Improved Manifold Learning with Competitive Hebbian Rule

Qiang Gan, Furao Shen*, Jinxi Zhao

***Abstract*—Manifold Learning methods aim to find meaningful low-dimensional structures hidden in their high-dimensional observations. Recently, they are faced with critical problems of how to reduce computational and space complexity in big data applications, how to determine neighborhood size adaptive to different data sets and how to deal with new observations in an out-of-sample mode. This paper presents a new method called TLOE (Topology Learning and Out-of-sample Embedding) to deal with the above three problems. TLOE uses the competitive Hebbian rule to construct the topology preserving network on a given manifold. It is capable of: 1) automatical selection of the right number and position of landmarks, 2) adaptive determination of neighborhood sizes for landmarks and 3) online embedding of new observations. Experiments on both synthetic and real-world data sets show its promising results.**

I. Introduction

Dimensionality reduction is to find a meaningful low-dimensional parameterization of high-dimensional observations. In other words, it associates low-dimensional coordinates to data while preserving their similarities in the high-dimensional space. Thus dimensionality reduction methods usually consist of two steps: similarity estimation and structure preserving embedding. In the first step, traditional linear methods such as PCA (Principal Component Analysis) and MDS (Multidimensional Scaling) regard Euclidean distances between data as their similarities. And in the second step, PCA maximizes the variance of data and MDS preserves pairwise Euclidean distances in the low-dimensional space. However, Euclidean distances are not able to estimate the similarities in data sets containing essential nonlinear structures, thus linear methods such as PCA and MDS fail to detect the true degrees of freedom of those data sets [1].

Isomap [1] is the first to introduce geodesic distance to measure the similarities between data in the high dimensional space. It is a globally optimal solution and opens up a new field called “manifold learning”. Although Isomap works well on many data sets, it has three drawbacks that hold up its real-world applications. The first is its computational inefficiency and space complexity. The computational complexity of shortest path distances in k-nearest-neighbor graph is $O(kN^2 \log N)$ and complexity of MDS is $O(N^3)$, where N is the number of data points [2][3]. The space complexity is $O(N^2)$, which will cause out-of-memory errors in big data application. The second is neighborhood determination. Large neighborhood size introduces short-circuit edges into the neighborhood graph, and small size makes the graph become too sparse to approximate geodesic paths accurately [4]. The third is new observations embedding. Isomap has to retrain the whole data set with new samples because the neighborhood graph is updated after the addition of new samples [5]. Many methods have been proposed to deal with those three drawbacks separately: L–Isomap methods to improve the computational efficiency [6][8], adaptive methods to select the local neighborhood sizes [12][13], and out-of-sample methods to embed new observations without retraining the whole data set [5][7]. However, there is still no method to address all the above three problems. Solutions to one problem may even amplify the other two problems. That makes existing manifold learning methods incapable for real-world applications.

This paper proposes a new method TLOE (Topology Learning and Out-of-sample Embedding) to deal with the above three problems at the same time. It constructs topology preserving graph G to select the right number and position of landmarks. Besides, the neighborhood size for each landmark is determined adaptively in G. Moreover, TLOE is able to embed new observations without retraining the whole data set. All calculations in TLOE are based on landmarks only, which reduces much of the computation and space complexity in real-world application.

II. Topology Learning and Out-of-sample Embedding

It is obvious that in real-world applications, we need landmarks to reduce the computational and space complexity. However, existing L–Isomap methods are not able to determine appropriate position and number of landmarks. That will magnify short–circuit errors [8]. Besides, the neighborhood size k is determined by users empirically, which is demanding and unreliable [4]. Furthermore, there may be considerable variance of data density and distribution on a given manifold. Thus each landmark needs to adapt the neighborhood size to the local data density, instead of using the same neighborhood size k as all the other landmarks. In TLOE, we use competitive Hebbian rule to construct network G on the manifold M. It is able to determine the reasonable number and position of landmarks automatically, and determine the neighborhood size of each landmark adaptively. It will prove later in Section 3 that with competitive Hebbian rule, the network G constructed by TLOE is a topology and path preserving network of M.

Although L–Isomap methods reduces computational complexity for Isomap, they do not reduce any space complexity. They still have to calculate the $N \times N$ Euclidean distance matrix D. Based on D they calculate the geodesic distances between landmarks. That will be a huge amount of space complexity, which may cause out-of-memory errors in big data applications. What’s more, large amounts of data are emerging every second in the age of big data. Due to the high computational and space complexity, it is not possible to retrain the whole data set in order to embed new data points. TLOE seeks to embed new observations onto the low-dimensional space without retraining the whole data set. It

Qiang Gan (Email:njucsgq@gmail.com), Furao Shen (corrresponding author, Email:frshen@nju.edu.cn) and Jinxi Zhao (Email:jxzhao@nju.edu.cn) are with the National Key Laboratory for Novel Software Technology, and Department of Computer Science and Technology, Nanjing University, China.

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

uses network G to preserve the topology of the data in the high-dimensional space. Based on the similarities between new observations and landmarks, the low-dimensional coordinates of new observations will be figured out easily.

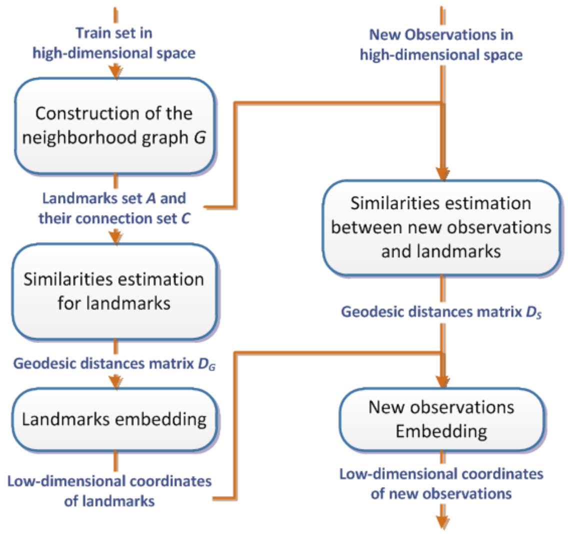


Fig. 1. Flowchart of Topology Learning and Out-of-sample Embedding (TLOE)

The entire work flow of TLOE (Fig 1) is as follows: given a training set distributed on a manifold M, TLOE constructs the neighborhood graph G based on competitive Hebbian rule. The graph G determines the landmarks and their connections. Then TLOE calculates the similarities between landmarks with shortest path algorithm and get the geodesic distances matrix D_G. After that, TLOE embeds the landmarks onto the low-dimensional space with MDS. The embedded landmarks preserve the topology of M in low-dimensional space. When a new observation ξ comes, TLOE estimates the similarities between ξ and landmarks in the high-dimensional space and gets the geodesic distances vector D_s. Afterwards, LMDS (Landmark MDS) [3] is used to embed ξ onto the low-dimensional space. In TLOE, we use geodesic distances to measure the similarities in the high-dimensional space, and we use Euclidean distances to measure the similarities in the low-dimensional space.

A. Construction of the neighborhood graph G

Consider a training set of N data points $\{x_1, x_2, \ldots, x_N\}$ distributed on manifold M from high-dimensional space. We will construct the neighborhood graph G by learning the topology of M. The construction algorithm is enlighted by Self-organizing incremental neural network (SOINN) [9][10]. The true topology of M is hidden behind the distribution of data points. G consists of landmark set A and connection set C. The landmark set A is initialized to contain two landmarks by: $A = \{L_1, L_2\}$,where L_1, L_2 are two randomly selected data points. C is the set of neighborhood relationships between landmarks. It is set to $\emptyset$ with no initial connection between L_1, L_2. Local accumulated number M_{L_i} is used to record the activation times of landmark L_i. We set $M_{L_1} = M_{L_2} = 0$. Their similarity thresholds $T_{L_1} = T_{L_2} = \|L_1 - L_2\|$, which denote the activation scopes for landmarks. Each connection has an age as an outdate degree and we initialize $age_{(L_1,L_2)} = 0$.

When the data point $x_{i(i=1,2,\ldots,N)}$ comes, we search landmark set A for the winner s_1 and second winner s_2 by:

$$s_1 = \arg\min_{x \in A} \|x_i - x\| \tag{1}$$

$$s_2 = \arg\min_{x \in A \setminus \{s_1\}} \|x_i - x\| \tag{2}$$

where $\|\cdot\|$ represents the Euclidean distance in the high-dimensional space.

If the similarities between x_i and first two winners are more than their similarities threshold:

$$\|x_i - s_1\| > T_{s_1} \quad or \quad \|x_i - s_2\| > T_{s_2} \tag{3}$$

that means x_i is a new landmark and we add x_i into A. Then we deal with the next coming data point x_{i+1}.

If condition (3) is not satisfied, x_i is not a new landmark. Then connection establishing and strengthening are conducted based on the competitive Hebbian learning rule. If there is no connection between s_1 and s_2, we create the connection (s_1, s_2) and add it to connection set C. Its age $age_{(s_1,s_2)}$ is set to 0 to represent the connection is newly established or strengthened:

$$C = C \cup (\mathrm{s}_1,\mathrm{s}_2) \quad and \quad age_{(\mathrm{s}_1,\mathrm{s}_2)} = 0 \tag{4}$$

Then we add 1 to the the local accumulated number of signals M_{s_1} and the age of all edges emanating from s_1:

$$age_{(s_1,L_i)} = age_{(s_1,L_i)} + 1 \quad and \quad M_{s_1} = M_{s_1} + 1 \tag{5}$$

that means s_1 has been activated one more time.

To learn the true topology of manifold M, the winner landmarks adapts their locations slowly to the input data by:

$$s_1 = s_1 + \varepsilon(t)\,\|x_i - s_1\| \tag{6}$$

$$s_2 = s_2 + \varepsilon'(t)\,\|x_i - s_2\| \tag{7}$$

where $\varepsilon(t) = \frac{1}{t}$, $\varepsilon'(t) = \frac{1}{100t}$.

After a period of learning, landmarks that are connected at an early stage may be far away. One indication is that their connection age becomes greater than a predefined threshold age age_{max}. We remove its connection from C to update the neighborhood graph G.

The threshold T_{s_1} and T_{s_2} are updated to the largest distance between s_1, s_2 and its neighbors respectively:

$$T_{s_1} = \arg\max_{(x,s_1)\in C} \|x - s_1\| \tag{8}$$

$$T_{s_2} = \arg\max_{(x,s_2)\in C} \|x - s_2\| \tag{9}$$

that means for each landmark, all neighbors are within its activation scope. After that, we deal with the next coming data point x_{i+1}.

In order to denoise in real-world application, we remove the outliers every λ data points are learned. For all landmarks in A, if L_i has only one neighbor and M_{L_i} is less than an adaptive threshold M_{min} , we remove it from A.

After all the data points are learned, we get the network G containing landmark set $A = \{L_{i(i=1,\cdots,n)}\}$ and the connection set C we needed. n is the number of landmarks that is determined automatically.

B. Similarities estimation for landmarks

We apply shortest path algorithm to each landmark $L_i(i = 1, \cdots, n)$ to calculate the geodesic distances between L_i and other landmarks in G. It will prove later in Section 3 that the union of all edges in G constructed by TLOE is a path preserving presentation of M.

At first, we initialize two sets by:

$$S = \{L_i\} \quad and \quad U = A - \{L_i\} \tag{10}$$

where S contains the landmarks whose geodesic distances to L_i has been calculated, while landmarks in U remain to be addressed.

The similarity matrix is denoted by D_G. Obviously, $D_G(i, i) = 0$. For all landmarks in U, if L_i and L_j are neighbors, which means $(L_i, L_j) \in C$, then $D_G(i, j) = \|L_i - L_j\|$; otherwise $D_G(i, j) = \infty$.

We select the ${L_i}'s$ nearest landmark L_j from U, which means $L_j = \arg\min D_G(i, j)$, and add L_j into S. Then we update S and U by:

$$S = S \cup \{L_j\} \quad and \quad U = U - \{L_j\} \tag{11}$$

For each landmark L_k remains in U, if L_j and L_k are neighbors and $D_G(i, j) + \|L_j - L_k\| < D_G(i, k)$, there is a shorter path from L_i to L_k via L_j. Then we update the distance from to L_i to L_k by:

$$D_G(i, k) = D_G(i, j) + \|L_j - L_k\| \tag{12}$$

We repeat the above operations until $S = A, U = \emptyset$. After that, the geodesic distances calculation between L_i and other landmarks in G is finished. We select the next landmark L_{i+1} to go through the same calculation process as L_i. When all landmarks finish their calculation, we get the needed similarity matrix D_G.

C. Landmarks embedding

Based on the principle of similarities preservation, we can get the low-dimensional coordinates of the landmarks. We apply MDS on the similarity matrix D_G to embed the landmarks.

Firstly, we get the squared distance matrix Δ_n and calculate its mean column vector by:

$$\Delta_n(i, j) = D_G(i, j) * D_G(i, j), (i, j = 1, \cdots, n) \tag{13}$$

$$\vec{\delta_\mu} = (\vec{\delta_1} + \vec{\delta_2} + \cdots + \vec{\delta_n})/n, \tag{14}$$

where $\vec{\delta_i}$ denotes ith column of Δ_n.

Then we construct the mean-centering matrix H_n and the mean-centered "inner-product" matrix B_n by:

$$H_n(i, j) = \delta(i, j) - \frac{1}{n} \tag{15}$$

$$B_n = -\frac{1}{2} H_n \Delta_n H_n \tag{16}$$

where $\delta(i, j)$ equals to 1 when $i = j$, and 0 when $i \neq j$.

After that, we compute d largest eigenvalues of B_n together with an orthonormal set of eigenvectors. $\lambda_1, \cdots, \lambda_d$ and $\vec{v_1}, \cdots, \vec{v_d}$ denote for the eigenvalues and corresponding eigenvectors. d is the dimensionality of low-dimensional space.

Finally, the landmarks are embedded by the obtained eigenvalues and eigenvectors:

$$L = \begin{bmatrix} \sqrt{\lambda_1} \cdot \vec{v_1}^T \\ \sqrt{\lambda_2} \cdot \vec{v_2}^T \\ \vdots \\ \sqrt{\lambda_d} \cdot \vec{v_d}^T \end{bmatrix}, \tag{17}$$

columns $\vec{l_1}, \cdots, \vec{l_n}$ of L (each vector has d features) are low-dimensional coordinates of landmarks.

D. Similarities Estimation between New Observations and Landmarks

One advantages of TLOE is the capability to deal with new observations. When the new observation ξ comes, we need to estimate its similarities to all landmarks in the high-dimensional space. The similarity vector is denoted by D_s.

We find the nearest landmark L_α and estimate similarities between ξ and all the other landmarks via L_α:

$$L_\alpha = \arg\min_{x \in A} \|\xi - x\| \tag{18}$$

$$D_s(\xi, L_i) = \|\xi - L_\alpha\| + D_G(\alpha, i) \tag{19}$$

E. New Observations Embedding

At this stage, we have gathered all the information needed to embed ξ onto the low-dimensional space. The embedding principle is similarities preservation based on LMDS [3]. In low-dimensional space, the similarities are indicated by Euclidean distances.

First of all, we compute squared distances between ξ and landmarks:

$$\vec{\delta_\xi}(i) = D_s(\xi, L_i) * D_s(\xi, L_i) \tag{20}$$

After that, the pseudoinverse transpose matrix is computed with eigenvalues and eigenvectors of B_n:

$$L^\# = \begin{bmatrix} \vec{v_1}^T / \sqrt{\lambda_1} \\ \vec{v_2}^T / \sqrt{\lambda_2} \\ \vdots \\ \vec{v_d}^T / \sqrt{\lambda_d} \end{bmatrix} \tag{21}$$

Finally, we embed the new observation ξ by:

$$l_\xi = -\frac{1}{2} L^\# (\vec{\delta_\xi} - \vec{\delta_\mu}) \tag{22}$$

where δ_μ is the mean vector by formula (14).

Now we get the low-dimensional coordinate of the new observation ξ. The embedding process is between ξ and landmarks only without retraining the whole data set.

III. Analysis

In this section, we prove the correctness and efficiency of TLOE. Firstly, we introduce briefly about how TLOE based on competitive Hebbian rule forms perfect topology preserving maps and path preservation representations of manifold. Secondly, we analyze the computational and space complexity of TLOE, compared with Isomap and L–Isomap methods.

A. Competitive Hebbian Rule Learns Topology of Manifolds

The proof in [11] that competitive Hebbian rule learns topology of manifolds can be summarised as follows:

The graph G formed by competitive Hebbian rule is Delaunay triangulation. If the distribution of the nodes in G is dense on the given manifold M, then G is the induced Delaunay triangulation $D^{(M)}$. Thus G forms a perfectly topology preserving map of M.

Besides, if the distribution of the nodes in G is dense on M, the induced Delaunay triangulation $D^{(M)}$ has the additional property that each edge of $D^{(M)}$ lies completely on M. The union of all edges of $D^{(M)}$ forms a path preserving representation of M.

In TLOE, we construct G with competitive Hebbian rule by formula (1),(2) and (4), thus G is Delaunay triangulation. To ensure the distribution of the landmarks of G is dense on M, we add new landmarks if condition (3) is satisfied. Condition (3) is used to meet the Definition 6 in [11]. Therefore, graph G constructed by TLOE is the induced Delaunay triangulation $D^{(M)}$ and forms a perfectly topology preserving map of M (by Theorem 3 in [11]). Since each edge of $D^{(M)}$ lies completely on M, the union of all edges of G forms a path preserving representation of M (by Theorem 4 in [11]).

B. Computational and Space Complexity Analysis

Isomap calculates the $N \times N$ Euclidean distance matrix D first, which takes $O(N^2)$ time. Then it uses Dijkstra algorithm to calculate the $N \times N$ shortest-path distance matrix D_N, which is in $O(kN^2 \log N)$ time. The neighborhood size k is determined by users empirically. In the final step, MDS eigenvalue calculation on the full D_N is in $O(N^3)$ time. The space complexity of Isomap is $O(N^2)$ to store D and D_N.

L–Isomap methods address the inefficiencies of Isomap by designating n landmarks, where $n \ll N$, in the first step. There are a few accepted ways to determine the position of landmarks. Random selection is in $O(1)$ time. Kmeans and Maxmin methods are both in $O(nN)$ time. In the second step, they calculate $D(N \times N)$ just as Isomap and takes $O(N^2)$ time. Thirdly, instead of calculating the $D_N(N \times N)$, they compute the $n \times N$ matrix $D_{n,N}$ of distances from each data points to the landmarks only. The third step takes $O(knN \log N)$ time. The neighborhood size k is still determined by users empirically. Finally, LMDS based on $D_{n,N}$ is in $O(n^2N)$ time. The space complexity of L–Isomap is also $O(N^2)$ to store D.

TLOE constructs the topology and path preserving graph G by competitive Hebbian rule. The construction step determines the right number and position of landmarks, and establishes neighborhood relations between landmarks. It takes $O(nN)$ time. In the second step, it calculates the $n \times n$ shortest-path distance matrix D_n between the landmarks only. That step takes $O(mn^2 \log n)$ time, where m is the average neighborhood size between landmarks in G. There is no k to be determined by users in TLOE. In the third step, MDS on D_n is in $O(n^3)$ time. To deal with N new observations, LMDS takes $O(n^2N)$ time. The space complexity of TLOE is only $O(n^2)$ to store D_n.

Based on the above analysis, if the number of landmarks n is much smaller that the number of data points N, TLOE reduces much of the computational and space complexity.

IV. Experiments

In this section, we present the results of TLOE on three data sets compared with three L–Isomap methods (with landmarks by random, maxmin and kmeans separately). The three data sets are Swiss roll, Swiss roll with noise and Mnist. On Swiss roll and Swiss roll with noise, we visualize the embedded 2D coordinates to compare the four methods intuitively. On Mnist, classification tasks are executed to compare those methods quantitatively. Note that L–Isomap methods are not out-of-sample methods, thus they are not capable of dealing with test data in classification tasks. All experiments are implemented in MATLAB R2010a on a PC with 3GB memory.

A. Visualization on Swiss Roll

1) Methodology: We designed three sets of experiments on this data set to compare the above methods comprehensively. In the first set of experiments, the data set contains only 8000 data points of Swiss Roll. After that, we conduct the second set of experiments by expanding the size of data set to 15000 and visualize the results. In the third set of experiments, we use TLOE to embed 5000 new data points. The embedding is based on the landmarks and their neighborhood relationships from the second set of experiments. It aims to show the capability of TLOE to deal with new observations. L–Isomap methods are unable to address new observations, since they need to add them to the whole data set and rebuild the $k$nn graph. In the following experiments, if not stated explicitly, the number of neighbor k is set to 7 in L–Isomap methods. To be fair, the number of landmarks is set the same in all the four methods.

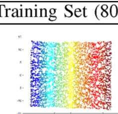
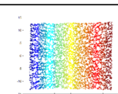
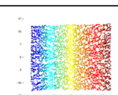
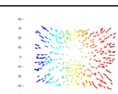
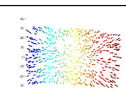
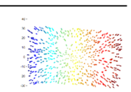
Method	Training Set (8000)	Training Set (15000)	Test Set (5000)
Random L–Isomap		out-of-memory	—
Maxmin L–Isomap		out-of-memory	—
Kmeans L–Isomap		out-of-memory	—
TLOE			

TABLE I. Visualization results on Swiss roll

2) Results: The first column of Table 1 reveals that in the small-scale noise-free data set, TLOE still preserves the basic topology of the manifold. Consider that TLOE utilizes the landmarks and their connections only, while L–Isomap methods build $k$nn graph on the whole data set. The loss of precision by TLOE is inevitable and acceptable.

When the data set contains 15000 data points, L–Isomap methods return out-of-memory errors due to the $O(N^2)$ space to store D. However, TLOE works properly and returns a satisfying result.

Based on the result of the second set of experiments, we add 5000 new data points to be embedded. L–Isomap methods is not capable of dealing with new data points. While TLOE produces an excellent result by embedding new data points.

There is a noteworthy phenomenon that the embedded data points by TLOE consist of small clusters. They distributed sparsely on low-dimensional space. One plausible explanation is that all landmarks form a Voronoi partition in the high-dimensional space. For all data points in the certain Voronoi cell V_i, their relative location information is ignored. The only difference between them is their distance difference to L_i. Therefore, all data points in V_i will be mapped around the embedding result of L_i and form a cluster. The number of clusters is exactly the number of landmarks.

B. Visualization on Swiss Roll with Noise

1) Methodology: It is known that Isomap algorithm is vulnerable to short–circuit errors if noise in the data moves the points slightly off the manifold [4]. We want to test whether L–Isomap methods and TLOE will be affected by noise. We add gaussian noise to the Swiss roll and conduct the three same sets of experiments as last subsection.

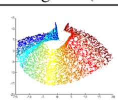
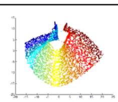
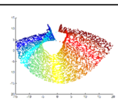
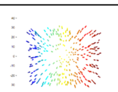
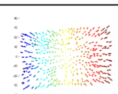
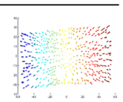
Method	Training Set (8000)	Training Set (15000)	Test Set (5000)
Random L–Isomap		out-of-memory	—
Maxmin L–Isomap		out-of-memory	—
Kmeans L–Isomap		out-of-memory	—
TLOE			

TABLE II. VISUALIZATION RESULTS ON SWISS ROLL WITH NOISE

2) Results: We can see from Table 2 that in small-scale data set with noise, L–Isomap methods are severely affected. Noise points connect distant data points wrongly and cause unacceptable short–circuit errors. When the data set becomes large, L–Isomap methods return out-of-memory errors. Furthermore, L–Isomap is unable to address new data points. On the contrary, TLOE outperforms its counterparts in reducing the short–circuit errors in the noise environment, regardless of the scale of the data set. For new data points, TLOE still preserves the basic topology of the manifold without retraining the whole data set.

C. Classification on Mnist

1) Methodology: The Mnist data set is formed by handwritten digits with 400 dimensions. We map them onto the 5-dimensional space and then execute the classification tasks. In the first set of experiments, the data set contains 5000 data points. L–Isomap methods and TLOE embed them separately to get their 5-dimensional coordinates. We run 10-fold cross validation 100 times on each of the results and get the average misclassification rates. In the second set of experiments, we expand the data set to contain 10000 data points and conduct the same experiments as the first set of experiments. In the third set of experiments, the training set is the embedded results by TLOE in last set of experiments, and the test set is the embedded results of 5000 new data points by TLOE. We use the 1-nearest-neighbor classifier and conduct the classification task. The experiments also repeat 100 times and we get the average misclassification rate.

TABLE III. AVERAGE MISCLASSIFICATION RATE ON MNIST

Method	Training Set (5000)	Training Set (10000)	Test Set (5000)
Random L–Isomap	0.1676	out-of-memory	—
Maxmin L–Isomap	0.1650	out-of-memory	—
Kmeans L–Isomap	0.1633	out-of-memory	—
TLOE	0.1596	0.1419	0.1924

2) Results: Table 3 reveals that in small-scale data set, TLOE outperforms L–Isomap methods in classification tasks. When dealing with large-scale data set, L–Isomap methods return out-of-memory errors but TLOE returns a satisfying classification result. Most importantly, TLOE is able to embed new observations onto a low-dimensional space and classify the results with an acceptable accuracy.

There is a notable phenomenon that misclassification rate becomes lower as the training set becomes large. The reason is that on a given manifold M, more training data points help to reveal the true topology and structure of M. Therefore, the topology preserving graph G is able to determine the landmarks and their connections more precisely. The results also reveals that the classification accuracy on test set is lower than on training set. One potential cause is that new observations may be off the learned manifold M slightly, but the selected landmarks did not adapt to the topology change. Solution to that problem will be our future work.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a new manifold learning method. It uses competitive Hebbian rule to address the landmarks selection and neighborhood determination problems. It reduces much of the computational and space complexity compared with Isomap and L–Isomap methods. Furthermore, it estimates the similarities between new observations and landmarks to realize out-of-sample embedding for new observations.

Our future work will be the incremental learning problem. The topology of the manifold may change as new observations come. However, current out-of-sample methods are unable to adapt to that change. We want to deal with that problem and extend manifold learning to obtain incremental learning ability.

ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation of China under Grant Nos. (61373001, 61375064) and Jiangsu NSF grant (BK20131279).

REFERENCES

[1] Tenenbaum J B, De Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5500): 2319-2323.

[2] Silva V D, Tenenbaum J B. Global versus local methods in nonlinear dimensionality reduction[C]//Advances in neural information processing systems. 2002: 705-712.

[3] De Silva V, Tenenbaum J B. Sparse multidimensional scaling using landmark points[R]. Technical report, Stanford University, 2004.

[4] Balasubramanian M, Schwartz E L. The isomap algorithm and topological stability[J]. Science, 2002, 295(5552): 7-7.

[5] Kim K, Lee D. Inductive manifold learning using structured support vector machine[J]. Pattern Recognition, 2014, 47(1): 470-479.

[6] Shi L, He P, Liu E. An incremental nonlinear dimensionality reduction algorithm based on ISOMAP[M]//AI 2005: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2005: 892-895.

[7] Raducanu B, Dornaika F. Embedding new observations via sparse-coding for non-linear manifold learning[J]. Pattern Recognition, 2014, 47(1): 480-492.

[8] Qiang G, Furao S, Jinxi Z. An Extended Isomap for Manifold Topology Learning with SOINN Landmarks[C]//International Conference on Pattern Recognition, 2014.

[9] Furao S, Hasegawa O. An incremental network for on-line unsupervised classification and topology learning[J]. Neural Networks, 2006, 19(1): 90-106.

[10] Furao S, Ogura T, Hasegawa O. An enhanced self-organizing incremental neural network for online unsupervised learning[J]. Neural Networks, 2007, 20(8): 893-903.

[11] Martinetz T, Schulten K. Topology representing networks[J]. Neural Networks, 1994, 7(3): 507-522.

[12] Zhang Z, Wang J, Zha H. Adaptive manifold learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(2): 253-265.

[13] Mekuz N, Tsotsos J K. Parameterless Isomap with adaptive neighborhood selection[M]//Pattern Recognition. Springer Berlin Heidelberg, 2006: 364-373.

[14] Geng X, Zhan D, Zhou Z. Supervised nonlinear dimensionality reduction for visualization and classification[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2005, 35(6):1098 - 1107.