DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 6
Special issue with selection of extended papers from 6th International Conference on Logistic, Informatics and Service Science LISS'2016

Sofia • 2016

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0088

Multiple Manifolds Clustering via Local Linear Analysis

Wei Zheng[1], *Shuo Chen*[2]

[1]*School of Computer Engineering, Jinling Institute of Technology, Nanjing 211169, China*
[2]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*
Emails: vividzheng@163.com functioncs@qq.com

***Abstract*:** *Clustering on multiple manifolds serves as an analysis of the data lying on multiple manifolds. The smoothness and local linearity of data samples are utilized to define the local linear degree which is motivated by Principal Component Analysis* (*PCA*) *and Depth First Search* (*DFS*). *Then, Multiple Manifolds Clustering* (*LMMC*) *is proposed on the base of the Local Linear Analysis* (*LLA*) *via this definition and neighbor-growing algorithm, which are especially effective under the condition of interactions. Instead of addressing problems of complex optimization and K-means operation, LMMC is simple and efficient compared with traditional manifold clustering. The algorithm can achieve superior performance on complex subspace and manifolds clustering datasets. Meanwhile, comparative experiments are given to show the effectiveness and efficiency of this algorithm.*

***Keywords*:** *Manifolds learning, clustering algorithm, PCA, DFS, neighborhood.*

1. Introduction

As an important research direction of machine learning and pattern recognition, cluster analysis is the task of grouping a set of data in such a way that objects in the same group are more similar to each other than to those in other groups [1]. As the theory of unsupervised learning flourishes and improves, research importance has been increasingly attached to cluster analysis as a branch. In the new era of big data, it is a meaningful matter of urgency to research the ways of efficient and regular data clustering.

Over the past decades, K-means clustering has been successfully applied to a reasonable number of practical cluster problems, albeit typical of lazy learning [11, 12]. However, in the big data era represented by complicated data samples in bulk, it is gradually beyond the scope of K-means clustering to meet current demands as a simple and conventional algorithm. For instance, K-means clustering always fails to solve problems arising from linear separability of a pair of sets correctly and rationally. Clustering on linearly separable data sets is surely all the way challenging.

The major difficulty lies in exploring the law of homogeneous data classification. Meanwhile, there is ambiguity in ways to group areas of overlap composed of data between classes in data space. With different types of data clustering for overlapped zones, areas but for those overlapped may be wrongly partitioned and the entire cluster outcome will be destroyed as a result. Given this, proper data clustering for overlapped areas determines the success of the whole cluster analysis.

In recent years, Spectral Clustering (SC) [2, 3] has attracted numerous research attentions. According to related research findings, massive challenging cluster problems have been addressed using SC. K-means clustering, remaining as the starting point of SC, uses the eigenvector of the affinity matrix constructed by SC instead of raw sample data. This simple starting point indeed tackles problems of basic Multiple Manifolds Clustering (MMC), such as the condition that homogeneous data obey the distribution on low-dimensional subspace or manifolds. However, a dominant prerequisite for SC is that no areas of overlap should be generated from data between classes. Any cluster problem handled by SC without this prerequisite will produce a result less effective than it can have been. Despite the said constraints, there remains heated discussion on SC theories. For example, the SC-based SMMC, as one of cluster algorithms [6], effectively groups data between classes in overlapped areas by training Probabilistic Principle Component Analysis (PPCA) factors on part of data samples. Nevertheless, super high spatiotemporal costs restrict its application to big data situations.

K-manifold Algorithm is another recently popular cluster algorithm [14]. For the first time, among all the attempts made, K-manifold Algorithm handles MMC problems in overlapped areas. However, it behaves badly for well-separated clusters. Meanwhile, the classic concept of divide and conquer has been employed to solve cluster problems [19], for which data is partitioned into single manifolds and overlapped manifolds. The latter one is further divided into overlapped areas and non-overlapped areas. Graphic models are established as the final step to complete clustering tasks. The effectiveness of the divide and conquer algorithm rests on discrimination of overlapped areas as well as estimation of local dimensions during the process of searching by graphic models. Parameter selection plays an important role at the same time. Thus the divide and conquer algorithm may encounter certain difficulties in practice [6].

Data in SC-based data clustering is mainly characterized by its distribution on multiple low-dimensional manifolds. Recently, some research work has been done on data clustering in several amounts of subspace, i.e., subspace clustering [2, 4, 5, 7, 8]. The predecessor of subspace clustering is the theory of subspace learning like sparse decomposition and Low Rank Representation (LRR). LRR-based subsequence clustering is a representative one, for which raw data matrices are decomposed on a low-rank basis before the obtained low-rank coefficients undergo K-means clustering so as to complete clustering. From the mathematical perspective, as subspace features linear manifolds, the corresponding subspace clustering is a matter of linear MMC. According to present-day research progress and previous mathematical experience, clustering of linear data is easier soluble than nonlinear types in most cases. What is more, many actual data spread over massive linear

subspaces. Therefore, it is of certain practical significance to conduct specific studies on data clustering in linear subspaces. However, this algorithm usually fails to find its application in manifold clustering. Despite the usage of some quadratic-fit-based methods [14] to approximate manifolds, it is still difficult to solve most complicated multi-manifold data cluster problems efficiently.

2. Multi-Manifold Cluster algorithm based on Local linear analysis (LMMC)

The main objective of MMC is to partition sets of label-free points $X = [\mathbf{x}_1, ..., \mathbf{x}_n] \in R^{d \times n}$ into $X_1, ..., X_K$ with the hope that every type of clustered points is visually or mathematically smooth. Fig. 1 is the main test data sets that involve MMC. First, we propose a MMC algorithm that specializes in non-overlapped areas. We then broaden its scope of application to overlapped areas under the guidance of local linear analysis.

Many documents have put forward feasible solutions to MMC in non-overlapped areas, such as SC and sparse LRR. In order for an unimpeded searching for solution to a local linear analysis model that we will expound later, we devise a neighbour-growing algorithm based clustering for MMC in non-overlapped areas. This cluster algorithm is as simple as referring to connectivity (graph theory), and handles non-overlapped MMC effectively.

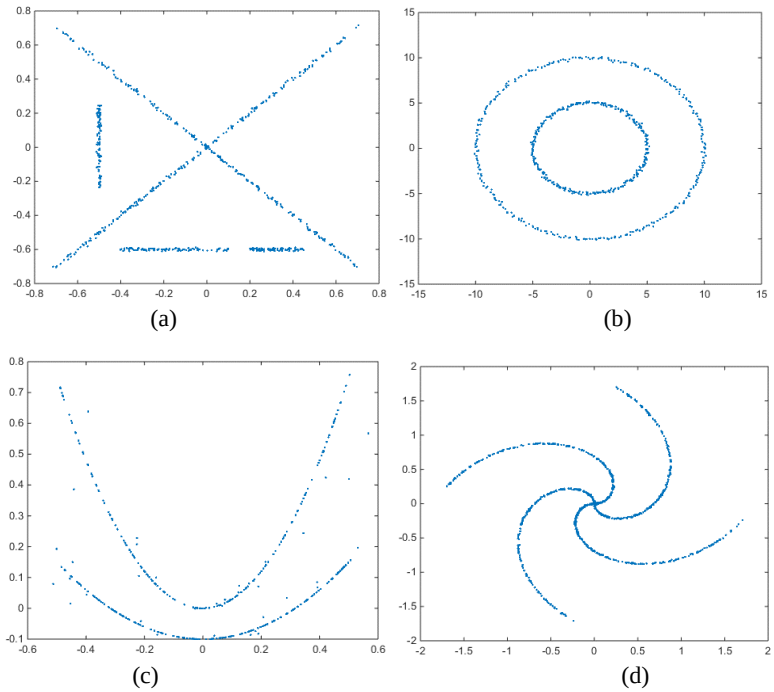


Fig. 1. Test data sets for MMC: A pair of intersecting lines in a two-dimensional surface (a); a triple of intersecting surfaces in three-dimensional space (b); a pair of smooth lines in a two-dimensional surface(c); a pair of intersecting S-shape curves in a two-dimensional surface (d)

Inspired by the thought of connected spatial domains proposed in document [9], we define the clustering-oriented concept of neighbour joining as follows.

Definition 1. We assume a pair of points $\mathbf{x}_1, \mathbf{x}_2 \in X$, $X \subseteq R^{d\times n}$; $\mathbf{x}_1, \mathbf{x}_2$ on X is δ-neighbour connected if and only if there exists a finite element sequence $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_t$ and a neighbour radius δ that let

$$N_\delta(\mathbf{x}_1) \subseteq X, \quad N_\delta(\mathbf{w}_i) \subseteq X, \quad i = 1, ..., t,$$

and

$$\mathbf{w}_1 \in N_\delta(\mathbf{x}_1), \mathbf{w}_2 \in N_\delta(\mathbf{w}_1), ..., \mathbf{w}_t \in N_\delta(\mathbf{w}_{t-1}), \mathbf{x}_2 \in N_\delta(\mathbf{w}_t).$$

The point set X itself is δ-neighbour connected if and only if a pair of arbitrary points in X is δ-neighbour connected. Furthermore, we define that a pair of point sets (X and Y) is δ-neighbour connected if and only if there exists $\mathbf{x}_0 \in X$ and $\mathbf{y}_0 \in Y$ that are δ-neighbour connected.

The above mentioned definition is the key premise of neighbour-growing algorithm. The major idea of neighbour-growing algorithm in non-overlapped areas [8] is built up on the basis that a single manifold is connected whilst manifolds are not connected to each other. Taking double-curve manifold data as an example (as shown in Fig. 2), the pair of curves is independently connected, but fails to connect to each other.

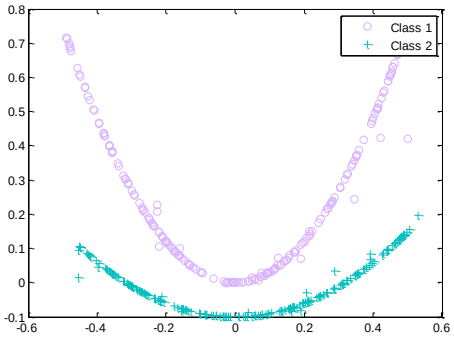


Fig. 2. The diagram of regional connectivity

The main steps of non-overlapped neighbor-growing algorithm are:

1. *Initialization*: Label a sample point as 0;
2. *Growth*: Let within-neighbor points labelled zero grow, and redistribute class labels to them;
3. *Render*: All the classes emerging in Step 2 merge into one.

We describe the process of the algorithm as:

Algorithm 1. Non-overlapped neighbour-growing algorithm

Input: Sample points set $X \subseteq R^{d\times n}$, neighbour radius δ, present class label No $D = 1$.

Initialization: Choose initial point $\mathbf{x} = \mathbf{x}_i$, $i = 1$, sample label $L_k = 0$, $k = 1, \ldots, n$.

Repeat:

Step 1. Check points falling in $N_\sigma(\mathbf{x}_i)$.

Step 2. If all the points falling in $N_\sigma(\mathbf{x}_i)$ have not been labelled, endow them with class label D, and let $D = D + 1$.

Step 3. If there exists a point y in $N_\sigma(\mathbf{x}_i)$ that belongs to label L', then label all other points as L'.

Step 4. Let point No $i = i + 1$.

Output: class label $\mathbf{L} = (L_1, \ldots, L_n)$.

The essence of non-overlapped multi-manifold neighbour-growing algorithm is: a depth-first transversal is done on all neighbour-joining points, and points in neighbour-joining point sets will be labelled the same; two point sets that are not neighbour-joining will be labelled differently after the execution of this algorithm. The following theorem can be summarized out.

Theorem 1. X and Y, a pair of δ-neighbour joining point sets, are not δ-neighbour connected to each other. Thus, if we input $Z = X \cup Y$ to Algorithm 1, the output of class label L must satisfy the following conditions:

$$\begin{cases} L(\mathbf{p}) = L(\mathbf{q}) & \mathbf{p}, \mathbf{q} \text{ fall in the same set,} \\ L(\mathbf{p}) \neq L(\mathbf{q}) & \mathbf{p}, \mathbf{q} \text{ fall in different sets.} \end{cases}$$

Proof:

As X is δ-neighbour joining, when we traverse points in X and take neighborhood into account, two arbitrary connected points will be connected by a finite neighborhood sequence before merging into one class. Therefore, all points in X will be grouped in the same class. Similarly, all points in Y will be grouped in the same class.

In addition, we assume that X and Y can merge into one, then it is sure that a certain finite neighborhood sequence [9] connects two points in X and Y, which is contradictory to the premise that X and Y are not δ-neighbour connected to each other. Therefore, we conclude that X and Y cannot be integrated into the same class. Until now, we have proved Theorem 1.

For non-overlapped areas, Algorithm 1 performs well. However, considering that our research focus is clustering of multiple manifolds with overlapped points, the use of Algorithm 1 may fail to conform to the requirement that multiple overlapped manifolds are partitioned into independent classes. In essence, the main problem is to categorize overlapped areas. To this end, prior analysis of overlapped areas is a necessity.

Given the smoothness and low-dimension of manifolds for us to cluster, especially of local areas, we take data sets a, b, c, d (see Fig. 1) as examples to detect the local, linear characteristics of every points in the data sets. The local characteristics are always embodied by linearity [10]. A point set that does not feature low dimension in a locality is definitely unsmooth and not manifold alike in the locality [10].

Principal component analysis is conducted on all points contained in the neighborhood of every point (or every row in the neighbour-joining matrix) in the said data sets. The reason is that when there is a point located in a low-dimensional

subspace, the eigenvalue of the sample's covariance matrix will promptly attenuate, or only a few is none-zero element. In this way, we measure the degree to which a data set fits a one-dimension subspace by using the contribution degree of the maximum eigenvalue among all eigenvalues. Similarly, we measure the degree to which a data set fits a two-dimension subspace by using the combined contribution degree of the first two largest eigenvalue among all eigenvalues.

Specifically, we conduct principal component analysis on points in the neighborhood with its radius of 0.4, 0.5, 0.2, 0.3, respectively, and accordingly plot the ratio of the maximum eigenvalue to the sum of eigenvalue of all points in the data set, as shown in Figs 3-6. On the right is the spatial distribution of points in each data set. On the left is the distribution of the said ratio. Points with deeper colors have larger ratio.

As can be seen from Figs 3-6, the ratio of eigenvalue in overlapped areas is extremely low, whilst quite high in non-overlapped areas. Such statistics (ratio of eigenvalue) with distinguishing features help find solutions to point clustering in overlapped areas. In addition, the final clustering result is not affected by a few outliers in non-overlapped areas due to low ratio of eigenvalue.

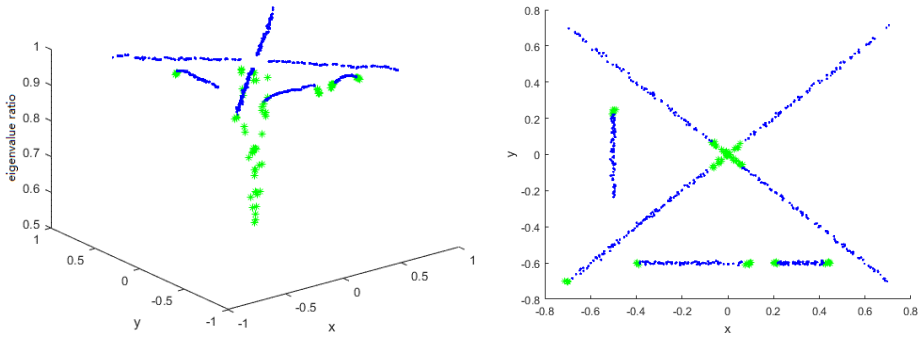


Fig. 3. Distribution of local linear degree (ratio of eigenvalue) for data a (Fig. 1)

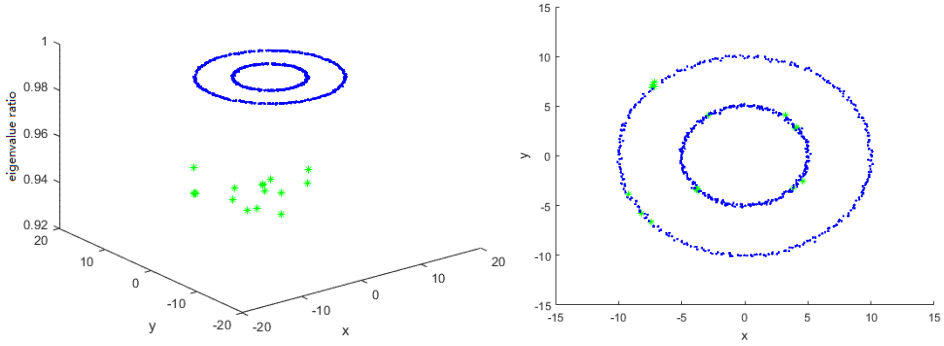


Fig. 4. Distribution of local linear degree (ratio of eigenvalue) for data b (Fig. 1)

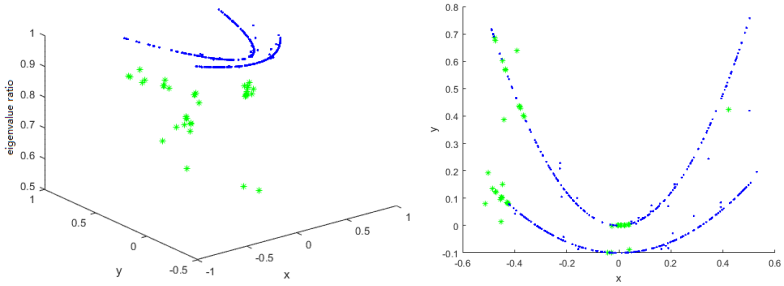


Fig. 5. Distribution of local linear degree (ratio of eigenvalue) for data c (Fig. 1)

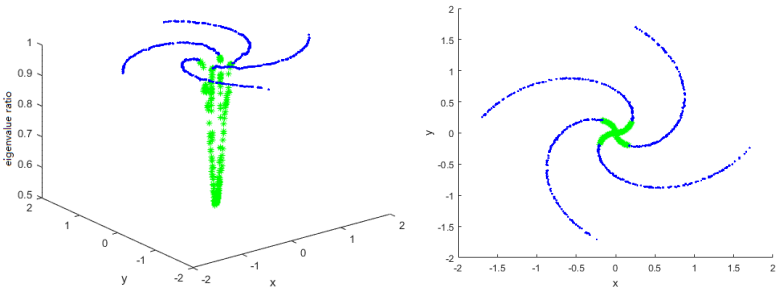


Fig. 6. Distribution of local linear degree (ratio of eigenvalue) for data d (Fig. 1)

The general mode of solution to the problem of multiple manifold clustering is: for a point set $X = [x_1, ..., x_n] \subseteq R^{d \times n}$ in sample space, one of our desires is to group points as $X = \bigcup_{i=1}^{K} X_i$, (and $\bigcap_{i=l}^{K} X_i = \varnothing$), and the other is to render every class as smooth as possible, so as to ensure points of every class to fall onto a smooth manifold. Against this background, it is particularly important to define the degree of smoothness of a point in a certain class. Considering the local linearity of a manifold, we measure the smoothness degree of a class sample space by inspecting the linearity of a sphere neighbourhood $N_\delta(x) = \{y | \, ||y - x||_2 < \sigma\}$ of a sample point in a class sample space and by integrating the linearity of all neighbourhoods.

Now we mainly use principal component analysis to detect the linear information of $N_\delta(x)$. By principal component analysis, we are capable of finding the major distribution law of a sample matrix in low-dimensional subspace. For example, a d-dimensional sample matrix that is proved to have only one non-zero eigenvalue by principal component analysis can be regarded as one-dimensional in essence, i.e. all the sample leaves are distributed in a one-dimensional manifold.

Accordingly, local linearity $R(A, p)$ is defined as the accumulated contribution degrees of the first p-dimensional principal components of a sample point matrix $A \in R^{d \times m}$, and can be expressed as

$$R(A, p) = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{d} \lambda_i}, \tag{1}$$

where $p < d$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ are the eigenvalues of covariance counterpart A.

As the whole sample point set is partitioned into a number of small neighbours, we can describe the smoothness of the point set by p-dimensional local linearity of every small neighbour. We then define the smoothness matrix $M(X)$ of the sample point set.

The total smoothness $M(\Omega)$ of a sample point set Ω is

$$M(\Omega) = \int_{\Omega} R(N_{\delta}(\omega), p)\, d\omega = \sum_{\omega \in \Omega} R(N_{\delta}(\omega), p) \Delta\omega. \tag{2}$$

Assuming that the label of the set X of all sample points is vector L, whose number of classes is K, $L_i = 1, 2, ..., K$, we then define the smoothness of sample point set X as

$$f(L, K) = \sum_{i=1}^{K} \frac{M(X_i)}{|X_i|}, \tag{3}$$

where $|X_i|$ is the cardinality of the set X_i (or the number of elements for a finite set). We eliminate the influence of data numbers on clustering effect by dividing the total smoothness by cardinality. In this way, we can rationally separate smooth manifolds into two independent classes which contain either few or many samples, without being interfered by the efforts the algorithm makes to balance sample volume. It is our expectation that the result of classification reaches its maximum smoothness. Thus, we construct an optimization model in relation to class label:

$$\begin{aligned} \max \quad & f(L, K) = \sum_{i=1}^{K} \frac{M(X_i)}{|X_i|}, \\ \text{s.t.} \quad & M(X_i) = \int_{X_i} R(\omega)\, d\omega, \\ & X_i = \{\mathbf{x}_j \,|\, L_j = i\}. \end{aligned} \tag{4}$$

This model involves a matter of combinatorial optimization. The optimization objective is the overall smoothness of each sample class, and the strategy variable [13] is class label. We attempt to find a best class label towards the highest smoothness of all classes of sample data.

However, this matter of combinatorial optimization is obviously NP-hard. The sole but unpractical access to optimal solution is an inch-by-inch search for the possible number of classes K and class label variance L. To this end, it is a necessity to find an approximate solution to that model so as to produce acceptable fruits within limited time. Fortunately, we have achieved success by modifying the aforementioned non-overlapped neighbour-growing clustering algorithm to a certain extent. Below is our exhibition of the reasonably modified technique.

When principal component analysis is done on the neighbourhood of every point, we obtain different values of local linearity R, as shown in Figs 3-6. Nevertheless, the fundamental law reflected by them lies in a high linearity of the smooth area and a low linearity of the overlapped part. An understandable explanation is that the points in the overlapped area come from multiple manifolds [14] and cannot be smoothed together. In other words, in order to maximize the

objective function, these points should not be grouped in the same class, or else the value of the objective function will plummet due to less smoothness. Given this, a natural idea strikes us that these points be omitted before we complete clustering smooth points.

The proposed LMMC Algorithm in this paper reflects the thought of greedy algorithm in essence. Every time we arrive at a tiny, snap decision at the local level as a second-best solution, albeit uncertain about whether we miss the optimal one. Specifically, the major thought of finding solutions to the problem of overlapped manifold clustering based on neighbour-growing algorithm is: We conduct principal component analysis on the neighbourhood of each point, and obtain the local linear degree of the points. Only those with high linearity are included in the neighbour-growing algorithm, and the rest are marked as "critical points". When the algorithm operates, principal component analysis is done on the neighbourhood of each point, whose principal component is the direction of its manifold. Finally, points with similar manifold directions are integrated into the same class.

Here are some necessary definitions concerning the algorithm.

Definition 2. The direction of **x**-manifold $D(\mathbf{x})$ is the principal component of points in $N_\delta(\mathbf{x})$ obtained by principal component analysis.

Definition 3. The distance of **x**-manifold from **y**-manifold, represented by Dist(**x**, **y**), is

$$\text{Dist}(\mathbf{x}, \mathbf{y}) = \frac{D(\mathbf{x})^{\mathrm{T}} D(\mathbf{y})}{\|D(\mathbf{x})\|_2 \|D(\mathbf{y})\|_2} = \cos\langle D(\mathbf{x}), D(\mathbf{y})\rangle. \tag{4}$$

Here is how we describe the multiple manifolds clustering algorithm based on local linear analysis:

Algorithm 2. LMMC clustering

Input: Sample point set $X \subseteq R^{d\times n}$, neighbor radius δ, current class label No $b = 1$.

Initialization: Initialize sample label $L_k = 0,\ k = 1,...,n.$

Define the threshold of contribution degree for principal component analysis Thr.

Define the inherent dimension of manifold Dim.

Define the set of critical points $T = \varnothing$.

Initialize $i = 1$, transverse $\mathbf{x}_i \in X$.

Repeat:

Step 1. Undertake principal component analysis on points in $N_\delta(\mathbf{x}_i)$, and obtain the local linear degree $R(N_\delta(\mathbf{x}_i), \text{Dim})$ and the direction of $\mathbf{x}$-manifold $D(\mathbf{x}_i)$.

Step 2. If $R(N_\delta(\mathbf{x}_i), \text{Dim})$ exceeds the threshold of contribution degree Thr, continue to Step 3; or otherwise we add $\mathbf{x}_i$ to the set of critical points *T*, and jump to Step 5.

Step 3. If no points in $N_\delta(\mathbf{x}_i)$ are grouped, renew the class labels of all points in $N_\delta(\mathbf{x}_i)$ as b, let $b = b+1$, and jump to Step 5.

Step 4. If there exists a point $\mathbf{y}$ in $N_\delta(\mathbf{x}_i)$ that is grouped to class L', then unify the class label of all points in $N_\delta(x_i)$ as L'.

Step 5. Let point No $i = i+1$.

Step 6. If $i > n$, the loop ends.

Initialize: $i = 1$, traverse $\mathbf{z}_i \in T$.

Repeat:

Step 1. Calculate manifold distance according to Definition 3, and extract two points $\mathbf{z}_1^*$, $\mathbf{z}_2^*$ from $N_\delta(\mathbf{z}_i)$, which have the nearest manifold distance and satisfy the condition that at least one of $L(\mathbf{z}_1^*)$, $L(\mathbf{z}_2^*)$ is non-zero (i.e., there is at least one point of them that falls out of the set of critical points *T*).

Step 2. If $L(\mathbf{z}_1^*) \neq L(\mathbf{z}_2^*)$, set all points of the classes to which $\mathbf{z}_1^*$ belongs to the same label.

Step 3. Let point No $i = i+1$. If $i > |T|$, the loop ends.

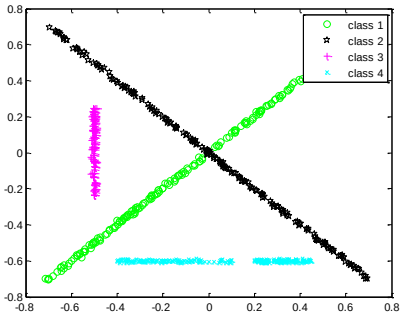
Output: Class label *L*.

Algorithm 2 contains two loops. The first one is to search for points in non-overlapped areas, followed by clustering analysis. By means of neighbour-growing, neighbour-joining points are grouped into the same class. The second one is to process points in overlapped areas (critical points). Under neighbourhood conditions, points in the critical point set are constantly sifted according to their distance of manifolds from labelled points in the non-critical point set, and the most approaching one is labelled the same. The proposed LMMC Algorithm is free from complicated arithmetical calculations, but is characterized by its simplicity of eigenvalue decomposition and low time complexity. Specifically, it is required for LMMC to decompose the eigenvalue of the set of local sample points, which does not slow the operation of the algorithm with generally small numbers. Moreover, no additional K-means clustering is demanded in this process. Instead, clustering analysis is directly done on sample point data using neighbour-searching.

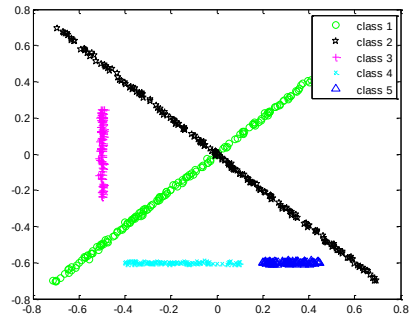
3. Experiments and simulation

In this section, we writes MATLAB programs to conduct experiments and simulation test on the proposed multi-manifold clustering algorithm using common manifold clustering data sets. The computer configuration is Intel I5 processor 4500 (1.8 GHz), 8GB DDR3 ROM, and the program debugging environment is MATLAB2014b.

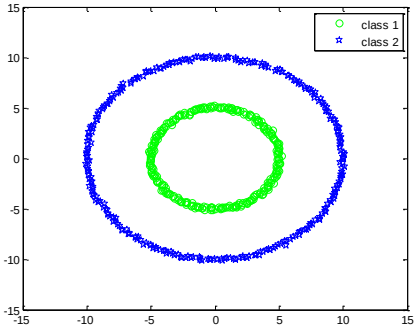
During the process of the experiment, we change related algorithm parameters within a certain scope, in order to find the best one by means of network searching. Fig. 7 is the clustering result of LMMC on each data set and the corresponding actual labels.



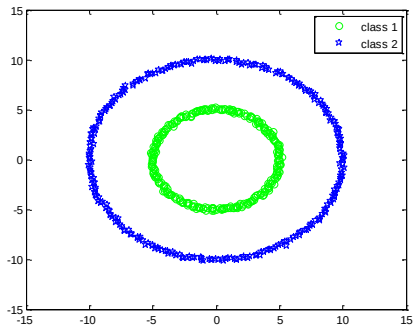
a(1) Experimental result



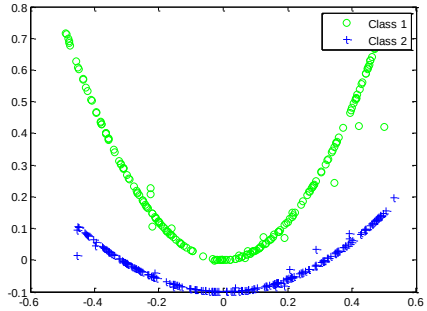
a(2) Actual result



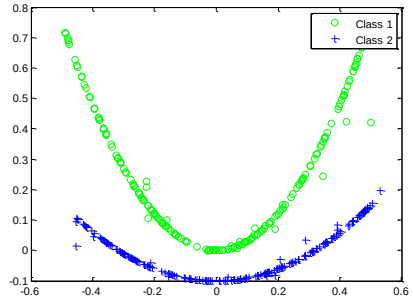
b(1) Experimental result



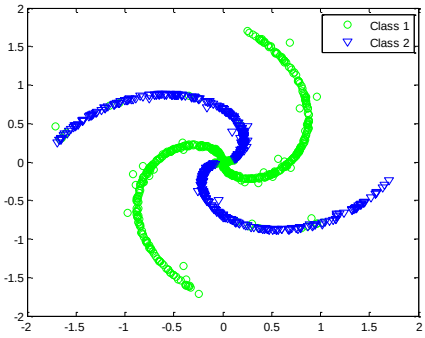
b(2) Actual result



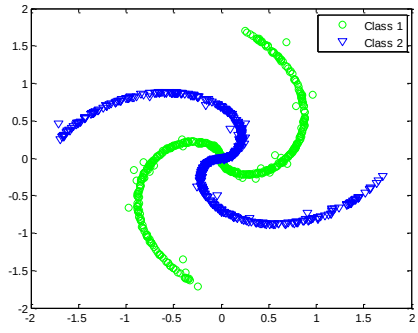
c(1) Experimental result



c(2) Actual result



d(1) Experimental result



d(2) Actual result

Fig. 7. The clustering result of LMMC on data set (a, b, c, d, Fig. 1) and the corresponding actual labels

Table 1 shows the accuracy rate of LMMC clustering and other clustering methods such as GPCA [20], SC [12], SCC [18], K-means [11], K-planes [21], K-manifolds [19]. Table 2 is the time (s) required for LMMC and other methods to run on each data set. Data set a, b, c and d is intersecting lines, hyperbola, concentric circles and a pair of intersecting S-shaped curves as shown in Fig. 1, respectively.

Table 1. Accuracy rate of LMMC clustering and other clustering methods (percentage)

Method	Data set (Fig. 1)			
	a	b	c	d
GPCA	78.7	98.3	50.0	52.4
SC	40.8	83.1	50.7	58.8
SCC	94.9	98.7	100.0	59.6
K-means	60.6	36.6	50.5	58.2
K-planes	98.3	58.9	50.5	55.2
K-manifolds	59.0	95.3	59.3	96.8
LMMC	**95.0**	**100.0**	**100.0**	**90.7**

Table 2. Time (s) required for LMMC and other methods to run on each data set

Method	Data set (Fig. 1)			
	a	b	c	d
GPCA	0.01	0.01	0.01	0.01
SC	2.39	3.29	1.38	4.29
SCC	3.06	1.95	0.56	0.71
K-means	0.01	0.02	0.01	0.01
K-planes	0.01	0.01	0.01	0.01
K-manifolds	144.60	837.21	59.39	261.39
LMMC	**1.12**	**3.54**	**1.12**	**5.34**

It can be seen from the above experimental result, that with steady clustering performance, LMMC Algorithm is an effective approach to addressing the problem of nonlinear clustering and overlapped manifold clustering. Despite the poor performance in handling outliers at times, the output of this algorithm does not greatly deviate from actual clustering results which will otherwise be caused by outliers. Meanwhile, the small number of outliers is powerless in weakening the overall clustering effect.

4. Conclusion

This paper starts from the local linearity of data lying on multiple manifolds, and proposes a simple but effective multiple manifolds clustering algorithm which reflects the thought of depth-first search, neighbour-joining and principal component analysis. The principal component theory provides a basis for this algorithm to define linearity. Highly-efficient clustering analysis is done on sample points using the concept of depth-first search and neighbour-joining. The experiment result shows that this algorithm has strong robustness in dealing with manifold data in overlapped areas, and performs well against non-linear manifold clustering. In terms of time complexity, this algorithm remains in a low complexity, albeit slightly slower than K-means algorithm and other extremely simple algorithms. The shortcoming of this

algorithm lies in its sensitivity to neighbour-size parameters. Thus, our follow-up research will be focused on improving parameter sensitivity using advantageous spectral clustering.

***Acknowledgements*:** The paper is sponsored by the Provincial University Natural Science Research Foundation of Jiangsu Education Department (Grant No 16KJB520012).

References

1. Duda, R. O., P. E. Hart, D. G. Stork. Pattern Classification. 2nd ed. New York, Wiley, 2000.
2. Vida, R. Subspace Clustering. – IEEE Signal Processing Magazine, Vol. **28**, 2011, No 2, pp. 52-68.
3. Shi, J., J. Malik. Normalized Cuts and Image Segmentation. – IEEE Transactions Pattern Analysis Machine Intelligence, Vol. **22**, 2000, No 2, pp. 888-905.
4. Liu, G., et al. Robust Recovery of Subspace Structures by Low-Rank Representation. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **68**, 2013, No 1, pp. 171-184.
5. Elhamifar, E., R. Vidal. Sparse Subspace Clustering: Algorithm, Theory, and Applications. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **35**, 2013, No 11, pp. 2765-2781.
6. Wang, Y., et al. Spectral Clustering on Multiple Manifolds. – IEEE Transactions on Neural Networks, Vol. **22**, 2011, No 7, pp. 1149-1161.
7. Cheng, B., et al. Multi-Task Low Rank Affinity Pursuit for Image Segmentation. – In: International Conference on Computer Vision, IEEE, 2011, pp. 2439-2446.
8. Lang, C., et al. Saliency Detection by Multi Task Sparsity Pursuit. – IEEE Transactions on Image Processing, Vol. **21**, 2012, No 3, pp. 1327-1338.
9. Bertsekas, D. P. Nonlinear Programming. Belmont, Athena Scientific, 1999.
10. Roweis, S. T., L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. – Science, Vol. **290**, 2000, No 5500, pp. 2323-2326.
11. Hartigan, J. A., M. A. Wong. A k-Means Clustering Algorithm. – Journal of the Royal Statistical Society, Series C (Applied Statistics), Vol. **28**, 1979, No 1, pp. 100-108.
12. Ng, A., M. Jordan, Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. – Advances in Neural Information Processing Systems, Vol. **2**, 2002, pp. 849-856.
13. Bishop, C. M. Pattern Recognition. Springer, 2006.
14. Goldberg, A. B., et al. Multi-Manifold Semi-Supervised Learning. – AISTATS, 2009, pp. 169-176.
15. Zhang, K., J. T. Kwok. Clustered Nyström Method for Large Scale Manifold Learning and Dimension Reduction. – IEEE Transactions on Neural Networks, Vol. **21**, 2010, No 10, pp. 1576-1587.
16. Haro, G., G. Randall, G. Sapiro. Translated Poisson Mixture Model for Stratification Learning. – International Journal of Computer Vision, Vol. **80**, 2008, No 3, pp. 358-374.
17. Zhang, Z, H. Zha. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. Department of Computer Science and Engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.
18. Chen, G., G. Lerman. Spectral Curvature Clustering (SCC). – International Journal of Computer Vision, Vol. **81**, 2009, No 3, pp. 317-330.
19. Wang, Y., et al. Multi-Manifold Clustering. – In: Pacific Rim International Conference on Artificial Intelligence, Springer Berlin Heidelberg, 2010, pp. 280-291.
20. Vidal, R., Y. Ma, S. Sastry. Generalized Principal Component Analysis (GPCA). – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **27**, 2005, No 12, pp. 1945-1959.
21. Cappelli, R., D. Maltoni. Multispace KL for Pattern Representation and Classification. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **23**, 2001, No 9, pp. 977-996.