



# Passage method for nonlinear dimensionality reduction of data on multi-cluster manifolds

Deyu Meng<sup>a,\*</sup>, Yee Leung<sup>b</sup>, Zongben Xu<sup>a</sup>

<sup>a</sup> Institute for Information and System Sciences and Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, PR China

<sup>b</sup> Department of Geography and Resource Management, The Chinese University of Hong Kong, Hong Kong, PR China

## ARTICLE INFO

### Article history:

Received 5 August 2011

Received in revised form

27 November 2012

Accepted 24 January 2013

Available online 8 February 2013

### Keywords:

Manifold learning

Multi-cluster manifolds

Nonlinear dimensionality reduction

Passage method

## ABSTRACT

Nonlinear dimensionality reduction of data lying on multi-cluster manifolds is a crucial issue in manifold learning research. An effective method, called the passage method, is proposed in this paper to alleviate the disconnectivity, short-circuit, and roughness problems ordinarily encountered by the existing methods. The specific characteristic of the proposed method is that it constructs a globally connected neighborhood graph superimposed on the data set through technically building the smooth passages between separate clusters, instead of supplementing some rough inter-cluster connections like some existing methods. The neighborhood graph so constructed is naturally configured as a smooth manifold, and hence complies with the effectiveness condition underlying manifold learning. This theoretical argument is supported by a series of experiments performed on the synthetic and real data sets residing on multi-cluster manifolds.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

The goal of nonlinear dimensionality reduction (NLDR) is to recover the intrinsic low-dimensional representations of data located in the high-dimensional space. In recent years, an NLDR problem, called “manifold learning”, has been specifically highlighted. The manifold learning issue is addressed on the high-dimensional data sampling a probability distribution on a smooth manifold with intrinsic low dimensionality. Currently, this issue has become one of the central problems in machine learning and pattern recognition, such as text categorization [1], image processing [2], remote sensing [3], etc.

A variety of methods aiming at solving the manifold learning issue have been recently developed. They include locally linear embedding (LLE [1]), isometric feature mapping (Isomap [2]), local tangent space alignment (LTSA [4]), etc. ([5–7]). The initial step of all these methods is to specify a neighborhood around each of the given data points and then construct a weighted neighborhood graph superimposed on the entire data set, with a collection of edges connecting the neighboring data points. In this step, the most commonly utilized neighborhood-specifying method is the  $k$ -NN or  $\varepsilon$ -NN method, which defines neighbors of a datum as its  $k$  nearest ones, or the ones away from the datum smaller than the threshold  $\varepsilon$  [1,2]. Under the assumption (the neighborhood assumption) that each neighborhood so specified is

configured as a local patch (the neighborhood patch) approximately residing on the underlying manifold, the neighborhood graph naturally forms the geometry of the underlying manifold. For data located on a 1-cluster manifold, the neighborhood graph so formed is globally connected and figures the smooth manifold underlying the given data. By utilizing the intrinsic neighborhood-preserving relationship between the manifold and its representation set, the NLDR of the data can then be appropriately realized.

The data from practical applications, however, are often distributed on multiple manifold clusters instead of only one as original manifold learning assumes. Current manifold learning methods generally cannot take effect in such multi-cluster cases. This is intrinsically conducted by the abnormality of the neighborhood graph constructed by the traditional  $k$ -NN or  $\varepsilon$ -NN technique. If a small neighborhood size  $k$  or  $\varepsilon$  is selected, it tends to construct an inter-cluster disconnected neighborhood graph on the entire data set. This always makes the manifold learning methods invalid or infeasible since they are supposed to be implemented on a connected neighborhood graph. The above “disconnectivity” problem can only be avoided under a large neighborhood size, but accompanied is the “short-circuit” problem, i.e., the intra-cluster neighborhood patches tend to deviate greatly from the underlying manifold. Such violations of the basic neighborhood assumption tend to adversely affect the performance of a manifold learning method. That is to say, the inter-cluster disconnectivity problem and the intra-cluster short-circuit problem cannot be easily compromised in multi-cluster cases.

\* Corresponding author. Tel.: +86 130 3290 4180; fax: +86 29 8266 8559.  
E-mail address: [dymeng@mail.xjtu.edu.cn](mailto:dymeng@mail.xjtu.edu.cn) (D. Meng).

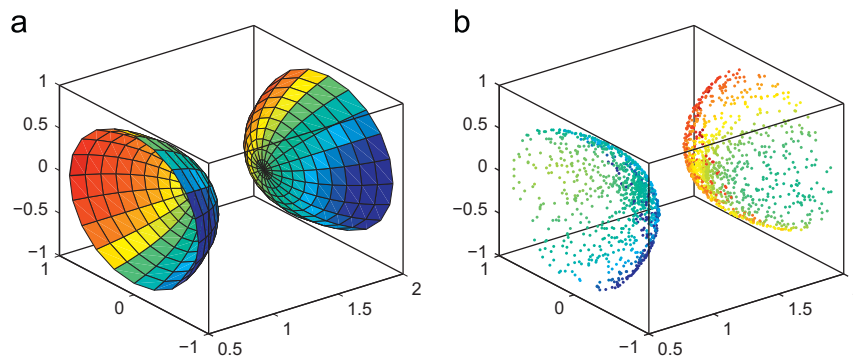


Fig. 1. (a) Manifold A composed by two semi-spherical surfaces; (b) Data A with 2000 points generated from Manifold A.

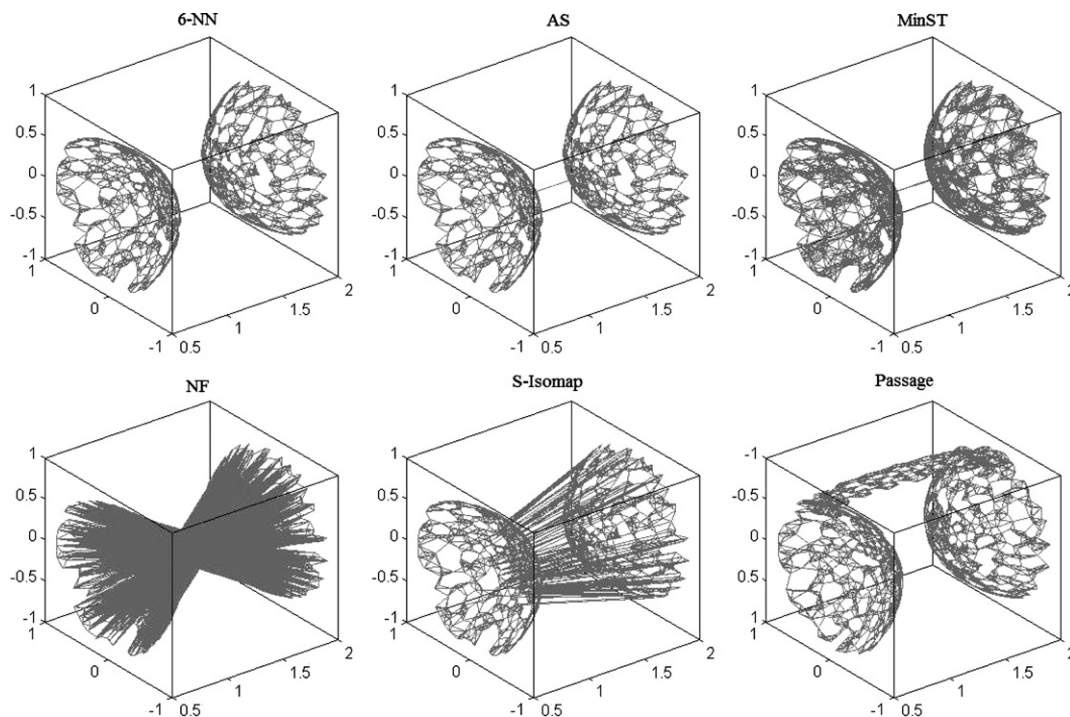
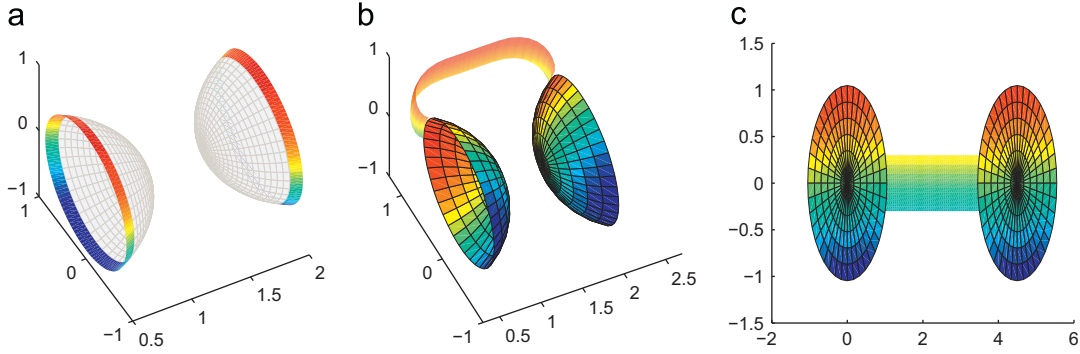


Fig. 2. The neighborhood graphs superimposed on Data A constructed by  $k$ -NN, AS, MinST, NF, S-Isomap, and the passage methods, respectively. All neighborhood sizes are set as 6.

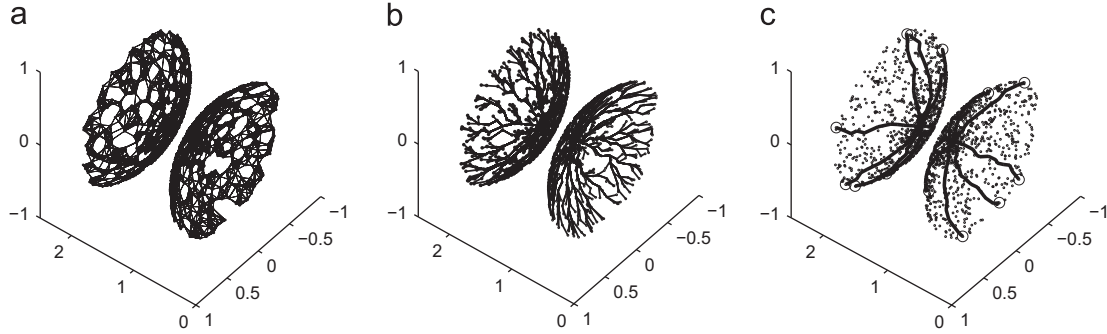
So far, the state of the art on this topic can be mainly represented by five approaches. The common feature of four among them is that they all need to first specify a connected neighborhood graph superimposed on the multi-cluster data. In particular, the first approach realizes this aim through two steps: constructing the  $k$ -NN or  $\varepsilon$ -NN neighborhood graph with a suitable neighborhood size  $k$  or  $\varepsilon$ , and then connecting inter-cluster nearest data pairs [8]. For convenience, we call this approach as the add-shortest-connection (AS) method. The second approach utilizes techniques of graph theory to construct  $k$ -vertex-connected or  $k$ -edge-connected neighborhood graph on the given data set under properly calibrated neighborhood size  $k$ . Three latest techniques of this approach are  $k$ -EC [9], MinST [10] and  $k$ -VC [11] methods. The third approach builds a special  $k$ -NN neighborhood graph by connecting each point with its  $k/2$  closest data and  $k/2$  farthest ones of the entire data set [12]. The method can thus be called the nearest-farthest (NF) method in short. The fourth approach is generally named supervised Isomap (S-Isomap) [13]. In the approach, the neighborhood graph of the input data is constructed according to a certain kind of dissimilarity between data points, which is specially designed to give a certain

chance to make the points in different clusters “more similar” than those in the same cluster. Another development for the multi-cluster issue is the decomposition–composition (D–C) method [14], so named due to its two involved processes: the decomposition and composition processes. The former process calculates the embeddings of each data cluster via its intra-cluster neighborhood structures, and the latter tunes the rigid-body transformations to appropriately locate and orient the embeddings of all clusters by utilizing their inter-cluster shortest connections.

For the former four approaches aforementioned, the disconnectivity problem can be evidently avoided, and the short-circuit problem can also be alleviated to a certain extent since the intra-cluster neighborhood patches specified by these approaches approximately reside on the underlying manifold. However, their performance is still always unsatisfactory in multi-cluster applications [13]. This can be explained by virtue of Fig. 2, which depicts the neighborhood graphs superimposed on the 2-cluster data (as shown in Fig. 1(b)) specified by the AS, MinST, NF, and S-Isomap methods, respectively. It is evident that under each of these approaches, the inter-cluster neighborhood patches make



**Fig. 3.** Graphical presentation of the process of the passage method. (a) Step I: Finding cluster edges; (b) Step II: Building inter-cluster passage; and (c) Step III: Data embedding.



**Fig. 4.** Graphical presentation of Step I of the passage method. (a) Step I.1: Constructing neighborhood graph; (b) Step I.2: Generating SPT trees; and (c) Step I.3: Calculating edge sets.

very rough transitions between manifold clusters, i.e., the manifold reflected by the constructed neighborhood graph has a low degree of smoothness. This is deviated from the fundamental assumption of manifold learning: the manifold underlying the input data is supposed to be smooth. This violation from the basic manifold assumption largely conducts the possible abnormal performance of the current approaches in multi-cluster applications. For convenience, we call this issue as the “roughness” problem.

Unlike the other four approaches, in the D–C method the intra-cluster neighborhood structures and the inter-cluster connections are employed in the decomposition and composition processes, respectively. The rough inter-cluster connections still have to be used to locate and orient each cluster’s embeddings in the composition process, and therefore, also tend to negatively affect the global multi-cluster structure of the entire embeddings. That is to say, the roughness problem still tends to occur in the calculation of the D–C method, as clearly depicted in the simulation results in Section 3.

Different from the existing approaches, the proposed passage method formulates smooth passages between separate manifold clusters to generate a connected neighborhood graph superimposed on the entire multi-cluster data, which can be naturally configured as a globally smooth manifold, as depicted in Fig. 2. The roughness problem, together with the disconnectivity and short-circuit problems, tend to be thus averted by applying the manifold learning methods to the data so constructed.

In what follows, the general idea and the implementation details of the passage method are first introduced in Section 2. Experimental results obtained by applying the passage method to a series of synthetic and real-world data sets are then analyzed and interpreted in Section 3. The paper is finally concluded with a summary and outlook for future research.

## 2. The passage method for multi-cluster manifold learning

Given the data set  $X = \{x_i\}_{i=1}^l$  distributed on multiple manifold clusters in the space  $R^n$ , the proposed method aims at calculating its intrinsic low-dimensional representation set  $Y = \{y_i\}_{i=1}^l \subset R^d$  ( $d < n$ ). For two separate clusters  $X_1$  and  $X_2$  of  $X$ , a passage between them is defined as a differentiable manifold [15] which forms a smooth connection between the edges of the two clusters and shares the similar tangent spaces with the clusters in the connection areas. The passage method aims to construct such smooth transitional passage between separate manifold clusters such that the union of the clusters and inter-cluster passage configures a globally smooth manifold. In brief, the passage method consists of the following basic steps:

- Step I. Finding the approximate edge sets of all manifold clusters.
- Step II. Building smooth passages between separate clusters.
- Step III. Calculating low-dimensional representations of the input data.

To facilitate our discussion, manifold A in Fig. 1 is employed to illustrate how our method is realized in Fig. 3 through the following procedure.

### 2.1. The procedure of the passage method

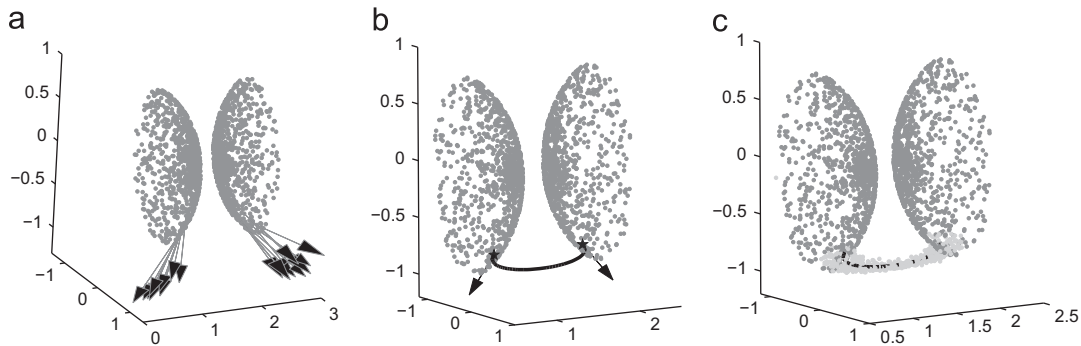
**Step I. Finding cluster edges:** Step I of the passage method aims to generate the clusters of the given data set and Calculate the edge representation set of each cluster (as depicted in Fig. 3(a)). Three sub-steps are involved and graphically depicted in Fig. 4.

**Step I.1. Constructing neighborhood graphs of the manifold clusters:** There are three goals to be attained in this step: (1) Generating the manifold clusters, (2) Constructing the neighborhood graphs superimposed on the clusters, and (3) Estimating their geodesic distance matrices.

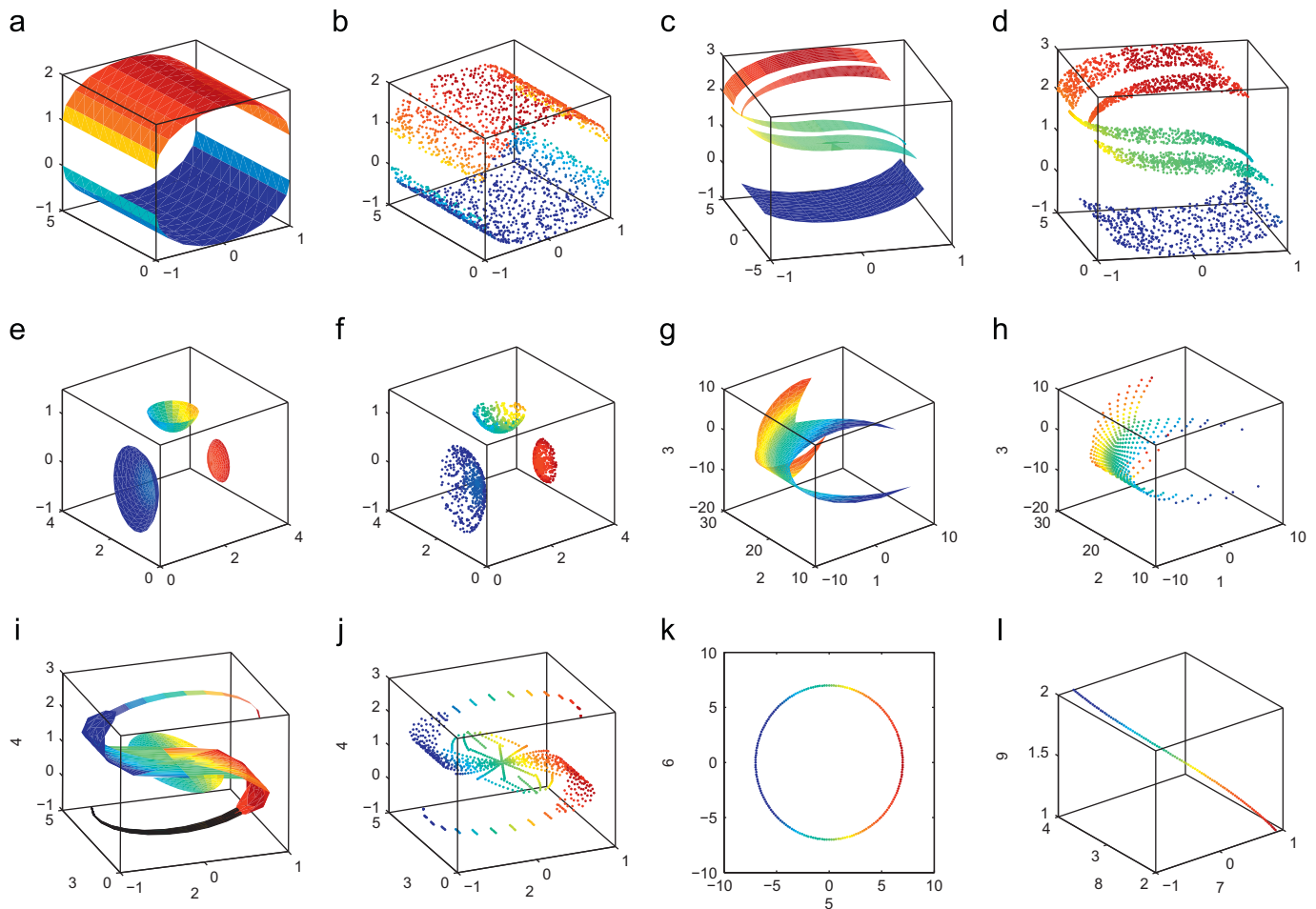
Any efficient cluster techniques can be employed to generate the clusters of the input data. For distinctly separate multi-cluster data, the  $k$ -NN or  $\varepsilon$ -NN approach with appropriately calibrated neighborhood size can be simply and effectively utilized. The byproduct of

this technique is that the neighborhood graph superimposed on each cluster data is also automatically constructed (as depicted in Fig. 4(a)). We denote the clusters of the data so generated as  $X^i = \{x_j^i\}_{j=1}^{l_i}$  ( $i = 1, 2, \dots, s$ ), where  $s$  is the number of manifold clusters and  $l_i$  is the number of data points in the  $i$ -th cluster.

The geodesic distance between a data pair in the  $i$ -th cluster  $X^i$  can then be estimated by calculating the length of the pairwise shortest path in the corresponding neighborhood graph. The standard shortest-path-finding algorithm, such as the Dijkstra's



**Fig. 5.** Graphical presentation of Step II of the passage method. (a) Step II.1: Calculating cluster tangent directions; (b) Step II.2: Formulating smooth inter-cluster path; and (c) Step II.3: Computing inter-cluster passage.



**Fig. 6.** (a) Manifold B composed by two clusters; (b) Data B with 2000 points generated from Manifold B; (c) Manifold C composed by five clusters, wrapped from the 2-D rectangle, diamond, sphere, square and triangle, respectively; (d) Data C with 3000 points generated from Manifold C; (e) Manifold D composed by three hemispherical surfaces with different radii; (f) Data D generated from Manifold D, number of points in each cluster is 600; (g) The first manifold cluster of the 4-cluster Manifold E, distributed in its 1–3 dimensional subspace; (h) Data with size 400 generated from the manifold (g); (i) The second manifold cluster of Manifold E distributed in its 2–4 dimensional subspace; (j) Data with size 800 generated from the manifold (i); and (k)(l) Data with size 200 and 100 generated from the third and forth manifold clusters of Manifold E distributed in its 5–6 and 7–9 dimensional subspaces, respectively.



algorithm [16] and the Floyd's algorithm [17,18], can be utilized for such estimation. The geodesic distance matrix so estimated is denoted as  $D_i = [D_i(j,k)]_{l_i \times l_i}$ , where  $D_i(j,k)$  is the approximate geodesic distance between  $x_j^i$  and  $x_k^i$ .

**Step I.2. Generating shortest-path trees for manifold clusters:** This step aims to yield a shortest-path tree (SPT) for each manifold cluster (as depicted in Fig. 4(b)). An SPT is a rooted spanning tree of a connected graph, whose branches (edges) are taken as the shortest paths from the root vertex to all other vertices of the graph. Each vertex of an SPT is connected to a single vertex immediately following it and leads in turn to the root vertex on the shortest path. Since the SPT is designated as long as the root vertex is selected, to construct an SPT for a manifold cluster, the root vertex must first be determined.

Without any prior knowledge about the underlying manifold, it is preferable to select the cluster center as the root. An easy way to obtain the approximate center of a cluster is to utilize its circumcenter. For a bounded close set  $\Omega$ , its circumcenter is defined as  $\arg \min_{x \in \Omega} (\max_{y \in \Omega} (\|x - y\|))$ . Correspondingly, the circumcenter of the  $i$ -th cluster can be approximated by

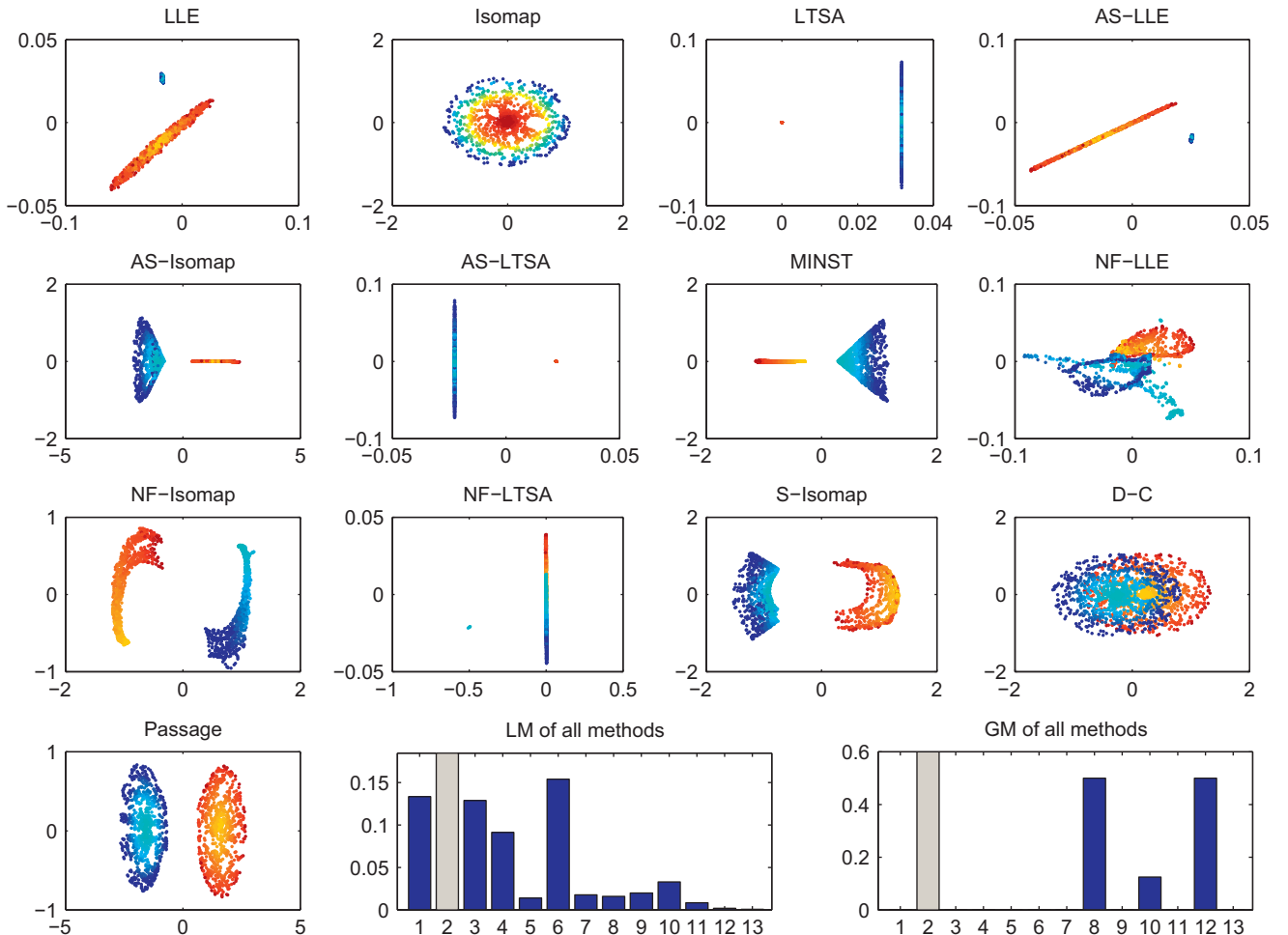
$$x_i^c = \arg \min_{x_i^c \in X^i} \left( \max_{x_k^i \in X^i} (D_i(j,k)) \right), \quad 1 \leq i \leq s,$$

where  $D_i(j,k)$  is the estimated geodesic distance between  $x_j^i$  and  $x_k^i$ . Utilizing  $x_i^c$  as the root vertex, the SPT of the  $i$ -th cluster can then be designated.

**Step I.3. Finding the main branches of the SPTs:** The aim of this step is to search for the main branches of each SPT generated by the previous step, such that the leafs of these branches can approximately represent the edge of the corresponding cluster (as depicted in Fig. 4(c)). Actually, if there are  $k$  neighbors around the estimated center (the root vertex)  $x_i^c$  of the  $i$ -th cluster, since each branch of the corresponding SPT definitely passes through one neighbor of the root  $x_i^c$ , all branches are naturally clustered into  $k$  categories by taking the neighbors they pass by as cluster labels [19]. Then it is reasonable to select the longest branch from each category as the representation of that category. The branch so selected approximately depicts the orientation tendency of all branches in this category directed from the root to their leaf vertices. The  $k$  branches so constructed are then configured as the main branches of the SPT, and their leafs can approximately represent the edge of the whole cluster. Therefore, after this step, the edge representation set of each cluster is obtained.

**Step II. Building inter-cluster passages:** The aim of this step is to construct the passage set such that the separate clusters can be smoothly connected (as depicted in Fig. 3(b)). There are also three sub-steps involved, as depicted in Fig. 5.

**Step II.1. Formulating tangent directions of manifold clusters:** For two separate manifold clusters, this step aims to calculate the start and end points of the inter-cluster passage located on both cluster edges, and compute the tangent directions towards the outside of the corresponding clusters at the points so calculated (as depicted in Fig. 5(a)). This is to be achieved via three steps.



**Fig. 7.** 2-D embeddings of Data A calculated by (1) LLE, (2) Isomap, (3) LTSA, (4) AS-LLE, (5) AS-Isomap, (6) AS-LTSA, (7) MinST, (8) NF-LLE, (9) NF-Isomap, (10) NF-LTSA, (11) S-Isomap, (12) D-C method, (13) passage method, respectively. The below two figures depict the values of LM and GM of all 13 methods. The gray bars mean the corresponding methods cannot find embeddings of the entire data set.

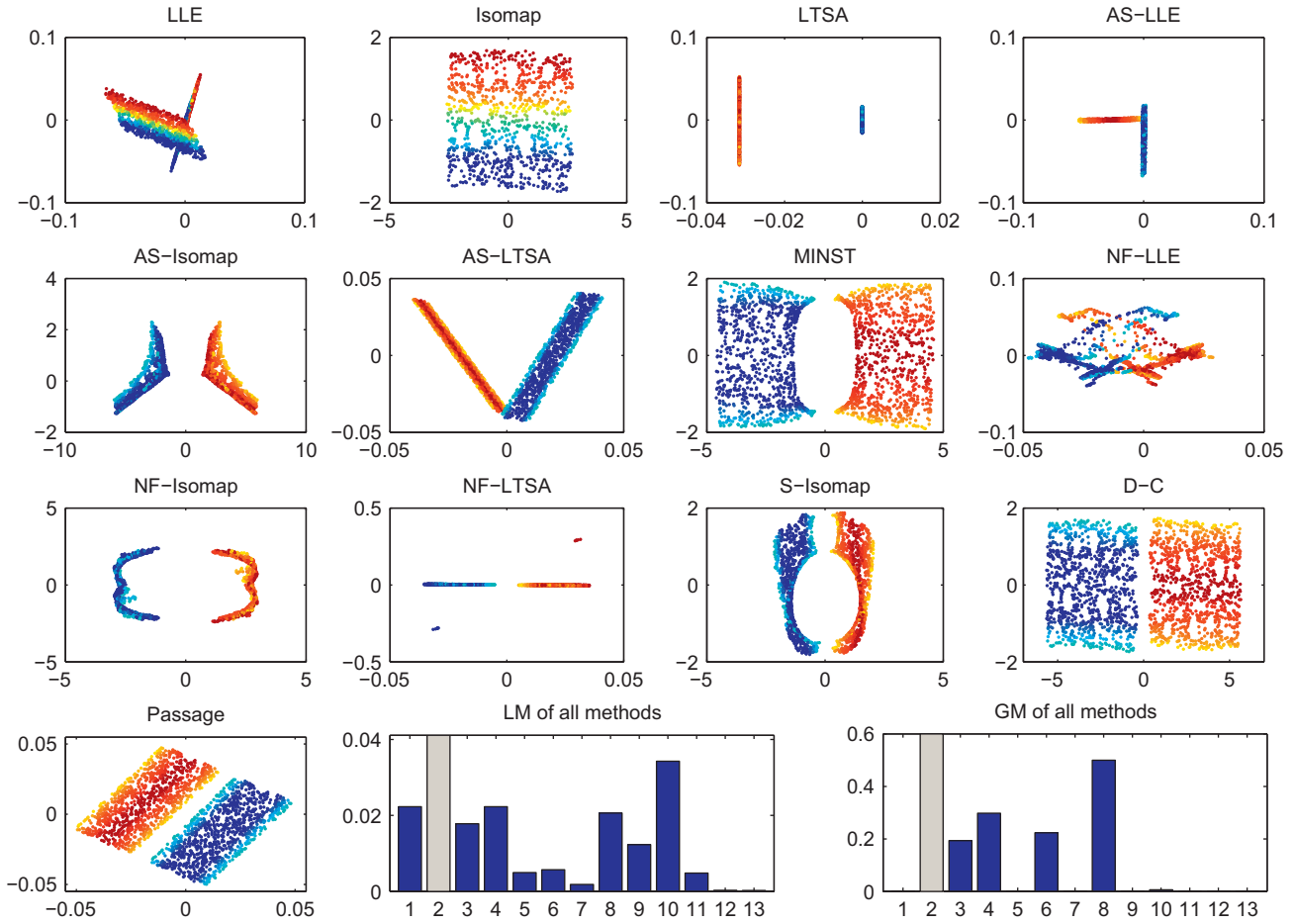


Fig. 8. 2-D embeddings of Data B calculated by 12 existing methods and the proposed passage method, respectively.

First, the nearest data pair, denoted as  $e_1$  and  $e_2$ , in the edge sets of the two clusters is calculated. Second, for each of the nearest pair, the neighborhood patch around it, i.e., the collection of its first  $k$  nearest data points, denoted as  $\Omega_1$  and  $\Omega_2$ , is computed. Third, based on the location of the neighborhood patch on the corresponding SPT, the tangent direction of the corresponding cluster is calculated. Specifically, for each datum ( $p_1$ ) on the neighborhood patch  $\Omega_1$  or  $\Omega_2$ , we need to search its immediate vertex ( $p_2$ ) on its located shortest path oriented to the root of the SPT and calculate the direction from  $p_2$  to  $p_1$  (i.e.,  $p_1 - p_2 / \|p_1 - p_2\|$ ). Since the neighborhood patches approximately reside on the underlying cluster manifold, the direction so formulated tends to be tangent to the manifold. Furthermore, since each of the branches in the SPT orients from the cluster center to the edge of the manifold, this formulated direction tends to be oriented towards the outside of the manifold. By this means,  $k$  tangent directions pointing towards the outside of the cluster at  $e_1$  or  $e_2$  can be approximately formulated for each of the clusters. In the algorithm, their average, denoted as  $v_1$  or  $v_2$ , respectively, is actually computed to improve the statistical stability of the method.

Accordingly, after this step, two outward orienting tangent directions,  $v_1$  and  $v_2$ , of the two separate clusters, starting from the nearest edge vertices  $e_1$  and  $e_2$ , are formulated. Besides, the neighborhood patches  $\Omega_1$  and  $\Omega_2$  around  $e_1$  and  $e_2$  are constructed, respectively.

**Step II.2. Calculating the smooth inter-cluster path:** This step aims to generate a smooth curve between two separate clusters (as depicted in Fig. 5(b)). By virtue of the output obtained from

the last step, this task is easily accomplished by formulating the following curve:

$$f(t) = At^3 + Bt^2 + Ct + D, \quad t \in [0, 1], \quad (1)$$

where  $A = 2e_1 - 2e_2 + v_1 - v_2$ ,  $B = -3e_1 + 3e_2 - 2v_1 + v_2$ ,  $C = v_1$ ,  $D = e_1$ . This curve is infinitely differentiable, and with the following property:

$$f(0) = e_1, \quad f'(0) = v_1, \quad f(1) = e_2, \quad f'(1) = -v_2. \quad (2)$$

That is, the curve starts from  $e_1$  with tangent direction  $v_1$ , and ends at  $e_2$  with tangent orientation  $-v_2$ . Evidently, the two clusters can be smoothly connected by this curve.

**Step II.3. Constructing the smooth inter-cluster passage:** This step aims to generate a collection of data points such that it forms a transitional passage smoothly connecting two separate clusters (as depicted in Fig. 5(c)). Before discussing the strategies for the realization of such aim, it is necessary to introduce some relevant theoretical and practical knowledge on differentiable (smooth) manifolds. For a smooth manifold, one can attach to every point  $x$  on the manifold a tangent space, containing all vectors at  $x$  tangent to the attached manifold. When only limited number of data points on the manifold are available, the tangent space at  $x$  can be approximated by the affine space spanned by the neighbors of  $x$  [4].

According to the above knowledge, the tangent spaces  $\pi_1$  and  $\pi_2$  at edge vertices  $e_1$  and  $e_2$  can be generated from the corresponding neighborhood patches  $\Omega_1$  and  $\Omega_2$ , respectively. Since the directions  $v_1$  and  $v_2$  calculated in Step II.1 are approximately tangent to the underlying cluster manifolds at  $e_1$  and  $e_2$ ,

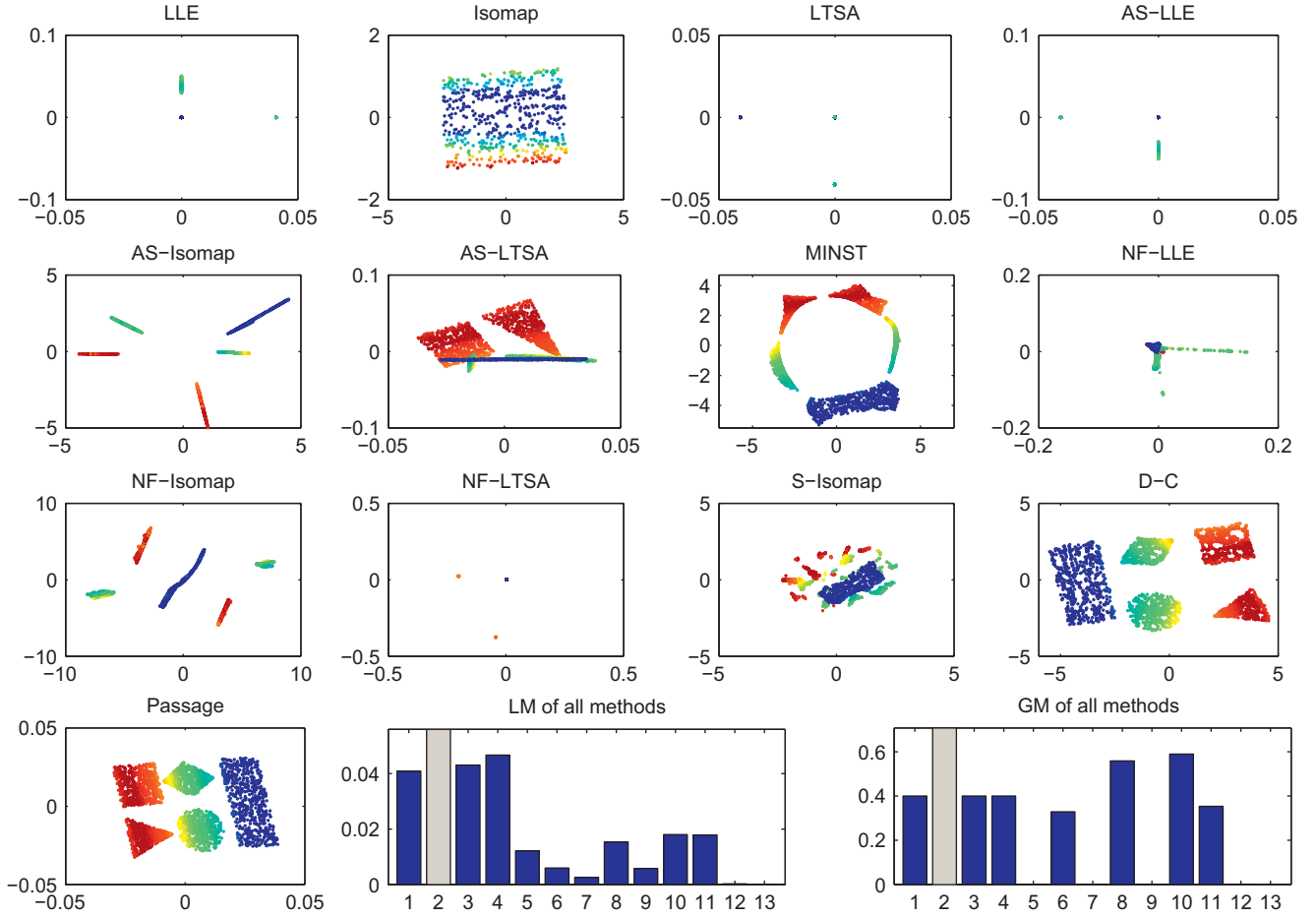


Fig. 9. 2-D embeddings of Data C calculated by 12 existing methods and the passage method, respectively.

they can be seen as two elements in  $\pi_1$  and  $\pi_2$ . Thus, two orthogonal coordinates systems of the tangent spaces  $\pi_1$  and  $\pi_2$ ,  $P = (v_1, p_1, p_2, \dots, p_{d_1-1})$  and  $Q = (-v_2, q_1, q_2, \dots, q_{d_2-1})$  ( $d_1$  and  $d_2$ , respectively, are the intrinsic dimensions of the manifold clusters  $X^i$  and  $X^j$ . Here  $d_i \geq d_j$  is the default assumption), can be approximately generated. In particular, the last  $d_i-1$  or  $d_j-1$  axes of  $P$  or  $Q$  are the first  $d_i-1$  or  $d_j-1$  principle directions of  $\Omega_1$  or  $\Omega_2$  after being projected onto the space orthogonal to  $v_1$  or  $v_2$ , respectively. Employing the coordinate systems  $P$  and  $Q$  so constructed, the inter-cluster smooth passage can then be constructed using the following procedure:

1. Compute the  $d_i$ -dimensional coordinates  $\gamma$  of  $\Omega_1 - e_1$  (i.e., centralizing the neighborhood patch  $\Omega_1$  at  $e_1$ ) on the coordinate system  $P$ .
2. Build a sequence of points  $f(t_1), f(t_2), \dots, f(t_m)$  ( $0 < t_1 < \dots < t_{m-1} < t_m = 1$ ) along the curve  $f(t)$  (Eq. (1)).
3. Generate a sequence of coordinate systems  $(P_1, P_2, \dots, P_m)$  along the curve  $f(t)$  as follows: for each  $t_i$  ( $1 \leq i \leq m$ ), set

$$o_s^i = \begin{cases} 3At_i^2 + 2Bt_i + C, & s = 1 \\ (1-t_i)p_{s-1} + t_i q_{s-1}, & 2 \leq s \leq d_j \\ (1-t_i)p_{s-1}, & d_j < s \leq d_i \end{cases}$$

and let  $P_i = (o_1^i, o_2^i, \dots, o_{d_i}^i)$ .

4. Calculate new data sets  $X_i^j$  ( $i = 1, 2, \dots, m$ ) by mapping the coordinates  $\gamma$  onto the coordinate systems  $P_i$ , respectively, and then centralize  $X_i^j$  at  $f(t_i)$  (i.e., let  $X_i^j = X_i^j + f(t_i)$ ).
5. Output the union  $X' = \bigcup_{i=1}^m X_i^j$  as the inter-cluster passage set.

It should be noted that the first axis  $o_1^i$  is set as  $f'(t_i)$  to make the affine space spanned from  $P_i$  be tangent to the curve  $f(t)$ .

By implementing the above procedure, the sequence of  $X_1^j, X_2^j, \dots, X_m^j$  composes the passage set  $X'$ . The passage so constructed has the following properties: it smoothly connects two separate clusters along the continuous curve  $f(t)$  (actually, along the point sequence  $f(t_i), i = 1, 2, \dots, m$ ), and the tangent spaces along the passage are continuously transferred from  $\pi_1$  to  $\pi_2$ . Specifically, the coordinate system  $P_i$  of the tangent space attached to  $f(t_i)$  smoothly changes from the coordinate system  $P$  of the tangent space  $\pi_1$  at point  $e_1$  to  $Q$  of  $\pi_2$  at  $e_2$  along the passage. Therefore, the passage smoothly connecting two separate clusters is generated after this step.

**Step III. Data embedding:** The aim of this step is to calculate the low-dimensional embedding set of the union of the original and the passage data sets,  $X \cup X'$ , and choose from the entire embeddings the subset  $Y$  corresponding to the original data set  $X$  as the final output (as depicted in Fig. 3(c)).

The pseudocode of the whole algorithm is listed in the appendix.

## 2.2. NLDR method selection

Aiming at designating an appropriate manifold learning method to implement NLDR in Step III of the passage method, it is necessary to construct a reasonable assessment criterion to quantitatively evaluate the performance of the method. Remind that the motivation of the passage method is to realize the NLDR of the multi-cluster data such that on one hand the local neighborhood topology of each cluster can be well preserved,

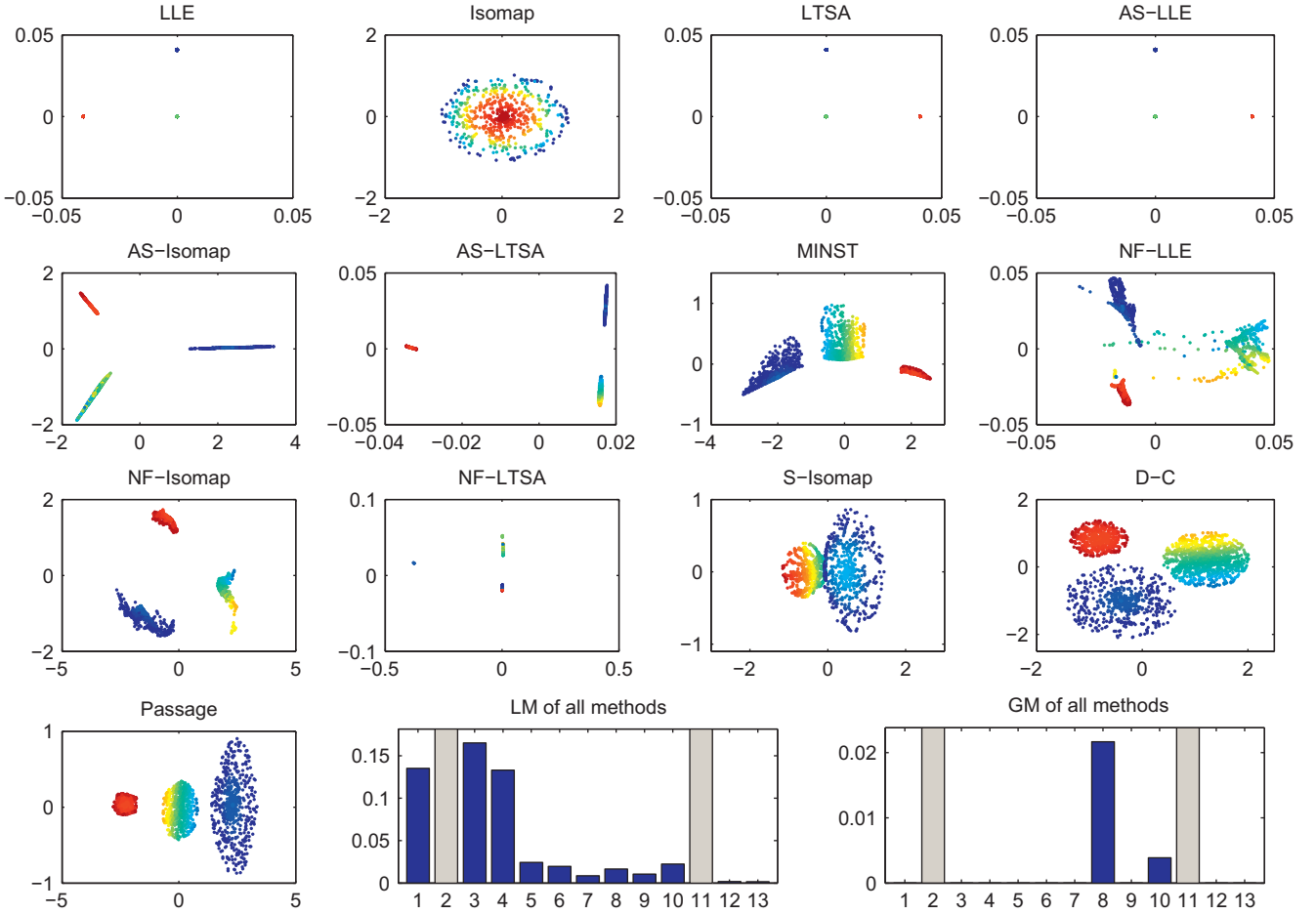


Fig. 10. 2-D embeddings of Data D calculated by 12 existing methods and the passage method, respectively.

and on the other hand the global multi-cluster structure can be finely recovered. Therefore, the criterion should be composed by two parts: local measure (*LM*) and global measure (*GM*). The former evaluates the local-neighborhood-preserving degree of each manifold cluster, and the latter scales the global-structure-holding degree between the calculated embedding data and the original data.

For *LM* evaluation, there are several methods for candidates. The latest developments along this line include the trustworthiness and continuity measure (T&C [8]), the local continuity metacriterion (LCMC [20]), the mean relative rank error (MRRE [21]). Here we employ the continuity measure utilized in T&C method for its easiness of calculation. In this measure, the performance of a dimensionality reduction is assessed by the extent that the  $k$  closest neighbors of a point in the original space are deviated from its neighbors on the reduction. Denote the  $k$  nearest neighbors of the point  $x_i$  as  $\Omega_i = \{x_{i1}, x_{i2}, \dots, x_{ik}\}$ , and denote  $\{y_{i1}, y_{i2}, \dots, y_{ik}\}$  as the  $\{I_{i1}, I_{i2}, \dots, I_{ik}\}$  nearest points from  $y_i$  in the reduction space. Then this measure is calculated as follows:

$$LM = \frac{\sum_{i=1}^l \sum_{j=1}^k I_{ij} - l \sum_{j=1}^k j}{H(k)}, \quad (3)$$

where  $l, k$  denote the data size and the neighborhood size, respectively, and  $H(k) = lk(l-k)$  is the normalization term. In specific, if the local neighborhood topology of the original multi-cluster data is well preserved by the embeddings obtained by the NLDR method, then *LM* so calculated is of a comparatively small value. Otherwise, if some neighborhood structures are

greatly distorted after NLDR implementation, the value of *LM* tends to be its largest value 1.

Comparatively, the *GM* estimation is easier to be attained through the following steps: first to obtain the cluster structure of the input data, then to apply a cluster algorithm to the embedding data to get its categorization, and further to compute the misclassification rate of this categorization based on the prior cluster knowledge. The misclassification rate so calculated can then be taken as the *GM* value. If the data points have already been categorized by an external source, the aforementioned first step can be directly skipped. Many effective clustering methods can be utilized in the above strategy, such as the Meanshift and  $k$ -means [22] methods. It should be noted that although another *GM* measure is presented in [19], it is proposed only for 1-cluster data and does not consider keeping global cluster structures underlying the multi-cluster manifolds. We thus do not employ this measure for *GM* evaluation.

By taking *LM* and *GM* as the assessment criterion, the optimal NLDR method to implement Step III of the passage method can be properly selected.

### 2.3. Computational complexity

The computational complexity is one of the basic issues in the implementation of the passage method. To clarify this point, the computational complexity of its related algorithm is analyzed in this section.

For the algorithm of the passage method, the time complexity of Step I is essentially determined by Step I.1, specifically, by the neighborhood graph construction and the geodesic distance



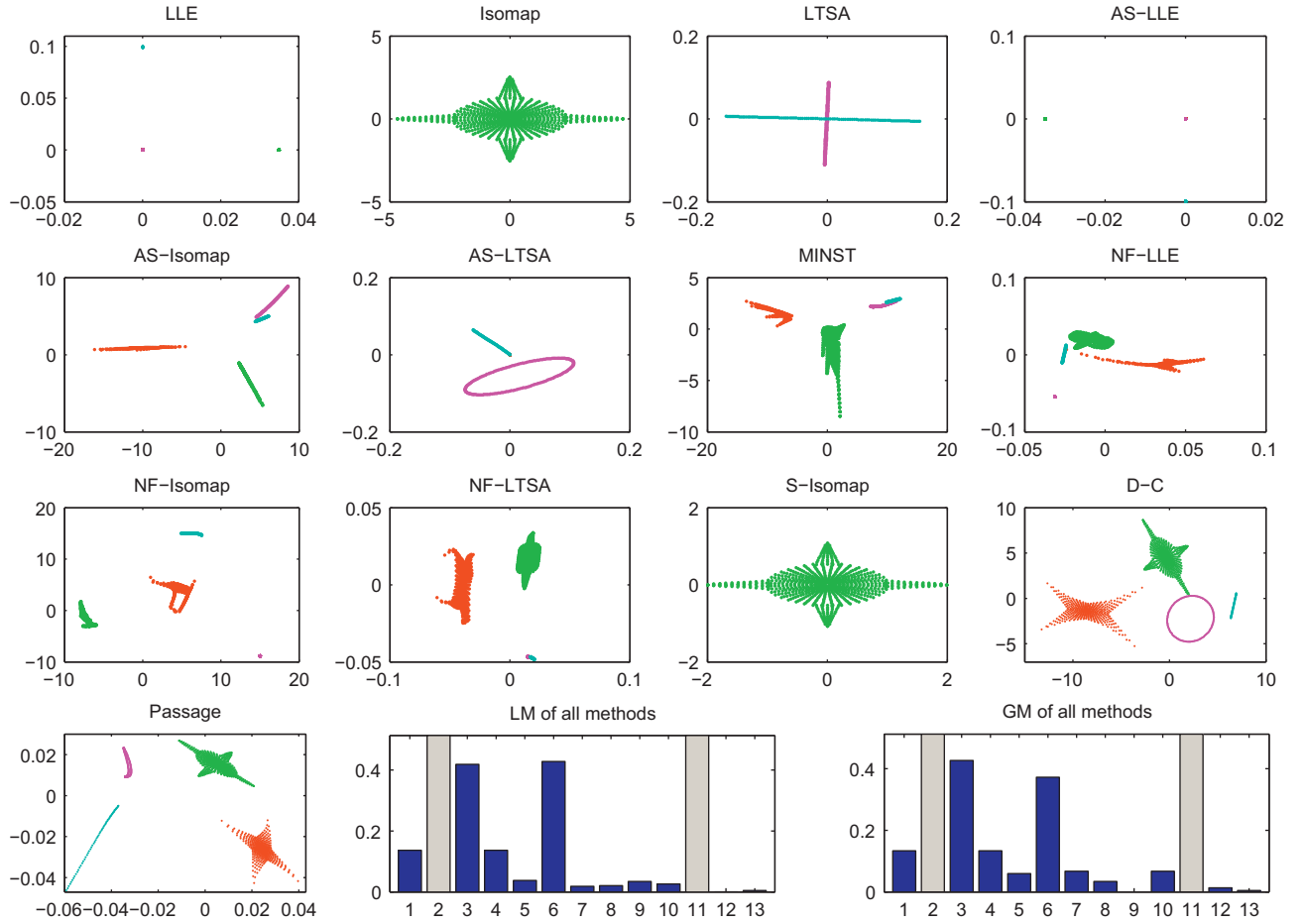


Fig. 11. 2-D embeddings of Data E calculated by 12 existing methods and the passage method, respectively.



Fig. 12. Typical images of the 3-cluster image data and 2-cluster handwritten data.

estimation processes. The former needs to compute the  $k$ -NN neighborhood graph superimposed on the entire data set, costing  $O(nl^2)$  time in the worst case [23]. The latter needs to estimate the shortest path between data pairs, at least taking  $O(l^2 k \log l)$  time by adopting the Floyd's algorithm [17]. Assuming the data set residing on the  $s$ -cluster manifolds ( $s \ll l$  and  $s \geq 2$ ), then Step II at most contains  $s(s-1)/2$  iterations. In each iteration, most of the computational cost is evidently consumed by searching the inter-cluster nearest data (Step II.1) and calculating the principle axes of the local neighborhood patches (Step II.3). By applying the well-known heap sorting algorithm [24], the computational time of the former process is around  $O(l^2/s^2 \log l)$ , and the latter process involves two eigenvector calculations, costing about  $O(k^3)$  time [25]. Thus, the total time complexity of Steps I and II is around  $O(nl^2 + kl^2 \log l)$ .

Actually, the computational speed of these two steps essentially determines the increased computational burden of the proposed method compared with the existing NLDR methods.

This is due to the fact that one of the existing NLDR methods, such as LLE, Isomap, and LTSA, is to be called in the final step of the passage method. Generally speaking, the speed impact of the first two steps of the passage method on the last NLDR step is immaterial. For instance, the time complexities of Isomap and LTSA are around  $O(nl^3 + kl^2 \log l)$  [18] and  $O(nkl^3)$  [4], respectively, higher than that of the first two steps, and the computational cost of LLE is about  $O(nl^2 + nlk^3)$  [1], slightly, but not substantially, lower than that of the first two steps. It should be noted that most of the current NLDR techniques for multi-cluster manifold learning, like the AS, graph-based, NF and S-Isomap methods, need to first construct a connected neighborhood graph on multi-cluster data and then run traditional NLDR methods, like Isomap, LLE and LTSA, on this graph, and thus their computational complexities are no less than these NLDR methods. This is to say, the passage method does not materially increase the time complexity of the existing NLDR methods, including the current multi-cluster manifold learning techniques.

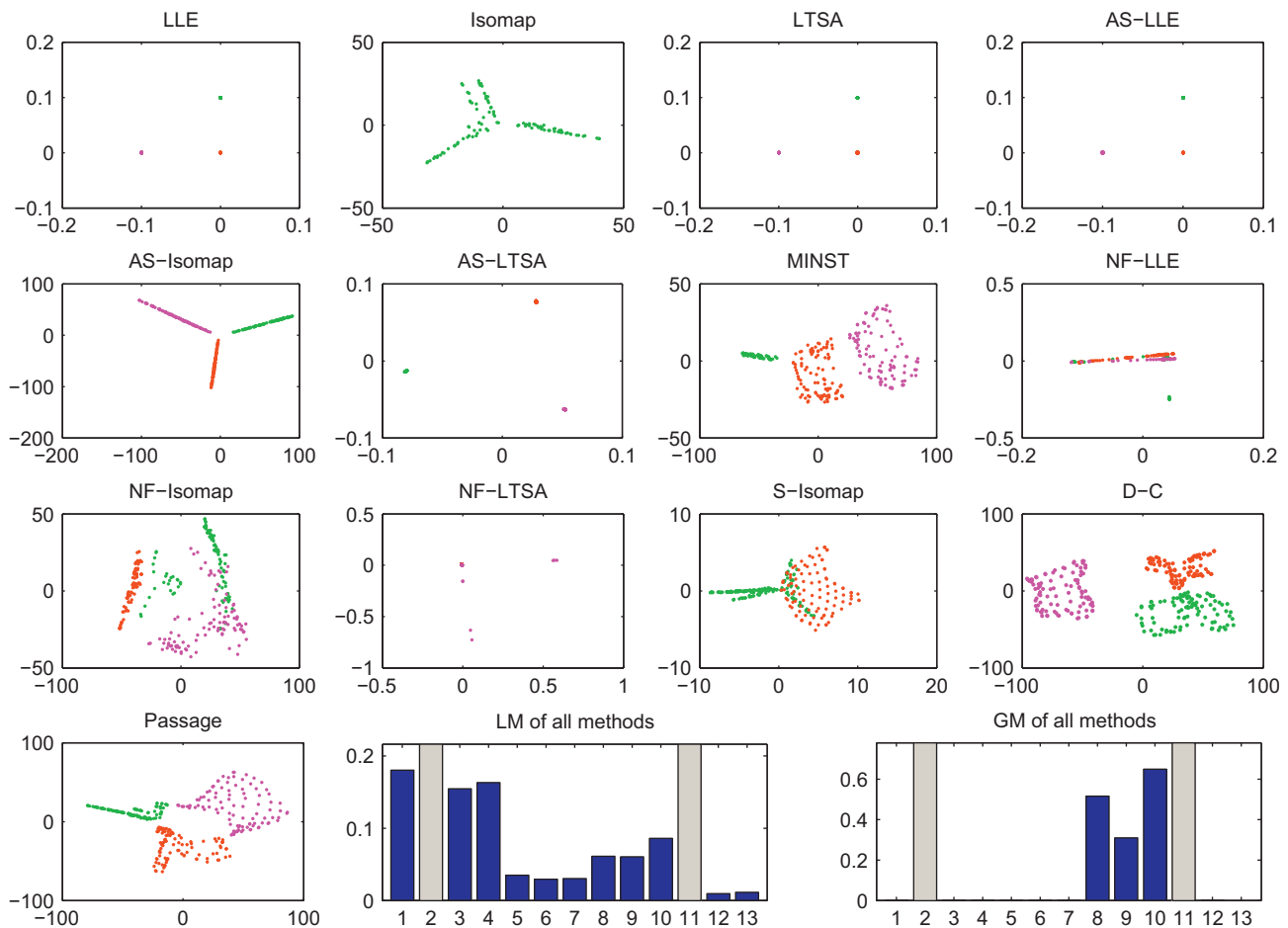


Fig. 13. The embeddings of 3-cluster image data calculated by 12 current methods and the proposed passage method, respectively.

### 3. Experimental results and interpretations

The proposed passage method was applied to multiple synthetic and real multi-cluster data sets. For comparison, 12 of the existing methods, including LLE, Isomap, LTSA, AS-LLE, AS-Isomap, AS-LTSA, MinST, NF-LLE, NF-Isomap, NF-LTSA, S-Isomap, and D-C methods, have also been utilized. The five synthetic data sets are first employed for evaluation, and the two real data sets are then analyzed for further substantiation.

#### 3.1. Simulations on synthetic data sets

The performance of the passage method was first evaluated on five synthetic multi-cluster data sets. The first two data sets (Data A and Data B, as depicted in Figs. 1(b) and 6(b)), both with 2000 points, were uniformly generated from the 2-cluster manifolds as shown in Figs. 1(a) and 6(a), respectively. The third 5-cluster data set contains 3000 points (Data C, as shown in Fig. 6(d)), distributed on the 5-cluster manifolds with a global 3-D S-curve form, composed by five separate sub-manifolds wrapped from 2-D rectangle, diamond, sphere, square and triangle, respectively, as shown in Fig. 6(c). Another 3-cluster data set (Data D, as shown in Fig. 6(f)) was generated from three separate hemispherical surfaces, with radii 1, 0.75, 0.5, respectively (as shown in Fig. 6(e)). Each cluster consists of 600 points, and thus evidently, the three clusters are of different densities. Comparatively, the last 9-D data set (Data E) is designed to test the performance of the proposed method on multi-cluster data with complex intrinsic structures [26]. Its 4 clusters are located in the 1–3, 2–4, 5–6,

7–9 dimensional subspaces of the whole space, respectively. Two of its clusters (as shown in Fig. 6(h) and (j)) are generated from Swiss-roll and S-shaped manifolds (as shown in Fig. 6(g) and (i)), wrapped from two star-like 2-D figures, respectively. Both of the other clusters lie in the manifolds with 1-D intrinsic dimension: one is with O-shape and the other is a curve, as depicted in Fig. 6(k) and (l), respectively.

The embeddings of the five multi-cluster data sets calculated by the passage method are shown in Figs. 7–11, respectively. For all of the five data sets, LLE, Isomap and LTSA were utilized as the NLD methods in Step III of the related algorithms, and the optimal one was evaluated by taking  $LM + GM/2$  as the assessment criterion. For comparison, results obtained by the existing 12 methods are also shown in the corresponding figures.

It can be easily observed from Figs. 7–11 that the proposed method outperforms the existing methods on all of the five multi-cluster data sets, and the disconnectivity, short-circuit and roughness problems all appear to be averted. In particular, the passage method has advantages mainly in the following three-fold aspects. (i) The local intra-cluster topologies are well preserved, which can be observed from the continuously changing colors of the local areas in the calculated embeddings. (ii) The global inter-cluster structure is correctly obtained. These two advantages are also accordantly reflected on the values of  $LM$  and  $GM$  (as shown in the below figures of Figs. 7–11) estimated by utilizing the methods provided in Section 2.2. Especially, for each of the five data sets,  $LM$  of the passage method is the smallest and its  $GM$  is approximately 0, quantitatively showing the good intra-cluster neighborhood-preserving and inter-cluster structure-holding

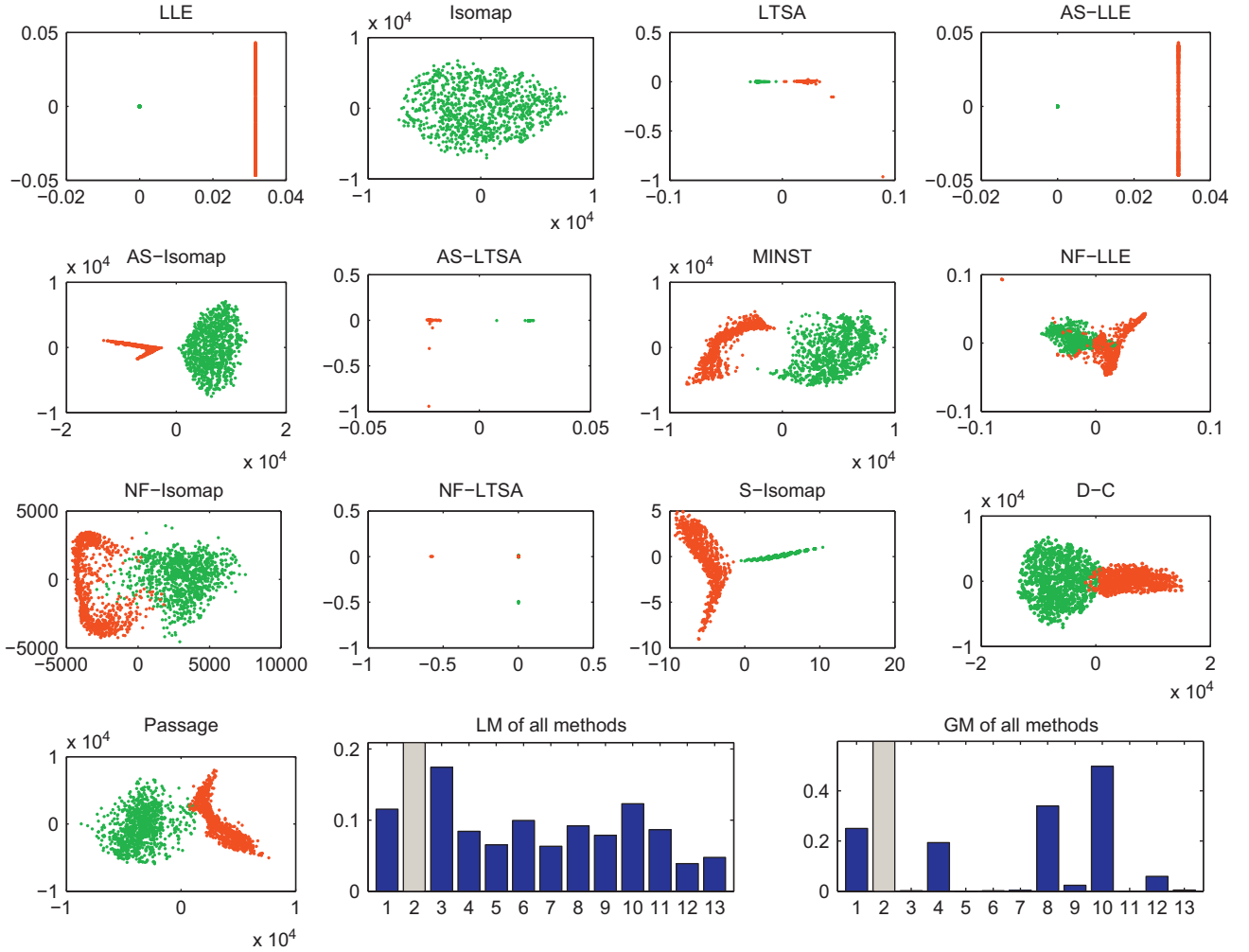


Fig. 14. The embeddings of 2-cluster handwritten data calculated by 12 current methods and the proposed passage method, respectively.

properties of the passage method. (iii) By means of graphical visualization, it is evident that the cluster shapes are properly recovered in all of the five cases by the proposed method. Specifically, embeddings of Data A–D are correctly shaped as two adjacent rectangles, two adjacent circles, five clusters configured as a rectangle, a diamond, a sphere, a square and a triangle, and three adjacent circles, respectively. The intrinsic figures of Data E are also recovered approximately, except that the O-shaped cluster is a little pinched. Based on these advantages, the effectiveness of the passage method in improving the capability of NLDR in multi-cluster manifold learning can be verified, even when the clusters are in different subspaces (e.g., Data E), with different shapes (e.g., Data C) or with different densities (e.g., Data D).

### 3.2. Simulations on real data sets

Two multi-cluster real data sets were employed to testify the performance of the proposed method. The first is a 3-cluster image data set, each containing 100 images of a cartoon panda, a terracotta soldier, and a horse in different poses, respectively. All of the images in the set are gray scale pictures of  $50 \times 60$  pixels (3000-dimension). The second contains 1000 handwritten “0”s and “1”s, each of which is a  $28 \times 28$ -pixel (784-dimension) gray scale picture, randomly selected from the benchmark MNIST database (downloaded from <http://yann.lecun.com/exdb/mnist>).

Some typical images in the two data sets are depicted in Fig. 12. The passage method and 12 existing multi-cluster manifold learning methods were applied to the data sets, and the results are depicted in Figs. 13 and 14, respectively. The LM and GM of the embedding results obtained by all of the methods were also evaluated, as shown in the below sub-figures of Figs. 13 and 14 for comparison.

For both data sets, the excellent performance, in both local and global aspects, is attained by the passage method. On one hand, LM value of the proposed method, as well as the D–C method, is smaller than those of the other 11 methods, implying that the result obtained by the passage method can finely preserve the intra-cluster neighborhood topologies of the original data. On the other hand, GM of the new method approximates 0 in each of the two cases, meaning that the low-dimensional embeddings calculated by the passage method correctly hold the global cluster structures of the input high-dimensional data. Both empirical results show the effectiveness of the passage method on NLDR of data residing on multiple manifold clusters.

## 4. Conclusion

In this paper, we have proposed a new method, called the passage method, for nonlinear dimensionality reduction of data lying on multi-cluster manifolds. The proposed method has been

formulated for the purpose of solving the disconnectivity, short-circuit, and roughness problems commonly encountered in the existing methods for multi-cluster manifold learning. The most distinguished characteristic of the new method is the construction of smooth transitional passages between separate clusters. The passages, together with the input data, can be configured as a smooth manifold such that the effectiveness condition underlying manifold learning can be satisfied, and multi-cluster manifold learning can thus be properly implemented.

There are, however, limitations of the passage method. For example, the proposed method is generally not appropriate when cluster boundaries are imprecise, i.e., very fuzzy. It is due to the implicit assumption of the passage method that clusters can be precisely separated so that they can be properly detected by the clustering method utilized in Step I.1. Besides, it is necessary to try more *LM* and *GM* measures to more reasonably evaluating the performance of the multi-cluster manifold learning methods in our future research, and the effectiveness of the proposed method should be further substantiated by more practical applications.

### Conflict of interest statement

None declared.

### Acknowledgement

This research was supported by the Geographical Modeling and Geocomputation Program under the Focused Investment Scheme at The Chinese University of Hong Kong, the National Grand Fundamental Research 973 Program of China under Grant No. 2013CB329404 and the China NSFC project under contract 11131006 and 61075054.

### Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.patcog.2013.01.028>.

### References

- [1] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [2] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (2000) 2319–2323.
- [3] C.M. Bachmann, T.L. Ainsworth, R.A. Fusina, Exploiting manifold geometry in hyperspectral imagery, *IEEE Transactions on Geoscience and Remote Sensing* 43 (2005) 441–454.
- [4] Z.Y. Zhang, H.Y. Zha, Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment, Technical Report, CSE-02-019, CSE, Penn State Univ., 2002.
- [5] J.A. Lee, A. Lendasse, M. Verleysen, Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis, *Neurocomputing* 57 (2004) 49–76.
- [6] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* 15 (2003) 1373–1396.
- [7] D.K. Agrafiotis, H. Xu, A self-organizing principle for learning nonlinear manifolds, *Proceedings of the National Academy of Sciences* 99 (2002) 15869–15872.
- [8] J. Venna, S. Kaski, Local multidimensional scaling, *Neural Networks* 19 (2006) 889–899.
- [9] L. Yang, Building k edge-disjoint spanning trees of minimum total length for isometric data embedding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005) 1680–1683.
- [10] L. Yang, Building k-edge-connected neighborhood graphs for distance-based data projection, *Pattern Recognition Letters* 26 (2005) 2015–2021.
- [11] L. Yang, Building k-connected neighborhood graphs for isometric data embedding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2006) 827–831.
- [12] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, N. Koudas, Non-Linear Dimensionality Reduction Techniques for Classification and Visualization, SIGKDD, Edmonton, Alberta, Canada, 2002.
- [13] X. Geng, D.C. Zhan, Z.H. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, *IEEE Transactions on Systems, Man, and Cybernetics: Part B* 35 (2005) 1098–1107.
- [14] D.Y. Meng, Y. Leung, T. Fung, Z.B. Xu, Nonlinear dimensionality reduction of data lying on the multi-cluster manifolds, *IEEE Transactions on Systems, Man, and Cybernetics: Part B* 4 (2008) 1111–1122.
- [15] H. Whitney, Differentiable manifolds, *Annals of Mathematics* 37 (1936) 645–680.
- [16] E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik* 1 (1959) 269–271.
- [17] R.W. Floyd, Algorithm 97: Shortest path, *Communications of the ACM* 5 (1962) 345.
- [18] V.D. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, *Advances in Neural Information Processing Systems* 15 (2003) 705–712.
- [19] D.Y. Meng, Y. Leung, Z.B. Xu, Evaluating nonlinear dimensionality reduction based on its local and global quality assessments, *Neurocomputing* 74 (2011) 941–948.
- [20] L. Chen, Local multidimensional scaling for nonlinear dimension reduction, graph layout and proximity analysis, Ph.D. Thesis, University of Pennsylvania, 2006.
- [21] J. Lee, M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer, Berlin, 2007.
- [22] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 603–619.
- [23] L.K. Saul, S.T. Roweis, Think globally, fit locally: unsupervised learning of nonlinear manifolds, *Journal of Machine Learning Research* 4 (2003) 119–155.
- [24] D.E. Knuth, *The Art of Computer Programming: Sorting and Searching*, vol. 3, Addison-Wesley, Massachusetts, 1973.
- [25] T. Cox, M. Cox, *Multidimensional Scaling*, Chapman & Hall, London, 1994.
- [26] X.X. Wang, P. Tino, M.A. Fardal, Multiple manifold learning framework based on hierarchical mixture density model, *Lecture Notes in Computer Science* 5212 (2008) 566–581.

**Deyu Meng** received the B.Sc., M.Sc., and Ph.D degrees in 2001, 2004, and 2008, respectively, from Xi'an Jiaotong University, Xi'an, China. He is currently an associate professor with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University. His current research interests include principal component analysis, nonlinear dimensionality reduction, feature extraction and selection, compressed sensing, and sparse machine learning methods.

**Yee Leung** received the B.Sc degree in geography from The Chinese University of Hong Kong in 1972, the M.Sc and Ph.D. degrees in geography and the MS degree in engineering from The University of Colorado, in 1974, 1977, and 1977, respectively. He is currently a professor of geography of the Department of Geography & Resource Management, The Chinese University of Hong Kong. His current research interests include specialization over the development and application of intelligent spatial decision support systems, spatial optimization, fuzzy sets and logic, neural networks, and evolutionary computation.

**Zongben Xu** received the M.Sc. degree in mathematics and the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1981 and 1987, respectively. In 1988, he was a postdoctoral researcher with the Department of Mathematics, The University of Strathclyde, Glasgow, U.K. He was a research fellow with the Information Engineering Department from February 1992 to March 1994, the Center for Environmental Studies from April 1995 to August 1995, and the Mechanical Engineering and Automation Department from September 1996 to October 1996, The Chinese University of Hong Kong, Shatin, Hong Kong. From January 1995 to April 1995, he was a research fellow with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. He is currently a professor with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University. His current research interests include manifold learning, neural networks, evolutionary computation, and multiple-objective decision-making theory.