International Journal of Machine Learning and Cybernetics (2022) 13:1317–1336
https://doi.org/10.1007/s13042-021-01449-8

ORIGINAL ARTICLE

An unsupervised multi-manifold discriminant isomap algorithm based on the pairwise constraints

Xiaofang Gao[1] · Jiye Liang[1,2] · Wenjian Wang[1,2] · Xuefei Bai[1] · Lina Jia[1]

Received: 22 October 2020 / Accepted: 8 October 2021 / Published online: 5 March 2022

Abstract

In this paper, an unsupervised multi-manifold Isomap algorithm, which is named UMD-Isomap, is proposed for the purpose of dimensionality reduction and clustering of multi-manifold data. First, the global pairwise constraints are constructed by training m mixtures of probabilistic principal component analyzers (MPPCA) and propagating their local tangent subspaces. At the same time, the sub-manifolds are also clustered, and their classes information are recorded in the pairwise constraints. If the number of sub-manifolds is known, a new pairwise constraints is computed by using a cluster ensemble algorithm, which creates a similarity matrix by accumulating c sets of pairwise constraints. Subsequently, a new objective function with pairwise constraints and two supervised solutions are proposed to achieve the dimensionality reduction of the multi-manifolds. The proposed UMD-Isomap algorithm achieved better performance in terms of dimensionality reduction and clustering accuracy than other commonly used methods and its effectiveness was verified.

Keywords Manifold learning · Pairwise constraints · Tangent space propagation · Clustering

1 Introduction

The effective use of high-dimensional data involves a strong framework and a flexible methodology to achieve a tight connection between information science and the original purpose of data analysis of datasets derived from various scientific disciplines. A number of manifold learning methods have been proposed for solving the problem of nonlinear dimensionality reduction (NLDR), e.g., isometric feature mapping (ISOMAP) [1], locally linear embedding (LLE) [2], and Laplacian eigenmaps (LE) [3]. These methods assume that the real-world data are intended for embedding into a lower dimensional space while preserving the geometrical structure [4]. If low-dimensional representations of the data can be obtained, then clustering, visualization, and searching are more convenient and effective. So the manifold learning algorithms have been widely exploited in a broad range of applications including image classification [5, 6], image retrieval [7], activity recognition [8] and biomedical information detection [9].

The classical manifold learning algorithms often assume that the data resides on a single manifold. However, a mixture of manifolds typically occurs in practice. For instance, in handwritten digit recognition, each digit forms its own manifold in the feature space; in computer vision, motion segmentation is an essential process for many computer vision algorithms. These manifolds may intersect or partially overlap, and they may contain different dimensionality, orientation, and density.

At present, several multi-manifold learning methods have been proposed. K-Manifolds [10] was first proposed to classify unorganized data roughly occurring on multiple intersecting nonlinear manifolds. Unfortunately, this method is limited to deal with intersecting manifolds because the estimation of geodesic distances fails when the clusters are widely separated. In contrast, the decomposition-composition (D-C) method [11] and DC-Isomap algorithm [12] can only classify well-separated multi-manifolds. Their decomposition processes classify the multi-manifold to several sub-manifolds, and their composition processes also enable low-dimensional embedding. There are many spectral clustering [13–15] and Multi-manifold clustering [16–18]

✉ Xiaofang Gao
gxfhtp@sxu.edu.cn

[1] College of Computer and Information Technology, Shanxi University, Taiyuan, China

[2] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Taiyuan, China

methods capable of clustering multi-manifolds into several sub-manifolds, however, these methods are not capable of dimensionality reduction for multi-manifolds.

t-SNE [19, 20] and Umap [21] are two dimensionality reduction and visualization techniques, which produce significantly better visualizations by reducing the tendency to crowd points together in the center of the map. In other word, they can cluster the high-dimensional data that lie on several different, but related, low-dimensional manifolds, and then visualize them in 2D or 3D space. But they are not suitable for the intersected manifolds, especially in low dimensional spaces. The points at the intersection are still very similar after mapping, and the "crowding problem" still exists. Some supervised or semi-supervised manifold learning methods have been proposed to cluster and visualize multi-manifolds. These methods typically fall into one of two methods [22]: distance-based and constraint-based. The distance-based methods use distance metrics to satisfy the objective function. The S-Isomap algorithm [23] utilizes class information to form a geodesic distance matrix that distorts the topology among multi-manifolds, although this method can be applied to a single-manifold with noise. The multi-manifold semi-supervised learning (MMSSL) algorithm [24] is a semi-supervised version of Isomap that cannot be assured to cross the intersection of the manifolds. The M-Isomap [25] first introduces local pairwise constraints to construct the intrinsic and penalty graphs and then uses the graph embedding framework to determine the final optimization criterion. However, M-Isomap only preserves the local geodesic distances, thus it only focuses on increasing the distances between the local interclass points but not the holistic interclass separation. The multi-manifold discriminant Isomap (MMD-Isomap) algorithm [26] is an improvement of the M-Isomap method that constructs global pairwise constraints and optimizes its objective function. M-Isomap and MMD-Isomap are all supervised versions of Isomap. The semi-supervised local multi-manifold Isomap (SSMM-Isomap) [27] improves these two methods by extending the manifold feature learning to a semi-supervised scenario, linear extension scenario and local feature learning scenario simultaneously. Although these constraint-based methods perform well for multi-manifold classifications, it is difficult to select the parameters and separate the embedding of the sub-manifolds, especially for more than three intersecting manifolds.

Compared to supervised learning methods, learning descriptors from unlabeled data is preferable because in many cases (e.g., image registration, template matching, or object tracking, especially for their applications involving images from relatively less commonly used sensors in the vision community, such as infrared, SAR, hyperspectral, medical images, etc.) collecting labels is expensive and even impractical while local image without labels are ubiquitous and easy to collect. [28]

In this paper, we propose an unsupervised multi-manifold discriminant Isomap (UMD-Isomap) algorithm, for the purpose of nonlinear dimensionality reduction and clustering of data lying on the multi-manifold. Our UMD-Isomap algorithm uses the global pairwise constraints by training m mixtures of probabilistic principal component analyzers (MPPCA) [29, 30], propagates the local tangent subspaces, and optimizes the objective function of MMD-Isomap [26] and SSMM-Isomap [27] to dimensionality reduction multi-manifolds. The pairwise constraints include the information of the sub-manifolds, and a priori determination of the number of classes or sub-manifolds is not required. The sub-manifolds are dynamically constructed using the process of tangent subspaces propagation. If the number of sub-manifolds is determined in advance, the pairwise constraints can be re-constructed by ensembling c sets of cluster results.It is noteworthy to express several aspects that demonstrate the contributions of the proposed UMD-Isomap method:

- An unsupervised method is used for the dimensionality reduction and clustering of multi-manifold data; this approach not only creates the global pairwise constraints for clustering the multi-manifold but also establishes a more suitable objective function for its dimensionality reduction.
- An effective method is utilized to compute the global pairwise constraints for multi-manifolds. First, m MPPCA "patches" are trained on the multi-manifold dataset. Then their tangent spaces are propagated to construct the global pairwise constraints. At the same time, the sub-manifolds are created. If the number of sub-manifolds is known, the pairwise constraints can be re-constructed by ensembling c sets of cluster results.
- A suitable objective function is optimized and two effective methods are used for dimensionality reduction of the multi-manifold in the lower dimensional spaces. The method not only optimizes MMD-Isomap and SSMM-Isomap, but it is also easy to determine the parameters and fit for more than three intersecting manifolds.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 describes the proposed method with details. Section 4 presents the experimental results and discussions. The conclusion is presented in Section 5.

2 Related works and necessary definitions

Suppose that the set of data points $X = \{x_1, ..., x_n\}$ is in high dimensional space R^D and the feature space is R^d.

2.1 Mixtures of probabilistic PCA

In the mixtures of probabilistic PCA (MPPCA) model proposed in [29, 30], each d-dimensional data vector x_n in the i.i.d sample $\mathbf{X} = \{x_n\}_{n=1}^{N}$ is generated with two steps. First, a natural number j is generated based on the distribution $p(j) = \pi_j, j = 1, ..., m$ under the constraint $\sum_{j=1}^{m} \pi_j = 1$. Second, given this j, x_n is generated from the restricted factor analysis model

$$\begin{aligned} x_n|j &= \mathbf{A}_j\mathbf{y}_{nj} + \mu_j + \epsilon_{nj}; \\ \mathbf{y}_{nj} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \epsilon_{nj} \sim \mathcal{N}(\mathbf{0}, \sigma_j^2\mathbf{I}), \end{aligned} \tag{1}$$

where $\mathbf{I}$ denotes some unit matrix, μ_j is a d-dimensional mean vector, $\mathbf{A}_j$ is a $d \times k_j$ factor loading matrix, $\mathbf{y}_{nj}$ is an independent k_j-dimensional latent factor vector and σ_j^2 is the noise variance associated with component j. Clearly, this is a mixture of m probabilistic PCA sub-models with mixture proportion π_j's. Unlike the traditional factor analysis model, which assumes that ϵ_{nj} has a diagonal covariance, MPPCA assumes a scalar covariance.

Let $\theta = \{\pi_j, \theta_j; j = 1, ..., m\}$, $\theta_j = \left(\mathbf{A}_j, \mu_j, \sigma_j^2\right)$, and $\mathbf{\Sigma}_j = \mathbf{A}_j\mathbf{A}_j' + \sigma_j^2\mathbf{I}$. Under the MPPCA model, the log likelihood of observing the data $\mathbf{X}$ is

$$\mathcal{L}(\mathbf{X}|\theta) = \sum\nolimits_{n=1}^{N} log\left[\sum\nolimits_{j=1}^{m} \pi_j p(x_n|\theta_j)\right], \tag{2}$$

where $p(x_n|\theta_j) = (2\pi)^{-d/2}|\mathbf{\Sigma}_j|^{-1/2} \cdot exp\{-\frac{1}{2}(x_n - \mu_j)^T\mathbf{\Sigma}_j^{-1}(x_n - \mu_j)\}$, The maximum likelihood estimate $\hat{\theta}$ is defined as

$$\hat{\theta} = \underset{\theta}{argmax}\, \mathcal{L}(\mathbf{X}|\theta). \tag{3}$$

If the prior $p(\theta)$ for model parameter θ is available, then the maximum a posteriori estimate $\hat{\theta}$ can be obtained as

$$\hat{\theta} = \underset{\theta}{argmax}\{\mathcal{L}(\mathbf{X}|\theta) + logp(\theta)\}. \tag{4}$$

Given the number of components m and the subspace dimension $\mathbf{k} = (k_1, k_2, ..., k_m)$, the parameter θ in MPPCA model can be estimated by the well known expectation-maximization (EM) algorithm. [31].

2.2 MMD-Isomap and SSMM-Isomap

MMD-Isomap is a supervised multi-manifold learning method over Isomap [26]. The algorithm uses the global pairwise constraints [32–36] to solve the optimization problem. In the pairwise constraints, some pairs of points are in same class and their relationships are recorded in the must-Link (ML) set. Other pairs of points are in different classes and their relationships are recorded in the cannot-Link (CL) set. The global pairwise ML and CL sets are defined as:

$$ML = \{(x_i, x_j)|i \neq j, l(x_i) = l(x_j)\}, \tag{5}$$

$$CL = \{(x_i, x_j)|i \neq j, l(x_i) \neq l(x_j)\}, \tag{6}$$

where $l(x_i)\epsilon\{1, 2, ..., c\}$ is the class label of $x_i, i = 1, 2, ..., N$, and c is the number of classes. Then, MMD-Isomap aims to preserve the global geometry structures of intra-class data and separate the inter-class data by solving the following two problems:

$$J_{ML} = \min_{Y} \sum\nolimits_{(x_i,x_j)\epsilon ML} (d(y_i, y_j) - d_G(x_i, x_j))^2, \tag{7}$$

$$J_{CL} = \max_{Y} \sum\nolimits_{(x_i,x_j)\epsilon CL} d^2(y_i, y_j) = \|y_i - y_j\|^2. \tag{8}$$

where $d^2(y_i, y_j)$ denotes the Euclidean distance between the low- dimensional representations y_i and y_j , and $d_G(x_i, x_j)$ is the shortest path distance for approximating the geodesic distance. Minimizing Eq. 2) is equivalent to preserving the pairwise distance between $d(y_i, y_j)$ and $d_G(x_i, x_j)$ for the sample pair $(x_i, x_j)\epsilon ML$, and maximizing Eq. 3) aims to separate all the sample pairs $(x_i, x_j)\epsilon CL$. Finally, MMD-Isomap solves the criterion:

$$\begin{aligned} &J = \min_{Y} \sigma(Y), \\ &\sigma(Y) = \frac{1-\alpha}{|ML|} \sum_{(x_i,x_j)\epsilon ML} (d(y_i, y_j) - d_G(x_i, x_j))^2 - \frac{\alpha}{|CL|} \sum_{(x_i,x_j)\epsilon CL} d^2(y_i, y_j), \end{aligned} \tag{9}$$

and $\alpha\epsilon(0, 1)$ is a weighting parameter for trading-off the effects of discrimination over the ML and CL constraints. Note that MMD-Isomap solves Eq. 4) using the Scaling by MAjorizing a COmplicated Function (SMACOF) [37, 38] approach.

SSMM-Isomap is an improvement of MMD-Isomap [27]. First, the objective function in MMD-Isomap is optimized. Then two solution schemes are presented: SMACOF and Eigen-decomposition. The main contribution in SSMM-Isomap is its objective function:

$$\begin{aligned} &J = \min_{Y,P} \sigma(Y, P), \\ &s.t. YY^T = I, \\ &\sigma(Y, P) = \frac{1-\alpha}{|ML|} \sum_{(x_i,x_j)\epsilon ML} (d(y_i, y_j) - d_G(x_i, x_j))^2 - \frac{\alpha}{|CL|} \sum_{(x_i,x_j)\epsilon CL} d^2(y_i, y_j) \\ &+ \beta \sum_{i=1}^{N} \|y_i - \sum_{x_j \in N(x_i)} W_{ij} y_j\|_2^2 + \gamma \sum_{i=1}^{N} \|Px_i - y_i\|_2^2, \end{aligned} \tag{10}$$

The first two items in SSMM-Isomap are the same as in MMD-Isomap. To enable the proposed model to compute low-dimensional local manifold features by using labeled and unlabeled data, the third item is added. The reconstruction weights matrix W of the training samples can be computed by minimizing the following LLE-style optimization problem [2]. To enable the SSMM-Isomap method to learn an explicit projection or feature extractor for handling the external new data, a fourth item is included, which encodes the mismatch between the embedded features using an extractor and the reduced manifold features, so that the learnt extractor P can embed the external new data efficiently by projection [27].

3 Unsupervised multi-manifold discriminant ISOMAP

3.1 Problem formulation

The proposed UMD-Isomap algorithm can be seen as a refinement of MMD-Isomap and SSMM-Isomap. The algorithm not only computes the pairwise constraints ML and CL by training MPPCA "patches" and propagating their local tangent spaces, but also improves the objective functions of MMD-Isomap and SSMM-Isomap. Furthermore, two solutions are proposed to minimize the objective function and obtain low-dimensional embedding results.

It is difficult to determine whether a connected component consists of a single manifold or multiple intersecting manifolds, and how many manifolds exists in the connected component. Though the data are located globally on or close to the multiple smooth nonlinear manifolds, locally, each point and its neighbors are located on a linear patch of the manifold [2, 39]. Moreover, the local tangent space at each point provides a good low-dimensional linear approximation to the local geometric structure of the manifold. As it will become clear shortly, at the intersections of different manifolds, points on the same manifold have similar local tangent spaces whereas points on different manifolds have dissimilar tangent spaces [16].

Generally, the tangent space at each point can be estimated by performing a singular value decomposition (SVD) of $x_1, ...x_n$ [40, 41]. When two points x and y are very close to each other their local tangent spaces are very similar even if they are from different manifolds. This occurs because their local neighborhood based on the Euclidean distance have a large overlap, resulting in similar local covariance matrices. Therefore, this traditional definition of the local tangent space does not work well for intersecting manifolds.

The global nonlinear manifolds can be well-approximated locally by a series of local linear manifolds, and principal component analyzers can successfully cross the intersecting linear manifolds [29]. Moreover, the points approximated by the same linear analyzer usually have similar local tangent spaces which can also be well-approximated by the principal subspace of the local analyzer [16]. Therefore, we can train many local linear analyzers to approximate the underlying manifolds. Subsequently. the local tangent space of a given sample is determined by the principle subspace of its corresponding local analyzer. Thus our UMD-Isomap trains m MPPCA "patches" first before propagating the points in these patches to create the sub-manifolds. At the same time, the points in the same sub-manifolds must be linked, and the points in the different sub-manifolds can not be linked. That is to say, the pairwise constraints ML and CL are created. Subsequently, the optimized objective functions is proposed based on the pairwise constraints, and the two solutions (UMDwithSMACOF and UMDwithEd) represent the final embedding results of the multi-manifold.

It is noteworthy that the number of classes created by the UMD-Isomap algorithm may be higher than that of the real sub-manifolds because the algorithm classifies the dataset according to its topological structure with no consideration of the classification. The number of classes does not have to be determined in advance in our algorithm because it is affected only by the parameter m and θ_0 in MPPCA and the propagation of the tangent subspaces. If the number of classes is known in advance, the proposed algorithm can merge its clustering results using clustering ensemble methods. For example, the MNIST dataset contains thousands of handwritten digit images of 10 digits, and the Coil20 data set contains 1440 images of 20 objects. After running the normal UMD-Isomap algorithm, the number of classes(submanifolds) may be higher than 10(or 20). The UMD-Isomap algorithm creates c sets of ML and CL and accumulates a consensus similarity matrix for a clustering ensemble algorithm. The result of the clustering ensemble algorithm consists of newer pairwise constraints with a fixed number of sub-manifolds.

```
Algorithm 1 (UMD-Isomap): Dimensionality reduction of the multi-manifold.
Input: X, the original multi-manifold dataset; $m$, the number of mixture models; $\alpha$, the
    trade-off parameter of the UMDwithSMACOF and UMDwithEd algorithms; $\theta_0$, the
    threshold of dissimilarity between tangent spaces; $\beta$, the adjusting parameter only for
    UMDwithEd; $\gamma$, the parameter to enlarge the distances between sub-manifolds; $l^*$, the
    number of sub-manifolds; $c^*$, the times of ensemble.
Output: $Y$,the embedding results.
    Step 1: Construct the neighborhood for all $x_i$;
    Step 2: (Construct the pairwise constraints):
    Step 2.1: Train $m$ $d$-dimensional local linear patches by using MPPCA to approximate
    the underlying manifolds;
    Step 2.2: Construct ML and CL by propagating the tangent subpaces (in Algorithm 2);
    Step 3: Re-construct ML and CL if the parameter $l^*$ and $c^*$ exist (in Algorithm 3);
    Step 4: (Compute the embedding results):
    Step 4.1: Construct the object function;
    Step 4.2: Use UMDwithSMACOF(in Algorithm 4) or UMDwithEd(in Algorithm 5) to
    compute the embedding results and visualize them.
```

3.2 Construct the neighborhood

In the UMD-Isomap algorithm (Algorithm 1), the neighborhood of x_i can be computed by the k-nearest neighbor (K-NN) or the dynamical neighborhood algorithm [42]. The K-NN is simple and efficient, whereas the parameter K is not easy to determine, and one global setting may not work well for the entire manifold. The selection of the neighborhood should be data-driven and mainly depends on its local topological structure. The dynamical neighborhood algorithm represents a more accurate neighborhood relationship because it considers the sampling density and manifold curvature. Thus, the neighborhood graph determined by the dynamical neighborhood algorithm improves the accuracy of propagating the sub-manifolds.

3.3 Construct the pairwise constraints

It is difficult to determine whether the points are in the same manifold or not by using only local geometric information. Intuitively, for two points in the same local area, if (a) they are close to each other and (b) have similar tangent spaces, then there is a high probability that they occur on the same manifold. And if they have different local tangent spaces, they are very likely to come from different manifolds [16].

UMD-Isomap defines the dissimilarity between the tangent spaces $\mathbf{T}_i$ and $\mathbf{T}_j$ as:

$$v_{ij} = \sqrt{1 - \sigma^2(U_i^T U_j)}, \tag{11}$$

where U_i and U_j are the basis of the tangent subspace $\mathbf{T}_i$ and $\mathbf{T}_z$, and σ is the minimum singular value of $U_i^T U_j$.

After m patches are trained by MPPCA, if there is a pair of points (x_i, x_j) that are neighbors and exist in different patches p_i and p_j, the patches p_i and p_j are adjacent to each other and are recorded as $\mathbf{N(p_i, p_j)}$. The pair of points (x_i, x_j) is also recorded as the boundary point pairs. Because the points in the "patches" belong to the same sub-manifold, the UMD-Isomap can only compute the dissimilarity between the boundary point pairs.

Before propagating the tangent subspaces of the points in the "patches", the relationship between these patches is defined first:

$$R_{ij} = \begin{cases} 1, & \textit{if } \mathbf{N(p_i, p_j)} \textit{ and } v(p_i, p_j) < \theta_0 \\ -1, & \textit{if } \mathbf{N(p_i, p_j)} \textit{ and } v(p_i, p_j) > \theta_0 \\ 0, & \textit{otherwise.} \end{cases} \tag{12}$$

where $v(p_i, p_j)$ is the dissimilarity between the tangent subspaces of two patches p_i and p_j) , and θ_0 is the threshold of the dissimilarity between the tangent spaces of these patches.

If $R_{ij} = 1$, patches p_i and p_j are neighbors, and it is possible that they exist in the same sub-manifold. If $R_{ij} = -1$, patches p_i and p_j are neighbors, and it is impossible that they exist in the same sub-manifold. That is to say, they exist in the intersecting area. If $R_{ij} = -1$, the patches p_i and p_j are not neighbors.

The first pair of "patches" to be propagated should be located at the point with the minimum local tangent subspace. That is to say,

$$p_{start} = \{(p_i, p_j) | \arg\min(v_{st}), v_{st} < \theta_0, x_s \in p_i, x_t \in p_j, R_{ij} = 1\} \tag{13}$$

After the first pair of "patches" is determined, the next patch is determined as follow:

$$\begin{aligned} p_{next} = \{&p_j | \arg\min(v_{st}), v_{st} < \theta_0, x_s \in p_i, \\ &x_t \in p_j, p_i \in Ped, p_j \notin Ped, R_{ij} \neq -1\}. \end{aligned} \tag{14}$$

where *Ped* is the set of “patches” that had been propagated.

If p_{next} is empty, the propagated points in *Ped* must be linked, because they are in the same sub-manifold. Thus $ML_{Ped,Ped} = 1$. These points cannot be linked to other points, because they are in different sub-manifolds. Thus, $CL_{Ped,nPed} = 1$, where *nPed* is the set of “patches” that are not propagated this time. These “patches” cannot be propagated the next time therefore, their relationships with other “patches” is $R_{ij} = -1$.

```
Algorithm 2 (ConstructPC): construct the pairwise constraints.
Input: X, the original dataset; $m$, the number of mixture models; $\theta_0$, the threshold of dissimilarity between tangent spaces;
Output: $ML$, the must-link sets; $CL$, the cannot-link sets;
    Step 1: Compute the dissimilarity between the boundary point pairs using Eq.(11) and construct the patches relationship matrix using Eq.(12).
    Step 2: Determine the start patch using Eq.(13);
    Step 3: Select the next patch using Eq.(14);
    Step 4: Repeat Step 3 until $p_{next} = \phi$, then the points in the current propagated “patches” creates a sub-manifold. Set $ML_{Ped,Ped} = 1$ , $CL_{Ped,nPed} = 1$ and $R_{ped,nPed} = -1$.
    Step 5: Repeat Step 2 until all patches are propagated, then the propagation process is ended.
```

3.4 Re-construct the pairwise constraints

The $s_{ij} \in ML$ indicates whether the pair of points x_i and x_j is in the same class or not. After running MPPCA and ConstructPC algorithms c^* times, the consensus similarity matrix *M* is constructed by accumulating c^* sets of ML. The $m_{ij} \in M$ denotes the number of the pair of points x_i and x_j in the same class. The ensemble clustering can be derived from *M* to obtain l^* classes in a number of ways, such as using hierarchical clustering or spectral clustering algorithms. Then the new pairwise constraints are re-constructed according to the l^* new classes.

```
Algorithm 3 (ReConstructPC): reconstruct the pairwise constraints.
Input: X, the original dataset; $m$, the number of mixture models; $\theta_0$, the threshold of dissimilarity between tangent spaces; $l^*$, the number of sub-manifolds; $c^*$, the times of ensemble.
Output: $ML$, the must-link sets; $CL$, the cannot-link sets;
    Step 1: Run MPPCA and ConstructPC algorithms $c^*$ times;
    Step 2: Accumulate $c^*$ sets of ML to compute a consensus similarity matrix $M$;
    Step 3: Form the ensemble clustering based on the similarity matrix $M$, and the number of classes is $l^*$;
    Step 4: Re-construct the pairwise constraints ML and CL according to the new classes.
```

3.5 Compute the Embedding Results

In order to tackle the dimensionality reduction of multi-manifold, MMD-Isomap defines the objective function as expressed in Eq. 4). The objective function attempts to preserve the global geometrical structures of the intra-class data points, therefore $d(y_i, y_j)$ is as close as possible to $d_G(x_i, x_j)$. Meanwhile, the objective function aims to separate the interclass data points ensuring that the distance between them is as large as possible. However, a question remains: how large are the distances between the interclass data points? The item $\sum_{(x_i,x_j)\epsilon CL} d^2(y_i, y_j)$ maybe very large due to the dimension of the dataset, but the item $\sum_{(x_i,x_j)\epsilon ML} (d(y_i, y_j) - d_G(x_i, x_j))^2$ maybe controllable. Therefore, it is difficult to set the threshold of $\sigma(Y)$ and to select a suitable parameter α. To accommodate this uncertainty, our proposed UMD-Isomap defines the following objective function:

$$
\begin{aligned}
J &= \min_{Y} \sigma(Y),\\
\sigma(Y) &= \frac{1-\alpha}{|ML|} \sum_{(x_i,x_j)\epsilon ML} (d(y_i, y_j) - d_G(x_i, x_j))^2 \\
&\quad - \frac{\alpha}{|CL|} \sum_{(x_i,x_j)\epsilon CL} (d(y_i, y_j) - C)^2.
\end{aligned}
\tag{15}
$$

where $C = \gamma * max(d_G(x_i, x_j))$, $max(d_G(x_i, x_j))$ represents the maximum distance in sub-manifolds and γ is an adjustment parameter to increase the distances between sub-manifolds. Thus, c is a constant that determines the distances between the interclass data points. Then, we can scale by majorizing a convex function (SMACOF) and eigen-decomposition techniques respectively to solve the above-mentioned problem.

3.5.1 Effective solution using SMACOF

Despite the slow convergence of the SMACOF algorithm, a large number of iterations may be required for achieving high-accuracy, depending on the size of the data set and the used metric [26].

$$\begin{aligned}\sigma(Y) &= \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\epsilon ML}(d(y_i,y_j)-d_G(x_i,x_j))^2\\ &\quad-\frac{\alpha}{|CL|}\sum_{(x_i,x_j)\epsilon CL}(d(y_i,y_j)-C)^2\\ &= \eta^2_{con}+\eta^2(Y)-2\rho(Y)\end{aligned}\tag{16}$$

where

$$\eta^2_{con} = \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\epsilon ML} d^2_G(x_i,x_j)-\alpha C^2,\tag{17}$$

$$\eta^2(Y) = \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\epsilon ML} d^2(y_i,y_j)-\frac{\alpha}{|CL|}\sum_{(x_i,x_j)\epsilon CL} d^2(y_i,y_j),\tag{18}$$

$$\rho(Y) = \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\epsilon ML} d(y_i,y_j)d_G(x_i,x_j)-\frac{\alpha C}{|CL|}\sum_{(x_i,x_j)\epsilon ML} d(y_i,y_j)\mathbf{1}.\tag{19}$$

Let

$$w^{ML}_{ij} = \begin{cases}\frac{1-\alpha}{|ML|}, & if(x_i,x_j)\in ML\\ 0, & otherwise\end{cases}, w^{CL}_{ij} = \begin{cases}\frac{\alpha}{|CL|}, & if(x_i,x_j)\in CL\\ 0, & otherwise\end{cases}.\tag{20}$$

then Eq. 18) can be rewritten as

$$\begin{aligned}\eta^2(Y) &= \sum_{i,j=1,i\neq j}^{N} w^{ML}_{ij}d^2(y_i,y_j)-\sum_{i,j=1,i\neq j}^{N} w^{CL}_{ij}d^2(y_i,y_j)\\ &= tr(Y(\sum_{i,j=1,i\neq j}^{N} w^{ML}_{ij}F^{ij})Y^T)-tr(Y(\sum_{i,j=1,i\neq j}^{N} w^{CL}_{ij}F^{ij})Y^T)\\ &= tr(YVY^T)\end{aligned}\tag{21}$$

where

$$V = V_{ML}-V_{CL}, V_{ML} = \sum_{i,j=1,i\neq j}^{N} w^{ML}_{ij}F^{ij}, V_{CL} = \sum_{i,j=1,i\neq j}^{N} w^{CL}_{ij}F^{ij},\tag{22}$$

and F_{ij} is a $N \times N$ matrix with elements $F^{ij}_{ii} = F^{ij}_{jj} = 1$, $F^{ij}_{ij} = F^{ij}_{ji} = -1$, and all other elements zero.

Now, consider Eq. 20). Let

$$wb^{ML}_{ij} = \begin{cases}\frac{(1-\alpha)d_G(x_i,x_j)}{|ML|d(z_i,z_j)}, & if(x_i,x_j)\in ML, d(z_i,z_j)\neq 0\\ 0, & otherwise\end{cases},\tag{23}$$

$$wb^{CL}_{ij} = \begin{cases}\frac{\alpha C\mathbf{1}}{|CL|d(z_i,z_j)}, & if(x_i,x_j)\in CL, d(z_i,z_j)\neq 0\\ 0, & otherwise\end{cases}.\tag{24}$$

Then we have

$$\begin{aligned}\rho(Y) &= \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\epsilon ML} d(y_i,y_j)d_G(x_i,x_j)\\ &\quad-\frac{\alpha C}{|CL|}\sum_{(x_i,x_j)\epsilon ML} d(y_i,y_j)\mathbf{1}\\ &= \sum_{\substack{(x_i,x_j)\epsilon ML\\ d(z_i,z_j)\neq 0}} wb^{ML}_{ij}d(y_i,y_j)d(z_i,z_j)\\ &\quad-\sum_{\substack{(x_i,x_j)\epsilon CL\\ d(z_i,z_j)\neq 0}} wb^{CL}_{ij}d(y_i,y_j)d(z_i,z_j).\end{aligned}\tag{25}$$

If we define

$$B(Z) = \sum_{i,j=1\ i\neq j}^{N} wb^{ML}_{ij}F^{ij}-\sum_{i,j=1\ i\neq j}^{N} wb^{CL}_{ij}F^{ij},\tag{26}$$

the Cauchy–Schwartz inequality implies that for all pairs of configurations Y and Z, we have $\rho(Y) \geq tr(YB(Z)Z)$. Then Eq. (16) can be rewritten as

$$\begin{aligned}\sigma(Y) &= \eta^2_{con}+\eta^2(Y)-2\rho(Y)\\ &\leq \eta^2_{con}+tr(YVY^T)-2tr(YB(Z)Z^T).\end{aligned}\tag{27}$$

If we define $\tau(Y,Z) = \eta^2_{con}+tr(YVY^T)-2tr(YB(Z)Z^T)$, then $\tau(Y,Z)$ is an auxiliary function of $\sigma(Y)$. The Y value for $\tau(Y,Z)$ attaining its minimum, can be calculated by setting the partial derivative equal to zero. So that $Y = ZB(Z)V^{-1}$.

The solution using SMACOF is described in the UMD-withSMACOF algorithm.

```
Algorithm 4 (UMDwithSMACOF): dimensionality reduction using SMACOF.
Input: $X$, the original dataset; $ML$, the Must-Link matrix; $CL$, the Cannot-Link matrix;
Output: $Y$,the embedding result;
  Step 1: Compute the geodesic distances $d_G(x_i, x_j)$ where $ML_{ij} = 1$;
  Step 2: Construct matrix $V$ using Eq.(22) and compute $V^{-1}$;
  Step 3: Produce a random matrix $y^0$, initialize $Z = Y^0$, then calculate $\sigma(Y^0)$ by Eq.(16)
  and set step $t = 0$;
  Step 4: Compute $B(Z)$ using Eq.(26), then set $Y^{t+1} = ZBV^{-1}$;
  Step 5: Compute $\sigma(Y^{t+1})$ using Eq.(16), if $\sigma(Y^t) - \sigma(Y^{t+1}) < \xi$, then stop; else $Z =$
  $Y^{t+1}, t = t + 1$ and go to Step 4.
```

3.5.2 Effective Solution using Eigen-decomposition

It is also feasible to solve the objective function in Eq. (16) using the eigen-decomposition technique. To facilitate the optimization, we first transform Eq. (16) into the following trace form-based expressions:

$$\begin{aligned}\sigma(Y) &= \frac{1-\alpha}{|ML|}\sum_{(x_i,x_j)\in ML}(d(y_i,y_j) - d_G(x_i,x_j))^2 - \frac{\alpha}{|CL|}\sum_{(x_i,x_j)\in CL}(d(y_i,y_j) - C)^2\\ &= tr(YR^{ML}Y^T) - tr(YR^{CL}Y^T)\\ &= tr(Y(R^{ML} - R^{CL})Y^T).\end{aligned} \tag{28}$$

where

$$R^{ML} = -\frac{1-\alpha}{2|ML|}(H(Q^{ML})H), \tag{29}$$

$$R^{CL} = -\frac{\alpha C}{2|CL|}(H(Q^{CL})H), \tag{30}$$

$$Q^{ML}_{ij} = d^{ML}_G(x_i, x_j), \tag{31}$$

$$Q^{CL}_{ij} = \begin{cases} 1, if(x_i, x_j) \in CL \\ 0, otherwise \end{cases}, \tag{32}$$

and $H = I - \mathbf{1}\mathbf{1}^T/N$ is a centering matrix, $\mathbf{1}$ denotes a vector of all ones.

It should be noted that HQH is used to solve the objective function $E = \sum((d(y_i, y_j) - d_G(x_i, x_j))^2)$ in MDS [43] and Isomap [1]. Therefore, a solution using the eigen-decomposition of $\sigma(Y)$ only reflects the distances between the sub-manifolds but does not reflect their topological structure. In addition, the iteration process $\sum_{(x_i,x_j)\in ML}(d(y_i, y_j) - d_G(x_i, x_j))^2$ in the solution using the SMACOF algorithm results in the sub-manifolds with better low-dimensional embedding. In order to optimize the solution with the eigen-decomposition and obtain accurate embedding results, the LLE-style item described in [2, 13] is added to the proposed objective function:

$$\beta\sum_{i=1}^{N}\|y_i - \sum_{x_j\in N(x_i)} W_{ij}y_j\|^2 = \beta tr(Y(I - W)(I - W)^T Y^T) = tr(YMY^T) \tag{33}$$

where β is a trade-off parameter, and $M = \beta(I - W)(I - W)^T$ describes the pairwise local weighting relationships. Subsequently, the new objective function expressed in Eq. (16) can be rewritten as

$$\sigma(Y) = tr(Y(R^{ML} - R^{CL} + M)Y^T). \tag{34}$$

Letting

$$V = R^{ML} - R^{CL} + M, \tag{35}$$

we can obtain the embedding results $Y = [\sqrt{\lambda_1}v_1, \sqrt{\lambda_2}v_2, ..., \sqrt{\lambda_d}v_d]^T$, where $v_i, i = 1, 2, ..., d$, are the standard eigenvectors corresponding to first d leading eigenvalues of the matrix V.

```
Algorithm 5 (UMDwithED): dimensionality reduction using eigen-decomposition.
Input: $X$, the original dataset; $\{\ell_1, ...\ell_n\}$, the set of labels which are the classification result
  of the classifyMtoS algorithm(Algorithm 1);
Output: $Y$,the embedding result;
  Step 1: Compute the geodesic distances $d_G(x_i, x_j)$ where $ML_{ij} = 1$;
  Step 2: Construct matrix $V = R^{ML} - R^{CL} + M$ by Eq.(35);
  step 3: Eigen-decompose $V$;
  Step 4: Obtain $Y = [\sqrt{\lambda_1}v_1, \sqrt{\lambda_2}v_2, ..., \sqrt{\lambda_d}v_d]^T$, $i = 1, 2, ..., d$;
```

3.6 Complexity analysis

In the proposed UMD-Isomap algorithm, the process to compute the pairwise constraints is achieved by MPPCA and ConstructPC (in Algorithm 2), and the process to dimensionality reduction is achieved by two algorithms: UMDwithSMACOF (in Algorithm 4) or UMDwithEd (in Algorithm 5).

The complexity of MPPCA is $O(mNDdt_1)$, where m denotes the number of "patches" trained by MPPCA, and t_1 represents the number of iterations. The complexity of ConstructPC is $O(\sum_{i=1}^{m} t_i)$, where t_i denotes only the time required to compute the dissimilarity of tangent subspaces between the points in "Ped" set and their neighbors in "nPed" set. Compared with $O(mNDdt_1)$, the time complexity of ConstructPC is smaller. Hence, the complexity of computing the pairwise constraints can be approximated as $O(mNDdt_1)$.

In the dimensionality reduction process, the geodesic distances between each pair of points in the ML set are computed by using Dijkstra's algorithm. Then the embedding results of the original samples are computed using SMACOF (in Algorithm 4) or eigen-decomposition (in Algorithm 5), respectively. The complexity of Dijkstra's algorithm is $O(N^2)$. therefore, the complexity of computing the geodesic distances matrix is $O(\sum_{sM_i} N_{sM_i}^2)$, where sM_i represents the sub-manifolds determined by the pairwise constraints ML and CL. The complexity of SMACOF is $O(t_2 dN^2)$, where t_2 is the number of iterations. Compared to the computation the geodesic distances matrix and the iterative calculation of the embedding results using SMACOF, the complexity of creating the pairwise constraints and he iterative calculation of the embedding using the eigen-decomposition is very small. Therefore, the complexities of Algorithm 4 and Algorithm 5 are approximately $O(\sum_{sM_i} N_{sM_i}^2 + t_2 dN^2)$ and $O(\sum_{sM_i} N_{sM_i}^2)$, respectively.

4 Experiments

In this section, the performance of the proposed UMD-Isomap is assessed using a series of synthetic and real-world datasets. First, computation of the pairwise constraints is analyzed using four artificial datasets and three real datasets. Because the pairwise constraints reflect the clustering results of the sub-manifolds, the clustering accuracy is used as the evaluation criterion to assess the performance of computing the pairwise constraints. Subsequently, two approaches for dimensionality reduction (i.e., UMDwithEd and UMDwithSMACOF) in the proposed UMD-Isomap are evaluated using two artificial datasets and three real datasets. The two-dimensional(2D) embedding results are visualized to assess the quality. After that, the influence of the parameters and the results of the time complexity analysis are also discussed.

All of the experiments were performed on a PC with an Intel Core based system with 3.6 GHz CPU and 8 GB RAM and using the Matlab platform.

4.1 Clustering analysis

Clearly, the higher the classification accuracy is, the better the algorithm performance is. In experiments, we evaluate the clustering performance with two standard clustering evaluation metrics, i.e. Clustering Accuracy (ACC) and Normalized Mutual Information (NMI). The clustering accuracy is defined as the maximum classification accuracy among all possible alignments:

$$ACC = \max_{align} \sum_{i=1}^{N} \delta(t_i = l_i)/N, \tag{36}$$

where t_i is the true label, l_i is the obtained label of x_i, and $\delta(\cdot)$ is the delta function. And the Normalized Mutual Information measures the similarity between the clustering results and the true classes:

$$NMI = \frac{\sum_{i=1}^{l^*} \sum_{j=1}^{l^*} p_{ij} log(\frac{p_{ij}}{p_i p_j})}{\sqrt{\sum_{i=1}^{l^*} p_i log p_i \sum_{j=1}^{l^*} p_j log p_j}}, \tag{37}$$

where l^* is the number of sub-manifolds (or classes), p_i is the probabilities that the sample belongs to the ith cluster, and p_{ij} is the joint probability that the sample belongs to the both the ith cluster and the jth cluster.

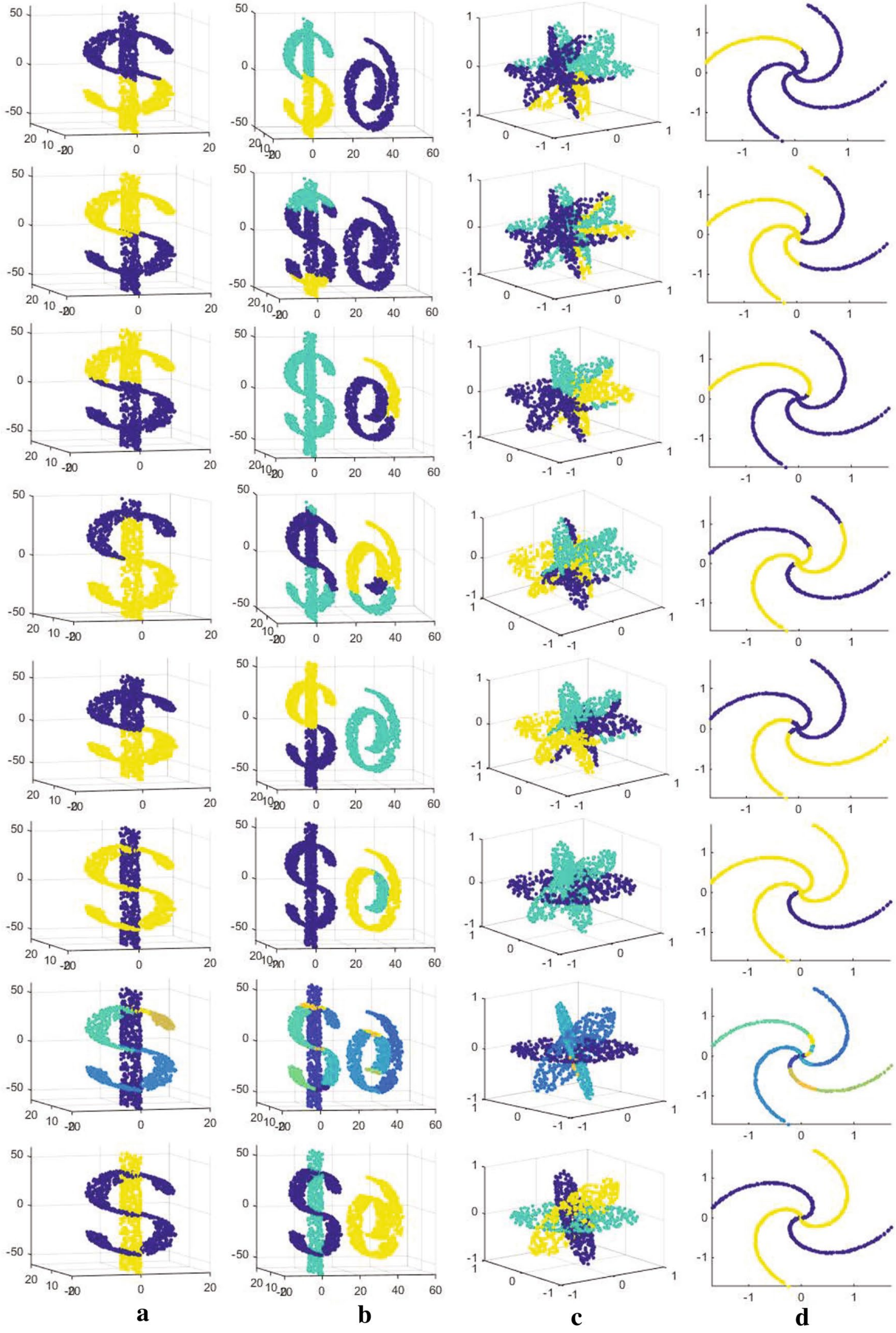


Fig. 1 The classification results of Isomap, K-manifolds, Ncut, t-SNE, Umap, SMMC and two versions of UMD-Isomap on four artificial data sets, the different colors represent the different sub-manifolds. **a** $ data; **b** Hybrid data; **c** three planes; **d** two spirals

Table 1 The Classification accuracy and Normal Mutual Information (mean ± std

	$ data		Hybrid data		Three-planes		Two-spirals	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
ISOMAP	0.667±0.0	0.0±0.0	0.800±0.0	0.799±0.0	0.438±0.0	0.125±0.0	0.627±0.0	0.191±0.0
	(0.667)	(0.0)	(0.800)	(0.799)	(0.438)	(0.125)	(0.627)	(0.191)
K-manifolds	0.667±0.0	0.0±0.0	0.559±0.021	0.214±0.027	0.351±0.076	0.017±0.0	0.524±0.013	0.002±0.0
	(0.667)	(0.0)	(0.568)	(0.223)	(0.353)	(0.0176)	(0.525)	(0.002)
Ncut	0.632±0.132	0.0±0.0	0.461±0.071	0.799±0.003	0.466±0.053	0.085±0.002	0.696±0.044	0.192±0.013
	(0.667)	(0.0)	(0.523)	(0.801	(0.498)	(0.087)	(0.743)	(0.195)
t-SNE	0.667±0.0	0.0±0.0	0.655±0.0	0.395±0.0	0.433±0.0	0.085±0.0	0.881±0.0	0.003±0.0
	(0.667)	(0.0)	(0.655)	(0.395)	(0.433)	(0.085)	(0.881)	(0.003)
Umap	0.654±0.0	0.0±0.0	0.691±0.0	0.402±0.0	0.441±0.0	0.087±0.0	0.764±0.0	0.002±0.0
	(0.654)	(0.0)	(0.691)	(0.402)	(0.441)	(0.087)	(0.764)	(0.002)
SMMC	0.978±0.044	0.986±0.0	0.92±0.081	0.892±0.015	0.804±0.017	0.639±0.021	0.739±0.020	0.325±0.014
	(1.0)	(0.986)	(0.925)	(0.901)	(0.867)	(0.654)	(0.829)	(0.336)
ConstructPC	**0.998±0.001**	**0.980±0.0**	**0.959±0.079**	**1.0±0.0**	**0.868±0.028**	**0.875±0.0**	**0.991±0.039**	**0.943±0.0**
	(1.0)	**(1.0)**	**(1.0)**	**(1.0)**	**(0.984)**	**(0.875)**	**(0.992)**	**(0.943)**
ReConstructPC	**1.0±0.0**	**0.986±0.061**	**0.987±0.061**	**1.0±0.0**	**0.978±0.311**	**0.885±0.0**	**0.993±0.012**	**0.930±0.0**
(c^*=5)	**(1.0)**	**(1.0)**	**(1.0)**	**(1.0)**	**(0.983)**	**(0.885)**	**(0.994)**	**(0.855)**

Followed by the highest values) for different methods using four data sets

Optimal experimental results and our experimental results are given in bold

Table 2 Classification accuracies (mean ± std. Followed by the highest accuracy) and computation time(s) for different methods using three data sets

	Coil20(1440×1024)		Mnist(10000×784)		USPS(9289×256)	
	ACC	NMI	ACC	NMI	ACC	NMI
ISOMAP	0.523±0.014	0.613±0.004	0.494±0.021	0.445±0.007	0.647±0.005	0.578±0.0
	(0.528)	(0.618)	(0.504)	(0.447)	(0.698)	(0.578)
K-manifolds	0.276±0.027	0.491±0.016	0.208±0.076	0.202±0.011	0.294±0.065	0.269±0.012
	(0.303)	(0.506)	(0.249)	(0.214)	(0.317)	(0.271)
Ncut	**0.874±0.044**	**0.934±0.052**	0.756±0.013	0.752±0.032	0.312±0.004	0.479±0.063
	(0.882)	**(0.957)**	(0.769)	(0.777)	(0.357)	(0.503)
t-SNE	0.798±0.0	0.889±0.0	0.819±0.0	0.776±0.0	0.873±0.0	0.893±0.0
	(0.798)	(0.889)	(0.819)	(0.776)	(0.873)	(0.893)
Umap	0.815±0.0	0.879±0.0	0.793±0.0	0.811±0.0	**0.914±0.0**	**0.903±0.0**
	(0.815)	(0.879)	(0.793)	(0.811)	**(0.914)**	**(0.903)**
SMMC	0.292±0.036	0.418±0.031	0.171±0.011	0.166±0.011	0.226±0.187	0.129±0.011
	(0.344)	(0.433)	(0.186)	(0.184)	(0.2276)	(0.138)
ConstructPC	**0.872±0.007**	**0.887±0.012**	**0.819±0.017**	**0.746±0.031**	**0.875±0.012**	**0.794±0.026**
	(0.882)	**(0.892)**	**(0.836)**	**(0.753)**	**(0.894)**	**(0.811)**
ReConstructPC	**0.788±0.013**	**0.750±0.014**	**0.762±0.011**	**0.641±0.007**	**0.844±0.019**	**0.784±0.022**
(c^*=5)	**(0.791)**	**(0.774)**	**(0.787)**	**(0.657)**	**(0.869)**	**(0.798)**

Optimal experimental results and our experimental results are given in bold

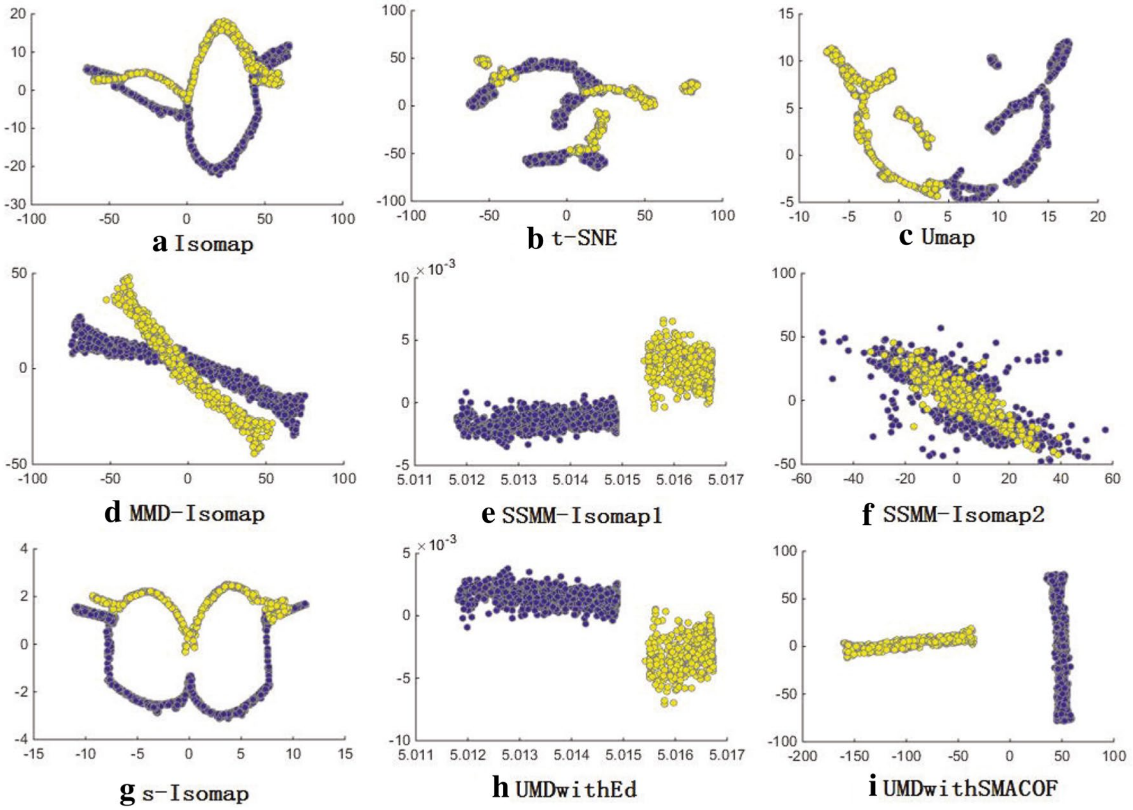
In this subsection, the computation results of the pairwise constraints in our UMD-Isomap are compared with those of several other methods, (i.e., the k-means on Isomap, *K*-manifolds [10], Ncut [14], t-SNE [19], Umap [20][1] and spctral multi-manifold clustering (SMMC) [16]). Neither Isomap nor *K*-manifolds performs well on intersecting nonlinear manifolds. However, since these methods are quite well-known, and in order to confirm that the proposed algorithm outperforms them, these two methods were included in the comparison.

[1] The experiment is carried out with the python version of Umap algorithm, and its results in figures are displayed by MATLAB.

Table 3 The running time(s) for different methods on seven data sets

	$ data	Hybrid data	Three-planes	Two-spirals	Coil20	Mnist	USPS
ISOMAP	26.862	122.695	14.422	7.722	23.802	8.157e+3	6.968e+3
K-manifolds	82.815	2.216e+3	41.385	25.946	122.227	1.629e+4	1.490e+4
Ncut	4.454	4.467	1.779	0.933	5.226	1.155e+3	867.997
t-SNE	55.181	161.486	32.066	20.357	47.651	2.405e+3	2.401e+3
Umap	—	—	—	—	—	—	—
SMMC	4.005	4.212	3.863	1.866	596.833	3.613e+3	1.725e+3
ConstructPC	**1.648**	**1.816**	**1.162**	**0.399**	**329.929**	**735.562**	**169.073**
ReConstructPC(c^*=5)	**5.825**	**10.596**	**5.396**	**2.398**	**2.112e+3**	**4.026e+3**	**1.128e+3**

Optimal experimental results and our experimental results are given in bold



a Isomap **b** t-SNE **c** Umap

d MMD-Isomap **e** SSMM-Isomap1 **f** SSMM-Isomap2

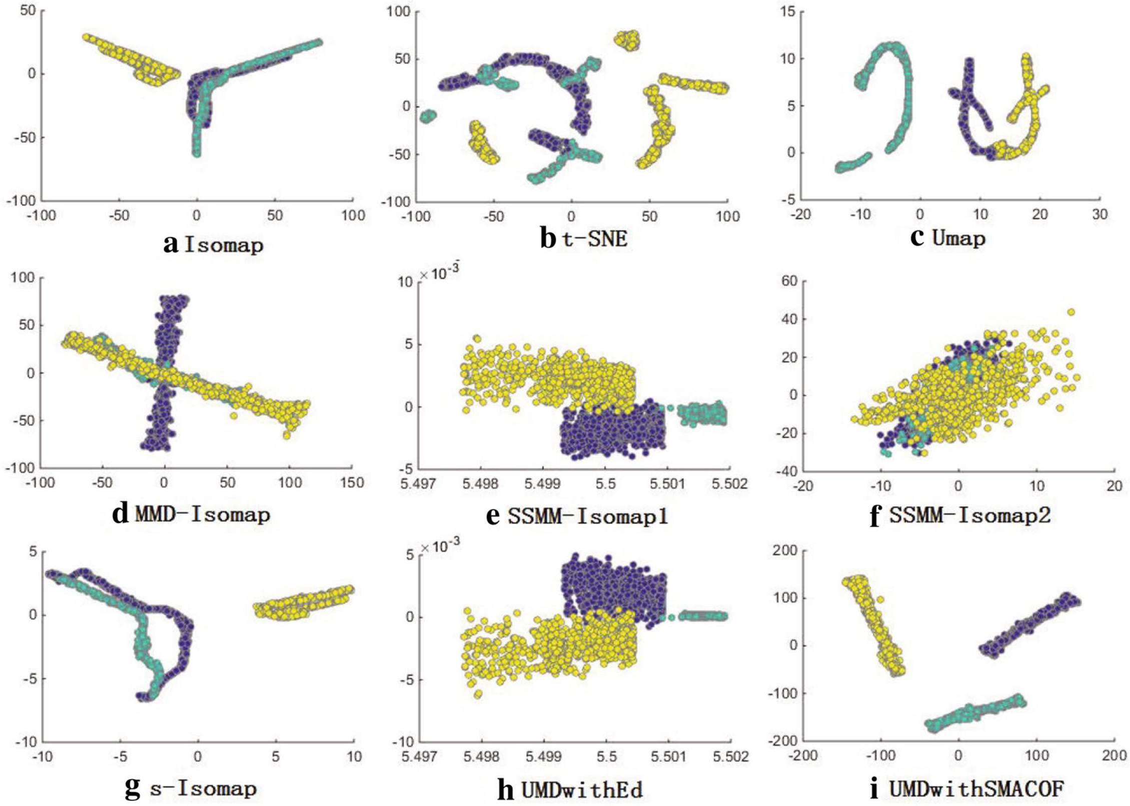
g s-Isomap **h** UMDwithEd **i** UMDwithSMACOF

Fig. 2 The 2D embedding results obtained by each method on the $ dataset

Note that the clustering performance of UMD-Isomap is represented by the results of the ConstructPC or ReConstructPC algorithms, without considering the clustering results of the low-dimensional embedding for the proposed UMD-Isomap. Some supervised and semi-supervised algorithms always use k-means to test the clustering or classification results of the low-dimensional embedding. Although the accuracy of clustering is high, the result only reflects dimensionality reduction performance but not the clustering performances because the supervised and semi-supervised algorithms use a large number of labels and cannot classify the multi-manifold themselves.

Four artificial datasets with different characteristics and complexities are selected and the clustering results are shown in Fig. 1 and Table 1. As illustrated in Fig. 1, the different colors denote different classes. In Table 1, the average clustering accuracy (ACC) and Normalized Mutual Information (NMI) of 10 independent runs the standard deviations, and the maximum values for the different methods are tabulated.

The following is observed: (1) The Isomap, *K*-manifolds, Ncut, t-SNE and Umap methods did not perform well on intersecting multi-manifolds. (2) The SMMC not only resulted in satisfactory performance for grouping



a Isomap **b** t-SNE **c** Umap

d MMD-Isomap **e** SSMM-Isomap1 **f** SSMM-Isomap2

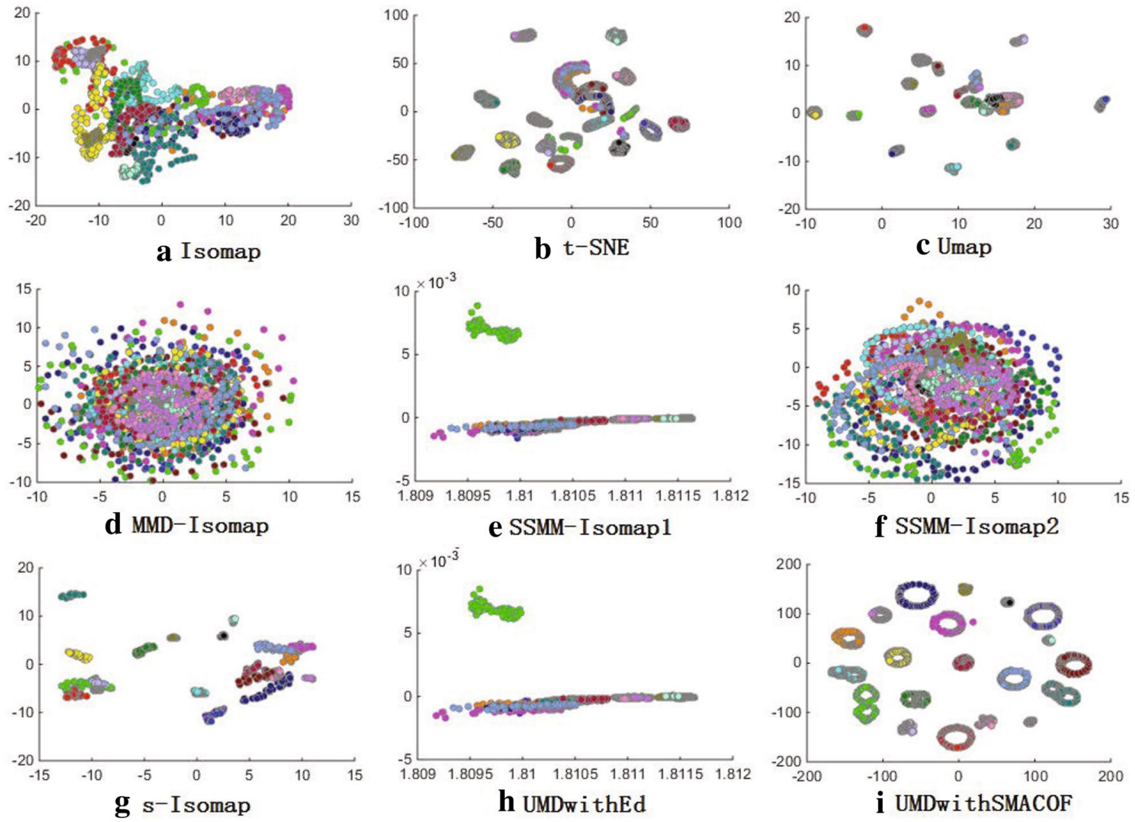
g s-Isomap **h** UMDwithEd **i** UMDwithSMACOF

Fig. 3 The 2D embedding results obtained by each method on the Hybrid dataset

samples generated from intersecting manifolds, but also performed well on the more general hybrid nonlinear manifold clustering problems. However, its results were not sufficiently robust because of the unstable results determined by MPPCA. (3) Both versions of UMD-Isomap resulted in excellent clustering accuracy and normal mutual information. However, the ConstructPC (in UMD-Isomap) algorithm is also not sufficiently robust for the same reason as the SMMC, and the number of its classes are created more than the real one(see the pictures of the sixth line in Fig. 1). (5) The results of the ReConstructPC (in UMD-Isomap) algorithm are more robust because it ensembles $c^* = 5$ sets of ML and computes the new clustering results with the parameter l^*.

Three widely-known real-world datasets are also selected to evaluate the efficiency and accuracy of the different algorithms. The first one is the Coil20 dataset, which consists of 1440 images of 20 objects, and each object is viewed under a full 360° of rotation. The MNIST dataset contains 10000 handwritten digit images ('0' to '9'). The USPS data set consists of 9289 handwritten digit images('0' to '9'). The clustering results of the real-world datasets cannot be visualized in $R2$ or $R3$ spaces. Hence, we only compared the clustering accuracies (ACC) and Normalized Mutual Information (NMI) for the seven algorithms on the Coil20, MNIST, and USPS datasets (in Table 2).

The experimental results revealed a number of significant achievements: (1) The clustering results of t-SNE and Umap on three real-world datasets are desirable, which proves the "crowding problems" in high-dimensional space is not as serious as that in low-dimensional space. And the clustering results of Umap on USPS is best in seven algorithms; (2) The clustering results of Ncut on Coil20 is best in seven algorithms, because the sub-manifolds in Coil20 are hardly intersect. But its clustering results on USPS is worse than those of t-SNE and our two algorithms, because the USPS dataset has more samples and lower dimensions than other two real-world datasets; (3) The SMMC algorithm is not appropriate for real-world datasets because its accuracy are even lower than those of the Isomap, k-manifolds and Ncut algorithms. (4) The clustering accuracy is higher for ConstructPC than for ReConstructPC because the number of classes created by the ConstructPC algorithm is more than the real one. The influence of the classed number on the clustering accuracy will be discussed in Sect. 4.3.1. (5) the clustering results are better for the two algorithms in our proposed UMD-Isomap than for the other algorithms, demonstrating the effectiveness of the proposed algorithm.



a Isomap **b** t-SNE **c** Umap

d MMD-Isomap **e** SSMM-Isomap1 **f** SSMM-Isomap2

g s-Isomap **h** UMDwithEd **i** UMDwithSMACOF

Fig. 4 The 2D embedding results obtained by each method on the Coil20 dataset

In addition, the average computation times are shown in Table 3, demonstrating a rough indication of their time complexity. (1)The running time of the ConstructPC algorithm in the UMD-Isomap is less than those of other algorithms. However, it is still a little higher than that of the Ncut algorithm for the Coil20 data set because the time complexity of the ConstructPC algorithm is not only related to the number of samples but also the dimensionality of the datasets; (2) The running time of the ReConstructPC algorithm is larger because it computes the similarity matrix by running c^* sets of ConstructPC algorithm; (3) The proposed method is more efficient than the other method, which is also confirmed by the time complexity results.

In brief, the experiments on both the artificial datasets and the real-world datasets verify the effectiveness of the proposed algorithm.

4.2 Dimensionality reduction and visualization

Herein, two solutions for dimensionality reduction process (i.e., UMDwithEd and UMDwithSMACOF) in the proposed UMD-Isomap are compared with Isomap [1], t-SNE [19], Umap [20], MMD-Isomap [26], SSMM-Isomap [27] algorithms and S-Isomap [23].

The 2D embedding results of $ and Hybrid datasets are visualized in Figs. 2 and 3, and those of Coil20, Mnist and USPS datasets are displayed in Figs. 4, 5 and 6. It was revealed that (1) Isomap cannot be applied to multi-manifolds, and the embeddings of sub-manifolds mix together; (2) t-SNE and Umap can construct better embedding for each sub-manifold for the high-dimension real-world manifolds, but it is not suitable for low-dimension intersect manifolds; (3) MMD-Isomap and SSMM-Isomap-2 can compute accurate embeddings for each sub-manifold. However, these embeddings overlap together; (4) S-Isomap can compute low-dimensional embeddings of multi-manifolds and visualize them as well, but the embedding for each sub-manifold is not accurate; (5) the proposed UMDwithEd algorithm achieves similar results with SSMM-Isomap-1, although they have different objective functions, which reflects that the solution based on an eigen-decomposition technique maybe not proper for multi-manifolds datasets; (6) the proposed UMDwithSMACOF algorithm not only indicates the most accurate embedding results, but also visualizes sub-manifolds separately. For example, Fig. 4(i) shows a two dimensional representation of Coil20 dataset discovered by UMDwithSMACOF, and the embedding of each object is

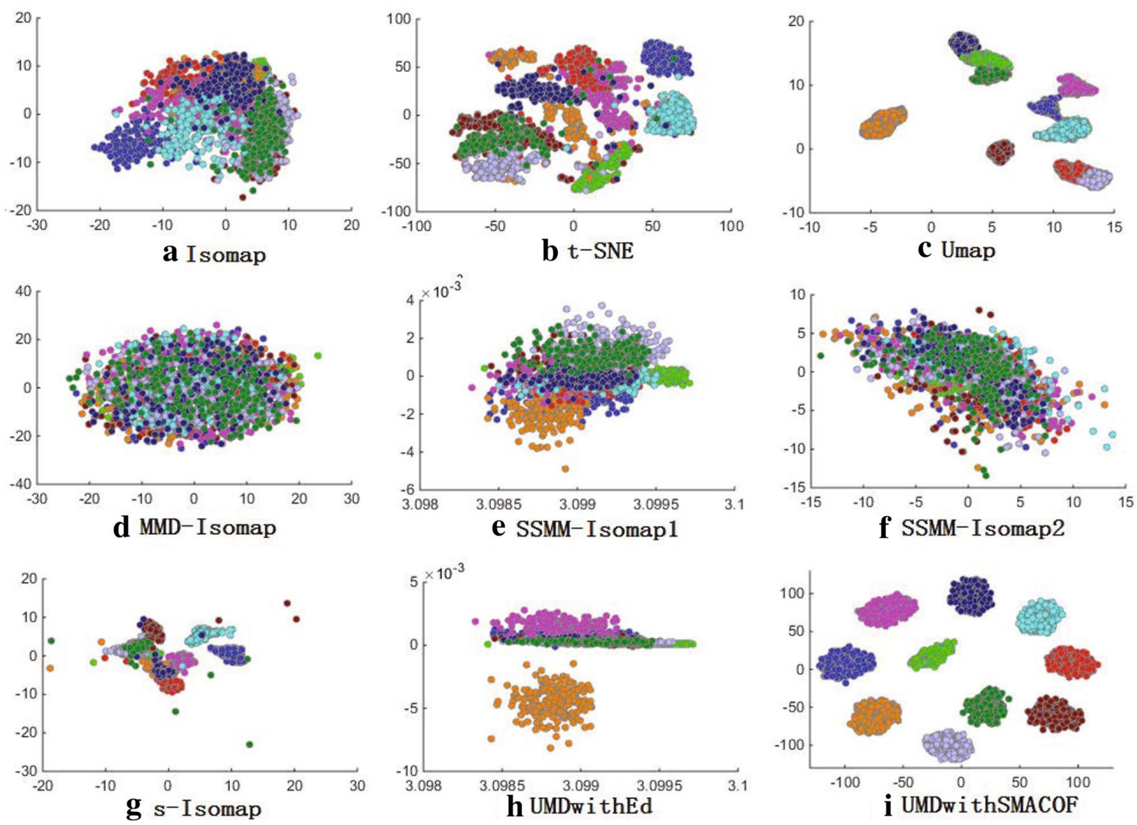


Fig. 5 The 2D embedding results obtained by each method on the Mnist dataset

visualized in a circle, which reflects the underlying rotational degree of freedom.

The time of running eight algorithms on five data sets is shown in Table 4. Based on the results it can be concluded that: (1)the running times of UMDwithSMACOF, SSMM-Isomap2 and MMD-Isomap algorithms are longer, because these algorithms all depend on iterations to reach objective functions; (2)the UMDwithEd algorithm has a better time complexity than other algorithms, although UMDwithSMACOF can achieve a more accurate embedding. Also, the running time on three real-world datasets in UMDwithEd is less than others obviously, because these datasets have more sub-manifolds and there is a smaller number of samples in each sub-manifold, which verifies the analysis of time complexity.

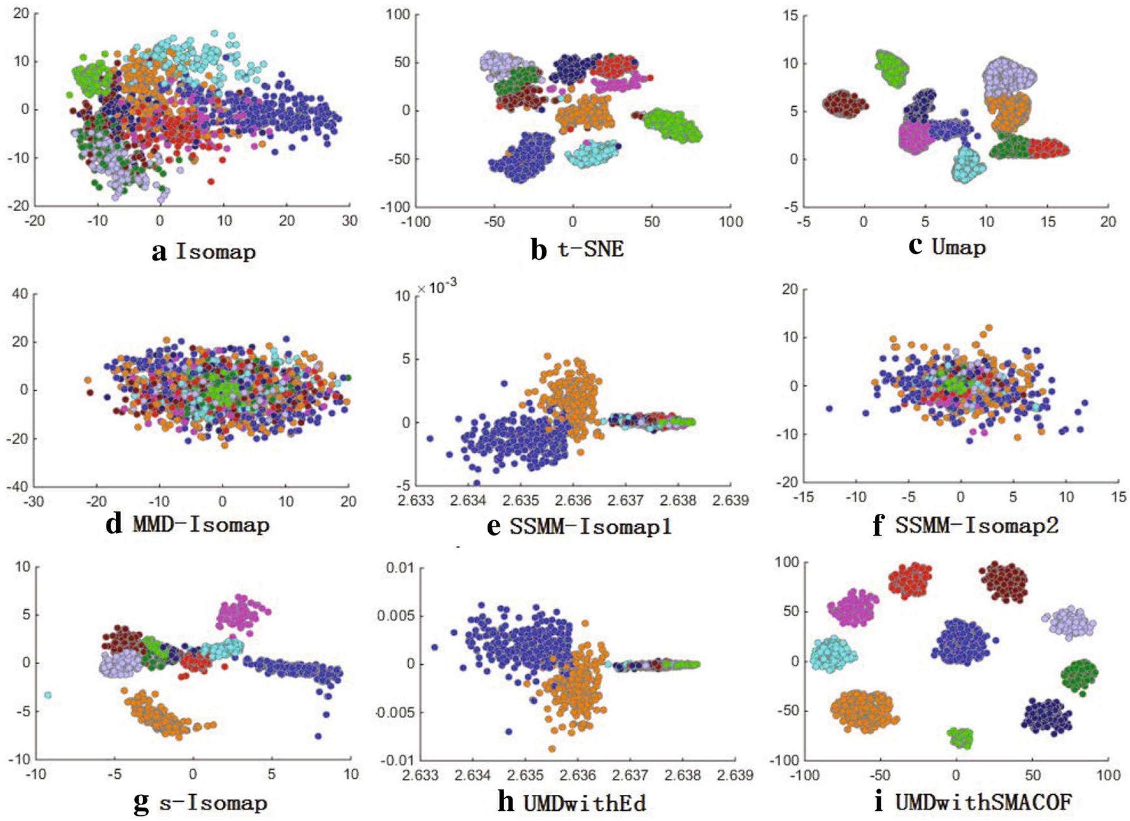
4.3 Discussion on the parameters

In the UMD-Isomap, m and θ_0 determine the clustering accuracy of the ConstructPC and ReConstructPC algorithms and the number of classes constructed by ConstructPC. At the same time, the classes number also impacts the clustering accuracy of the ConstructPC. α is the trade-off parameter for the UMDwithSMACOF and UMDwithEd algorithms. β is the adjusting parameter only for UMDwithEd, and γ is a parameter to enlarge the distances between sub-manifolds. In the subsection, the influence of m and θ_0 and the classes number on the clustering accuracy will be discussed firstly. Then the parameters of α, β, and γ will be discussed respectively. Finally, the iteration processes of MPPCA and SMACOF will be discussed, because they are the bottleneck of the time complexity in ConstructPC and UMDwithSMACOF.

4.3.1 The Parameters of m and θ_0

m is the number of MPPCA "patches". The propagation process propagates the MPPCA "patches" to construct the sub-manifolds. So the classes number must be less than m. θ_0 is the threshold of the dissimilarity of tangent subspaces. The larger θ_0 is, the more the classes number is. The smaller θ_0 is, the less the classes number is.

m and θ_0 determine the clustering accuracy of the ConstructPC and ReConstructPC algorithms and the number of classes constructed by ConstructPC. The number of classed and the clustering accuracy constructed by ConstructPC algorithm on $ dataset is shown in Fig. 7. Now we know the real classes number of $ dataset is 2. So the number of classed in



a Isomap **b** t-SNE **c** Umap

d MMD-Isomap **e** SSMM-Isomap1 **f** SSMM-Isomap2

g s-Isomap **h** UMDwithEd **i** UMDwithSMACOF

Fig. 6 The 2D embedding results obtained by each method on the USPS dataset

Table 4 Comparison of the running time of different algorithms on five data sets

	$ data	Hybrid data	Mnist	Coil20	USPS
Isomap	27.114	125.631	64.832	24.252	27.357
t-SNE	56.579	161.133	100.713	50.223	53.222
Umap	–	–	–	–	–
MMD-Isomap	198.028	356.886	86.309	31.678	56.331
SSMM-Isomap1	31.876	20.562	19.511	10.824	12.224
SSMM-Isomap2	233.288	438.101	49.873	31.717	53.622
s-Isomap	84.896	203.631	112.841	49.249	61.583
UMDwithEd	**32.778**	**67.815**	**20.208**	**9.872**	**11.872**
UMDwithSMACOF	**297.548**	**640.308**	**516.216**	**269.023**	**266.363**

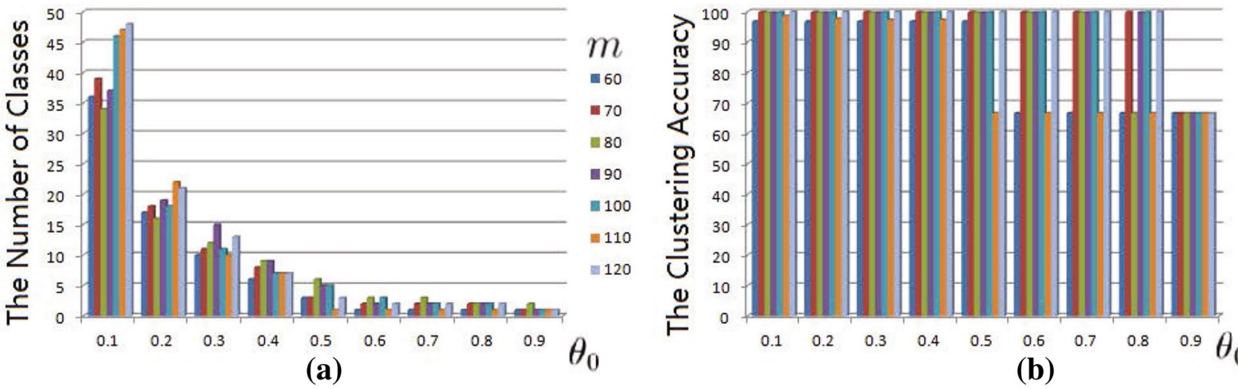
Optimal experimental results and our experimental results are given in bold

Fig. 7(a) should be 2. If the number of classes decomposed by ConstructPC algorithm is slightly larger than 2, the value of m and θ_0 can be accepted. So the value of m can be set between 70 and 120, and the value of θ_0 can be set between 0.6 and 0.8.

4.3.2 The Parameters of α, β, and γ

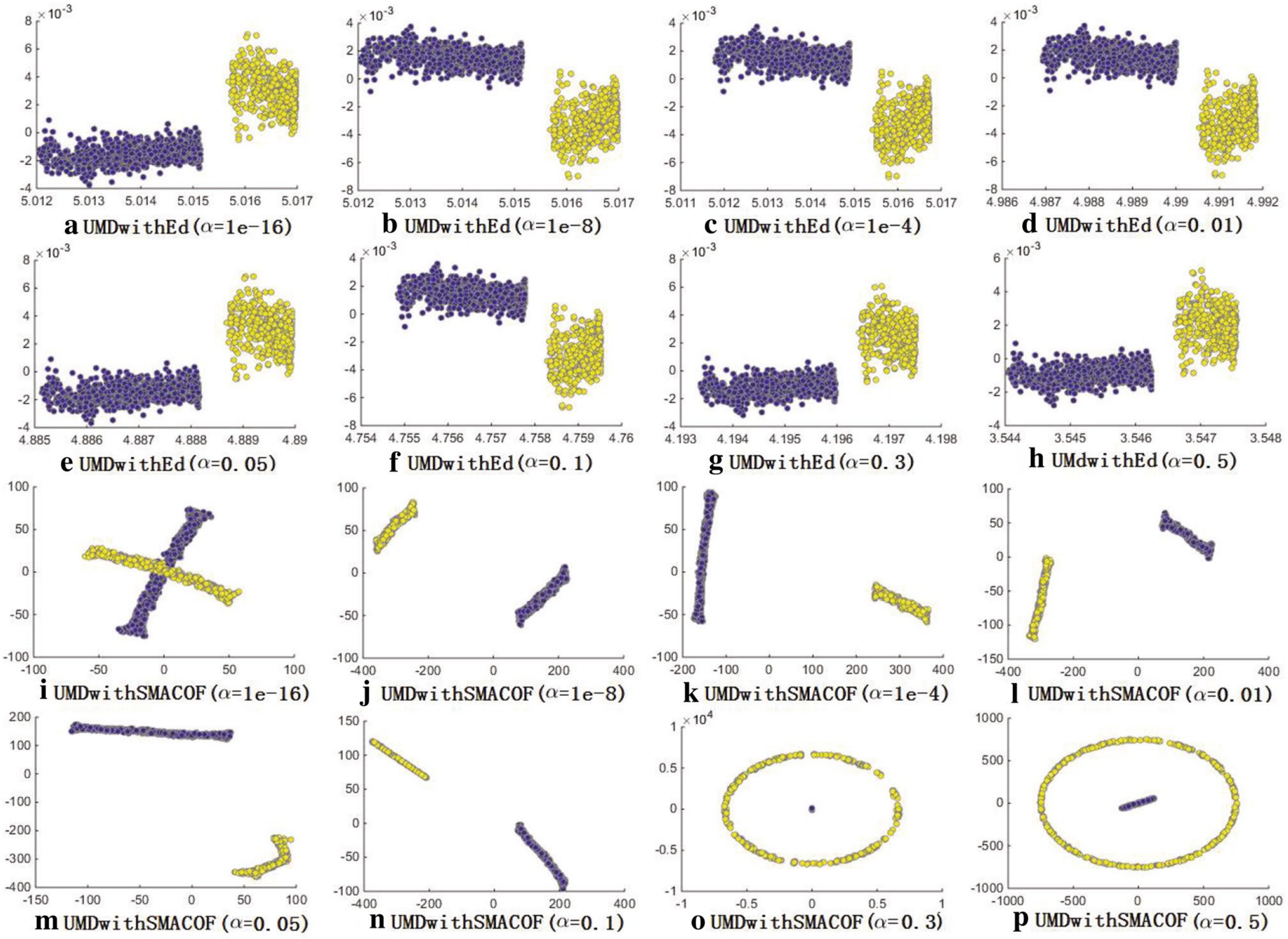
Our objective function attempts to preserve the global geometrical structures of the intra-class data points and separate the interclass data points. The parameter α plays key role in the algorithm, determining the dimensionality reduction and visualization results of multi-manifold. The smaller α is, the better the global geometrical structures of the intra-class points preserves. However, the interclass data points cannot be distinguished if the parameter is too small. So a smaller value for α less than 0.5 is better, and it is usually set to 0.01. The bigger the parameter γ is, the larger the interclass distances become. And it is usually set to 3 or 4. The parameter β can adjust the accuracy of single-manifold in UMDwithEd. When the parameter β is small, the UMDwithEd algorithm only show the distance relationship between point-pairs. When the parameter β becomes large, the topological structure constructed by the weight matrix in LLE-style item is demonstrated. And the parameter is usually set to 0.1.

The three parameters are analyzed in Figs. 8, 9 and 10, respectively. It can be seen that (1) the UMDwithEd algorithm is not sensitive to α and γ, and it is only sensitive to β; (2) the UMDwithSMACOF algorithm is more sensitive to α, and it can construct better embedding if the parameter



(a) (b)

Fig. 7 The influence of m and θ_0 on (a)the number of classes ; (b)the clustering accuracy



a UMDwithEd (α=1e−16) **b** UMDwithEd (α=1e−8) **c** UMDwithEd (α=1e−4) **d** UMDwithEd (α=0. 01)

e UMDwithEd (α=0. 05) **f** UMDwithEd (α=0. 1) **g** UMDwithEd (α=0. 3) **h** UMdwithEd (α=0. 5)

i UMDwithSMACOF (α=1e−16) **j** UMDwithSMACOF (α=1e−8) **k** UMDwithSMACOF (α=1e−4) **l** UMDwithSMACOF (α=0. 01)

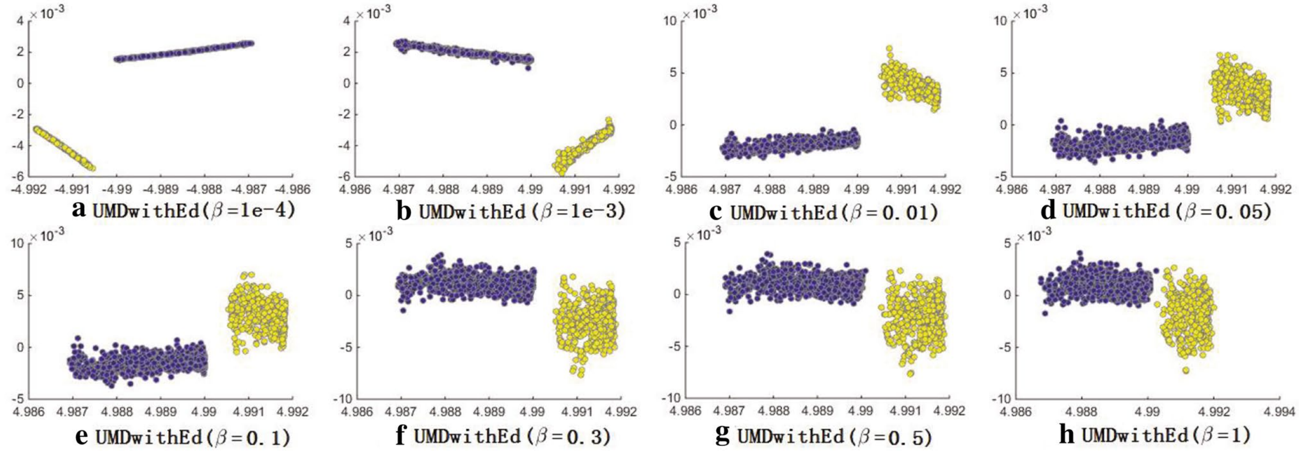
m UMDwithSMACOF (α=0. 05) **n** UMDwithSMACOF (α=0. 1) **o** UMDwithSMACOF (α=0. 3) **p** UMDwithSMACOF (α=0. 5)

Fig. 8 The influence of α on the UMDwithEd and UMDwithSMACOF algorithms

α is between 1e-8 and 0.1; (3) the interclass distances in UMDwithSMACOF algorithm can be adjusted by γ; (4) our algorithms are very robust to the parameters, they nearly can be seen parameter-free approaches.

4.3.3 Convergence analysis

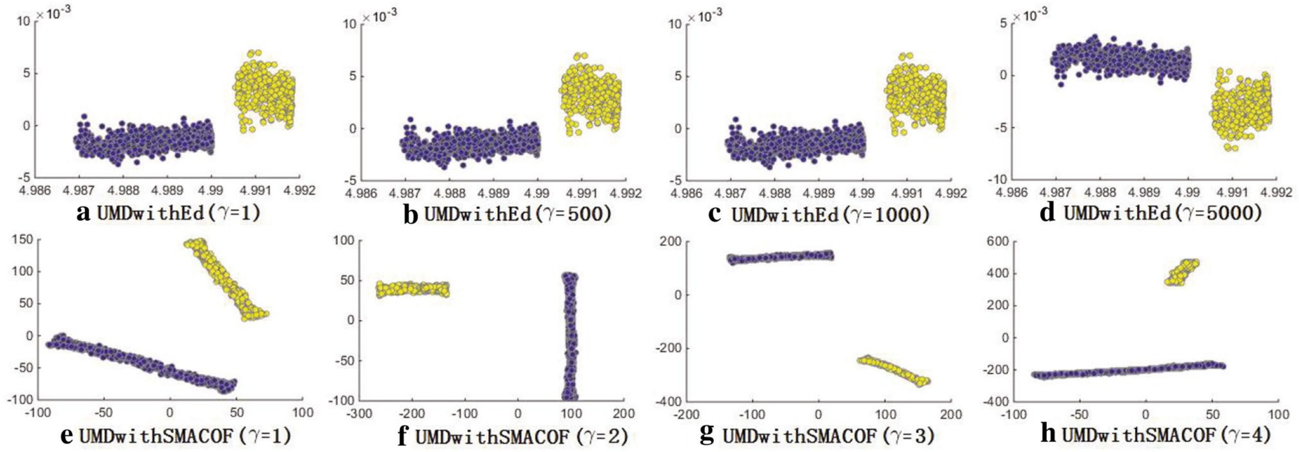
The bottleneck of the time complexity in ConstructPC and UMDwithSMACOF is in the iteration process of MPPCA



a UMDwithEd (β=1e-4) **b** UMDwithEd (β=1e-3) **c** UMDwithEd (β=0.01) **d** UMDwithEd (β=0.05)

e UMDwithEd (β=0.1) **f** UMDwithEd (β=0.3) **g** UMDwithEd (β=0.5) **h** UMDwithEd (β=1)

Fig. 9 The influence of β on the UMDwithEd algorithms



a UMDwithEd (γ=1) **b** UMDwithEd (γ=500) **c** UMDwithEd (γ=1000) **d** UMDwithEd (γ=5000)

e UMDwithSMACOF (γ=1) **f** UMDwithSMACOF (γ=2) **g** UMDwithSMACOF (γ=3) **h** UMDwithSMACOF (γ=4)

Fig. 10 The influence of γ on the UMDwithEd and UMDwithSMACOF algorithms

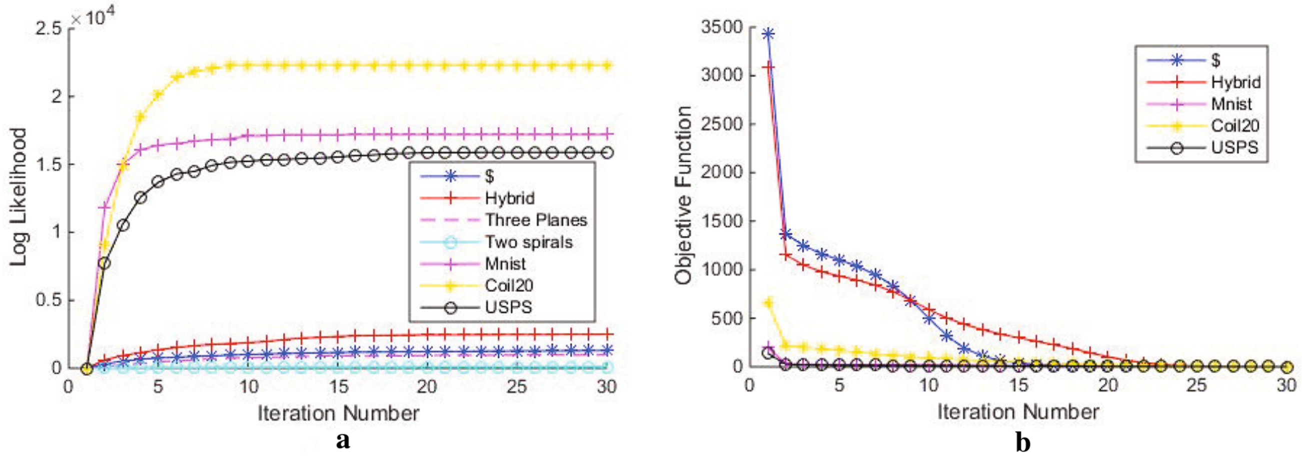


Fig. 11 The time complexity vs. the iteration process. **a** The log likelihood and the iteration number in MPPCA; (**b**)the objective function and the iteration number in SMACOF

and SMACOF. Fortunately, Fig. 11 shows that two algorithms have a rapid convergence rate and use fewer iterations to achieve a good performance. It is noted that each set of the log likelihood and the objective function subtracts their minimum values for better comparison.

5 Conclusions

In this paper, a novel algorithm is proposed, named UMD-Isomap, for the purpose of nonlinear dimensionality reduction and clustering of data lying on the multi-manifold. Different from most existing manifold learning algorithms, the proposed algorithm are used for the dimensionality reduction of multi-manifold. Each sub-manifold can not only get its low-dimensional embedding, but also all low-dimensional embeddings of multi-manifold do not overlap.

Our proposed algorithm is an unsupervised algorithm, which runs ConstructPC (or ReConstructPC) to cluster the data lying on the multi-manifolds, and transforms the clustering results into the pairwise constraints. Compared with the existing manifold clustering and spectral clustering algorithms, our proposed algorithms achieved better performance in terms of clustering accuracy and normalized mutual information. After that, Our proposed algorithm runs UMDwithSMACOF or UMDwithEd to construct the low-dimensional embeddings for multi-manifold. The UMDwithSMACOF can represent more accurate embeddings for multi-manifolds, however, its time complexity is higher because of its iteration process. The UMDwithEd represents the embedding more effectively, and it separates sub-manifolds distinctly, as it is not sensitive to the parameter.

Since the pairwise constraints can be transformed into label vectors, the ConstructPC and ReConstructPC algorithms can be applied to any supervised learning algorithm to construct corresponding unsupervised learning algorithm. And the UMDwithSMACOF and UMDwithEd algrithms can visualize 2D or 3D embeddings of any clustering results.

In future, there are many directions we need to further study. At first, the time complexity of the proposed algorithm is related to the dimension of the dataset, the optimization of solution will be explored in our future work. Secondly, the convergence of the model is only proved by some experiments in this paper, and the theoretical proof will be our next task. Finally, we will also explore other applications of UMD-Isomap, such as image recognition, scene classification and speech recognition.

Acknowledgements This work is partially supported by the National Natural Science Foundation of China (No. 61703252); Applied Basic Research Programs of Shanxi Province (201701D121053) and Research Project Supported by Shanxi Scholarship Council of China (2016-002).

References

1. Tenenbaum J, Silva DD, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
2. Roweis S, Saul L (2000) Nonlinear dimentionality reduction by locally linear embedding. Science 290(5500):2323–2326
3. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
4. Lin T, Zha HB (2008) Riemannian manifold learning. IEEE Trans. Pattern Anal. Mach. Intell. 30(5):796–809
5. Liu Y, Nie F et al (2019) Flexible unsupervised feature extraction for image classification. Neural Netw. 115:65–71
6. Liu Y, Gao Q, Yang Z et al (2018) Learning with Adaptive Neighbors for Image Clustering. In: Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18. 2018, 2483-2489
7. Fan B, Kong QQ et al (2020) Efficient nearest neighbor search in high dimensional hamming space. pattern recognition. https://doi.org/10.1016/j.patcog.2019.107082
8. Zheng Y, Liu X, Chen S et al (2020) Multi-task deep dual correlation filters for visual tracking. IEEE Trans. Image Process. 29:9614–9626
9. Zheng YH, Jeon B et al (2018) Student's t-Hidden Markov Model for Unsupervised Learning Using Locallized Feature Selection. IEEE Transactions on Circuits and Systems for Video Technology 28(10):2586–2598
10. Souvenir R, Manifold Pless R (2005) Clustering. Proceedings 1(1):648–653
11. Meng D, Leung Y, Fung T, Xu Z (2008) Nonlinear dimensionality reduction of data lying on the multicluster manifold. IEEE Trans Syst Man Cybernet-Part B 38(4):1111–1122
12. Gao XF, Liang JY (2013) Manifold learning algorithm DC-ISOMAP of data lying on well-separated multi-manifold with same intrinsic dimension. J Comput Res Dev 50(8):1690–1699
13. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. in Advances in Neural Information Processing Systems 14. 2001: 849-856
14. Shi JB, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
15. Zhang K, Kwok JT (2010) Clustered Nyström method for large scale manifold learning and dimension reduction. IEEE Trans Neural Netw 21(10):1576–1587
16. Wang Y, Jiang Y, Zhou ZH (2011) Spectral clustering on multiple manifolds. IEEE Trans Neural Netw 22(7):1149–1161
17. Wang Y, Jiang Y, Wu Y, et al (2011) Local and Structural Consistency for Multi-manifold Clustering. Proc of IJCAI 2011. Menlo Park, CA: AAAI Press, 1559-1564
18. Babaeian A, Bayestehtashk A, Bandarabadi M. Multiple Manifold Clustering Using Curvature Constrained Path. PLoS ONE 10(9): e0137986. https://doi.org/10.1371/journal.pone.0137986
19. van der Maaten L, Hinton GE (2008) Visualizing high-dimensional data using t-SNE. J Mach Learn Res 9:2579–2605
20. van der Maaten L (2014) Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245
21. McInnes L, Healy J, Melville J (2018) UMAP: uniform manifold approximation and projection for dimension reduction. J Open Source Softw 3(29):861
22. Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: advanced in algorithms, theory, and applications. CRC Press, Boca Raton
23. Geng X, Dc Zhan, Zhou ZH (2005) supervised nonlinear dimensionality reduction for visualization and classification. IEEE Trans Syst Man Cybernet-Part B 35(6):1098–1107

24. Goldberg A, Zhu X et al (2009) Multi-manifold semi-supervised learning. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, 2009, 169-176
25. Zhang Z, Chow TW, Zhao M (2013) M-Isomap:orthogonal constrained marginal Isomap for nonlinear dimensionality reduction. IEEE Trans. Cybern 43(1):180–191
26. Yang B, Xiang M, Zhang Y (2016) Multi-manifold Discriminant Isomap for visualization and classification. Pattern Recognit 2016(55):215–230
27. Zhang Y, Zhang Z et al (2017) Semi-supervised local multi-manifold Isomap by linear embedding for feature extraction. Pattern Recoginit 2017(000):1–17
28. Fan B, Liu HM et al (2020) Deep Unsupervised Binary Descriptor Learning through Locality Consistency and Self Distinctiveness. IEEE Transactions on Multimedia. https://doi.org/10.1109/TMM.2020.3016122
29. Tipping ME, Bishop CM (1999) Mixtures of probabilistic principal component analysers. Neural Comput. 11(2):443–482
30. Zhao J (2014) Efficient Mofel selection for mixtures of probabilistic PCA via hierarchical BIC. IEEE Trans Cybernet 44(10):1871–1883
31. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological) 39(1):1–38
32. Sun D, Zhang D (2010) Bagging constraint score for feature selection with pairwise constraints. Pattern Recognit 43(6):2106–2118
33. Baghshah M S , Shouraki S B. Semi-supervised metric learning using pairwise constraints. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2009, 9: 1217-1222
34. Wang F (2011) Semi-supervised metric learning by maximizing constraint margin. IEEE Trans Syst, Man Cybernet Part B-Cybernet 41(4):931–939
35. Joint Optimization for Pairwise Constraint Propagation, IEEE TNNLS, 2020
36. (2020) Pairwise Constraint Propagation With Dual Adversarial Manifold Regularization. IEEE TNNLS
37. Leeuw J de. Applications of Convex Analysis to Multidimensional Scaling. In JR Barra, F Brodeau, G Romier, B van Cutsem (eds.), Recent Developments in Statistics (1977) 133–145. North Holland Publishing Company, Amsterdam
38. de Leeuw I, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. J Stat Softw 31(3):1–30
39. Saul L, Roweis S (2004) Think globally, fit locally: unsupervised learning of low dimensional manifolds. J Mach Learn Res 4(2):119–155
40. Donoho D, Grimes C (2003) Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. PNAS. 100(10):5591–5596
41. Zhang ZY, Zha HY (2005) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J Sci Comput 26(1):313–338
42. Gao XF, Liang JY (2011) The dynamical neighborhood selection based on the sampling density and manifold curvature for isometric data embedding. Pattern Recognit Lett 32:202–209
43. Steyvers M (2002) Multidimensional scaling, In Encyclopedia of. Cognitive Science