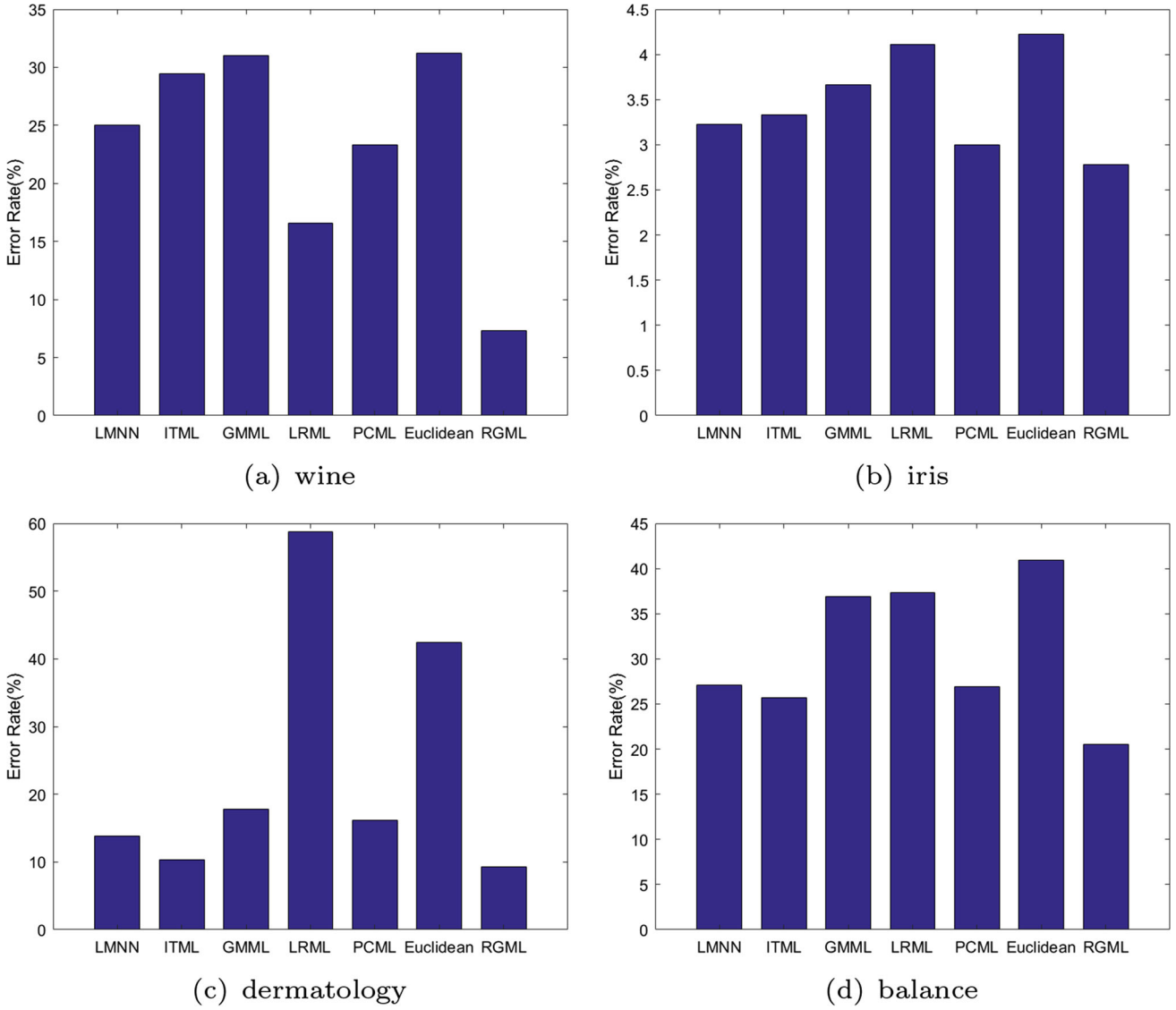
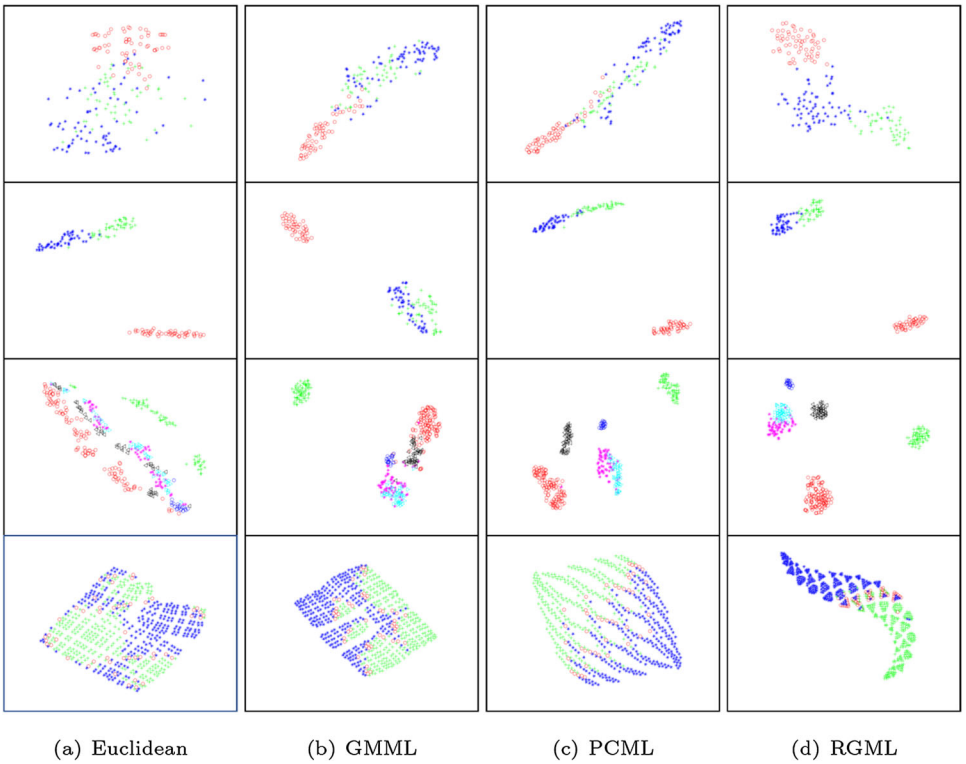


Fig. 2 Test error rates for UCI data sets obtained by different methods

$\sigma = \min D + (1/v)(\max D - \min D)$. $\min D$ and $\max D$ are the minimum and maximum values of the distance between samples.

Figure 2 shows all classification results for four UCI data sets by 1NN classification method under the metric A obtained by different methods. From this figure, we can see that except for the individual results (LRML algorithm on "dermatology" data set), all metric learning algorithms improve the classification results of the Euclidean metric. What's more, the proposed method (RGML) has the best classification results on each UCI data set.

To visualize the effect of the proposed metric learning, we project all results to 2 dimensional space by t-SNE [50]. Visualization results for four UCI data sets (from top to bottom is "wine", "iris", "dermatology", "balance") under different metrics by different methods are shown in Fig. 3. From left to



(a) Euclidean (b) GMML (c) PCML (d) RGML

Fig. 3 Visualization results for UCI data sets ("wine", "iris", "dermatology", "balance" from top to bottom) by different methods

right, they are Euclidean metric without any metric learning, GMML, PCML and RGML. Different colors represent the samples from different classes, and we can clearly see that different classes of points are clearly distinguished by our algorithm, which is superior to other methods.

How to choose the trade-off parameter λ is also a key point for improving our method's performance. Figure 4 shows its influence on recognition error rate under different intervals for four different UCI data sets. For "wine", "iris" and "dermatology", the range of λ in the experiment is $0.1 \sim 2\times10^{-3}$, and for "balance", the range is $0.1 \sim 2 \times 10^{-4}$. For "wine", "iris", "dermatology" and "balance", the best λ value is 1.5×10^{-3}, 0.1×10^{-3}, 0.7×10^{-3}, and 1.1×10^{-4}, respectively.

Furthermore, Fig. 5 shows the variation trend of objective function along with the iteration by Riemannian gradient descent (RGD) algorithm for four UCI data sets. It can be seen that with the number of iterations increases, the values of the objective function keep decreasing and gradually converge. especially for "wine" data set, its convergence speed is the fastest among the four UCI data sets.

4.2 USPS data set

The USPS data set contains 20,000 samples, and all samples in USPS data set are 16×16 gray-level images of 10 types of handwritten digits. In our experiment, 30% of the data are randomly selected as the labeled data. The regularizer used for USPS data set is the same as the UCI data sets, and the parameters are also same.

Firstly, for the parameter λ, we also conduct the experiment for its selection. Figure 6 shows the recognition error rate under different λ, from which we can see that the range of the λ is $0.5 \sim 10\times10^{-6}$ and the best λ for USPS data set is 1.5×10^{-6}, which will be used in the following comparison experiments.

Figure 7 shows the recognition error rates for USPS dataset by 1NN classification method under the metric A obtained by different methods. We can see that LRML performs badly in this data set, and the reason why may be that it only considers the pairwise constraints for labeled data and its regularization term only considers the relations among their neighbors rather than among their local topology. And the performance of the proposed RGML method is comparable to other methods such as ITML, GMML, PCML and LMNN, particularly superior to LRML. Note that LMNN has the lowest error rate, since it enables a large margin between different class points while for RGML there is no such clear margin, which may be helpful in the classification task.

Furthermore, Fig. 8 gives all recognition results of different methods. Note that the first row shows the test samples and the rest rows display their nearest neighbours obtained by different metric methods. From this figure, we can find that the proposed method owns a very high classification accuracy compared with other methods, for example, LMNN, ITML,

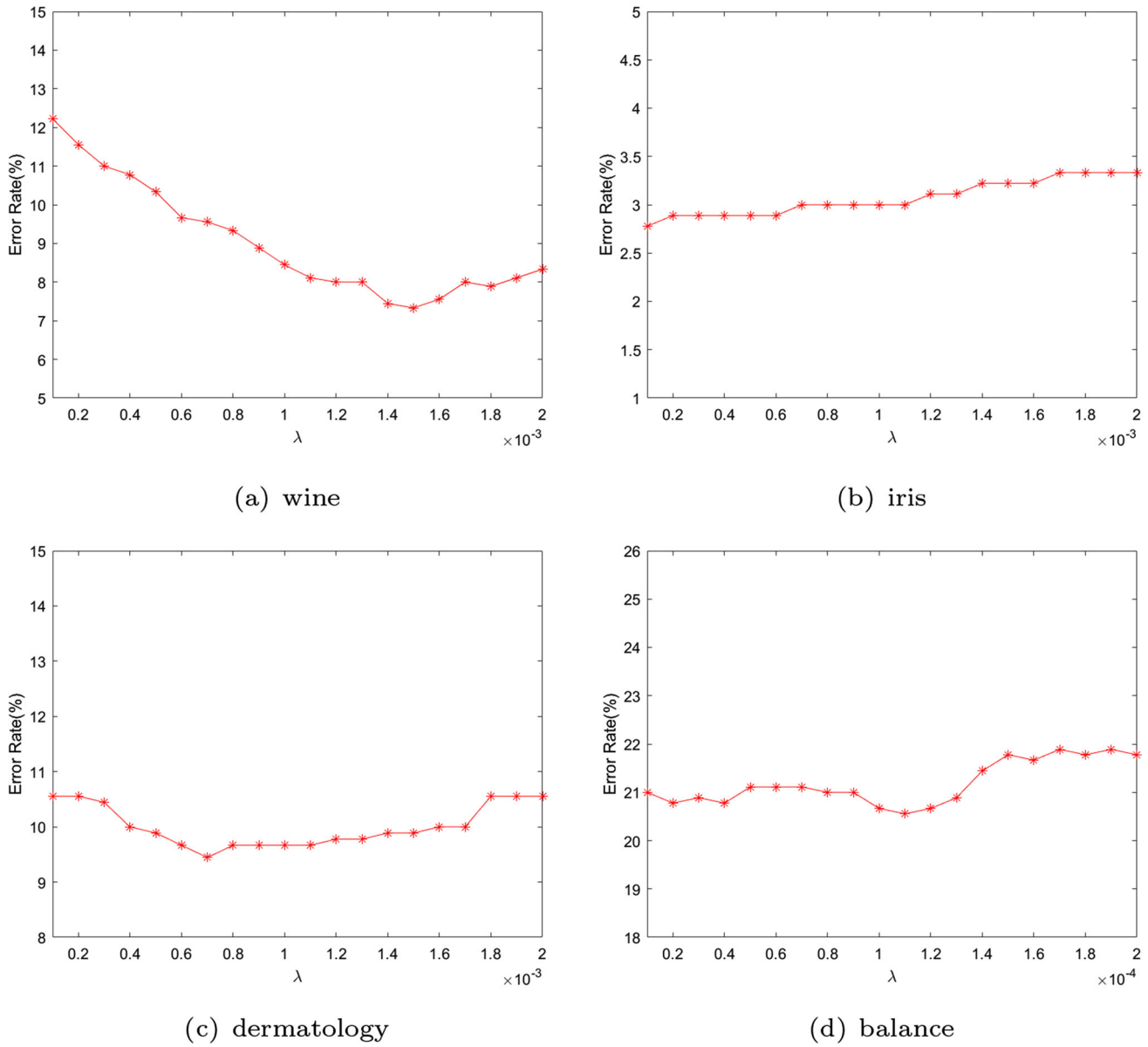


Fig. 4 The influence of λ on recognition error rates under different intervals for UCI data sets by RGML

LRML and PCML, which all misclassify some handwritting digits such as 3, 5 or 8.

4.3 Facial data sets

YaleB and ORL are two classic facial data sets, where YaleB data set contains 2414 frontal-face images over 38 subjects and about 64 images per subject, which are captured under different lighting conditions and various facial expressions, and ORL database contains 400 images from 40 distinct subjects, in which for some subjects, the images are taken at different times, varying the lighting, facial expressions and facial details. All images in both data sets are cropped and resized to the size of 32×32. Similarly, 30% of the data are randomly selected as the labeled data, and we also use the same regularizer as the UCI data sets.

Initially, we conduct experiments to determine the optimal selection of the parameter λ. Figure 9 shows all recognition error rates under different λ for two facial data sets. The range of λ for "yaleB" data set is $0.5 \sim 10 \times 10^{-7}$ and the best λ value is 1×10^{-7}. For "ORL", the range of λ is $0.5 \sim 10 \times 10^{-6}$ and the best one is 5×10^{-7}. Similarly, we choose the optimal λ for our method RGML in the following experiments to compare with other methods.

Figure 10 shows recognition error rates using kNN classification algorithm by different metric learning methods on

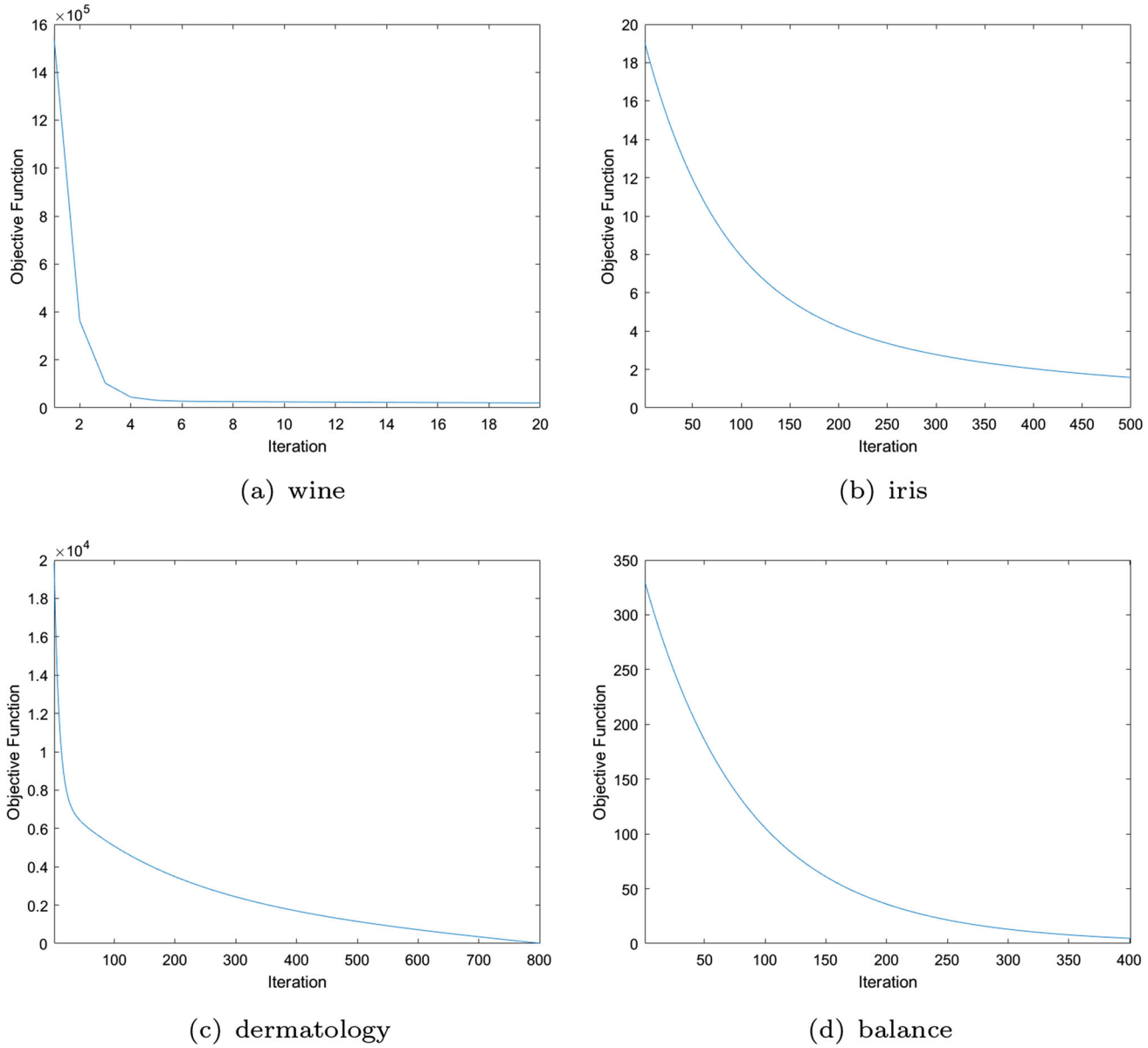


Fig. 5 Variation of the objective function of RGML along with the iterations for UCI data sets

two data sets. We find that for two face data sets our method have best performance among all the metric learning method for every k. Moreover, the classification effect of the proposed RGML algorithm is almost unaffected by the value of k, which shows its robustness for different k.

4.4 Hyperspectral image datasets

In this section, we will perform classification experiments on two hyperspectral data such as Indian pines data set and Kennedy Space Center (KSC) data set. Note that Indian pines data set consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range $0.4 \sim 2.5 \times 10^{(-6)}$ meters. Here the number of bands are reduced to 200 by removing bands covering the region of water absorption: [104–108], [150–163], 220. The KSC data, acquired from an altitude of approximately 20 km, has a spatial resolution of 18 m. After removing water absorption and low SNR bands, 176 bands are used for the analysis. For classification, 13 classes representing various land cover types that occur in this environment are defined for the site. The detailed information of these two data sets is shown in Tables 2 and 3.

During these experiments about HSI data sets, 30% of the data are randomly selected from each class as the labeled data. The regularizer used here is the spatial regularizer in (27), in which the parameter σ will have a great impact on

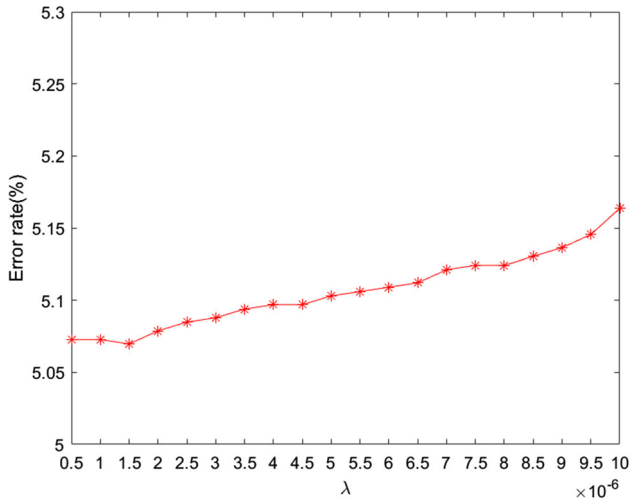


Fig. 6 The influence of λ on recognition error rates under different intervals for USPS data set by RGML

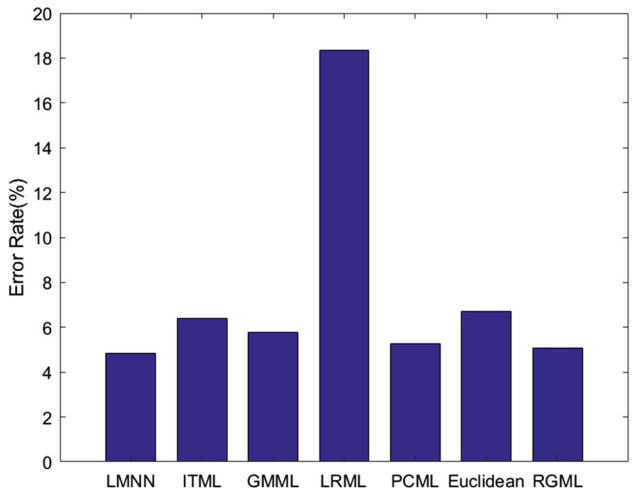


Fig. 7 Recognition error rates by different methods for USPS data set

the classification accuracy. The following Fig. 11 will demonstrate how to choose σ. Note that all experiments are repeated 10 times and we use three classification measurements, i.e., average classification accuracy of all the classes (AA), overall classification accuracy (OA) and kappa coefficient (KC).

Figure 11 shows their OAs of two data sets under the different parameters in the regularizer. Window size represents the range to determine spatial neighbors of each point. In our experiment, we set three different window sizes by "3 × 3", "5 × 5" and "7 × 7". For each window size, we conduct classification experiments under σ ranging in 0.1, 0.25, 0.5, 0.75, 0.9, 1. The results show that for Indian Pines data set when "window size" $= 7 \times 7$ and $\sigma = 0.25$, we can get the best result. For KSC data set, the best parameters are "window size"$= 5 \times 5$ and $\sigma = 0.1$.

Similarly, as for the trade-off parameter λ, we also carry out the experiments on it. Figure 12 shows the OAs of two data sets under different λ. For Indian pines data set, the range of λ is 0.9 1.1, we find out that the best λ is 1.03. However, for KSC data set, the value of λ ranges from 0.08 to 0.1, and the experiment shows that the best λ is 0.081.

Based on these optimal parameters, we will compare the proposed method with other algorithms in Tables 4 and 5.

From Table 4 and 5, we can see all classification results for Indian Pines and KSC data sets. In these two tables, we list all classification accuracy indexes such as AA, OA and KC, for all algorithms in each data class. Notably, the highest value for each index is highlighted in bold. And the proposed RGML method shows its superiority in most cases. Specifically, for Indian Pines data set, OA and KC of ours are the highest among all algorithms involved in comparisons. For KSC data set, all of three classification indexes are the highest.

Furthermore, Figs. 13 and 14 show the classification maps for two HSI data sets. The Ground Truth was shown at the top left corner, where the pixels in different colors represent different features. The predicted results of different algorithms for each pixel are presented in the classification maps by different colors. By comparing the classification maps of all algorithms with the Ground Truth, we can see that our algorithm performs best among all methods involved in comparisons.

5 Conclusion

In this work, point-to-class distance is used to investigate the semi-supervise metric learning problem, which can better describe the differences between samples than the point-to-point structure to some extent. The goal of metric learning is to narrow the distance of inner-class data and meanwhile to enlarge the distance of inter-class data, which can be achieved by the triplet constraint. For traditional point-to-point triplet constraint, if we want to consider the triplet constraint between all the labeled samples, the number of triples in the model is going to be huge. Usually, an attempt is only to use the distances from one point to its neighbors rather than to all other points, however, which will cause the information to be lost in practice. One alternative triplet constraint formulated by the point-to-class structure is raised in this work, in which we consider the distance from one point to the center of each class. In addition, to make our model more accurately to capture those distance that attempts to become larger during metric learning, we put different weights on the point-to-class triplet constraint to measure the interclasses dissimilarity. For each point, the class-centers that correspond to the larger weights will be preferentially considered for enlarging the distances between it and its different classes. By adding weights, we can automatically figure out which class centers are close enough to be prior considered. Then,

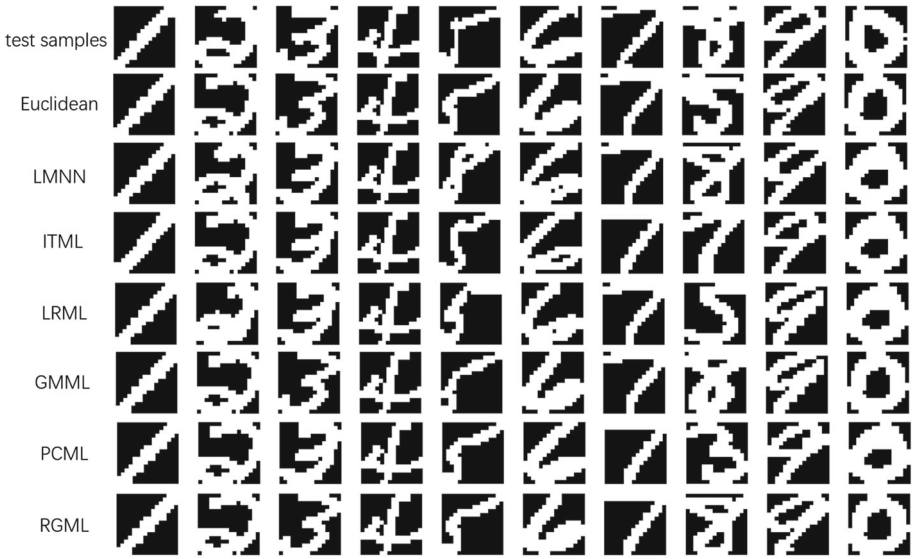
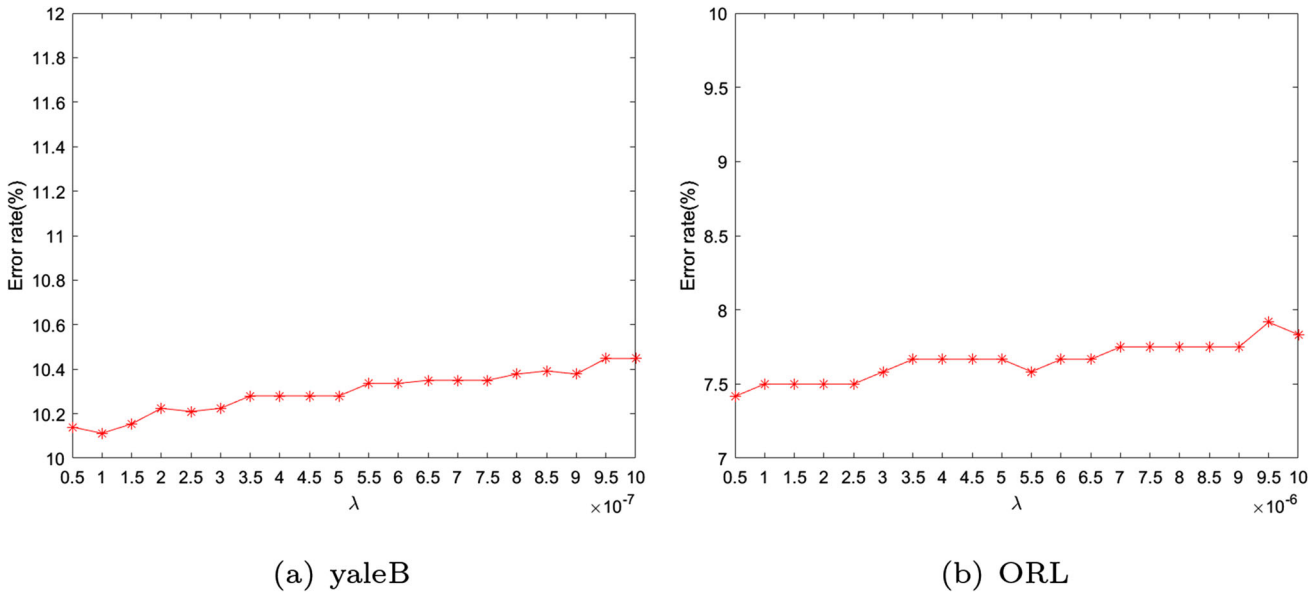


Fig. 8 Recognition results for the test samples under different metrics for USPS data set



(a) yaleB (b) ORL

Fig. 9 The influence of λ on recognition error rates under different intervals for yaleB and ORL data sets by RGML

regularization term is also incorporated into the proposed model in order to mitigate the overfitting phenomenon and preserve the topological structure of the data. For different data types, two kinds of regularizer are utilized in this work, for example, in term of hyperspectral images data, since their spatial neighbors are usually in the same class, the added spatial regularizer could better preserve their such structure during metric learning. Moreover, since the objective function in the proposed model is smooth on an SPD manifold, Riemannian gradient descent algorithm is given, which owns higher computational efficiency and is one appropriate choice for the solution of the model. Finally, we use the learned metrics to classify different data sets and compare the classification performance to state-of-the-art methods. The experimental results show that the proposed method has a better performance in classification of various kind of data such as hyperspectral image data.

The future work will primarily focus on reducing computational complexity of metric learning to reduce the time cost of the model when handling large datasets. Moreover, for the Riemannian manifold optimization method in this work, we plan to introduce other Riemannian metrics, such as the log-Euclidean metric, in conjunction with other Riemannian manifold optimization methods like Riemannian conjugate gradient method for further research.

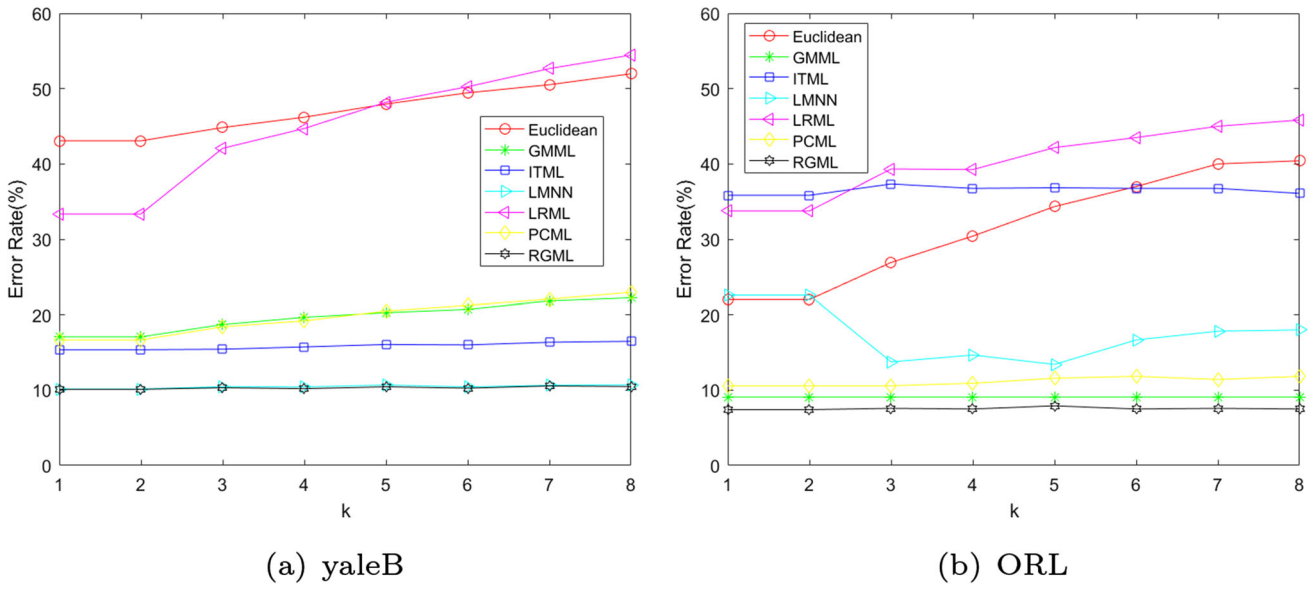


Fig. 10 Recognition error rates for yaleB and ORL by different methods for k from 1 to 8

Table 2 Number of samples of each class for the Indian Pines data set

Class NO	Land cover	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-tree	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybeans-notill	972
11	Soybeans-mintill	2455
12	Soybeans-clean	593
13	Wheat	205
14	Woods	1265
15	Bldg-grass-tree-drives	386
16	Stone-steel-towers	93

Table 3 Number of samples of each class for the KSC data set

Class NO	Land cover	Samples
1	Scurb	761
2	Willow-swamp	243
3	Cabbage-palm-hammock	256
4	Cabbage-palm/oak-hammock	252
5	Slash-pine	161

Table 3 continued

Class NO	Land cover	Samples
6	Oak/broadleaf-hammock	229
7	Hardwood-swamp	105
8	Graminoid-marsh	431
9	Spartina-marsh	520
10	Cattail-marsh	404
11	Salt-marsh	419
12	Mud-flats	503
13	Water	927

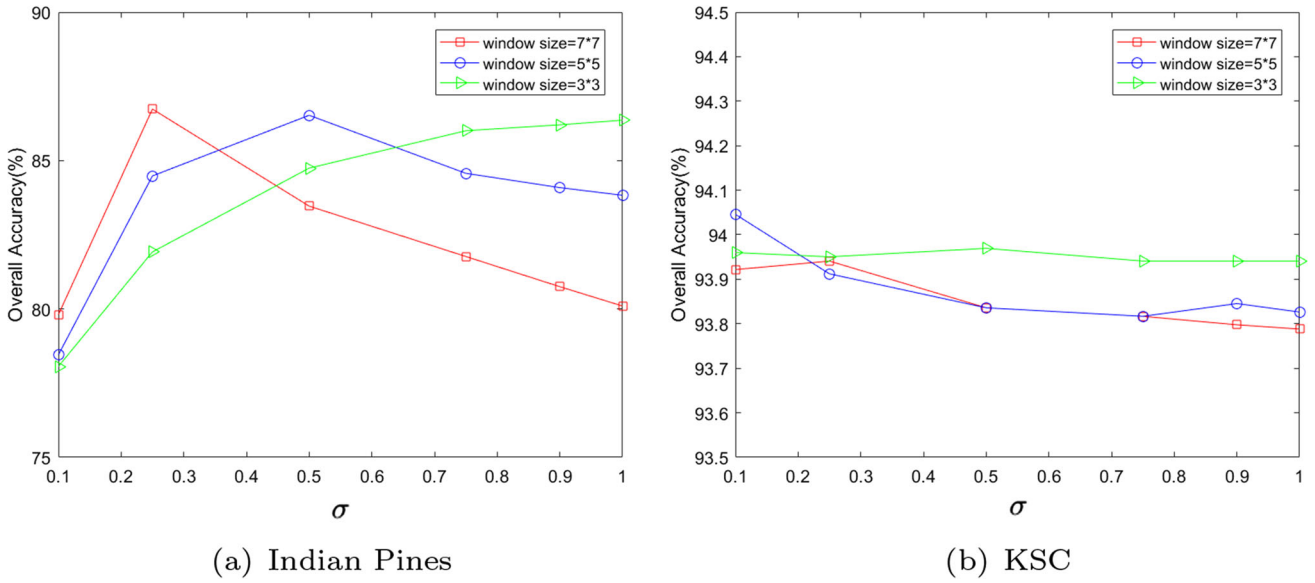


Fig. 11 Influence of different parameters in the regularizer for two HSI data sets

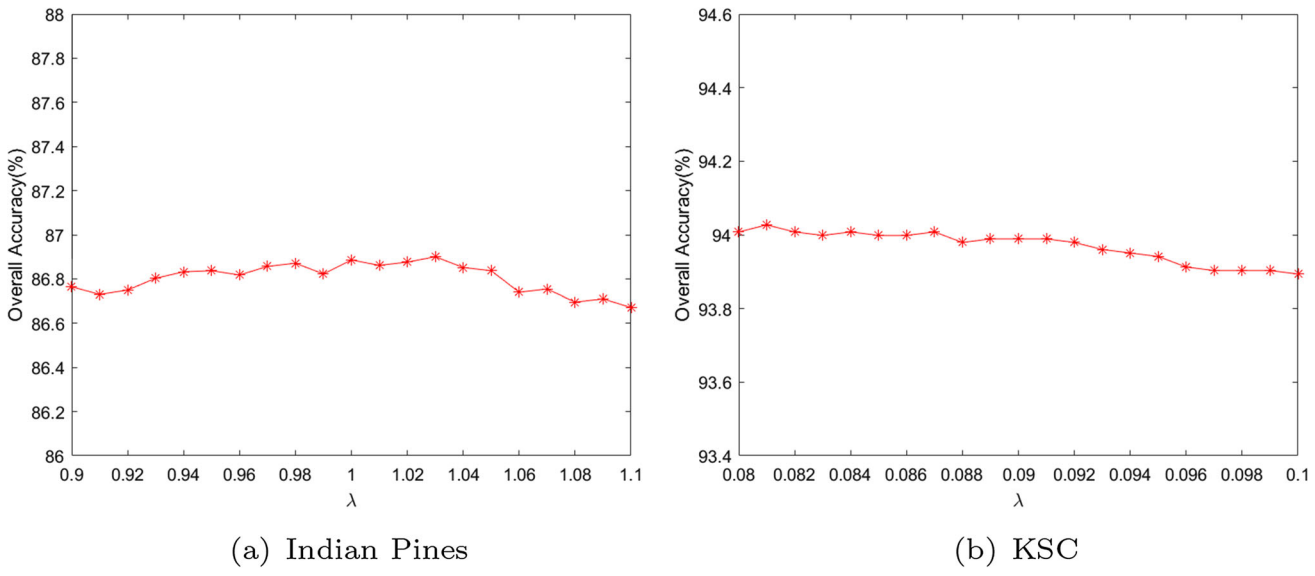


Fig. 12 Influence of different parameter λ for two HSI data sets

Table 4 Classification results for Indian Pines data set by 1NN under different metrics learned by different methods

Class NO	Euclidean	GMML	ITML	LMNN	LRML	PCML	RGML
1	58.00 ± 12.29	59.00 ± 12.87	60.00 ± 17.64	74.00 ± 17.13	71.00 ± 8.76	88.00 ± 6.32	70.00 ± 14.14
2	61.64 ± 3.29	65.98 ± 3.29	71.01 ± 4.16	74.16 ± 2.47	61.33 ± 4.14	78.32 ± 2.77	84.55 ± 2.61
3	60.06 ± 3.70	66.14 ± 2.29	65.42 ± 3.46	72.65 ± 2.81	60.06 ± 3.18	77.59 ± 1.39	77.83 ± 3.39
4	47.71 ± 6.25	50.83 ± 4.41	56.67 ± 7.97	63.96 ± 6.81	44.38 ± 6.29	73.96 ± 5.66	66.88 ± 5.51
5	88.45 ± 3.25	88.87 ± 2.27	83.4 ± 4.44	91.55 ± 2.42	87.32 ± 3.30	93.09 ± 2.75	90.41 ± 3.15
6	94.04 ± 1.07	94.18 ± 1.26	94.04 ± 2.68	96.85 ± 0.87	94.38 ± 1.24	97.33 ± 1.14	97.60 ± 1.30
7	93.33 ± 11.65	93.33 ± 11.65	85.00 ± 21.44	91.67 ± 16.2	95.00 ± 8.05	95.00 ± 8.05	95.00 ± 8.05
8	96.35 ± 2.32	96.88 ± 1.77	96.77 ± 2.27	98.44 ± 1.01	95.10 ± 2.31	99.38 ± 0.73	99.06 ± 0.77
9	57.50 ± 26.48	57.50 ± 23.72	77.50 ± 27.51	82.50 ± 16.87	55.00 ± 22.97	87.50 ± 17.68	67.50 ± 28.99
10	69.38 ± 2.85	72.97 ± 3.28	72.67 ± 2.84	76.62 ± 3.20	63.38 ± 4.74	82.62 ± 3.12	84.77 ± 3.35
11	74.87 ± 2.60	77.84 ± 1.43	78.35 ± 1.67	81.18 ± 1.30	79.04 ± 1.66	84.83 ± 1.42	87.86 ± 0.81
12	52.27 ± 3.54	57.23 ± 3.46	64.45 ± 5.78	74.54 ± 2.38	53.11 ± 3.06	77.56 ± 2.86	80.50 ± 3.71
13	95.37 ± 3.14	95.61 ± 2.52	95.85 ± 5.15	98.54 ± 1.71	95.61 ± 3.95	98.78 ± 1.72	98.78 ± 1.72
14	89.88 ± 1.72	90.51 ± 2.51	90.59 ± 1.84	91.86 ± 1.80	92.13 ± 1.48	94.19 ± 1.05	95.26 ± 1.43
15	46.15 ± 5.09	47.44 ± 4.60	53.46 ± 8.40	63.08 ± 5.98	49.49 ± 4.61	60.26 ± 7.05	66.03 ± 3.15
16	87.37 ± 8.3	91.05 ± 6.59	86.32 ± 7.92	87.37 ± 8.30	83.68 ± 9.43	91.05 ± 7.46	86.84 ± 8.32
AA	73.27 ± 2.19	75.34 ± 2.04	76.97 ± 3.92	82.43 ± 2.34	73.75 ± 2.14	**86.22 ± 1.45**	84.30 ± 1.77
OA	73.60 ± 0.83	76.33 ± 0.68	77.56 ± 0.74	81.73 ± 0.78	74.30 ± 0.50	85.02 ± 0.92	**86.90 ± 0.65**
KC	0.6985 ± 0.0093	0.7297 ± 0.0079	0.7438 ± 0.0087	0.7917 ± 0.009	0.7052 ± 0.0058	0.8291 ± 0.0105	**0.8504 ± 0.0075**

The bold indicates the highest value for each classification accuracy indexes

Table 5 Classification results for KSC data set by 1NN under different metrics learned by different methods

Class NO	Euclidean	GMML	ITML	LMNN	LRML	PCML	RGML
1	92.94 ± 1.22	92.68 ± 1.74	93.53 ± 2.14	94.51 ± 1.08	94.25 ± 1.79	90.33 ± 2.30	95.82 ± 1.75
2	88.16 ± 5.25	86.33 ± 5.36	87.35 ± 4.98	90.82 ± 2.20	92.04 ± 4.56	87.14 ± 4.31	92.24 ± 4.17
3	89.23 ± 3.53	89.42 ± 4.56	88.85 ± 4.60	92.5 ± 2.47	86.15 ± 4.03	85.58 ± 4.64	89.62 ± 3.29
4	68.24 ± 7.44	64.31 ± 7.50	66.08 ± 6.67	73.92 ± 6.54	76.27 ± 8.13	58.82 ± 4.53	71.57 ± 6.22
5	61.82 ± 9.92	57.88 ± 9.42	62.73 ± 10.40	68.18 ± 10.13	76.67 ± 8.21	58.18 ± 8.90	79.70 ± 4.05
6	49.57 ± 6.94	48.91 ± 7.12	55.00 ± 9.17	58.48 ± 3.90	63.04 ± 9.61	41.09 ± 9.64	70.00 ± 7.45
7	85.71 ± 5.94	81.43 ± 7.60	83.33 ± 9.32	88.1 ± 6.83	81.43 ± 9.38	79.52 ± 11.68	81.90 ± 6.66
8	87.93 ± 3.64	84.37 ± 3.88	89.08 ± 2.72	92.53 ± 2.38	92.41 ± 2.88	85.98 ± 1.7	95.52 ± 2.20
9	96.54 ± 2.09	95.19 ± 2.60	96.44 ± 1.76	97.21 ± 1.78	96.92 ± 2.56	95.77 ± 1.37	97.50 ± 1.22
10	96.67 ± 2.10	96.67 ± 1.75	99.26 ± 0.86	98.52 ± 1.13	99.75 ± 0.52	97.65 ± 0.91	99.75 ± 0.52
11	98.10 ± 1.79	97.86 ± 1.84	98.33 ± 1.15	98.81 ± 1.25	98.45 ± 1.59	97.5 ± 1.73	98.93 ± 1.04
12	94.16 ± 2.49	93.66 ± 2.73	95.74 ± 1.55	97.92 ± 1.09	99.50 ± 0.52	95.54 ± 2.3	99.41 ± 0.69
13	99.89 ± 0.23	99.89 ± 0.23	100.00 ± 0.00	100 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
AA	85.30 ± 1.41	83.74 ± 1.47	85.82 ± 1.47	88.58 ± 1.19	88.99 ± 1.13	82.55 ± 1.10	**90.15 ± 0.69**
OA	90.30 ± 0.82	89.26 ± 0.89	90.92 ± 0.70	92.76 ± 0.63	93.09 ± 0.57	88.56 ± 0.68	**94.03 ± 0.64**
KC	0.892 ± 0.0091	0.8804 ± 0.0099	0.8989 ± 0.0078	0.9194 ± 0.0070	0.9231 ± 0.0064	0.8726 ± 0.0075	**0.9335 ± 0.0071**

The bold indicates the highest value for each classification accuracy indexes

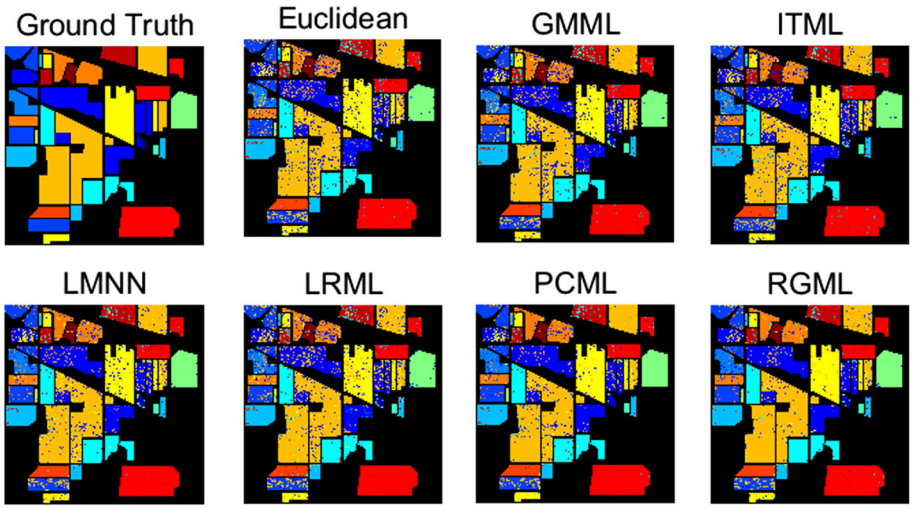


Fig. 13 Classification maps for Indian Pines data set by different methods

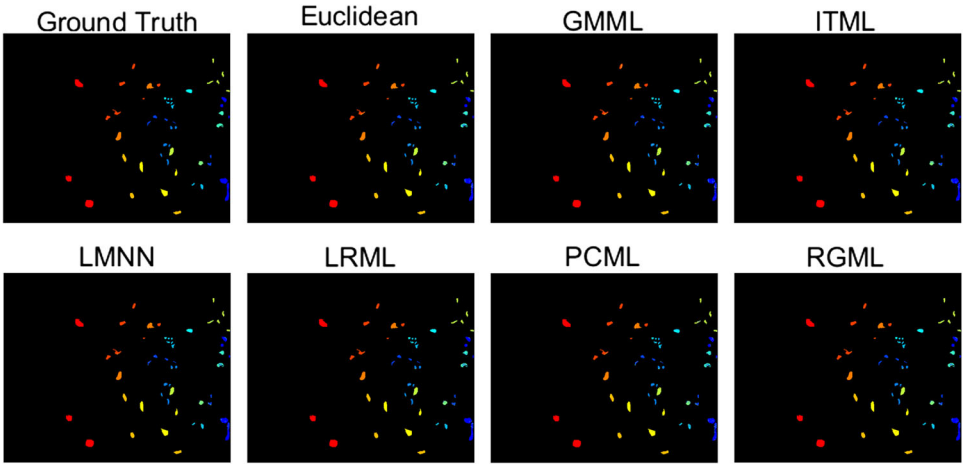


Fig. 14 Classification maps for KSC data set by different methods

Acknowledgements Thank all the referees and the editorial board members for their insightful comments and suggestions, which improved our paper significantly. This study was funded by the National Natural Science Foundation of China under the Grants No.11501351.

Data availibility The data that support the findings of this study are openly available in UCI Machine Learning Repository [49], [http://archive.ics.uci.edu/ml],Codes and Datasets for Feature Learning, [http://www.cad.zju.edu.cn/home/dengcai/Data/data.html] and Hyperspectral Remote Sensing Scenes, [https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes].

Declarations

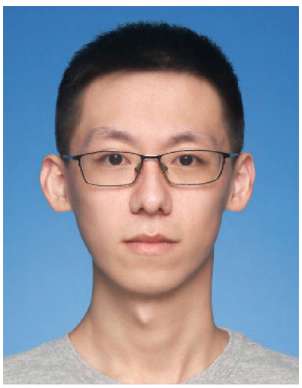
Conflict of interest The authors declare no Conflict of interest.

References

1. Chen, P.-H., Lin, C.-J., Schölkopf, B.: A tutorial on ν-support vector machines. Appl. Stoch. Model. Bus. Ind. **21**(2), 111–136 (2005)
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
3. Bro, R., Smilde, A.K.: Principal component analysis. Anal. Methods **6**(9), 2812–2831 (2014)
4. Cox, M.A., Cox, T.F.: Multidimensional scaling. In: Chen, C.H., Härdle, W.K., Unwin, A. (eds.) Handbook of data visualization, pp. 315–347. Springer, Berlin, Heidelberg (2008)
5. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
6. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
8. He, X., Niyogi, P.: Locality preserving projections. Adv. Neural Inform. Process. Syst. **16** (2003)
9. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learn. Res. **4**(Jun), 119–155 (2003)
10. He, X., Cai, D., Yan, S., Zhang, H.-J.: Neighborhood preserving embedding. In: Tenth IEEE international conference on computer vision (ICCV'05) Volume 1, vol. 2, pp. 1208–1213. IEEE (2005). https://doi.org/10.1109/ICCV.2005.167
11. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Proceedings of the 15th international conference on neural information processing systems. NIPS'02, pp. 521–528. MIT Press, Cambridge, MA (2002)

12. Globerson, A., Roweis, S.: Metric learning by collapsing classes. Adv. Neural Inform. Process. Syst. **18** (2005)
13. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**(2), 207–244 (2009)
14. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th international conference on machine learning, pp. 209–216 (2007)
15. Goldberger, J., Hinton, G.E., Roweis, S., Salakhutdinov, R.R.: Neighbourhood components analysis. Adv. Neural Inform. Process. Syst. **17** (2004)
16. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. J. Mach. Learn. Res. **8**(5), 1027–1061 (2007)
17. Zadeh, P., Hosseini, R., Sra, S.: Geometric mean metric learning. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd international conference on machine learning. Proceedings of machine learning research, vol. 48, pp. 2464–2471. PMLR, New York, New York (2016). https://proceedings.mlr.press/v48/zadeh16.html
18. Zuo, W., Wang, F., Zhang, D., Lin, L., Huang, Y., Meng, D., Zhang, L.: Distance metric learning via iterated support vector machines. IEEE Trans. Image Process. **26**(10), 4937–4950 (2017)
19. Hoi, S.C., Liu, W., Chang, S.-F.: Semi-supervised distance metric learning for collaborative image retrieval and clustering. ACM Trans. Multimed. Comput. Commun. Appl. **6**(3), 1–26 (2010)
20. Wang, Q., Yuen, P.C., Feng, G.: Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions. Pattern Recogn. **46**(9), 2576–2587 (2013)
21. Sugiyama, M., Idé, T., Nakajima, S., Sese, J.: Semi-supervised local fisher discriminant analysis for dimensionality reduction. Mach. Learn. **78**(1), 35–61 (2010)
22. Ying, S., Wen, Z., Shi, J., Peng, Y., Peng, J., Qiao, H.: Manifold preserving: an intrinsic approach for semisupervised distance metric learning. IEEE Trans. Neural Netw. Learn. Syst. **29**(7), 2731–2742 (2018). https://doi.org/10.1109/TNNLS.2017.2691005
23. Ji, S., Zhang, Z., Ying, S., Wang, L., Zhao, X., Gao, Y.: Kullback–Leibler divergence metric learning. IEEE Trans. Cybern. **52**(4), 2047–2058 (2022). https://doi.org/10.1109/TCYB.2020.3008248
24. Wang, R., Wu, X.-J., Kittler, J.: Graph embedding multi-kernel metric learning for image set classification with grassmannian manifold-valued features. IEEE Trans. Multimed. **23**, 228–242 (2021). https://doi.org/10.1109/TMM.2020.2981189
25. Ren, Z., Kong, X., Zhang, Y., Wang, S.: Ukssl: Underlying knowledge based semi-supervised learning for medical image classification. IEEE Open J. Eng. Med. Biol. (2023)
26. Zhu, W., Zhang, X., Hu, C., Zhao, B., Peng, S., Yang, H.: A comfort quantification method based on semi-supervised learning for automated vehicle at lane change scenarios. IEEE Trans. Intell. Veh. **8**(5), 3375–3383 (2022)
27. Kim, G., Choi, J.G., Ku, M., Lim, S.: Developing a semi-supervised learning and ordinal classification framework for quality level prediction in manufacturing. Comput. Ind. Eng. **181**, 109286 (2023)
28. Karimi, Z., Ghidary, S.S.: Semi-supervised metric learning in stratified spaces via intergrating local constraints and information-theoretic non-local constraints. Neurocomputing **312**, 165–176 (2018)
29. Li, Y., Tian, X., Tao, D.: Regularized large margin distance metric learning. In: 2016 IEEE 16th international conference on data mining (ICDM), pp. 1015–1022 (2016). https://doi.org/10.1109/ICDM.2016.0129
30. Wang, Z., Li, Y., Tian, X.: Semi-supervised coefficient-based distance metric learning. In: International conference on neural information processing, pp. 586–596. Springer (2017)
31. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press, Princeton (2009). https://doi.org/10.1515/9781400830244
32. Chen, Y., Zhao, F., Zhang, N., Zhou, C.: Semi-supervised subspace metric learning. In: 2021 IEEE 3rd international conference on frontiers technology of information and computer (ICFTIC), pp. 66–76 (2021). https://doi.org/10.1109/ICFTIC54370.2021.9647276
33. Ma, L., Ma, A., Ju, C.H., Li, X.: Graph-based semi-supervised learning for spectral-spatial hyperspectral image classification. Pattern Recognit. Lett. **83**, 133–142 (2016)
34. Li, X., Zhang, L., You, J.: Locally weighted discriminant analysis for hyperspectral image classification. Remote Sens. **11**(2), 109 (2019)
35. Wen, J., Tian, Z., Liu, X., Lin, W.: Neighborhood preserving orthogonal pnmf feature extraction for hyperspectral image classification. IEEE J. Select. Top. Appl. Earth Obs. Remote Sens. **6**(2), 759–768 (2012)
36. Peng, J., Zhang, L., Li, L.: Regularized set-to-set distance metric learning for hyperspectral image classification. Pattern Recogn. Lett. **83**, 143–151 (2016)
37. Xiao, Z.: Non-negative matrix factorization with local preservation for hyperspectral image dimensionality reduction. Remote Sens. Lett. **5**(9), 793–802 (2014)
38. Roy, S.K., Mhammedi, Z., Harandi, M.: Geometry aware constrained optimization techniques for deep learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4460–4469 (2018)
39. Bécigneul, G., Ganea, O.-E.: Riemannian adaptive optimization methods. arXiv:1810.00760 (2018)
40. Kasai, H., Jawanpuria, P., Mishra, B.: Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In: International Conference on Machine Learning, pp. 3262–3271. PMLR (2019)
41. Gao, Z., Wu, Y., Fan, X., Harandi, M., Jia, Y.: Learning to optimize on Riemannian manifolds. IEEE Trans. Pattern Anal. Mach. Intell. **45**(5), 5935–5952 (2022)
42. Boumal, N.: An introduction to optimization on smooth manifolds. (2022)
43. Meyer, G., Bonnabel, S., Sepulchre, R.: Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. J. Mach. Learn. Res. **12**, 593–625 (2011)
44. Usevich, K., Markovsky, I.: Optimization on a Grassmann manifold with application to system identification. Automatica **50**(6), 1656–1662 (2014)
45. Hu, J., Liu, X., Wen, Z.-W., Yuan, Y.-X.: A brief introduction to manifold optimization. J. Op. Res. Soc. China **8**(2), 199–248 (2020)
46. Gao, B., Absil, P.-A.: A Riemannian rank-adaptive method for low-rank matrix completion. Comput. Optim. Appl. **81**(1), 67–90 (2022)

47. Zhu, X., Sato, H.: Riemannian conjugate gradient methods with inverse retraction. Comput. Optim. Appl. **77**(3), 779–810 (2020)
48. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. **29**(1), 328–347 (2007)
49. Asuncion, A., Newman, D.: UCI machine learning repository. Irvine, CA (2007)
50. Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)



YiZhe Xia received the B.S. degree from the Department of Mathematics, Shanghai University, China, in 2021, and the M.S. degree from the Department of Mathematics, Shanghai University, China, in 2024. His research interests include image classification, pattern recognition and hyperspectral remote sensing.

Hongjuan Zhang received the M.S. degree from the Department of Applied Mathematics, Dalian University of Technology, Dalian, China, in 2005, and the Ph.D. degree in mathematics from the Dalian University of Technology, in 2009. In 2012, she was a Visiting Scholar with The University of Aizu, Aizuwakamatsu, Japan. From 2016 to 2017, she visited the University of Rochester, Rochester, NY, USA, where she was engaged in research in music data's processing problem. She is currently an Associate Professor with the Department of Mathematics, Shanghai University, Shanghai, China. Her research interests cover blind signal processing, pattern recognition, and systems optimization.