Improved D–C Isomap on data set A

O1

O2

O3
Improved D–C Isomap on data set B

O1
O2
O3
Improved D–C Isomap on data set C
Improved D–C Isomap on teapot data set
Isometric Multi-Manifolds Learning

Abstract—Isometric feature mapping (Isomap) is a promising manifold learning method. However, Isomap fails to work on data which distribute on clusters in a single manifold or manifolds. Many works have been done on extending Isomap to multi-manifolds learning. In this paper, we proposed a new multi-manifolds learning algorithm (M-Isomap) with the help of a general procedure. The new algorithm preserves intramanifold geodesics and multiple inter-manifolds edges faithfully. Compared with previous approaches, this algorithm can isometrically learn data distribute on several manifolds. Some revisions have been made on the original multi-cluster manifold learning algorithm called D-C Isomap [24] such that the revised D-C Isomap can learn multi-manifolds data. Finally, the features and effectiveness of the proposed multi-manifolds learning algorithms are demonstrated and compared through experiments.

Index Terms—Isomap, nonlinear dimensionality reduction, manifold learning, pattern analysis, multi-manifolds learning.

I. INTRODUCTION

Challenges, known as "the curse of dimensionality", are usually confronted when scientists are doing researches on high dimensional data. Dimensionality reduction is a promising tool to circumvent these problems. Principal component analysis (PCA) [1] and multidimensional scaling (MDS) [2] are two important linear dimensionality reduction methods. Due to their linear model assumption, both of the methods will fail to discover nonlinear intrinsic structure of data.

Recently, there are more and more interests in nonlinear dimensionality reduction (NLDR). NLDR is used to learn nonlinear intrinsic structure of data, which is considered to be the first step of "machine learning and pattern recognition: observe and explore the phenomena" [3]. Two interesting nonlinear dimensionality reduction methods based on the notion of manifold learning [6], isometric feature mapping (Isomap) [4] and local linear embedding (LLE) [5], have been introduced in SCIENCE 2000. LLE assumes that data points locally distribute on a linear patch of a manifold. It preserves local linear coefficients, which best reconstruct each data point by its neighbors, into a lower dimensional space. Isomap is based on classical MDS method. Instead of preserving pairwise Euclidean distance, it preserves geodesic distance on the manifold. The geodesic between two data points is approximated by the shortest path on a constructed graph. Both of the methods are computational efficient and able to achieves global optimality. Besides, there are many other important nonlinear dimensionality reduction methods. Laplacian eigenmap [7] utilizes the approximation of the Laplace-Beltrami operator on manifold to provide an optimal embedding. Hessian LLE [8] resembles Laplacian eigenmap by using the approximation of Hessian operator instead of Laplacian operator. Local tangent space alignment (LTSA) [9] method learns local geometry by constructing a local tangent space of each data point and then aligns these local tangent spaces into a single global coordinates system with respect to the underlying manifold. Diffusion maps [10] applies diffusion semigroups to produce multiscale geometries to represent complex structure. Riemannian manifold learning (RML) [11] method uses the constructed Riemannian normal coordinate chart to map the input data into a lower dimensional space. NLDR is fast developed and has been proved very useful in many fields and applications, such as classification using Isomap [16] and laplacian eigenmap [17], geometric based semi-supervised learning method using laplacian eigenmap [18], data visualization [19], time series manifold learning [20], [21] and so on.

As Isomap emphasizes on the global geometric relationship of data points, it is very illustrative in data visualization and pattern analysis [13]. Although Isomap algorithm implicitly requires the data set is convex, it still demonstrates very meaningful results on non convex data sets. Therefore, in this paper, we will focus attention on extending Isomap to multi-manifolds learning. The first step of Isomap algorithm is to construct a neighborhood graph which connects all the data points. This step is of vital importance because the success of following steps depend on how well the constructed neighborhood graph is. However, it is hard to build a totally connected neighborhood graph and guarantee the topological stability of classical Isomap algorithm when points of the data set distribute on clusters in a manifold or manifolds (multiple manifolds). Many works have been done on extending Isomap to multi-manifolds data. Some methods try to do this by providing new neighborhood graph construction algorithms. Yiming Wu et al [23] introduced a split and augment procedure for neighborhood graph construction which could produce a totally connected neighborhood graph. Li Yang [26]–[29] introduced several neighborhood graph construction algorithms using techniques from discrete mathematics, graph theory. Deyu Meng et al [24] proposed a decomposition and composition Isomap (D-C Isomap).

The rest of the paper is organized as follows: In Section II, main issues and limitations of classical Isomap algorithm are presented. The problem of multi-manifolds learning is also investigated. In Section III, previous methods on multi-manifolds learning are briefly introduced and discussed. In Section IV, a general procedure for designing multi-manifolds learning algorithms is first proposed. With the proposed procedure, a new multi-manifolds learning algorithm (M-Isomap) is designed and analyzed. With some revisions on the original algorithm, the main limitations of D-C Isomap are resolved. Finally, in Section V, the effectiveness of these multi-manifolds learning algorithms has been demonstrated by experiments. Comparisons of these algorithms have also been made.

II. CLASSICAL ISOMETRIC FEATURE MAPPING AND ITS LIMITATIONS

Isomap is able to recover the intrinsic geometric structure and converge as the number of data points increases [4] if data lie on a manifold. Like PCA and MDS, Isomap has the advantage of simple implementation and computational efficiency. The algorithm also guarantees a globally optimal solution.

It is assumed that data set \( X = \{x_1, x_2, \cdots, x_N\} \) is in high dimensional space \( R^D \) and the feature space is \( R^d \). The classical Isomap algorithm has three steps.

Step 1: Identify the neighbors for all the data points to construct a neighborhood graph. With the given parameter
and “discontinuity” problem can be circumvented by carefully distributing uniformly on one manifold. Both the “short circuit” stability of Isomap algorithm [22].

edges will have a significant negative influence on the topological $k$ be satisfyingly approximated. Otherwise, if neighborhood size the neighborhood graph will be very sparse. Thus geodesics cannot be approximated. This usually happens in “data mixture” cases. Second, learning different data manifolds may need different value of input parameters, i.e. appropriate neighborhood size ($k$ or $\epsilon$) and intrinsic dimensionality $d$ for each data manifold. The case when data points distribute on pieces of a single manifold is referred to as multi-cluster manifold learning; meanwhile, the case when data points distribute on multiple manifolds is referred to as multi-manifolds learning. In this paper, we will concentrate on designing multi-manifolds learning algorithms to data with these properties.

III. PREVIOUS WORKS ON MULTI-MANIFOLDS LEARNING

A. Multi-manifolds learning by new neighborhood graph construction method

Wu and Chan [23] proposed a split-augment approach to construct a neighborhood graph. Their method can be regarded as a variation of the k-NN method and can be summarized as below:

1. k-NN method is applied to the data set. Every data point is connected with its neighbors. If the data lies on multiple manifolds, several disconnected graph components (data manifolds) will be formed.
2. Each couple of graph components are connected by their nearest couple of inter-components points.

This method is simple to implement and has the same computational complexity as k-NN method. However, as there is only one edge connecting every two graph components, geodesics across components are poorly approximated; meanwhile, their low dimensional embedding can be rotated arbitrarily. This method can not be directly applied to data lying on three or more data manifolds. If more than two graph components exist, intra-component shortest distances may be changed in the totally connected graph.

Li [26]–[29] introduced four methods to construct a connected neighborhood graph. The k minimum spanning trees (k-MST) [26] method repeatedly extracts k minimum spanning trees (MSTs) from the complete Euclidean graph of all data points. Edges of the k MSTs form a k-connected neighborhood graph. Instead of extracting k MSTs, the minimum-k-spanning trees (min-k-ST) [27] method finds k edge-disjoint spanning trees from the complete Euclidean graph, and the sum of the total edge length of the k edge-disjoint spanning trees attains a minimum. The k-edge-connected (k-EC) [28] method constructs a connected neighborhood graph by adding edges in a non-increasing order from the complete Euclidean graph. An edge is added if its two end vertices do not have k edge-disjoint paths connected with each other. The k-vertices-connected (k-VC) [29] method add edges in a non-increasing order from the complete Euclidean graph, an edge is added if its two end vertices would be disconnected by removing some $k - 1$ vertices. Finally, the constructed neighborhood graph would not be disconnected by removing any $k - 1$ vertices.

The methods introduced in [26]–[29] advantage over k-NN method for two reasons: First, the local neighbor relationship is affected by the global distribution of data points. This is beneficial for adaptively preservation of the global geometric metrics. Second, these methods could guarantee that the constructed neighborhood graph is totally connected. Compared with k-NN method, Li’s methods construct a neighborhood graph with more edges corresponding to the same neighborhood size $k$. This property can assure the quality of the neighborhood graphs.
B. Multi-manifolds learning by decomposition-composition Isomap

In [24], Meng et al. proposed a decomposition-composition method (D-C Isomap) which extends Isomap to multi-cluster manifold learning. The purpose of their method is to preserve intra-cluster and inter-cluster distances separately. Because a revised version of D-C Isomap will be introduced in the next section, we present the details of D-C Isomap algorithm in the following:

Step I: decomposition process

1. Given neighborhood size \( k \) or \( \epsilon \), if the data is of multi-cluster, several disconnected graph components can be identified when every data point is connected with its neighbors by k-NN or \( \epsilon \)-NN.
2. Assuming that there are \( M \) components, the \( m \)-th component is also denoted as a cluster \( X^m = \{x_{i}^{m}, \ldots, x_{n}^{m}\} \). Clusters \( X^m \) and \( X^n \) are connected by their nearest inter-cluster data points \( n_{x}^{m} \) and \( n_{x}^{n} \) whose pairwise distance is assumed to be \( d_{m,n}^{m} \).
3. Apply k-NN Isomap or \( \epsilon \)-NN Isomap on each cluster \( X^m \). Denote the geodesic distance matrix for \( X^m \) as \( D^m = (d_{j,l}^{m}) \), the corresponding low dimensional embedding as \( y_{m}^{e} = (y_{1}^{m}, \ldots, y_{n}^{m}) \), and the embedding point corresponding to \( n_{x}^{m} \) as \( y_{n}^{m} \), where \( y_{n}^{m} \in Y^m \).

Step II: composition process

1. The set of centers of clusters is denoted as \( CX = \{cx_{1}, \ldots, cx_{M}\} \), where every center is computed by
   \[
   cx_{m} = \arg \min_{x \in X^{m}} \left( \max_{j \in CX} D_{j}^{m} \right) \quad m = 1, \ldots, M.
   \]
2. The distance matrix for \( CX \) can be computed by
   \[
   \tilde{D} = \{\tilde{D}_{mn}\}, \quad \tilde{D}_{mn} = \begin{cases} d_{m,n} + d_{m,m} + d_{n,m} & m \neq n \\ 0 & m = n \end{cases}
   \]
   where \( d_{m,n} \) is the distance of shortest path between \( cx_{m} \) and \( n_{x}^{m} \) on the graph component \( X^m \).
3. Plug distance matrix \( \tilde{D} \) and neighborhood size \( d \) into classical Isomap algorithm. The embedding of \( CX \) is denoted by \( CY = \{cy_{1}, \ldots, cy_{M}\} \subset \mathbb{R}^{d} \). The low dimensional representation corresponding to \( n_{x}^{m} \) is computed as
   \[
   SY_{m}^{e} = cy_{m} + \frac{d_{m,m} + d_{m,n} + d_{n,m}}{d_{m,m}} (cy_{m} - cy_{n})
   \]
   where \( d_{m,m} \) is the distance of shortest path between every center of clusters by \( d_{m,n} \) and \( d_{n,m} \); \( m = 1, \ldots, M \).
4. Construct the rotation matrix \( \mathcal{A} \) for \( Y^{m}, m = 1, \ldots, M \). Assuming that \( QN_{m} \) is the principal component matrix of \( NY_{m} = (ny_{x}^{m}, \ldots, ny_{x}^{m}) \) and \( QS_{m} \) is the principal component matrix of \( SY_{m} = (sy_{x}^{m}, \ldots, sy_{x}^{m}) \), then the rotation matrix for \( Y^{m} \) is \( \mathcal{A}_{m} = QS_{m} QN_{m}^{T} \).
5. Transform \( Y^{m}, m = 1, \ldots, M \) into a single coordinate system by Euclidean transformations
   \[
   FY^{m} = \{fy_{i} = \mathcal{A}_{m} ny_{i}^{m} + cy_{m}, i = 1, \ldots, l_{m}\}, \quad m = 1, \ldots, M.
   \]
Then \( Y = \bigcup_{m=1}^{M} FY^{m} \) is the final output.

Firstly, the D-C Isomap reduces the dimensionality of clusters separately; meanwhile, it preserves a skeleton of the whole data. Secondly, using Euclidean transformations, embedding of each cluster is placed into the corresponding position by referring to the skeleton. In this way, intra-cluster geodesics are exactly preserved. As D-C Isomap method uses circumcenters to construct the skeleton of whole data, its learning results unstably depend on the mutual position of these circumcenters. It is known that at least \( d + 1 \) reference points are needed to anchor a \( d \)-dimensional simplex. However, in D-C Isomap algorithm, the number of the reference data points is limited by the number of clusters.

C. Constrained Maximum Variance Mapping

There is also a newly proposed algorithm called constrained maximum variance mapping (CMVM) [25] for multi-manifolds learning. CMVM method is proposed on the notion of maximizing dissimilarities between classes while holding up the intra-class similarity.

IV. ISOMETRIC MULTI-MANIFOLDS LEARNING

A. The general procedure for isometric multi-manifolds learning

Many previous methods extend Isomap for multi-manifolds learning by revising the neighborhood graph construction step of the Isomap algorithm [23], [26]–[29]. However, shortest paths across clusters or data manifolds are bad approximations of geodesics. In Isomap, bad local approximation always leads to the deformation of global low dimensional embedding.

Under the continuous assumption, it is assumed that \( \Omega \) is an open, convex and compact set in \( \mathbb{R}^{d} \), and \( f : \Omega \rightarrow \mathbb{R}^{d} \) is a continuous mapping. \( f(\Omega) = M \) is defined as a \( d \) dimensional parameterized manifold. \( K(x, y) \in M \) is a specially defined kernel and a reproducing kernel Hilbert space (RKHS) \( \mathcal{H} \) is constructed with this kernel. \( \phi(x) \) is the eigenfunction corresponding to the \( j \)-th largest eigenvalue \( \lambda_{j} \) of \( K(x, y) \in \mathcal{H} \), which is also the \( j \)-th element of Isomap embedding. The geodesic distance on manifold \( M \) is written as \( d^{2}(x, y) = d^{2}(f(\tau), f(\hat{\tau})) = \alpha||\tau - \hat{\tau}|| + \eta(\tau, \hat{\tau}) \), where \( \tau, \hat{\tau} \in \Omega \). \( \alpha \) is a constant and \( \eta(\tau, \hat{\tau}) \) is the deviation from isometry. The constant vector

\[
C = \int_{\Omega} \rho(x) dx = \int_{\Omega} \tau \mathcal{H}(\tau) d\tau
\]

where \( \rho(x) \) and \( \mathcal{H}(\tau) \) are density functions of \( M \) and \( \Omega \). With the assumptions above, the following theorem is proved by Zha et al [12].

**Theorem 4.1**. There is a constant vector \( P_{j} \) such that \( \phi_{j}(x) = P_{j} (\tau - C) + e_{j}(\tau) \), where \( e_{j}(\tau) = e_{j}^{(0)} - e_{j}(\tau) \) has zero mean, i.e.,

\[
\frac{1}{l_{j}} \int_{\Omega} \mathcal{H}(\tau) e_{j}(\tau) d\tau = 0,
\]

with \( e_{j}(\tau) = \frac{1}{2l_{j}} \int_{\Omega} \eta(\tau, \hat{\tau}) \mathcal{H}(\tau) \phi_{j}(x) d\hat{\tau} \).

By theorem 4.1, even if the deviation \( \eta(\tau, \hat{\tau}) \) is not zero with only a limited range of \((\tau, \hat{\tau})\). The coordinate of the low dimensional embedding \( \phi_{j}(x) \) is still deformed, and the deformation is measured by \( e_{j}(\tau) \).

In order to get a better understanding of multi-manifolds data, it is profitable to preserve intra-manifold relationship (where \( \eta(\tau, \hat{\tau}) = 0 \)) and inter-manifolds relationship (where \( \eta(\tau, \hat{\tau}) \neq 0 \)) separately. This is because that sometimes we care more about the information within the same data manifold. Here we propose
a general procedure for isometric multi-manifolds learning algorithms.

Step I: The decomposition process

1. Cluster the whole data set. If data distribute on different clusters in a manifold or manifolds, the clusters or manifolds should be identified. Many clustering methods could be used, such as K-means, Isodata and methods introduced in [15] [33]. Even if the manifolds overlay with each other, they can still be identified and clustered [39]. At this step, data set $X$ is clustered into several components and every component is considered as a data manifold.

2. Estimate parameters of data manifolds. For intrinsic dimensionality estimation, many methods can be used: the fractal based method [34], MLE method [35], [36], and the incising ball method [37]. Assume that $d_m$ is the intrinsic dimensionality of the $m$-th data manifold. Let $d = \max_{m} d_m$.

For neighborhood size, [32] introduces a method on automatically generating parameters for Isomap on one single data manifold. For convenience, appropriate neighborhood sizes ($k_m$ or $r_m$ for $X^m$) could be given manually for data manifolds.

3. Learn the data manifolds individually. One data manifold can be learned by traditional manifold learning algorithms. Here, we propose to rebuild a graph on each data manifold with new neighborhood size to better approximate intra-manifold geodesics. Methods of Li’s works [26]–[29] or the $k$-CC method is preferred, where the $k$-CC graph construction method will be described later. It is assumed that the low dimensional embedding for $X^m$ is $Y^m$.

Step II: The composition process

1. Preserve a skeleton $I$ of the whole data set in low dimensional space $\mathbb{R}^d$. The skeleton $I$ should be carefully designed such that it can represent the global structure of $X$. Let $RY_I$ be the low dimensional embedding of $I$.

2. Transform $Y^m$s into a single coordinate system by referring to $I$. In order to faithfully preserve intra-manifold relationship, Euclidean transformations could be constructed and used. Using embedding points $RY^m \subset RY_I$ and corresponding points from $Y^m$, we can construct an Euclidean transformation from $Y^m$ to the coordinate system of $I$.

Although the idea about decomposition-composition is not new, which is first used by Wu et al. [23] in their split-augment process and well developed and used in [24]. The procedure we proposed here aims to solve a more general problem. Step I.I permits that the designed learning method has a good ability to identify data manifolds. Step I.2 gives a guideline on learning manifolds with different intrinsic dimensionality and neighborhood sizes. Step I.3 learns data manifolds individually so that the intra-manifold relationship can be faithfully preserved. Step II.1 is the most flexible part of the procedure which allows us to design new multi-manifolds learning algorithms. A well designed skeleton $I$ could better represent the inter-manifolds relationship. In the following, we will introduce a new multi-manifolds learning algorithm and revise the original D-C Isomap algorithm with the help of this general procedure.

A. new algorithm for isometric multi-manifolds learning

Based on the proposed procedure, we designed a new multi-manifolds learning algorithm. As an extension of Isomap method for multi-manifolds data, the method will be referred to as multi-manifolds Isomap (M-Isomap). It is assumed that $X$ is also interchangeable to represent the matrix $\{x_1, x_2, \cdots, x_N\}$, where $\{x_i, i = 1, \cdots, N\}$ are column vectors in $\mathbb{R}^D$.

I) Using $k$-CC method to construct a neighborhood graph and identify manifolds: Table III shows the time complexity of $k$-NN, K-Min-ST, k-EC and k-VC. TC stands for time complexity and IL stands for the time complexity for incremental learning.

| Algorithm | k-NN | k-MST | Minig-k-ST | k-EC | k-VC |
|-----------|------|-------|-------------|-----|-----|
| TC        | $O(kN^2)$ | $O(k^2N^2)$ | $O(k^2N^3)$ | $O(kN^3)$ | $O(N^3)$ |
| IL        | $O(kN)$ | $O(N log N)$ | $O(N log N)$ | $O(N log N + kN)$ |

TABLE III

For computational complexity comparison of $k$-NN, k-MSTs, Minig-k-ST, k-EC and k-VC, TC stands for time complexity and IL stands for the time complexity for incremental learning.
of Li’s methods becomes insignificant when the neighborhood size $k$ for k-NN method increases. More importantly, k-NN implicitly has the property of clustering to multi-manifolds data. Data points of the same manifold tend to be connected by paths and disconnected otherwise when every data point is connected with its neighbors by edges. Although k-NN is not a robust clustering algorithm, it is computationally efficient for both clustering and graph construction. Therefore, we introduce a variation of k-NN method which inherits computational advantage of k-NN method. The method is also able to identify data manifolds and construct a totally connected neighborhood graph. In the rest of the paper, the proposed neighborhood graph construction method will be referred as k-edge connected components (the k-CC method).

The summary of k-CC algorithm: First, given a neighborhood size $k$ or $\varepsilon$, every data point is connected with its neighbors. If the data points distribute on several clusters or manifolds, several disconnected graphs will be constructed. Data points are assigned to the same data manifold if there is a path connects them on the graphs. Then, we connect each pair of graphs by $k$ nearest pairs of data points. Concerning about robustness of the algorithm, every data point is only allowed to have one inter-manifolds edge at most.

**Algorithm:** (k-CC method)

**Input:** Euclidean distance matrix $D$, whose $(i,j)$-th entry is $||x_i - x_j||$. Neighborhood size $k$ or $\varepsilon$.

**Output:** Graph $G = (V,E)$, number of clusters $M$, label of the data.

**Initialization:**

1. for $i = 1$ to $N$ do
2. Identify nearest neighbors $\{x_{i1},\ldots,x_{ik}\}$ for $x_i$ by $k$-nearest-neighbors or $\varepsilon$-nearest-neighbors. Let $E = E \cup \{e_{i1},\ldots,e_{ik}\}$
3. end for
4. Set $M = 1$
5. while $|V|$ is not empty do
6. $x \in V$, in-Queue[x], label(x) = $M$, $V' = V - \{x\}$
7. while Queue is not empty do
8. $x = \text{de-Queue}$
9. $\forall y$ is connected with $x$ by an edge
10. in-Queue[y], label(y) = $M$, $V' = V' - \{y\}$
11. end if
12. end while
13. $M = M + 1$
14. end while
15. $M = M - 1$
16. if ($M \geq 2$) then
17. $k = \text{average}(\{l_i\}_{i=1}^M)$
18. for $i = 1$ to $M$ do
19. Find $k$ shortest inter-manifolds edges $e_{i1},\ldots,e_{ik}$ between data manifolds $i$ and $j$ and make sure that their ending vertices are not identical. Let $E = E \cup \{e_{i1},\ldots,e_{ik}\}$
20. end for
21. end for
22. end if

The main difference between k-NN and k-CC is lines (4-25), which identify components (data manifolds) and connect different components of the graph. This change makes the constructed graph totally k-edge connected. Compared with the method proposed in [23], k-CC method constructs a neighborhood graph with $k$ inter-manifolds edges, which is able to control the rotation of the embedding of data manifolds. In Section V, the method, which uses k-CC to construct a totally connected graph and then perform classical Isomap on the graph, will be referred to as k-CC Isomap. It can be easily inferred that k-CC Isomap suffers the limitation which has been shown by Theorem 4.1.

At this step, it is assumed that $X$ is clustered into data manifolds $\{X^m\}_{m=1}^M$ and $\{x^m_i\}_{i=1}^{n_m}$ is the subset of $X^m$ whose data points connect with $X^{m(i)}, i = 1, \ldots, k$.

2) Learn data manifolds individually: As $X^m$ is considered as a single data manifold in $\mathbb{R}^D$, it is possible to find its intrinsic parameters. The incising ball method [37] is utilized to estimate the intrinsic dimensionality, which is simple to implement and always outputs an integer result. Assume that $d$ is the highest intrinsic dimensionality of data manifolds. Neighborhood size $k_m$ or $\varepsilon_m$ of each data manifold is given manually and the graph on data manifold $X^m$ is rebuilt. It is hoped that the new neighborhood graph on $X^m$ can make better approximations of intra-manifold geodesics. The approximated geodesic distance matrix for $X^m$ is written as $D_m$. By applying classical MDS on $D_m$, the low dimensional embedding for $X^m$ can be got as $Y^m = \{y_i^m\}_{i=1}^{n_m}$.

3) Preserve a skeleton of data manifold $X$: First, inter-manifolds distances are computed. Assuming that $x_p^m$ and $x_q^m$ are any data points with $x_p^m \in X^m$ and $x_q^m \in X^n$, their distance can be computed by

$$d(x_p^m, x_q^m) = \min_{1 \leq i \leq k} \{d_m(x_p^m, x_{m(i)}) + ||x_{m(i)} - x_{m(ir)}|| + d_m(x_{m(ir)}, x_q^m)\},$$ (1)

where $d(x_p^m, x_{m(i)})$ is the shortest path on the neighborhood graph of $X^m$. Although $d(x_p^m, x_{m(i)})$ may not be the shortest path on the totally connected graph of $X$, Eq. 1 is an efficient way to approximate distances across manifolds. $D_{mn}$ is assumed to be the distance matrix across over $X^m$ and $X^n$. The furthest inter-manifolds data points are computed by

$$\{f(x_n^m, f(x_n^m)) = \arg \max d(x_n^m, x_p^m), \quad d(x_n^m, x_p^m) \in D_{mn}\}.\quad (2)$$

Without lose of generality, we assume $I^m = \{x_{m(i)}^{n_m}\} = \bigcup_{m=1}^M \{x_{m(i)}^{n_m}\} = I$. $I^m$ is considered as the global skeleton of $X$. On the data manifold $X$, it can be seen that the skeleton $I$ formulates a sparse graph. We assume that $D_I = (d_I(i,j))$ is the distance matrix of $I$, where

$$d_I(i,j) = \begin{cases} d(x_i^m, x_j^m) \in D_{mn}, & x_i \in X^m, x_j \in X^n \\ d(x_i^n, x_j^n) \in D_{mn}, & x_i, x_j \in X^n \end{cases} \quad (3)$$

By applying classical MDS algorithm on $D_I$, the low dimensional embedding of $I$ can be got as $R_I$. It is assumed that $R_I^m \subset R_I$ is the embedding of $I^m$ and $R_I^m = \{r_i^m\}_{i=1}^{n_m}$. 4) Euclidean transformations: Assuming that $Y^n = \{y_i^n\}_{i=1}^{n_m}$ corresponds to $x_i^n$, an Euclidean transformation from $Y^n$ to $R_I^m$ could be constructed.
The general Euclidean transformation can be written as

\[ ry = Ay + \beta, \]

where \( A \) is an orthonormal matrix and \( \beta \) is a position translation vector. For the \( m \)-th data manifold, it is assumed that the Euclidean transformation is

\[ ry^m_i = A_m y^m_i + \beta_m i = 1, \cdots, l_m. \]

Generally, it could be written in form of matrix

\[ RY^m_i = A_m Y^m_i + \beta_m e^T = (A_m \quad \beta_m) \begin{pmatrix} y^m_i \\ e^T \end{pmatrix} \]

where \( e \) is a vector with all ones. Problem (4) can be solved by the least square strategy, and the solution is

\[ (A_m, \beta_m) = RY^m_i \begin{pmatrix} y^m_i \\ e^T \end{pmatrix} \left( (A_m y^m_i + \beta_m e^T)^T (A_m y^m_i + \beta_m e^T) + \lambda I \right)^{-1} \]

where \( I \) is the identity matrix and \( \lambda \) is the regularization parameter.

Input: \( X = \{x_i\}_{i=1}^n \) with \( x_i \in \mathbb{R}^d \). Initial neighborhood size \( k \) or \( \varepsilon \).

Step I.1 Perform k-CC on \( X \). Data manifolds \( \{x^m_i\}_{m=1}^M \) and the set of inter-manifolds points \( \{x^m_{m+1}\}_{i=1}^M \) of \( X^m \) can be obtained.

Step I.2 Estimate parameters of data manifolds. It is assumed that intrinsic dimensionality \( d_m \) and neighborhood size \( (k_m \text{ or } \varepsilon_m) \) are parameters for \( X^m \). Let \( d = \text{max}(d_m) \) and rebuild neighborhood graph on \( X^m \).

Step I.3 Classical Isomap is performed on the new graphs superimposed on \( X^m, m = 1, \cdots, M \). The corresponding low dimensional embedding of \( X^m \) is denoted as \( y^m \).

Step II.1 Inter-manifolds distance matrix \( D_{nm} \) is computed by Eq. (1), thus \( \{f x^m_i\}_{m+n}^M \) can be found by Eq. (2). Distance matrix \( D_I \) for the skeleton \( I \) is computed by Eq. (3). Applying classical MDS on \( D_I \), we denote the low dimensional embedding of \( I \) as \( RY_I \). \( RY^m_i \subset RY_I \) and \( RY^m \) is assumed to be the embedding of \( I^m \).

Step II.2 Construct Euclidean transformations by Eq. (5-7). Using the Euclidean transformations, it is assumed that \( Y^m, m = 1, \cdots, M \) are transformed to \( RY^m, m = 1, \cdots, M \).

Step II.3 \( Y = \{Y^m_i\}_{m=1}^M \) \( RY^m \) is the final output.

C. Computational complexity of M-Isomap method

Computational complexity is a basic issue for application. For M-Isomap method, k-CC method needs \( O((k + 1)N^2) \) time to construct a totally connected graph and identify the manifolds. Computing the shortest path on every data manifold needs \( O(M^2 \sum_{m=1}^M S_m^3 \log S_m) \) time, and performing classical MDS on the distance matrices of data manifolds needs \( O(M^2 \sum_{m=1}^M S_m^3) \) time. The time complexity of computing the shortest path across data manifolds is \( O(M \sum_{m\neq n}^M k_m S_m S_n) \) and finding \( f x^m_i, f x^n_i \) is \( O(M^2 \sum_{m\neq n}^M S_m S_n) \). Performing classical MDS on skeleton \( I \) needs \( O(M \sum_{m=1}^M l_m^3) \) computational time. The time complexity of least square solution and QR decomposition process for \( M \) data manifolds is \( O(Md^3) \). Finally, transforming \( Y^m \)'s into a single coordinate system needs \( O(d^2 N) \) computational time.

Therefore, the total time complexity of the M-Isomap method is

\[ O((k + 1)N^2 + \sum_{m=1}^M (S_m^3 + S_m^2 \log S_m) + \sum_{m<n} (k + 1)S_m S_n + \sum_{m=1}^M l_m^3 + Md^3 + d^2 N) \]

For a large data set when \( N \gg M \) and \( N \gg d \), the overall time complexity of M-Isomap can be approximated by

\[ O((k + 1)N^2 + \sum_{m=1}^M (S_m^3 + S_m^2 \log S_m) + \sum_{m<n} (k + 1)S_m S_n) \]

D. The revised D-C Isomap method

D-C Isomap applies the decomposition-composition procedure. Therefore, it is able to preserve intra-cluster distances faithfully.
Here on graph between center to inter-cluster points can be calculated as optimization problem: where

\[ \sum_{i=1}^{m} \| O_1 - O_i \| \]

shows the case when \( \| O_1 - O_i \| \) is in the line which connects centers \( O_m \) and \( O_n \). Thus it is more sensible that \( O_m \) is chosen by referring to the inter-cluster points. Fig. 1 illustrates two basic cases about the relationship of the center and inter-cluster points. Although the points \( nx_1^1, nx_2^1, nx_3^1 \) and \( O_1 \) do not have to really lie on the same plane in the ambient space. It is assumed that these points formulate a triangle in the low dimensional space. Fig. 1(a) shows the case when \( \angle nx_1^1 nx_2^1 nx_3^1 + \angle nx_1^1 nx_3^1 nx_2^1 < 180^\circ \).

In triangle \( \Delta O_1 nx_1^1 nx_2^1 \), the edge \( d(nx_1^1, nx_2^1) \) can be computed as \( \| nx_1^1 - nx_2^1 \| \). We also have

\[ \angle O_1 nx_1^1 nx_3^1 = \arccos \left( \frac{nx_1^1 - nx_2^1 \cdot nx_1^1 - nx_3^1}{\| nx_1^1 - nx_2^1 \| \| nx_1^1 - nx_3^1 \|} \right) \]

\[ \angle O_1 nx_2^1 nx_3^1 = \arccos \left( \frac{nx_2^1 - nx_3^1 \cdot nx_2^1 - nx_1^1}{\| nx_2^1 - nx_3^1 \| \| nx_2^1 - nx_1^1 \|} \right) \]

Subsequently, the length of edges \( d(O_1, nx_1^1) \) and \( d(O_1, nx_2^1) \) can be calculated by the Law of Sines in \( \Delta O_1 nx_1^1 nx_2^1 \). Suggested distances between center to inter-cluster points can be calculated as

\[ d'(O_1, nx_1^1) = d(O_1, nx_1^1) - \| nx_1^1 - nx_2^1 \| \]

\[ d'(O_1, nx_2^1) = d(O_1, nx_2^1) - \| nx_2^1 - nx_1^1 \| \]

For a cluster with intrinsic dimensionality 2, it is sufficient to estimate position of \( O_1 \) in the cluster by solving the following optimization problem:

\[ O_1 = \arg \min_{o \in X_1} f(o) \]  \hspace{1cm} (8)

where

\[ f(o) = \sum_{i=1}^{n} \| d(O_1, o) - d'(O_1, nx_i^1) \| \]

Here \( d(O_1, nx_i^1) \) is the length of shortest path between \( O_1 \) and \( nx_i^1 \) on graph \( X_1 \). For a cluster with intrinsic dimensionality \( d_m \), at least \( d_m \) distances \( d'(O_1, nx_i^1), i = 1, \cdots, d_m \) are needed to estimate the position of center \( O_1 \), and in this case \( f(o) \) is given by

\[ f(o) = \sum_{i=1}^{d_m} \| d(O_1, nx_i^1) - d'(O_1, nx_i^1) \| \]

If we can not find sufficient \( d'(O_1, nx_i^1) \) to locate the center, there must be many inter-cluster points located in space as illustrated in Fig. 1(b). In this case, we have \( \angle nx_1^1 nx_2^1 nx_3^1 + \angle nx_1^1 nx_3^1 nx_2^1 \geq 180^\circ \), when the center \( O_1 \) could never be in the line of \( nx_1^1 nx_2^1 \) and \( nx_1^1 nx_3^1 \). In order to get a better preservation of inter-cluster relationship, \( O_1 \) should be placed as far as possible from these inter-cluster points. For a cluster with intrinsic dimensionality 2, it is suggested that \( O_1 \) should be chosen by

\[ O_1 = \arg \max_{o \in X_1} g(o) \]  \hspace{1cm} (9)

where

\[ g(o) = d(o, nx_1^1) + d(o, nx_2^1) - \| d(o, nx_1^1) - d(o, nx_2^1) \| \]

If the intrinsic dimensionality of \( X_1 \) is \( d_m \) and \( \{ nx_i^1, i = 1, \cdots, d_m \} \) is the set of inter-cluster points in \( X_1 \), the function \( g(o) \) should be:

\[ g(o) = \sum_{i=1}^{d_m} \left( d(o, nx_i^1) + d(o, nx_i^1) - \| d(o, nx_i^1) - d(o, nx_i^1) \| \right) \]

2) Degenerative and unworkable cases: As original D-C Isomap algorithm relies on the position of centers to preserve inter-cluster relationship, the algorithm can not work on data under some circumstances. Considering about a simple case of two data clusters with \( d = 2 \), the method does not work because that it implicitly requires an another data cluster to provide sufficient rotation reference data points. Because that the low dimensional embedding of each clusters are relocated by referring to position of the centers. When there are three or more clusters and the centers of them are nearly in a line, the original D-C Isomap can not find the exact rotation matrix.

Therefore, we propose an algorithm to solve the problems by adding new clusters. This algorithm applies a trial and error procedure to determine the position of the new clusters. In the following, the case about two clusters is used as an example.

![Fig. 1](image1.png)

**Fig. 1**

**Two basic cases of the relationship of the center and inter-manifolds points**

![Fig. 2](image2.png)

**Fig. 2**

**An illustration of how to add a new cluster for D-C Isomap algorithm.**
As shown in Fig. 2, the nearest couple of inter-cluster points of clusters \( X^1 \) and \( X^2 \) are assumed to be \( n_x^1 \) and \( n_x^2 \), \( m_1 \) is the middle point of \( n_x^1 \) and \( n_x^2 \). The second nearest couple of inter-cluster points are \( n_x^1 \) and \( n_x^2 \), \( m_2 \) is the middle point of them. Then the third cluster \( X^3 \) is suggested to be produced by

\[
X^3 = m_1 + \gamma \| n_x^1 - n_x^2 \| \times \frac{m_2 - m_1}{\| m_2 - m_1 \|}
\]

The parameter \( \gamma \) can be decided by a trial and error procedure. Given a positive value \( \beta > 1 \), it is assumed that \( X^3 \) should satisfies

\[
\frac{1}{\beta} < \frac{\| X^3 - X^1 \|}{\| X^3 - X^2 \|} < \beta
\]

where \( \| X^3 - X^1 \| \) is the shortest distance between data points from clusters \( X^1 \) and \( X^3 \). If condition (10) is not satisfied, \( \gamma \) changes in a pre-given range like \{ \cdots , -3, -2, -1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{10}, \frac{1}{9}, 1, 2, \cdots \}. When there are \( M \) clusters in the data set with \( M < d + 1 \), we can start from the couple of clusters with maximum nearest inter-cluster distance. Assume that \( X^1 \) and \( X^2 \) satisfy

\[
\| X^1 - X^2 \| = \max \min |X^i - X^j|
\]

and \( n_x^1, n_x^1, m_1, n_x^2, n_x^2, m_2 \) are defined as above. The \( M + 1 \)-th cluster \( X^{M+1} \) could be generated as

\[
X^{M+1} = m_1 + \gamma \| n_x^1 - n_x^2 \| \times \frac{m_2 - m_1}{\| m_2 - m_1 \|}
\]

If \( X^p \) and \( X^q \) are the two nearest clusters of \( X^{M+1} \), given \( \beta > 0 \), it is assumed that \( X^{M+1} \) should satisfies

\[
\frac{1}{\beta} < \frac{\| X^{M+1} - X^p \|}{\| X^{M+1} - X^q \|} < \beta.
\]

If \( M + 1 < d + 1 \), replace \( M \) by \( M + 1 \) and repeat the generating procedure presented above.

PCA can be used to find out the dimensionality of the subspace on which the centers are lying. A new cluster is also needed if the dimensionality of the subspace is lower than \( d \). New clusters should be added until the centers could anchor a \( d \) dimensional simplex.

3) The complete algorithm of the revised D-C Isomap:
To give a compact representation of the revised D-C Isomap algorithm, and compare the difference between the revised and original D-C Isomap algorithm, the integral revised D-C Isomap algorithm is presented as bellow:

**Input:**

\[ X = \{ x_i \}_{i=1}^N, \text{ with } x_i \in \mathbb{R}^D. \text{ Initial neighborhood size } k \text{ or } \varepsilon. \]

**Step I.1**

The same to Step I.1 of the original D-C Isomap algorithm.

**Step I.2**

Estimate parameters, intrinsic dimensionality \( \{d_m\}_{m=1}^M \) and neighborhood sizes \( \{k_m\}_{m=1}^M \) of clusters. Let \( d = \max d_m \) and rebuild neighborhood graphs on clusters. If \( M < d + 1 \), new clusters should be added until \( M \geq d + 1 \).

**Step I.3-4**

The same to Step I.2-3 of the original D-C Isomap algorithm.

**Step II.1**

Centers of clusters are computed by (8) or (9). New clusters should be added until centers could anchor a \( d \) dimensional simplex.

**Step II.2-4**

The same as Step II.2-4 of the original D-C Isomap. It is assumed that \( Y^m \) is transformed to \( TY^m \).

**Step II.5**

\[ Y = \cup_{m=1}^M TY^m \text{ is the final output.} \]

V. Experiments

A. 3-D data sets

In this subsection, we compare k-CC Isomap, M-Isomap and the revised D-C Isomap on three 3-D data sets. It should be noted that during all experiments, the size of the neighborhood is chosen corresponding to the best performance of each algorithm.

Fig. 3(a) is a two-manifolds data set with \( N = 1200 \) data points, and the data set is generated by the following matlab code:

\[
t = (1^* \pi/6)^*(1+2^* \text{rand}(1,N));
xx = t.^* \text{cos}(t);yy = t.^* \text{sin}(t);
zz = [\text{unifrnd}(1,10,1,N/2) \text{ unifrnd}(16,25,1,N/2)];
X = [xx;zz;yy];
\]

It can be seen that each data manifold is intrinsically a rectangular region with 600 data points. Fig. 3(b) shows the result got by k-CC Isomap, whose neighborhood graph is constructed by using 8-CC method. It can be seen that the embedding shrinks along the edges in low dimensional space and edges of the embedding turn into noisy. Fig. 3(c) shows the result got by M-Isomap method with neighborhood size \( k = 8 \). As it can be seen, each of data manifold is exactly unrolled, and the inter-manifolds distance is precisely preserved. Fig. 3(d) illustrates the initialization step of the revised D-C Isomap algorithm. First, two data manifolds \( X^1 \) and \( X^2 \) are identified. Then the third data cluster \( X^3 \) is constructed, where the parameter \( \lambda = 0.1 \). Finally, centers \( O_1 \) and \( O_2 \) of the data manifolds are computed by referring to the nearest neighbors. The center of \( X^3 \) is also the data point \( X^3 \). Fig. 3(e) shows the result of the revised D-C Isomap method. It can be seen that the embedding exactly preserves both the intra-manifold distances and inter-manifolds distances.

Fig. 4(a) is another two-manifolds data set with \( N = 1200 \) data points, and the data set is generated by the following matlab code:

\[
t = (1^* \pi/6)^*(1+2^* \text{rand}(1,N));
xx = t.^* \text{cos}(tt);yy = t.^* \text{sin}(tt);
zz = [\text{unifrnd}(1,10,1,N/2) \text{ unifrnd}(16,25,1,N/2)];
X = [xx;zz;yy];
\]

Each data manifold has 600 data points. One data manifold is a
Fig. 3
Experiments on two-manifolds data set. (a) The original data set. (b) The result by k-CC Isomap. (c) The result by M-Isomap. (d) Illustration of the procedure of D-C Isomap. (e) The result by D-C Isomap.

Fig. 4
Experiments on two-manifolds data set. (a) The original data set. (b) The result by k-CC Isomap. (c) The result by M-Isomap. (d) Illustration of the procedure of D-C Isomap. (e) The result by D-C Isomap.

rectangular region and another data manifold is a round region. Fig. 4(b) shows the result got by k-CC Isomap with neighborhood size $k = 10$. It can be seen that the rectangular region bent outwards and the round region is prolonged. Fig. 4(c) shows the result got by M-Isomap method with the neighborhood size $k = 8$. As it can be seen, every data manifolds is exactly unrolled, and the inter-manifolds relationship is precisely preserved. Fig. 4(d) illustrates the initialization step of the revised D-C Isomap algorithm. The parameter $\lambda = 0.5$ for production of the new cluster $X^2$. Fig. 4(e) shows the result of the revised D-C Isomap method with neighborhood size $k = 5$. It is can be seen that the embedding exactly preserves both the intra-manifold distances and inter-manifolds distances.

Fig. 5(a) shows a three-manifolds data set with $N = 1600$ data points on the Swiss roll manifold. The data set is generated by the following matlab code:
There are three rectangular regions on the Swiss roll manifold. The longest data manifold has 800 data points, the other two manifolds contain face images of five persons. It can be observed that data manifolds of faces are mixed up, and the intra-face information is also not preserved. Fig. 6(c) is the result got by k-CC Isomap method with \( k = 3 \). As it can be seen, although the data points are clustered, their inter-face distances are not well preserved. The five lines mix up at one of their endings. Fig. 6(d) shows the result got by M-Isomap method with \( k = 3 \). Due to the limitation of k-NN method in clustering, only two data manifolds are identified. Although the data set is not well clustered, the result of M-Ismap shows that the low dimensional embedding can be separated up easily. Fig. 6(e) is the result got by the original D-C Isomap method, where the faces are split up beforehand. The circumcenters are used as their centers. However, as it can be seen, two faces are mixed up.

**B. Real world data sets**

Fig. 6(a) shows some samples of the faces data [38] which contains face images of five persons. The dataset consists of 153 images and has 34, 35, 26, 24, 34 images corresponding to each face. These images are gray scale with resolution of 112\times92. They are transformed into vectors in 10304-dimensional Euclidean space. In order to show the inter-manifolds relationship with more details, we embed the data into 3-dimensional space. Fig. 6(b) is the three dimensional embedding by PCA method. It can be observed that data manifolds of faces are mixed up, and the intra-face information is also not preserved. Fig. 6(c) is the result got by k-CC Isomap method with \( k = 3 \). As it can be seen, although the data points are clustered, their inter-face distances are not well preserved. The five lines mix up at one of their endings. Fig. 6(d) shows the result got by M-Isomap method with \( k = 3 \). Due to the limitation of k-NN method in clustering, only two data manifolds are identified. Although the data set is not well clustered, the result of M-Ismap shows that the low dimensional embedding can be separated up easily. Fig. 6(e) is the result got by the original D-C Isomap method, where the faces are split up beforehand. The circumcenters are used as their centers. However, as it can be seen, two faces are mixed up.

Fig. 7(a) shows some samples of the teapot data set with 300 data points, where ‘\( \square \)’ stands for the teapot bird-view images, ‘\( \Delta \)’ stands for the teapot back-forth rotation images and ‘\( \circ \)’ stands for the teapot side-view images. Each of the images is an 80\times60\times3 RGB colored picture, i.e. a vector in 14400 dimensional input space. Because the data points do not distribute on a single global manifold, this problem will poses a great challenge to classical manifold learning methods. The experiment shows that the three data manifolds can be identified by k-CC method. In order to show their exact embedding, Fig. 7(b)-(d) present the embedding of each data manifold by classical Isomap with neighborhood size \( k = 3 \). Fig. 7(e) shows the result got by PCA method. It can be seen that the data set is clustered, but the shape of each embedding is deformed because of the linear characteristic of the PCA method. Fig. 7(f) is the result got by k-CC Isomap method with neighborhood size \( k = 3 \). The bad approximations of inter-manifolds geodesics lead to the deformation of the embedding in low dimensional space. Fig. 7(g) shows the result by M-Isomap method with neighborhood size \( k = 3 \). The data set is clearly clustered and intra-manifolds relationships are exactly preserved.

\[
t1 = [\text{unifrnd}(\pi/5, 16/12, 1, N/4)];
t2 = [\text{unifrnd}(\pi/18, 12/6, 1, N/4)];
t3 = (5\pi/6) \times (1 + 7/5 \times \text{rand}(1, N/2));
a1 = t1.\ast \cos(t1); \quad b1 = t1.\ast \sin(t1);
c1 = [\text{unifrnd}(-1, 3, 1, N/4)];
a2 = t2.\ast \cos(t2); \quad b2 = t2.\ast \sin(t2);
c2 = [\text{unifrnd}(-1, 3, 1, N/4)];
a3 = t3.\ast \cos(t3); \quad b3 = t3.\ast \sin(t3);
c3 = [\text{unifrnd}(6, 10, 1, N/2)];
x1 = [a1; c1; b1]; \quad x2 = [a2; c2; b2]; \quad x3 = [a3; c3; b3];
Z = [x1, x2, x3];
\]

\[http://www.cs.toronto.edu/~roweis/data.html\]
input dimension of the teapot data set, we increase the dimension of scales of two embedding can be comparable. Teapot data vectors scale of the teapot data set should also be changed such that the IsoFACE set by adding zeros to the bottom of these vectors. The 1 to 5 scaled by 100, i.e. the scale of teapot data points shrink with neighborhood size $k$. Fig. 7(h) shows the result got by revised D-C Isomap method. We can see that the shape of IsoFACE data is distorted badly. Fig. 8(e) is the result got by M-Isomap method with neighborhood size $k=5$. The performance is significantly improved compared with k-CC Isomap. Fig. 8(f) is the result got by the revised D-C Isomap method.

**C. Discussion**

In our experiments, there are several important properties which should be considered:

1) As k-CC Isomap tries to preserve poor and good approximations of geodesics simultaneously, its low dimensional embedding is usually deformed. This method works well if each data manifold has comparable number of data points and the data manifolds can not be very far from each other, and the algorithm does not work well otherwise.
The generalization performance of classical Isomap, k-CC Isomap, Original D-C Isomap, revised D-C Isomap and M-Isomap for multi-manifolds learning.

| method            | density | dimensionality | Isometric | generalization |
|-------------------|---------|----------------|-----------|----------------|
| classical         | Δ       | Δ              | Δ         | Δ              |
| k-CC              | ○       | ○              | Δ         | ○              |
| Original D-C      | ○       | ○              | ○         | ○              |
| revised D-C       | ○       | ○              | ○         | ○              |
| Multi-            | ○       | ○              | ○         | ○              |

2) The revised D-C Isomap overcomes its original limitations, meanwhile, the robustness of the algorithm is also enhanced by adding a new cluster.

3) M-Isomap connects data manifolds with multiple edges, which can control the rotation of the low dimensional embedding, and at the same time, better preserve inter-manifolds distance. Like the D-C Isomap algorithm, it can also isometrically preserve intra-manifold geodesics and inter-manifolds distances.

To sum up, Table III shows the comparison of the general performance of the five versions of Isomap algorithms: classical Isomap, k-CC Isomap, Original D-C Isomap, revised D-C Isomap and M-Isomap. The labels "Δ" stands for poor performance, "○" stands for not bad and "□" stands for good. Density means the generalize ability on manifolds with different density, i.e. different neighborhood sizes; dimensionality means the generalization ability on manifolds with different intrinsic dimensionality; isometric means the property of isometry in preserving the inter and intra-manifold relationship; finally the generalization means the overall generalization ability to learn data from multiple manifolds.

VI. Conclusion

In this paper, the problem of multi-manifolds learning is presented and defined for the first time. A general procedure for isometric multi-manifolds learning is proposed. The procedure can be used to build multi-manifolds learning algorithms which are not only able to faithfully preserve intra-manifold geodesic distances, but also the inter-manifolds geodesic distances. M-Isomap is an implementation of the procedure and shows promising results in multi-manifolds learning. Compared with k-CC Isomap, it has the advantage of low computational complexity. With the procedure, the revised D-C Isomap becomes more effective in learning multi-manifolds data sets. Future work will be conducted on the applications of the multi-manifolds learning algorithms.

REFERENCES

[1] I. T. Jolliffe, “Principal Component Analysis,” Springer-Verlag, New York, 1989. ISBN 0-387-96269-7
[2] T. F. Cox, M. A. Cox, “Multidimensional Scaling,” Chapman & Hall, London, 2001. ISBN 1-58488-094-5.
[3] Eric Mjolsness, Dennis DeCoste, “Machine Learning for Science: State of the Art and Future Prospects,” Science, vol. 293, pp. 2051-2055, Sep. 2001.
[4] J.B. Tenenbaum, V. de Silva, and J. C. Landford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, pp. 2319-2323, Dec. 2000.
[5] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by local linear embedding,” Science, vol. 290, pp. 2323-2326, Dec. 2000.
[6] H. S. Seung, D. D. Lee, “The manifold ways of perception,” Science, vol. 290, pp. 2268-2269, Dec. 2000.
[7] M. Belkin and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation,” Neural Computation, vol. 15, no 6, pp. 1373-1396, June 2003.
[8] D. Donoho, C. Grimes, “Hessian Eigenmaps; New Locally Linear Embedding Techniques for High-Dimensional Data,” Proc. National Academy of Sciences, vol. 100, no. 10, pp. 5591-5596, 2003.
[9] Zhenyue Zhang, Hongyuan Zha, “Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment,” SIAM Journal on Scientific Computing , vol. 26, issue. 1, pp. 313-338, 2005.
[10] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, S. W. Zuck, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” Proc. National Academy of Sciences, vol. 102, no. 21, pp. 7426-7431, May. 2005.
[11] Tong Lin, Hongyuan Zha, “Riemannian Manifold Learning,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pp. 796-809, May 2008.

[12] Hongyuan Zha, Zhenyue Zhang, “Continuum Isomap for manifold learning,” Computational Statistics & Data Analysis, vol. 52, issue. 1, pp. 184-200, Sep. 2007.

[13] Martin H. C. Law, A. K. Jain, “Incremental Nonlinear Dimensionality Reduction by Manifold Learning,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pp. 377-391, March 2006.

[14] M. Bernstein, V. de Silva, J. C. Langford, J. B. Tenenbaum, “Graph approximations to geodesics on embedded manifolds,” Technical report, Dept. of Psychology, Stanford Univ., Dec. 2000.

[15] Maurizio Filippone, Francesco Camastra, Francesco Masulli, Stefano Rovetta, “A survey of kernel and spectral methods for clustering,” Pattern Recognition, vol. 41, pp. 176-190, May 2007.

[16] Ming-Hsuan Yang, “Extended Isomap for Pattern Classification,” ICPR 2002: Proceedings - International Conference on Pattern Recognition, vol. 3, pp. 30615, Aug. 2002.

[17] Xiaofei He, Shuicheng Yan, Tuixiao Hu, P. Niyogi, Hong-jian Zhang, “Face recognition using Laplacianfaces,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, issue.3, pp. 328-340, March 2005.

[18] M. Belkin, V. Sindhwani, and P. Niyogi, “Manifold Regularization: a Geometric Framework for Learning from Examples,” Journal of Machine Learning Research, vol. 7, pp. 2399-2434, Dec. 2006.

[19] Xin Geng, De-Chuan Zhang, Zhi-hua Zh, “Supervised nonlinear dimensionality reduction for visualization and classification,” IEEE Trans on Systems, Man, and Cybernetics Part B, vol. 35, no. 6, pp. 1098-1107, Dec 2005.

[20] O. C. Jenkins, Maja J. Mataric, “A Spatio-temporal Extension to Isomap Nonlinear Dimension Reduction,” ACM. Proc. 21st Int’l Conf. on Machine learning, vol. 69, pp. 56-66, 2004.

[21] A. Rahimi, B. Recht, T. Darrell, “Learning to Transform Time Series with a Few Examples,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp. 1759-1775, Oct. 2007.

[22] J.B. Tenenbaum, V. de Silva, and J. C. Landford “Respond to Comments on the Isomap Algorithm and Topological Stability,” Sciences, vol. 295, no. 5552, pp. 7, Jan. 2002.

[23] Yiming Wu, Kap Luk Chan “An Extended Isomap Algorithm for Learning Multi-Class Manifold,” Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3429-3433, Aug. 2004.

[24] Deyu Meng, Yee Leung, Tung Fung, Zongben Xu “Nonlinear Dimensionality Reduction of Data Lying on the Multicluster Manifold,” IEEE Trans on Systems, Man, and Cybernetics Part B, vol. 38, issue. 4, pp. 1111-1122, Aug. 2008.

[25] Bo Li, De-Shuang Huang, Chao Wang, Kun-Hong Liu “Feature extraction using constrained maximum variance mapping,” Pattern Recognition, vol. 41, pp. 3287-3294, May 2008.

[26] Li Yang, “K-Edge Connected Neighborhood Graph for Geodesic Distance Estimation and Nonlinear Projection,” Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04), vol. 1, pp. 196-199, 2004.

[27] Li Yang, “Building k-Edge-Disjoint Spanning Trees of Minimum Total Length for Isometric Data Embedding,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1680-1683, Oct. 2005.

[28] Li Yang, “Building k-Edge-connected neighborhood graph for distance-based data projection,” Pattern Recognition Letters, vol. 26, issue. 13, pp. 2015-2021, Oct. 2005.

[29] Li Yang, “Building k-Connected Neighborhood Graphs for Isometric Data Embedding,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, issue. 5, pp. 827-831, May 2006.

[30] Dongfang Zhao, Li Yang, “Incremental Construction of Neighborhood Graphs for Nonlinear Dimensionality Reduction,” International Conference on Pattern Recognition, vol. 28, issue. 5, pp. 827-831, May 2006.

[31] Dongfang Zhao, Li Yang, “Incremental Isometric Embedding of High-Dimensional Data Using Connected Neighborhood Graphs,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 1, pp. 86-98, Jan. 2009.

[32] O. Samko, A. D. Marshall, P. L. Rosin “ Selection of the optimal parameter value for the Isomap algorithm,” Pattern Recognition Letters vol. 27, issue. 9, pp. 968-979, Feb. 2006.

[33] R. Xu, D. Wunsch II, “Survey of clustering algorithms,” IEEE Trans on Neural Networks, vol. 16, no. 3, pp. 645-678, May. 2005.

[34] F. Camastra and A. Vinciarelli, “Estimating the intrinsic dimension of data with a fractal-based approach,” IEEE Trans. Pattern Analysis and Machine Intelligence vol. 24, no. 10, pp. 1404-1407, Oct. 2002.

[35] E. Levina and P. J. Bickel “Maximum likelihood estimation of intrinsic dimension,” NIPS 04: Neural Information Processing Systems
(a)
kcc Isomap on data set A
kcc Isomap on data set B
kcc Isomap on data set C
kcc Isomap on teapot data set
Isometric Multi-Manifolds Learning

Mingyu Fan, Hong Qiao, Senior Member, IEEE, and Bo Zhang

Abstract—Isometric feature mapping (Isomap) is a promising manifold learning method. However, Isomap fails to work on data which distribute on clusters in a single manifold or manifolds. Many works have been done on extending Isomap to multi-manifolds learning. In this paper, we first proposed a new multi-manifolds learning algorithm (M-Isomap) with help of a general procedure. The new algorithm preserves intra-manifold geodesics and multiple inter-manifolds edges precisely. Compared with previous methods, this algorithm can isometrically learn data distributed on several manifolds. Secondly, the original multi-cluster manifold learning algorithm first proposed in [24] and called D-C Isomap has been revised so that the revised D-C Isomap can learn multi-manifolds data. Finally, the features and effectiveness of the proposed multi-manifolds learning algorithms are demonstrated and compared through experiments.

Index Terms—Isomap, nonlinear dimensionality reduction, manifold learning, pattern analysis, multi-manifolds learning.

I. INTRODUCTION

Challenges, known as "the curse of dimensionality", are usually confronted when scientists are dealing with high dimensional data. Dimensionality reduction is a promising tool to circumvent these problems. Principal component analysis (PCA) [1] and multidimensional scaling (MDS) [2] are two important linear dimensionality reduction methods. Due to their linear model assumption, both of the methods will fail to discover nonlinear intrinsic structures of data.

Recently, there are more and more interests in nonlinear dimensionality reduction (NLDR). NLDR is used to learn nonlinear intrinsic structures of data, which is considered to be the first step of machine learning and pattern recognition [3]. Two interesting nonlinear dimensionality reduction methods based on the notion of manifold learning [6], isometric feature mapping (Isomap) [4] and local linear embedding (LLE) [5], have been introduced in SCIENCE 2000. LLE assumes that data points locally distribute on a linear patch of a manifold. It preserves local linear coefficients, which best reconstruct each data point by its neighbors, into a lower dimensional space. Isomap is based on the classical MDS method. Instead of preserving pairwise Euclidean distance, it preserves the geodesic distance on the manifold.

The geodesics between two data points is approximated by the shortest path on a constructed graph. Both of the methods are computationally efficient and able to achieve global optimality. There are also many other important nonlinear dimensionality reduction methods. Laplacian eigenmap [7] utilizes an approximation of the Laplace-Beltrami operator on manifolds to provide an optimal embedding. Hessian LLE [8] resembles Laplacian eigenmap by using an approximation of the Hessian operator instead of Laplacian operator. The local tangent space alignment (LTSA) [9] method learns the local geometry by constructing a local tangent space of each data point and then aligns these local tangent spaces into a single global coordinates system with respect to the underlying manifold. Diffusion maps [10] applies diffusion semigroups to produce multi-scale geometries to represent complex structures. Riemannian manifold learning (RML) [11] method uses the constructed Riemannian normal coordinate chart to map the input data into a lower dimensional space. NLDR is a fast growing research activity and has been proved very useful in many fields and applications, such as classification using Isomap [16] and Laplacian eigenmap [17], geometry based semi-supervised learning method using Laplacian eigenmap [18], data visualization [19] and time series manifold learning [20], [21].

As Isomap emphasizes on the global geometric relationship of data points, it is very illustrative in data visualization and pattern analysis [13]. Although Isomap algorithm implicitly requires the data set to be convex [8], it still provides very meaningful results on non-convex data sets. In this paper, we will focus on extending Isomap to multi-manifolds learning. The first step of Isomap algorithm is to construct a neighborhood graph which connects all the data points. This step is of vital importance because the success of the following steps depends on how well the constructed neighborhood graph is. However, it is hard to build a totally connected neighborhood graph in order to guarantee the topological stability of the classical Isomap algorithm when points of the data set distribute on clusters in a manifold or manifolds (multiple manifolds). It should be remarked that several methods have been proposed to extend Isomap to multi-manifolds data, and some of them are based on providing new neighborhood graph construction algorithms. For example, Wu et al [23] introduced a split and augment procedure for neighborhood graph construction which could produce a totally connected neighborhood graph. In a series of papers [26]–[29]. Yang introduced several neighborhood graph construction algorithms using techniques from discrete mathematics, graph theory, Meng et al [24] proposed a decomposition and composition Isomap (D-C Isomap).

The rest of the paper is organized as follows. In Section II the main issues and limitations of the classical Isomap algorithm are presented. The problem of multi-manifolds learning is also investigated. In Section III previous methods on multi-manifolds learning are briefly introduced and discussed. In Section IV a general procedure for designing multi-manifolds learning algorithms is first proposed. With the proposed procedure, a new multi-manifolds learning algorithm (M-Isomap) is then designed and analyzed. In addition, the original D-C Isomap algorithm is revised to overcome its main limitation. In Section V the effectiveness of these multi-manifolds learning algorithms has been demonstrated by experiments. Comparisons of these algorithms have also been made. Some concluding remarks are provided in Section VI.
II. CLASSICAL ISOMETRIC FEATURE MAPPING AND ITS LIMITATIONS

Isomap is an efficient NLDR algorithm to recover the intrinsic geometric structure of a data set if the data points lie on a single manifold [4]. Assume that the data set $X = \{x_1, x_2, \cdots, x_N\}$ is in a high dimensional space $\mathbb{R}^D$ and the feature space is $\mathbb{R}^d$. Then the classical Isomap algorithm has the following three steps [4].

Step 1: Identify the neighbors of all the data points to construct a neighborhood graph. With the given parameter $k$ or $\epsilon$, there are two ways to construct a neighborhood graph for $X$:
- if $x_j$ is one of $x_i$’s $k$ nearest neighbors, they are connected by an edge (the k-NN method).
- if $x_i$ and $x_j$ satisfy $\|x_i - x_j\| < \epsilon$, they are connected by an edge (the $\epsilon$-NN method).

Step 2: Use Dijkstra’s or Floyd-Warshall’s algorithm to compute the length of the shortest path $d_G(x_i, x_j)$ between any two data points $x_i$ and $x_j$ on the graph. It is proved that $d_G(x_i, x_j)$ is a good approximation of the geodesic distance $d_D(x_i, x_j)$ on the manifold as the number of data points increases.

Step 3: Perform the classical MDS on the graph distance matrix $D_G$ whose $(i, j)$-th element is $d_G(i, j)$. Minimize the cost function

$$E(Y) = \|\tau(D_G) - \tau(D_Y)\|_F,$$

The operator $\tau$ is defined as $\tau(D) = -\frac{1}{2} HSH$, where $H = I - \frac{1}{N} ee^T$ with $I$ the identity matrix and $e = (1, 1, \cdots, 1)^T$, $S = (D^2_{ij})$ with $D_{ij}$ being the $(i, j)$-th element of $D$ and $D_{ij} = (\|y_i - y_j\|)$. Assume that, in descending order, $\lambda_i$ is the $i$-th eigenvalue of $\tau(D_G)$ with the corresponding eigenvector $v_i$. Then the low-dimensional embedding $Y$ is given by:

$$Y = [y_1, y_2, \cdots, y_N] = \begin{bmatrix} \sqrt{\lambda_1}v_1^T \\ \vdots \\ \sqrt{\lambda_d}v_d^T \end{bmatrix}$$

The property of the Isomap algorithm is well understood [12], [14]. However, the success of the Isomap algorithm depends on two issues. One is how to choose the correct intrinsic dimensionality $d$. Setting a lower dimensionality $d$ will lead to a loss of data structure information. On the other hand, setting a higher dimensionality $d$ will make some redundant information to be kept. This issue has been well investigated so far. The other issue is the quality of the constructed neighborhood graph. It is known that constructing an appropriate neighborhood graph is still a tricky task. Both the k-NN and $\epsilon$-NN methods have their limitations. Under the assumption that data points distribute on a single manifold, if the neighborhood size $k$ or $\epsilon$ is chosen to be too small, the constructed neighborhood graph will be very sparse and therefore the geodesics cannot be satisfactorily approximated. On the other hand, if the neighborhood size $k$ or $\epsilon$ is chosen to be too large, short-circuit edges may occur which will have a significant negative influence on the topological stability of the Isomap algorithm [22].

Nonetheless, if data points distribute uniformly on one manifold, then both the “short circuit” problem and the “discontinuity” problem can be circumvented by carefully choosing an appropriate neighborhood size $k$ or $\epsilon$. However, if data points distribute on several clusters or manifolds, then neither of the k-NN method and the $\epsilon$-NN method can guarantee that the whole data set is totally connected and the geodesics is satisfactorily approximated.

However, both data missing and data mixture are common problems in practical data analysis. These two cases often cause data points to distribute on different clusters in a manifold or manifolds. Data points on different manifolds may have different input dimensionality $D$ (the dimensionality of the ambient space). This usually happens in the case of data mixture. On the other hand, learning different data manifolds may need different values of input parameters, that is, appropriate neighborhood size ($k$ or $\epsilon$) and intrinsic dimensionality $d$ for each data manifold. In this paper, we will focus on designing new multi-manifolds learning algorithms for data distributing on multiple manifolds. The case when data points distribute on pieces of a single manifold is referred to as multi-cluster manifold learning, while the case when data points distribute on multiple manifolds is referred to as multi-manifolds learning.

III. PREVIOUS WORKS ON MULTI-MANIFOLDS LEARNING

A. Multi-manifolds learning by new neighborhood graph construction method

Wu and Chan [23] proposed a split-augment approach to construct a neighborhood graph. Their method can be regarded as a variation of the k-NN method and can be summarized as below:

1. the k-NN method is applied to the data set. Every data point is connected with its neighbors. If the data lies on multiple manifolds, several disconnected graph components (data manifolds) will be formed.
2. Each couple of graph components are connected by their nearest couple of inter-components points.

This method is simple to implement and has the same computational complexity as the k-NN method has. However, as there is only one edge connecting every two graph components, geodesics across components are poorly approximated and, meanwhile, their low dimensional embedding can be rotated arbitrarily. This method cannot be directly applied to data lying on three or more data manifolds. If more than two graph components exist, the intra-component shortest distances may be changed in the totally connected graph.

Yang [26]–[29] introduced four methods to construct a connected neighborhood graph. The $k$ minimum spanning trees (k-MST) method [26] repeatedly extracts $k$ minimum spanning trees (MSTs) from the complete Euclidean graph of all data points. Edges of the $k$ MSTs form a $k$-connected neighborhood graph. Instead of extracting $k$ MSTs, the minimum-$k$-spanning trees (min-$k$-ST) method [27] finds $k$ edge-disjoint spanning trees from the complete Euclidean graph, and the sum of the total edge length of the $k$ edge-disjoint spanning trees attains a minimum. The $k$-edge-connected ($k$-EC) method [28] constructs a connected neighborhood graph by adding edges in a non-increasing order from the complete Euclidean graph. An edge is added if its two end vertices do not have $k$ edge-disjoint paths connected with each other. The $k$-vertices-connected ($k$-VC) method [29] adds edges in a non-increasing order from the complete Euclidean graph, where an edge is added if its two end vertices would be disconnected by removing some $k - 1$ vertices. And the constructed neighborhood graph would not be disconnected by removing any $k - 1$ vertices.

The methods introduced in [26]–[29] have the following advantages over the k-NN method. First, the local neighbor relationship
is affected by the global distribution of data points. This is beneficial for adaptively preservation of the global geometric metrics. Secondly, these methods can guarantee that the constructed neighborhood graph is totally connected. Compared with the k-NN method, Yang’s methods construct a neighborhood graph with more edges corresponding to the same neighborhood size \( k \). This property ensures the quality of the neighborhood graphs.

B. Multi-manifolds learning by decomposition-composition Isomap

In [24], Meng et al. proposed a decomposition-composition method (D-C Isomap) which extends Isomap to multi-cluster manifold learning. The purpose of the method is to preserve intra-cluster and inter-cluster distances separately. In the next section, we will introduce a revised version of the D-C Isomap to extend the application range of the original D-C Isomap. To this end, we present the details of the D-C Isomap algorithm as follows.

Step I: Decomposition process

1. Given an appropriate neighborhood size \( k \) or \( \varepsilon \), if data is comprised of multiple clusters, several disconnected graph components will be formed when each data point is connected with its neighbors by the \( k \)-NN or \( \varepsilon \)-NN method.

2. Assume that there are \( M \) graph components and a graph component is also considered as a cluster. Data points of the \( m \)-th cluster is denoted as \( X^m = \{x^m_1, \ldots, x^m_{n^m}\} \). Clusters \( X^m \) and \( X^n \) are connected by their nearest inter-cluster edge, whose ending vertices are assumed as \( x^m_{n^m} \) and \( x^n_{n^m} \) and edge length as \( d^m_{n^m} \).

3. Apply the \( k \)-NN Isomap or \( \varepsilon \)-NN Isomap on cluster \( X^m \). Denote by \( D^m = (d^m_{i,j}) \) the geodesic distance matrix for \( X^m \). By \( Y^m = \{y^m_1, \ldots, y^m_{n^m}\} \) the corresponding low-dimensional embedding to \( X^m \) and by \( y^m_{n^m} \) the embedding point corresponding to \( n^m_{n^m} \), where \( n^m_{n^m} \in Y^m \).

Step II: Composition process

1. The set of centers of clusters is denoted as \( CX = \{cx_1, \ldots, cx_M\} \), where each center is computed by

\[
    cx_m = \arg \min_{x \in X^m} \left( \max_{1 \leq i \leq n^m} (D^m_{i,j}) \right) \quad m = 1, \ldots, M.
\]

2. The distance matrix for \( CX \) can be computed by

\[
    \tilde{D} = \{\tilde{D}_{mn}\}, \quad \tilde{D}_{mn} = \begin{cases}
    d_{m,n} + d^m_{n^m} + d^m_{m,n} & m \neq n \\
    0 & m = n
\end{cases}
\]

where \( d_{m,n} \) is the distance of the shortest path between \( cx_m \) and \( x^m_{n^m} \) on the graph component \( X^m \).

3. Plug the distance matrix \( \tilde{D} \) and the neighborhood size \( d \) into the classical Isomap algorithm. The embedding of \( CX \) is denoted by \( CY = \{cy_1, \ldots, cy_M\} \subset \mathbb{R}^d \) (\( CY \) is called the translation reference set). Assume that the \( d \) nearest neighbors of \( cx_m \) are \( \{cx_{m_1}, \ldots, cx_{m_d}\} \). Then the low-dimensional representation corresponding to \( n^m_{n^m} \) can be computed as

\[
    Y^m_{n^m} = cy_m + \frac{d_{m,n^m} + d^m_{n^m} + d^m_{m,n}}{2} (cy_m - cy_n),
\]

where \( i = 1, \ldots, d \) and \( m = 1, \ldots, M \).

4. Construct the rotation matrix \( A_m \) for \( Y^m \) with \( m = 1, \ldots, M \). Assume that \( QN_m \) is the principal component matrix of

\[
    NY_m = \{ny^m_{1}, \ldots, ny^m_{n^m}\} \text{ and } QS_m \text{ is the principal component matrix of } SY_m = \{sy^m_{1}, \ldots, sy^m_{n^m}\} \text{. Then the rotation matrix for } Y^m \text{ is } A_m = QS_m QN_m^{-1}.
\]

5. Transform \( Y^m (m = 1, \ldots, M) \) into a single coordinate system by Euclidean transformations:

\[
    FY^m = \{fy^m = A_m y^m + cy_m, i = 1, \ldots, l_m\} \quad m = 1, \ldots, M.
\]

Then \( Y = \bigcup_{m=1}^{M} FY^m \) is the final output.

First, the D-C Isomap algorithm reduces the dimensionality of clusters separately and meanwhile, preserves a skeleton of the whole data. Secondly, using the Euclidean transformations, the embedding of each cluster is placed into the corresponding position by referring to the skeleton. In this way, the intra-cluster geodesics are exactly preserved. Since the D-C Isomap method uses circumcenters to construct the skeleton of the whole data, its learning results depend on the mutual position of these circumcenters, which would make the learning results unstable. On the other hand, it is known that at least \( d + 1 \) reference points are needed to anchor a \( d \)-dimensional simplex. However, in the D-C Isomap algorithm, the number of the reference data points is limited by the number of clusters.

C. Constrained Maximum Variance Mapping

Recently in [25], Li et al. proposed the constrained maximum variance mapping (CMVM) algorithm for multi-manifolds learning. The CMVM method is proposed based on the notion of maximizing the dissimilarity between classes while holding up the intra-class similarity.

IV. ISOMETRIC MULTI-MANIFOLDS LEARNING

In this section, we first introduce a general procedure for the design of isometric multi-manifolds learning algorithms and then present our new multi-manifolds learning algorithm called M-Isomap. Finally, we make a revision of the original D-C Isomap algorithm to extend its application range.

A. The general procedure for isometric multi-manifolds learning

Many previous methods extend Isomap to multi-manifolds learning by revising the neighborhood graph construction step of the Isomap algorithm [23], [26]–[29]. However, the shortest paths across clusters or data manifolds are bad approximations of geodesics. In Isomap, bad local approximation always leads to deformation of the global low-dimensional embedding.

Assumed that \( \Omega \) is an open, convex and compact set in \( \mathbb{R}^d \) and \( f : \Omega \rightarrow \mathbb{R}^d \) is a continuing mapping, where \( d << D \). Then \( f(\Omega) = M \) is defined as a \( d \) dimensional parameterized manifold. Let \( K(x,y) = d_{\Omega}(x,y) \) be a specially defined kernel. Then re-producing kernel Hilbert space (RKHS) \( \mathcal{H} \) is constructed with this kernel. Denote by \( \phi_j(x) \) the eigenfunction corresponding to the \( j \)-th largest eigenvalue \( \lambda_j \) of \( K(x,y) \) in \( \mathcal{H} \), which is also the \( j \)-th element of the Isomap embedding. The geodesic distance on the manifold \( M \) is written as

\[
    d^2(x,y) = d^2(f(\tau), f(\hat{\tau})) = \alpha ||\tau - \hat{\tau}|| + \eta(\tau, \hat{\tau}),
\]

where \( \tau, \hat{\tau} \in \Omega \), \( \alpha \) is a constant and \( \eta(\tau, \hat{\tau}) \) is the deviation from isometry. The constant vector

\[
    C = \frac{\int_M x p(x) dx}{\int_M \rho(x) dx} = \frac{\int_{\Omega} \tau H(\tau) d\tau}{\int_{\Omega} \rho(\tau) d\tau}
\]
TABLE I
Symbols and Variables used in the Algorithms

| Symbol | Definition |
|--------|------------|
| X = \{x_i\}_{i=1}^N | The total data set, with \(x_i \in \mathbb{R}^D\) |
| X^m = \{x^m_i\}_{i=1}^{m_n} | The m-th data manifold, where \(m = 1, \ldots, M\) |
| Y^m = \{y^m_i\}_{i=1}^{m_n} | Low dimensional embedding of \(X^m\) |
| \(D_{mn} = (d(x^m_i, x^n_j))\) | Matrix of geodesic distances across data manifolds \(X^m\) and \(X^n\) |
| \(f^m_{xm}, f^m_{xn}\) | The furthest couple of data points in \(X^m\) and \(X^n\), with \(f^m_{xm} \in X^m, f^m_{xn} \in X^n\) |
| \{x^m_{\hat{M}(i)}\}_{i=1}^{m_n} | Subset of \(X^m\) whose elements are the nearest data points to \(X^n\) |
| \(I^m = \{x^m_i\}_{i=1}^{m_n}\) | The selected data points from \(X^m\) to construct the skeleton \(I\) |
| \(Y^m_I\) | Low-dimensional embedding of \(I^m\) with \(Y^m_I \subset Y^m\) |
| \(I = \bigcup_{m=1}^M I^m\) | Points which are used to construct a skeleton of \(X\) |
| \(R_{Y_I}\) | Low-dimensional embedding of the skeleton \(I\) |
| \(R_{Y_I}^m\) | Low-dimensional embedding of \(I^m\) with \(R_{Y_I}^m \subset R_{Y_I}\). \(R_{Y_I}^m\) will also be referred to as the transformation reference for \(Y^m\) |
| \(n_{x^m_{\hat{M}^j}}\) | The point in \(X^m\) which is the nearest to \(X^j\) or the inter-cluster point in the D-C Isomap algorithm. |

where \(\rho(x)\) and \(\mathcal{H}(\tau)\) are density functions of \(M\) and \(\Omega\). With the above notations, the following theorem is proved by Zha et al [12].

**Theorem 4.1:** There is a constant vector \(P_j\) such that \(\phi_j(x) = P_j^T(\tau - C) + \epsilon_j(\tau)\). Here \(\epsilon_j(\tau) = \epsilon_j(0) - \epsilon_j(\tau)\) has zero mean, that is,

\[
\int_{\Omega} \mathcal{H}(\tau)\epsilon_j(\tau)d\tau = 0 ,
\]

\[
\epsilon_j(\tau) = \frac{1}{2J} \int_{\Omega} \eta(\tau, \hat{\tau})\mathcal{H}(\hat{\tau})\phi_j(x)d\hat{\tau},
\]

\[
\epsilon_j(0) = \frac{1}{2J} \int_{\Omega} \mathcal{H}(\tau)d\tau \int_{\Omega} \epsilon_j(\tau)\mathcal{H}(\tau)d\tau.
\]

By Theorem 4.1 even if the deviation \(\eta(\tau, \hat{\tau})\) is not zero with only a limited range of \((\tau, \hat{\tau})\), then the coordinate of the low-dimensional embedding \(\phi_j(x)\) is still deformed with the deformation being measured by \(\epsilon_j(\tau)\).

In order to get a better understanding of multi-manifolds data, it is profitable to preserve intra-manifold relationship (where \(\eta(\tau, \hat{\tau}) = 0\)) and inter-manifolds relationship (where \(\eta(\tau, \hat{\tau}) \neq 0\)) separately. This is because sometimes we care more about the information within the same data manifold. Here we propose a general procedure for the design of isometric multi-manifolds learning algorithms.

**Step I:** The decomposition process

1. Cluster the whole data set. If data distribute on multiple clusters in a manifold or manifolds, the clusters or manifolds should be identified. Many clustering methods can be used for this; for example, K-means, Isodata and other methods introduced in [15], [33]. Even if the manifolds overlay with each other, they can still be identified and clustered [39]. At this step, the data set \(X\) is clustered into several components, and each component is considered as a data manifold.

2. Estimate parameters of data manifolds. For intrinsic dimensionality estimation, many methods can be used: for example, the fractal based method [34], the MLE method [35], [36] and the incising ball method [37]. Assume that \(d_m\) is the intrinsic dimension of the \(m\)-th data manifold. Let \(\rho = \max d_m\). For the neighborhood size, [32] introduces a method on automatically generating parameters for Isomap on one single data manifold. For convenience, appropriate neighborhood sizes \((k_m\) or \(\epsilon_m\) for \(X^m\)) can be given manually for data manifolds.

3. Learn the data manifolds individually. One data manifold can be learned by traditional manifold learning algorithms. Here, we propose to rebuild a graph on each data manifold with a new neighborhood size to better approximate the intra-manifold geodesics. In doing so, Yang’s methods [26]-[29] and the\(\text{ek-CG}\) graph construction method are preferred, where the\(\text{ek-CG}\) graph construction method will be described later. It is assumed that the low-dimensional embedding for \(X^m\) is \(Y^m\).

**Step II:** The composition process

1. Preserve a skeleton \(I\) of the whole data set in a low-dimensional space \(\mathbb{R}^d\). The skeleton \(I\) should be carefully designed so it can represent the global structure of \(X\). Let \(R_I\) be the low-dimensional embedding of \(I\).

2. Transform \(Y^m\) into a single coordinate system by referring to \(R_I\). In order to faithfully preserve the intra-manifold relationship, Euclidean transformations can be constructed and used. Using the embedding points \(R_{Y_I}^m \subset R_{Y_I}\) and the corresponding points from \(Y^m\), we can construct an Euclidean transformation from \(Y^m\) to the coordinate system of \(R_I\).

The idea of using a decomposition-composition procedure is not new, which was first used by Wu et al. [23] in their split-augment process and well developed and used in [24]. The procedure we proposed here aims to solve a more general problem. Step I.1 permits that the designed learning algorithm has a good ability to identify data manifolds. Step I.2 gives a guideline on learning manifolds with different intrinsic dimensionality and neighborhood sizes. Step I.3 learns data manifolds individually so that the intra-manifold relationship can be faithfully preserved. **Step II.1 is the most flexible part of the procedure which allows us to design new isometric multi-manifolds learning algorithms.** A well designed skeleton \(I\) can better represent the inter-manifolds relationship. In the following subsections, we will introduce a new multi-manifolds learning algorithm and revise the original D-C Isomap algorithm with the help of this general procedure.
B. A new algorithm for isometric multi-manifolds learning

Based on the proposed procedure, we design a new multi-manifolds learning algorithm. As an extension of the classical Isomap method to multi-manifolds data, the method will be referred to as multi-manifolds Isomap or M-Isomap. It is assumed that X is also interchangeable to represent the matrix \(x_1, x_2, \cdots, x_N\), where \(x_i, i = 1, \cdots, N\) are column vectors in \(\mathbb{R}^D\).

1) Using the k-CC method to construct a neighborhood graph and identify manifolds: Table II shows the time complexity of the k-NN, k-MSTs, Min-k-ST, k-EC and k-VC methods on the neighborhood graph construction. As shown in the table, the k-NN method has the lowest computational complexity \(O(kN^2)\).

For incremental learning, the computational complexity of the k-NN, k-MSTs and k-VC methods are \(O(kN)\), \(O(N \ln N)\) and \(O(N \ln N + kn)\), respectively [30], [31]. The computational complexity of the Min-k-ST and k-EC methods for incremental learning are unavailable. For data on one single manifold, the improvement of performance of Yang’s methods becomes insignificant when the neighborhood size \(k\) increases for the k-NN method. More importantly, the k-NN method implicitly has the property of clustering to multi-manifolds data. Data points of the same manifold tend to be connected by paths and disconnected otherwise when each data point is connected with its neighbors by edges. Although the k-NN method is not a robust clustering algorithm, it is computationally efficient for both clustering and graph construction. Therefore, we introduce a variation of the k-NN method which inherits the computational advantage of the k-NN method. The method is able to identify data manifolds and construct a totally connected neighborhood graph. In the rest of the paper, the proposed neighborhood graph construction method will be referred as the k-edge connected graph method (the k-CG method).

The main difference between the k-NN method and the k-CG method is lines 4 to 25, which identify components (data manifolds) and connect different components of the graph. This change makes the constructed graph totally k-edge connected. Compared with the method proposed in [23], the k-CG method constructs a neighborhood graph with \(k\) inter-manifolds edges, which is able to control the rotation of the embedding of the data manifolds. In Section V the k-CG Isomap method, which uses the k-CG method to construct a totally connected graph and then perform the classical Isomap on the graph, is compared with the M-Isomap method. It can be easily seen that the k-CG Isomap suffers the limitation which has been shown by Theorem 4.1. We assume that \(\{x_n^{(m)}\}_{i=1}^k\) is the subset of \(X^n\) whose data points connect with manifold \(X^{n+1}\), \(i = 1, \cdots, k\).

2) Learn data manifolds individually: As \(X^n\) is considered as a single data manifold in \(\mathbb{R}^D\), it is possible to find its intrinsic parameters. The incising ball method [37] is utilized to estimate the intrinsic dimensionality, which is simple to implement and always outputs an integer result. Assume that \(d\) is the highest intrinsic dimensionality of data manifolds. The neighborhood size \(k_n\) or \(ε_n\) of each data manifold is given manually and the graph on the data manifold \(X^n\) is rebuilt. It is expected that the new neighborhood graph on \(X^n\) can give better approximations to the intra-manifold geodesics. The approximated geodesic distance matrix for \(X^n\) is written as \(D_m\). By applying the classical MDS on \(D_m\), the low-dimensional embedding for \(X^n\) can be obtained as \(Y^n = \{y_i^n\}_{i=1}^{D_n}\).

3) Preserve a skeleton of the data manifold X: First, inter-manifolds distances are computed. Assuming that \(x_i^n\) and \(x_j^n\) are any data points with \(x_i^n \in X^n\) and \(x_j^n \in X^n\), their distance can be
computed by
\[
d_G(x_n^m, y_n^m) = \min_{i=1}^k \{d_G(x_n^m, x_{n(i)}^m) + ||x_{n(i)}^m - y_n^m|| + d_G(y_n^m, x_n^m)\},
\]
where \(d_G(x_n^m, x_{n(i)}^m)\) is the shortest path on the neighborhood graph of \(X^m\). Although \(d_G(x_n^m, x_{n(i)}^m)\) may not be the shortest path on the totally connected graph of \(X\), Eq. (\ref{eq:dist}) is an efficient way to approximate distances across manifolds. Assume that \(D_{mn}\) is the distance matrix across over \(X^m\) and \(X^n\). Then the furthest inter-manifolds data points are computed by
\[
\{f x_n^m, f x_n^m\} = \arg \max d_G(x_n^m, y_n^m), \quad d_G(x_n^m, y_n^m) \in D_{mn}.
\]
Without loss of generality, we may assume
\[
I^m = \{x_n^m\}_{i=1}^m = \bigcup_{n=1}^M \{x_n^m(1), \ldots, x_n^m(k)\}.
\]
Then \(I = \bigcup_{m=1}^M I^m\) is considered as the global skeleton of \(X\). On the data manifold \(X\), it can be seen that the skeleton \(I\) formulates a sparse graph. We assume that \(D_I = \{d_I(i, j)\}\) is the distance matrix of \(I\), where
\[
d_I(i, j) = \begin{cases} 
  d_G(x_n^m, y_n^m) \in D_{mn}, & x_i \in X^m, x_j \in X^n \\
  d_G(x_n^m, x_n^m) \in D_{nn}, & x_i, x_j \in X^m 
end{cases}
\]
By applying the classical MDS algorithm on \(D_I\), the low-dimensional embedding \(R Y^m\) of \(I\) can be obtained. It is assumed that \(R Y = \{y_i\}_{i=1}^M \subset R Y\) is the embedding of \(I^m\).

4) Euclidean transformations: Assume that \(Y^m = \{y_i\}_{i=1}^M \subset Y^m\) and \(y_i^m\) corresponds to \(x_i^m\). Then the Euclidean transformation from \(Y_I^m\) to \(R Y^m\) can be constructed as follows.

The general Euclidean transformation can be written as
\[
r Y = A y + \beta,
\]
where \(A\) is an orthonormal matrix and \(\beta\) is a translation vector. For the \(m\)-th data manifold, it is assumed that the Euclidean transformation is
\[
r y^m = A y^m + \beta_m, \quad i = 1, \ldots, l_m.
\]
The above Euclidean transformation can be rewritten in the matrix form:
\[
R Y^m = A y^m + \beta_m e^T = (A_m \beta_m) \begin{pmatrix} Y^m & e \end{pmatrix}
\]
where \(e\) is a vector with all ones. Equation (\ref{eq:eucl_trans}) can be solved using the least square method, and the solution is given by
\[
(A_m \beta_m) = R Y^m \begin{pmatrix} Y^m & e \end{pmatrix} ^{-1}
\]
where \(I\) is the identity matrix and \(\lambda\) is a regularization parameter in the singular case. However, the least square solution does not necessarily provide an orthonormal matrix \(A_m\). We now propose to use the QR decomposition to get the orthonormal matrix \(A_m\). The QR process can be written as
\[
(A_m R) = Q R(\bar{A}_m)
\]
where the diagonal elements of \(R\) are forced to be nonnegative. Then \(\beta_m\) can be recomputed by minimizing the cost function
\[
C(\beta_m) = \sum_{i=1}^{l_m} ||A_m y_i^m + \beta_m - r y_i^m||^2.
\]
Solving the equation \(\frac{\partial C(\beta_m)}{\partial \beta_m} = 0\) gives
\[
\beta_m = \frac{1}{l_m} \sum_{i=1}^{l_m} (r y_i^m - A_m y_i^m)
\]
The low-dimensional embeddings \(Y_m (i = 1, \ldots, M)\) can be formed into a global coordinate system using the constructed Euclidean transformations.

5) The M-Isomap algorithm: The detailed M-Isomap algorithm is summarized in the following table.

**Input:** \(X = \{x_i\}_{i=1}^N\) with \(x_i \in \mathbb{R}^D\). Initial neighborhood size \(k\) or \(\epsilon\).

**Step I.1** Perform the \(k\)-CG algorithm on \(X\). Data manifolds \(X^m\) and the set of inter-manifolds points \(\{x_{n(i)}^m\}_{i=1}^M\) of \(X^m\) can be obtained.

**Step I.2** Estimate parameters of the data manifolds. Assume that the intrinsic dimension \(d_m\) and neighborhood size \(k_m\) or \(e_m\) are parameters for \(X^m\). Let \(d = \max d_m\) and rebuild the neighborhood graph on \(X^m\).

**Step I.3** Classical Isomap algorithm is performed on \(X^m\) with new neighborhood graph, \((m = 1, \ldots, M)\). The corresponding low-dimensional embedding of \(X^m\) is denoted by \(Y^m\).

**Step II.1** Inter-manifolds distance matrix \(D_{mn}\) is computed by Eq. (\ref{eq:dist}); thus \(\{f x_n^m\}_{m=1}^M\) can be found by Eq. (\ref{eq:dist}). Distance matrix \(D_I\) for the skeleton \(I\) is computed by Eq. (\ref{eq:dist}). Classical MDS is performed on \(D_I\) to obtain the low-dimensional embedding of \(I\), which is written as \(R Y_I\). Assume that \(R Y_I^m \subset R Y_I\) is the embedding of \(I^m\).

**Step II.2** Construct Euclidean transformations by Eqs. (\ref{eq:eucl_trans})-\(\ref{eq:eucl_trans} \). Use the Euclidean transformations to transform \(Y^m\) into \(R Y^m\), \(m = 1, \ldots, M\).

**Step II.3** \(Y = \bigcup_{m=1}^M R Y^m\) is the final output.

C. Computational complexity of the M-Isomap algorithm

Computational complexity is a basic issue for application. In the M-Isomap method, the \(k\)-CC algorithm needs \(O((k + 1)N^2)\) time to construct a totally connected graph and identify the manifolds. It needs \(O(S_m^3 + S_m^3 \ln S_m)\) time to compute the shortest path on each data manifold and \(O(S_m^3)\) time to perform the classical MDS on the distance matrix of data manifolds. The time complexity of computing the shortest path across data manifolds is \(O(S_m^3)\) and that of finding \(f x_i^m, f x_j^m\) is \(O(S_m^3)\). Performing the classical MDS on the skeleton \(I\) needs \(O(S_m^3)\) computational time. The time complexity of finding the least square solution and processing the QR decomposition for \(M\) data manifolds is \(O(Md^3)\). Finally, transforming \(Y^m\)’s into a single coordinate system needs \(O(d^2N)\) computational time.

Therefore, the total time complexity of the M-Isomap method is
\[
O((k + 1)N^2 + \sum_{m=1}^M (S_m^3 + S_m^3 \ln S_m) + \sum_{m=1}^M (k + 1)S_m^3 S_n + \sum_{m=1}^M l_m^3 + M d^3 + d^2N).
\]
For a large data set where \( N >> M \) and \( N >> d \), the overall time complexity of the M-Isomap algorithm can be approximated by

\[
O((k + 1)N^2 + \sum_{m=1}^{M} (S_m^2 + S_m^2 \ln S_m) + \sum_{m=n}^{M} (k + 1)S_mS_n).
\]

D. The revised D-C Isomap method

D-C Isomap applies the decomposition-composition procedure. Therefore, it is able to preserve intra-cluster distances correctly. However, this method suffers from several limitations. In the following, the original D-C Isomap algorithm will be revised to overcome its limitations.

1) Selection of centers: D-C Isomap implicitly assumes that the inter-cluster point \( nx_m^\circ \) is on the line which connects centers \( O_m \) and \( nx_m^\circ \). Thus it is more sensible that \( O_m \) is chosen by referring to the inter-cluster points. Fig. 1 illustrates two basic cases about the relationship of the center and inter-cluster points. Although the points \( n_1x, n_2x, n_3x, n_1x^3 \) and \( O_1 \) do not have to really lie on the same plane in the ambient space. It is assumed that these points formulate a triangle in the low-dimensional space. Fig. 1(a) shows the case when \( \angle nx_1x^3 nx_1x^2 < 180^\circ \).

In the triangle \( \Delta O_1nx_1^3nx_1^2 \), the edge \( d(nx_1^3, nx_1^2) \) can be computed as \( \|nx_1^2 - nx_1^3\| \). We also have

\[
\angle O_1nx_1^3nx_1^2 = \arccos \frac{d(nx_1^3, nx_1^2)^2 - d(nx_1^2, nx_1^3)^2 - d(nx_1^2, nx_1^3)^2}{2d(nx_1^2, nx_1^3)d(nx_1^2, nx_1^3)}
\]

\[
\angle O_1nx_1^3nx_1^2 = \arccos \frac{d(nx_1^3, nx_1^2)^2 - d(nx_1^2, nx_1^3)^2 - d(nx_1^2, nx_1^3)^2}{2d(nx_1^2, nx_1^3)d(nx_1^2, nx_1^3)}
\]

Subsequently, the length of edges \( d(O_1, nx_1^3) \) and \( d(O_1, nx_1^2) \) can be calculated by the Law of Sines in the triangle \( \Delta O_1nx_1^3nx_1^2 \). Suggested distances between the center \( O_1 \) to the inter-cluster points can be calculated as

\[
d'(O_1, nx_1^3) = d(O_1, nx_1^3) - \|nx_1^3 - nx_1^2\|
\]

\[
d'(O_1, nx_1^2) = d(O_1, nx_1^2) - \|nx_1^2 - nx_1^3\|
\]

For a cluster with the intrinsic dimension 2, it is sufficient to estimate the position of \( O_1 \) in the cluster by solving the following optimization problem:

\[
O_1 = \arg \min_{o \in X_1} f(o),
\]

where

\[
f(o) = \sum_{i=1}^{2} \|d(O_1, nx_1^i) - d'(O_1, nx_1^i)\|.
\]

Here, \( d(O_1, nx_1^i) \) is the length of the shortest path between \( O_1 \) and \( nx_1^i \) on the graph \( X^1 \). For a cluster with intrinsic dimension \( d_m \), at least \( d_m \) distances \( d'(O_1, nx_1^i), i = 1, \ldots, d_m \), are needed to estimate the position of the center \( O_1 \). In this case, \( f(o) \) is given by

\[
f(o) = \sum_{i=1}^{d_m} \|d(O_1, nx_1^i) - d'(O_1, nx_1^i)\|
\]

If we cannot find sufficiently many distances \( d'(O_1, nx_1^i) \) to locate the center, there must be many inter-cluster points located in space, as illustrated in Fig. 1(b). In this case, and when the center \( O_1 \) is never on the line passing through \( nx_1^2nx_1^3 \) and \( nx_1^3nx_1^2 \), we have

\[
\angle nx_1^1nx_1^3nx_1^2 + \angle nx_1^2nx_1^3nx_1^1 \geq 180^\circ
\]

In order to get a better preservation of the inter-cluster relationship, \( O_1 \) should be placed as far as possible from these inter-cluster points. For a cluster with intrinsic dimension 2, it is suggested that \( O_1 \) should be chosen as

\[
O_1 = \arg \max_{o \in X_1} g(o)
\]

where

\[
g(o) = d(o, nx_1^1) + d(o, nx_1^2) - \|d(o, nx_1^1) - d(o, nx_1^2)\|
\]

If the intrinsic dimension of \( X^1 \) is \( d_m \) and \( \{nx_1^i, i = 1, \ldots, d_m\} \) is the set of inter-cluster points in \( X^1 \), then the function \( g(o) \) should be given as

\[
g(o) = \sum_{i<j} (d(o, nx_1^i) + d(o, nx_1^j) - \|d(o, nx_1^i) - d(o, nx_1^j)\|)
\]

2) Degenerate and unworkable cases: Since the original D-C Isomap algorithm relies on the position of the center of each cluster to preserve the inter-cluster relationship, the algorithm does not work under certain circumstances. Consider a simple case of two data clusters with the intrinsic dimension \( d = 2 \). The method does not work in this case because it implicitly requires an another data cluster to provide sufficient rotation reference data points. Since the low-dimensional embedding of each cluster is relocated by referring to the position of the center of each cluster.
In the case when there are three or more clusters and their centers are nearly on a line, the original D-C Isomap cannot find the exact rotation matrix.

This issue is solved in this paper by adding fictitious clusters. The algorithm applies a trial and error procedure to determine the position of the fictitious clusters. As an example, we now consider the case of two clusters. As shown in Fig. 2, the nearest couple of inter-cluster points of the clusters $X^1$ and $X^2$ are assumed to be $nx^1$ and $nx^2$, and $m_1$ is the middle point between $nx^1$ and $nx^2$. The second nearest couple of inter-cluster points are $nx^1_2$ and $nx^2_2$, and $m_2$ is the middle point between them. The third fictitious cluster $X^3$ is then suggested to be given by

$$X^3 = m_1 + \gamma ||nx^1 - nx^1|| \frac{m_2 - m_1}{||m_2 - m_1||},$$

where the parameter $\gamma$ can be decided by a trial and error procedure. Given a positive value $\beta > 1$, $X^3$ is assumed to satisfy that

$$\frac{1}{\beta} < \frac{||X^3 - X^1||}{||X^3 - X^2||} < \beta,$$

(10)

where $||X^3 - X^1||$ is the shortest distance between the data points from clusters $X^1$ and $X^3$. If condition (10) is not satisfied, then $\gamma$ can be chosen in a pre-defined range such as

$$\{\cdots, -3, -2, -1, -\frac{1}{2}, -\frac{1}{3}, \cdots, \frac{1}{3}, \frac{1}{2}, 1, 2, \cdots \}.$$

In the case when there are $M$ clusters in the data set with $M < d + 1$, we can start from the couple of clusters with the maximum nearest inter-cluster distance. Assume that $X^1$ and $X^2$ satisfy that

$$||X^1 - X^2|| = \max_i \min_j ||X^i - X^j||$$

with $nx^1_1, nx^1_2, m_1, nx^2_1, nx^2_2, m_2$ being defined as above. Then the $(M + 1)$-th cluster $X^{M+1}$ can be generated as

$$X^{M+1} = m_1 + \gamma ||nx^1_1 - nx^1|| \frac{m_2 - m_1}{||m_2 - m_1||}.$$

If $X^P$ and $X^Q$ are the two nearest clusters of $X^{M+1}$, then, given $\beta > 0$, it is assumed that $X^{M+1}$ should satisfy

$$\frac{1}{\beta} < \frac{||X^{M+1} - X^P||}{||X^{M+1} - X^Q||} < \beta.$$

If $M + 1 < d + 1$, then replace $M$ by $M + 1$ and repeat the generating procedure presented above.

PCA can be used to find the dimensionality of the subspace on which the centers are lying. If the dimensionality of the subspace is smaller than $d$, then fictitious clusters should be added until the centers of the clusters can anchor a $d$-dimensional simplex.

3) The revised D-C Isomap algorithm: The revised D-C Isomap algorithm can be given as follows.

**Input:** $X = \{x_i\}_{i=1}^N$, with $x_i \in \mathbb{R}^d$. Initial neighborhood size $k$ or $\epsilon$.

**Step I.1** Same as Step I.1 of the original D-C Isomap algorithm.

**Step I.2** Estimate the parameters, intrinsic dimension $|d_m|_{m=1}^M$ and neighborhood sizes ($|k_m|_{m=1}^M$ or $|e_m|_{m=1}^M$), of the clusters. Let $d = \max_m d_m$ and rebuild the neighborhood graph for each cluster.

**Step I.3-4** Same as Step I.2-3 of the original D-C Isomap algorithm.

**Step II.1** Centers of the clusters are computed by [8] or [9]. Fictitious clusters should be added until centers of the clusters can anchor a $d$-dimensional simplex.

**Step II.2-4** Same as Step II.2-4 of the original D-C Isomap. Assume that $Y^m$ is transformed into $T^m$.

**Step II.5** $Y = \bigcup_{m=1}^M T^m$ is the final output.

V. Experiments

A. 3-D data sets

In this subsection, we compare the k-CC Isomap, the M-Isomap and the revised D-C Isomap on three 3-D data sets. It should be noted that in all experiments, the neighborhood size is chosen corresponding to the best performance of each algorithm.

Fig. 3(a) is a two-manifolds data set with $N = 1200$ data points, and the data set is generated by the following matlab code:

```matlab
xx=t.*cos(t);yy=t.*sin(t);
zz=unifrnd(1,25,1,N);
X=[xx;zz;yy];
```

It can be seen that each data manifold is exactly unrolled, and the inter-manifolds distance is precisely preserved. Fig. 3(b) shows the result obtained by the k-CC Isomap, whose neighborhood graph is constructed by using the 8-CC method. It can be seen that the embedding shrinks along the edges in the low-dimensional space and the edges of the embedding become noisy. Fig. 3(c) shows the result obtained by the M-Isomap method with the neighborhood size $k = 8$. As can be seen, each data manifold is exactly unrolled, and the inter-manifolds distance is precisely preserved. Fig. 3(d) illustrates the initialization step of the revised D-C Isomap algorithm. First, the two data manifolds $X^1$ and $X^2$ are identified. Then the third data cluster $X^3$ is constructed, where the parameter $\lambda = 0.1$. Finally, the centers $O_1$ and $O_2$ of the data manifolds are computed by referring to the nearest neighbors. The center of $X^3$ is also the data point $X^3$. Fig. 3(e) shows the result of the revised D-C Isomap method. It is seen that the embedding exactly preserves both the intra-manifolds distances and inter-manifolds distances.

Fig. 4(a) is another two-manifolds data set with $N = 1200$ data points, and the data set is generated by the following matlab code:

```matlab
t = (1*pi/6) * (1 + 2*rand(1, N));
xx = t.*cos(t); yy = t.*sin(t);
zz = unifrnd(1,10,1,N/2) unifrnd(16,25,1,N/2); X=[xx;zz;yy];
```

Each data manifold has 600 data points. One data manifold is a rectangular region and the other one is a round region. Fig. 4(b) shows the result obtained by the k-CC Isomap with the
Experiments on a two-manifolds data set. (a) The original data set. (b) The result obtained by the k-CC Isomap. (c) The result obtained by the M-Isomap. (d) Illustration of the procedure of the revised D-C Isomap. (e) The result obtained by the revised D-C Isomap.

neighborhood size $k = 10$. It can be seen that the rectangular region bent outwards and the round region is prolonged. Fig. 4(c) shows the result obtained by the M-Isomap method with the neighborhood size $k = 8$. As can be seen, each data manifold is exactly unrolled, and the inter-manifolds relationship is precisely preserved. Fig. 4(d) illustrates the initialization step of the revised D-C Isomap algorithm. The parameter $\lambda = 0.5$ for the production of the new cluster $X^3$. Fig. 4(e) shows the result of the revised D-C Isomap method with the neighborhood size $k = 5$. It can be seen that the embedding exactly preserves both the intra-manifold distances and inter-manifolds distances.

Fig. 5(a) shows a three-manifolds data set with $N = 1600$ data points on the Swiss roll manifold. The data set is generated by the following matlab code:

```matlab
t1 = [unifrnd(pi*5/6,pi*16/12,1,N/4)];
t2 = [unifrnd(pi*18/12,pi*12/6,1,N/4)];
t3=(5*pi/6)*(1+7/5*rand(1,N/2));
a1=t1.*cos(t1); b1=t1.*sin(t1);
c1=[unifrnd(-1,3,1,N/4)];
a2=t2.*cos(t2); b2=t2.*sin(t2);
c2=[unifrnd(-1,3,1,N/4)];
a3=t3.*cos(t3); b3=t3.*sin(t3);
c3=[unifrnd(6,10,1,N/2)];
x1=[a1;c1;b1]; x2=[a2;c2;b2]; x3=[a3;c3;b3]
Z=[x1 x2 x3];
```

There are three rectangular regions on the Swiss roll manifold.
The longest data manifold has 800 data points, and each of the other two shorter data manifolds contains 400 data points. Fig. 5(b) shows the result obtained by the k-CC Isomap algorithm with the neighborhood size \( k = 10 \). Due to the bad approximation of the inter-manifolds geodesics, edges of the data manifolds bend outwards. Fig. 5(c) shows the result obtained by the M-Isomap method, where the neighborhood size \( k \) is set to be 8. As can be seen, all data manifolds are exactly unrolled, and the inter-manifolds relationships of the three data manifolds are precisely preserved. Fig. 5(d) illustrates the initiation step of the revised D-C Isomap algorithm. The result of the revised D-C Isomap method is presented in Fig. 5(e). As seen in Fig. 5(e), the embedding does not exactly preserve the inter-manifolds distances. This is because the shape of the data manifolds are very narrow. The selected reference data points can not efficiently relocate each piece of the data manifold.

**B. Real world data sets**

Fig. 6(a) shows samples of the faces data [38] which contains face images of five person\(^1\). The data set consists of 153 images and has 34, 35, 26, 24, 34 images corresponding to each face. These images are gray scale with resolution of \( 112 \times 92 \). They are transformed into vectors in a 10304-dimensional Euclidean space. In order to show the inter-manifolds relationship with more details, the data is embedded into a three-dimensional space. Fig. 6(b) is the three-dimensional embedding by the PCA method. It can be observed that data manifolds of faces are mixed up, and the intra-face information is not preserved. Fig. 6(c) is the result obtained by the k-CC Isomap method with \( k = 3 \). As seen from Fig. 6(c), although the data points are clustered, their inter-face distances are not well preserved. The five lines are mixed up at one of their endings. Fig. 6(d) shows the result obtained by the M-Isomap method with \( k = 3 \). Due to the limitation of the k-NN method in clustering, only two data manifolds are identified. Although the data set is not well clustered, the result of the M-Isomap shows that the low-dimensional embedding can be separated easily. Fig. 6(e) presents the result obtained by the original D-C Isomap method, where the faces are split up beforehand. The circumcenters are used as their centers. However, as can be seen, two faces are mixed up.

Fig. 6(a) presents samples of the teapot data set with 300 data points, where "\( \Box \)" stands for the teapot bird-view images, "\( \Delta \)" stands for the teapot back-forth rotation images and "\( \bigcirc \)" stands for the teapot side-view images. Each image is an \( 80 \times 60 \times 3 \) RGB colored picture, that is, a vector in a 14400-dimensional input space. The data points do not distribute on a single global manifold, which is a great challenge to the classical manifold learning methods. The experiments show that the three data manifolds can be identified by the k-CC method. In order to show their exact embedding, Fig. 6(b)-(d) present the embedding of each data manifold by the classical Isomap with the neighborhood size \( k = 3 \). Fig. 6(e) gives the result obtained by the PCA method. It can be seen that the data set is clustered but the shape of each embedding is deformed because of the linear characteristic of the PCA method. Fig. 6(f) is the result obtained by the k-CC Isomap method with the neighborhood size \( k = 3 \). The bad approximations of the inter-manifolds geodesics lead to the deformation of the embedding in the low-dimensional space. Fig. 6(g) shows the result obtained by the M-Isomap method with the neighborhood size \( k = 3 \). From Fig. 6(g), it is seen clearly that the data set is clearly clustered and the intra-manifolds relationships are exactly preserved. Fig. 6(h) gives the result obtained by the revised D-C Isomap method with the neighborhood size \( k = 3 \). It is seen that the revised D-C Isomap algorithm also produces a satisfactory result.

Fig. 8(a) shows samples of the IsoFACE and teapot rotation bird-view data set. The IsoFACE data set consists of 698 images and each image is a \( 64 \times 64 \) (4096-dimensional) gray scale picture. Since the input dimension of the IsoFACE data set is different from that of the teapot data set, the dimension of the IsoFACE

---

\(^1\)http://www.cs.toronto.edu/~roweis/data.html
Isomap with the neighborhood size $k$ on the teapot data set shrinks to $(\frac{100}{100}) = 1$. The result obtained by the M-Isomap method, which is also satisfactory.

C. Discussion

In our experiments, there are several important features which should be considered:

1) Since the k-CC Isomap tries to preserve poor and good approximations of geodesics simultaneously, its low-dimensional embedding is usually deformed. This method works well if each data manifold has comparable number of data points and the data manifolds can not be very far from each other. The algorithm does not work well otherwise.

2) The revised D-C Isomap overcomes some limitations of the original D-C Isomap. Meanwhile, the robustness of
the algorithm is also enhanced by adding a new fictitious cluster.

3) The M-Isomap connects data manifolds with multiple edges, which can control the rotation of the low-dimensional embedding, and at the same time, better preserves inter-manifolds distance. Similarly to the D-C Isomap algorithm, the M-Isomap can also isometrically preserve intra-manifold geodesics and inter-manifolds distances.

To sum up, Table I shows the comparison of the general performance of the five versions of Isomap algorithms: classical Isomap, k-CC Isomap, Original D-C Isomap, revised D-C Isomap and M-Isomap. The labels "A" stands for poor performance, "o" stands for not bad and "o" stands for good. "Density" means the generalization ability on manifolds with different density, that is, different neighborhood sizes, "Dimensionality" means the generalization ability on manifolds with different intrinsic dimensionality, "Isometric" means the property of isometry in preserving the inter and intra-manifold relationship and "Generalization" means the overall generalization ability to learn from multiple data manifolds.

VI. Conclusion

In this paper, the problem of multi-manifolds learning is discussed. A general procedure for isometric multi-manifolds learning is proposed. The procedure can be used to build multi-manifolds learning algorithms which are able to faithfully preserve not only intra-manifold geodesic distances but also the inter-manifolds geodesic distances. The M-Isomap is an implementation of the procedure and shows promising results in multi-manifolds learning. Compared with the k-CC Isomap which was also introduced in this paper based on the general procedure, the M-Isomap has the advantage of low computational complexity. With the procedure, the D-C Isomap proposed in [24] was revised to overcome some of its limitations. Compared with the original D-C Isomap, the revised D-C Isomap is more effective in learning multi-manifolds data sets. Experiments have also been conducted on both synthetic and images data sets to illustrate the efficiency of the above multi-manifolds learning algorithms. Future work will be conducted on the application of the multi-manifolds learning algorithms.

Acknowledgment

The work of the second author (HQ) was partly supported by the Chinese Academy of Sciences through the Hundred Talents Program, the NNSF of China under grant 60675039 and grant 60621001, the 863 Program of China under grant 2006AA04Z227 and the Outstanding Youth Fund of the NNSF of China under grant 60725310. The third author (BZ) was partly supported by the Chinese Academy of Sciences through the Hundred Talents Program, the 863 Program of China under grant 2007AA04Z228, the 973 Program of China under grant 2007CB311002 and the NNSF of China under grant 90820007.

References

[1] I. T. Jolliffe, “Principal Component Analysis,” Springer-Verlag, New York, 1989.
[2] T. F. Cox and M. A. Cox, “Multidimensional Scaling,” Chapman & Hall, London, 2001.
[3] E. Mjolsness and D. De Coste, “Machine learning for science: State of the art and future prospects,” Science, vol. 293, pp. 2051-2055, Sep. 2001.
[4] J.B. Tenenbaum, V. de Silva and J. C. Landford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, pp. 2319-2323, Dec. 2000.
[5] S.T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by local linear embedding,” Science, vol. 290, pp. 2323-2326, Dec. 2000.
[6] H.S. Seung and D.D. Lee, “The manifold ways of perception,” Science, vol. 290, pp. 2268-2269, Dec. 2000.
[7] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural Computation, vol. 15, no. 6, pp. 1373-1396, June 2003.
[8] R. R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner and S. Warner, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” Proc. Nat. Acad. Sci. USA, vol. 102, no. 11, pp. 5591-5596, 2003.
[9] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment,” SIAM J. Comput., vol. 26, pp. 313-338, 2004.
[10] R. W. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner and S. Warner, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” Proc. Nat. Acad. Sci. USA, vol. 102, no. 21, pp. 7426-7431, May. 2005.
[11] T. Lin and H. Zha, “Riemannian Manifold Learning,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pp. 796-809, May. 2008.
[12] H. Zha and Z. Zhang, “Continuum Isomap for manifold learning,” Computational Statistics & Data Analysis, vol. 52, issue. 1, pp. 184-200, Sep. 2007.
[13] M.H.C. Law and A.K. Jain, “Incremental nonlinear dimensionality reduction by manifold learning,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pp. 377-391, March. 2006.
[14] M. Bernstein, V. de Silva, J.C. Langford and J.B. Tenenbaum, “Graph approximations to geodesics on embedded manifolds,” Technical report, Dept. of Psychology, Stanford Univ., Dec. 2000.
[15] M. Filippone, F. Camastra, F. Masulli and S. Rovetta, “A survey of kernel and spectral methods for clustering,” Pattern Recognition, vol. 41, pp. 176-190, May. 2007.
[16] M.-H. Yang, “Extended Isomap for pattern classification,” ICPAR 2002: Proc. Int. Conf. Pattern Recognition, vol. 3, Aug. 2002.
[17] X. He, S. Yan, T. Hu, P. Niyogi and H. Zhang, “Face recognition using Laplacianfaces,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 328-340, March. 2005.
[18] M. Belkin, V. Sindhwani and P. Niyogi, “Manifold regularization: a geometric framework for learning from examples,” J. Machine Learning Research, vol. 7, pp. 2399-2434, Dec. 2006.
[19] X. Geng, D. Zhang and Z. Zhou, “Supervised nonlinear dimensionality reduction for visualization and classification,” IEEE Trans. Systems, Man and Cybern. B, vol. 35, no. 6, pp. 1098-1107, Dec. 2005.
[20] O.C. Jenkins and M.J. Mataric, “A Spatio-temporal extension to Isomap nonlinear dimension reduction,” ACM, Proc. 21st Int. Conf. on Machine learning, vol. 69, pp. 56-66, 2004.
[21] A. Rahimi, B. Recht and T. Darrell, “Learning to transform time series with a few examples,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 10, pp. 1759-1775, Oct. 2007.
[22] J.B. Tenenbaum, V. de Silva and J.C. Landford “Response to Comments on the Isomap algorithm and Topological Stability,” Sciences, vol. 295, no. 5552, pp. 7, Jan. 2002.
[23] Y. Wu and K.L. Chan “An extended Isomap algorithm for Learning multi-class manifold,” Proc. of the 2004 Int. Conf. on Machine Learning and Cybernetics, vol. 6, pp. 3429-3433, Aug. 2004.
[24] D. Meng, Y. Lueng, T. Fung and Z. Xu “Nonlinear dimensionality reduction of data lying on the multiclass manifold,” IEEE Trans. Systems, Man and Cybern. B, vol. 38, issue. 4, pp. 1111-1122, Aug. 2008.
[25] B. Li, D. Huang, C. Wang and K. Liu “Feature extraction using constrained maximum variance mapping,” Pattern Recognition, vol. 41, pp. 3287-3294, May. 2008.
[26] L. Yang, “k-Efficient Connected Neighborhood Graph for Geodesic Distance Estimation and Nonlinear Projection,” Proc. of 17th Int. Conf. on Pattern Recognition (ICPR’04), vol. 1, pp. 196-199, 2004.
[27] L. Yang, “Building k-Edge-Disjoint Spanning Trees of Minimum Total Length for Isometric Data Embedding,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1680-1683, Oct. 2005.
(d)
Twice Isomap on data set A
Twice Isomap on data set C
Twice Isomap on teapot data set
Two-dimensional twice Isomap embedding