2016 IEEE 28th International Conference on Tools with Artificial Intelligence

A Fast Manifold Learning Algorithm for Dimensionality Reduction

Yu Liang[1], Furao Shen[1,*], Jinxi Zhao[1], Yi Yang[1]
[1]National Key Laboratory for Novel Software Technology, and Department of Computer Science and Technology, Nanjing University, China
Email: mg1533023@smail.nju.edu.cn, frshen@nju.edu.cn, jxzhao@nju.edu.cn, yangyi868@gmail.com

***Abstract*—This paper proposes a new manifold learning method called "Soinnmanifold". Traditional manifold learning method needs a lot of computation and appropriate priori parameters. This has somewhat restricted the domains in which manifold learning can potentially be applied. However, with the high-dimensional inputs, our method can generate a low-dimensional manifold in the high-dimensional space and determine the intrinsic dimension automatically. Then we will use this manifold to do dimensionality reduction quickly. Experiments demonstrate that our method can get promising results with less time and memory.**

***Index Terms*—Manifold learning, Intrinsic Dimension, Dimensionality Reduction**

I. Introduction

With the coming of the era "Big Data", the growing dimension of the data has led to a lot of problems such as curse of dimensionality. Therefore, how to represent the high-dimensional data in a lower-dimensional space becomes more and more important. As a consequence, dimensionality reduction has attracted many researchers in recent years.

Up to now, many classical algorithms have been proposed. For dimensionality reduction, Principal Component Analysis (PCA) can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized [1]. Besides, MDS (Multidimensional Scaling) is to minimize the difference between corresponding distances in high and mapped low dimensional space [2]. PCA and MDS are both linear methods, they can not work well on nonlinear data. Hence, kernel method was proposed. Kernel method is based on the idea that linearly inseparable problem can be linearly separable in high dimensional space [3]. What's more, scientists have proposed a new method which is called manifold learning. Manifold learning holds the idea that the original high dimensional data may lie on a low dimensional manifold.

Isomap [4] is the first manifold learning method which uses geodesic distance to measure the similarity between data in high dimensional space [5]. Locally Linear Embedding (LLE) is another powerful algorithm in manifold learning. LLE supposes that each point can be represented by a linear combination of its several neighbor points. It computes the the local combination weight matrix W in the original space firstly. Then it minimizes the reconstructed error in the new space, in which each point is reconstructed by W. [6].

However, LLE adopts k-nearest neighbors method to generate the manifold. Choosing a very small neighborhood is not a satisfactory solution, which may fragment the manifold into a large number of disconnected regions [7].Choosing a too large neighborhood may cause "short-circuit". Besides, target dimension also has an effect on the performance of LLE. A too small dimension can not preserve the structure and relationships in the high-dimensional space after data are mapped into a low-dimensional space.

In this paper, we propose a novel method called "Soinnmanifold". It can generate the manifold with the given input automatically. With the manifold, we can find the intrinsic dimension which is implicit in the input data without any parameters. For the reason that the manifold consists of a few landmarks, we can do dimensionality reduction on the manifold, which can save enormous amounts of time and memory. Afterwards, input data can be represented in the low-dimension space which is the result we want to get. Compared with other algorithms such as PCA, LLE, Isomap, experiments of Soinnmanifold demonstrate its promising results on swiss-roll data set, swisshole data set and Adult data set.

II. Dimensionality Reduction

Our method is composed of three components: manifold generation, intrinsic dimension estimation and dimensionality reduction. We defer the these components and implementation details to subsequent sections.

A. Manifold Generation

We adopt a self-organizing neural network (SOINN) algorithm to generate the manifold[8] [9]. SOINN can generate the appropriate manifold without the parameter k, which is the number of the neighbors.

1) Initialization:

- Initialize the node set A to contain two nodes, x_1 and x_2, which are chosen randomly from all the nodes.
- Initialize the connection set C, C $\subset$ A $\times$ A, to the empty set. The output manifold is composed of A and C.
- Initialize the local accumulated error $E_{x_1} = 0$ and $E_{x_2} = 0$.
- Initialize the local accumulated number of nodes $M_{x_1} = 0$ and $M_{x_2} = 0$.

Corresponding author is Furao Shen.

2375-0197/16 $31.00 © 2016 IEEE
DOI 10.1109/ICTAI.2016.149

- Initialize the similarity threshold T_{s_1} and T_{s_2} to an infinite number.

2) Input and Find Winner: Input a new node x_i. Search the nearest node s_1 and the second nearest node s_2 by

$$s_1 = \underset{s_c \in A}{\operatorname{argmin}} \|x_i - s_c\| \tag{1}$$

$$s_2 = \underset{s_c \in A \setminus \{s_1\}}{\operatorname{argmin}} \|x_i - s_c\| \tag{2}$$

3) Nodes Update: If the distance between x_i and s_1 are greater than T_{s_1} and the distance between x_i and s_2 are greater than T_{s_2}, it means that the the node x_i is far away from all the nodes in A, hence we add the new node x_i to A. Then go to step 2 to process next node.

If a connection between s_1 and s_2 does not exist already, create it and set the age of the connection $age(s_1, s_2)$ to 0, then add it to connection set C.

For all edges emanating from s_1:

$$age(s_1, s_i) = age(s_1, s_i) + 1 \tag{3}$$

In which s_i is next to s_1. If $age(s_1, s_i)$ is too large, it means that the edge between s_i and s_1 is created long time ago, hence we remove edges with an age greater than a predefined threshold age age_{dead} and delete it from C.

For the winner s_1, add the Euclidian distance between the node x_i and the winner s_1 to local accumulated error E_{s_1}, add 1 to the local accumulated number of signals M_{s_1}. Then, adapt the s_1 and its direct topological neighbors to the node x_i by a certain fraction ε.

4) Thresholds Update: Adjust the thresholds T_{s_1}/T_{s_2} to the maximum distance between s_1/s_1 and its neighbors.

5) Noise Elimination: If the number of input generated so far is an integer multiple of parameter λ, for all s_i in A, if s_i has no or only one neighbor and M_{s_1} is less than an predefined threshold, remove s_i from the node set A.

6) Manifold Generation: After the process above, we can get the manifold consisting of the node set A and the connection set C. However, sometimes the number of edges may be too small, which would fragment the manifold into some disconnected components. Therefore, for all nodes in A, if the number of neighbors k_i of the node s_i is less than the average number of neighbors k, then add the connection between the node s_i and its nearest node which are not adjacent to it until k_i becomes k.

Then, the number of the neighbors will be appropriate, which means that the manifold will not be fragmented into a large number of disconnected regions and will not cause "short-circuit" problem.

B. Intrinsic Dimension Estimation

For the reason that the input data lies on a low-dimensional manifold and we have got the appropriate manifold after the above process, we can estimate the intrinsic dimension of the input data. The manifold is composed of a lot of local linear space, therefore we can estimate the dimension of the local linear space, then calculate the average value of the dimension, which is the intrinsic dimension of the input data. We adopt the locally principal component analysis algorithm to do the dimension estimation.

Algorithm 1 Dimension Estimation

1.Input:the node set A and the connection set C.
2.For all the node $s_i \in A$, find the node set N, which consists of the neighbors of s_i, put the s_i into N, suppose the size of N be M.
3.Compute the mean $\overline{s} = \frac{\sum_{i=1}^{M} n_i}{M}$.
4.Compute the covariance matrix $X = Cov(s) = \frac{1}{N}\sum_{i=1}^{M}(x_i - \overline{x})(x_i - \overline{x})^T$ and find the eigenvalues of X $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$.
5.k_i=$argmin_k \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{d}\lambda_i} \geq 0.9$,$k_i$ is the intrinsic dimension around the node s_i
6.$k = \frac{\sum_{s_i \in A} k_i}{|A|}$. k is the intrinsic dimension of the input data.

Because of the small number of the neighbors, calculating the local dimension of the node s_i can be run in O(1) time. Therefore all the algorithm can be run in O(N) time, that is acceptable.

C. Dimensionality Reduction

Now, we have the manifold of the input data and the target dimension, hence we can do the dimensionality reduction. For the reason that the input data lies on the manifold, we can do dimensionality reduction on the manifold made up of the node set A and the connection C. Because the number of nodes is much less than the original input, we can finish the dimensionality reduction with a high speed.

In this paper, we adopt Locally Linear Embedding(LLE) algorithm to do dimensionality reduction. LLE calculates the linear coefficients that reconstruct each data point from its neighbors [6]. Reconstruction errors are measured by the cost function:

$$\varepsilon(W) = \sum_i |s_i - W_{ij} s_j|^2 \tag{4}$$

If the s_j is not adjacent to s_i, $W_{ij} = 0$. Calculate all the weight w_{ij} by minimizing the Eq 4. Then compute the low-dimensional embedding node that can be best reconstructed by minimizing the cost function:

$$\phi(M) = \sum_i |m_i - W_{ij} m_j|^2 \tag{5}$$

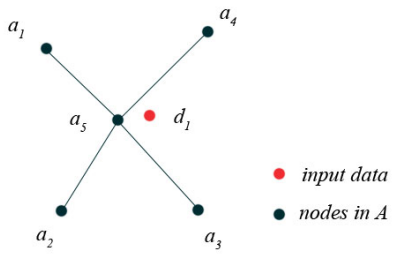

Fig. 1. In this figure, the nearest neighbor to d_1 is a_5, hence we calculate the weights that reconstruct d_1 from a_1,a_2,a_3 and a_4. Then keep the weights and reconstruct d_1 in the low-dimensional space.

Then, we can map the high-dimensional node s_i into the low-dimensional node m_i.

However, what we want to get is the data in the low-dimensional space mapped from the input data x_i. Hence, for the data x_i, find its nearest neighbor s_j in node set A. As shown in Fig 1. Like the process above, we calculate the reconstruction weight of the s_j's neighbors and then keep the weight and do reconstruction in the low-dimensional space by the m_j's neighbors. In this way, we can get the low-dimensional data mapped from the original input space one-to-one.

III. Experiment

In this section, we will show the results of our method compared with PCA, Isomap and LLE.

A. Artificial Data Set

First of all, we do dimensionality reduction on the artificial data called "swishroll", "swishhole", which are widely used in manifold learning. As we can see from Fig 2(a), the input data is made up of 5000 three-dimensional points. These 5000 points lie on a low-dimensional manifold. We would do dimensionality on these points with different algorithms.

For PCA, we set the target dimension to 2. For LLE, we set the target dimension to 2 and set the number of neighbors to 15, more neighbors almost have no influence on the results but too many neighbors would cause short-circuit. For our method, we can detect the target dimension is 2 and generate the manifold automatically.

As shown in the Fig 2(b), The input data lies on or near the low-dimensional manifold generated by our method. What's more, the manifold generated by our method don't have short-circuit and have not been fragmented the manifold into some disconnected regions. Which means that manifold generated by our method is appropriate.

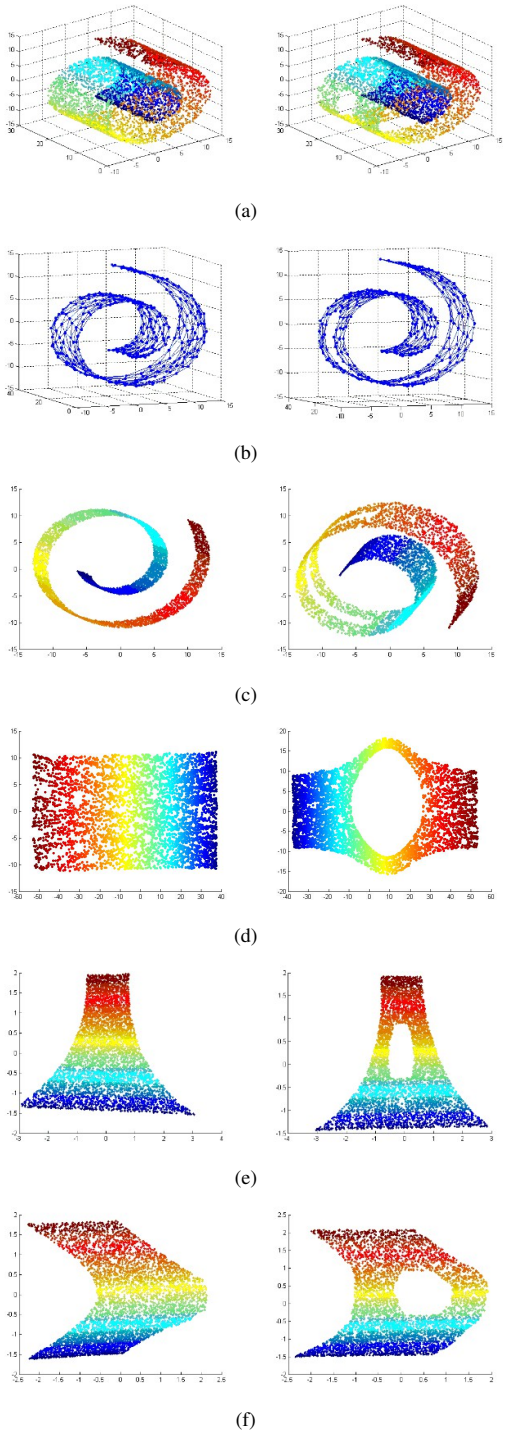

Fig. 2. Figure (a) shows that the distribution of the artificial data, figure (b) shows the manifold generated by our method, figure (c) shows the result of the PCA, figure (d) shows the result of the isomap, figure (e) shows the result of the LLE, figure (f) shows the result of our method.

As we can see from the Fig 2, PCA gets the worst result for the reason that PCA is a linear algorithm, it cannot handle the nonlinear dimensionality reduction problem. However, it runs

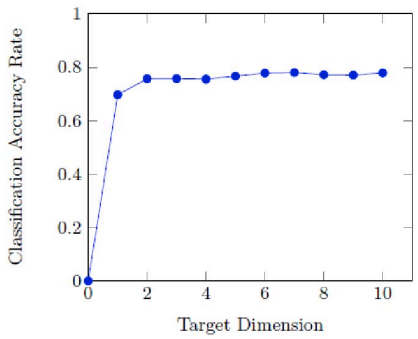

Fig. 3. The relationship between the target dimension and classification accuracy rate

quickly, only takes 2.49×10^{-3} seconds, that is the advantage of linear dimensionality reduction. Isomap achieves best result on "swishroll", but it takes 722.74 seconds to run, which is unacceptable in practical application. Besides, it works not well on "swishhole", the hole on the manifold would become large. Compared to the original data, the result of LLE has a little difference from the original data, but that is acceptable. Our method gets a similar result with LLE but in a higher speed. LLE takes 1.96 seconds and our method only takes 0.84 seconds.

B. Real Data Set

In this experiment, we do the dimensionality reduction on the real data set called Adult. Adult is a classification data set, we randomly select 10000 points from its training set and 5000 points from its testing set to do the test. For LLE and Isomap, we set the number of neighbors to 15 to generate the mannifold, more neighbors almost have no influence on the classification accuracy rate,in addition, too many neighbors will reduce the classification accuracy rate.

We use the train set to train the parameters and then do the dimensionality reduction on the training set and testing set. However, for LLE and Isomap, we can not do like this, LLE and Isomap can only do the dimensionality reduction on the training set and testing set at the same time, which is unreasonable in the practical life. That is also one of their major drawbacks. When we get the low-dimensional data, we use the nearest neighbor algorithm to judge the classification accuracy rate of the testing set.

The estimated intrinsic dimension calculated by our method is 6. As shown in the Fig 3, the target dimension 6 is appropriate for the reason that more dimension will only gets a little increase in the classification accuracy rate.

As shown in the Table 1, these three methods have similar result but with different speed. Isomap needs to calculate the geodesic distance between data in high dimensional space, which takes a lot of time. LLE spends less time than Isomap, but it still takes more time than ours. LLE should compute matrix eigenvalues and eigenvectors with the size $15000\times$ 15000. In our method, the manifold is made up of 540 nodes. We only need to calculate matrix eigenvalues and eigenvectors with the size 540×540, that will save a lot of time.

Method	Classification Accuracy Rate	Running time
Isomap	0.7836	21724.59s
LLE	0.7753	172.59s
Our Method	0.7804	6.41s

TABLE I
THE RUNNING TIME AND CLASSIFICATION ACCURACY RATE

When we run LLE or Isomap on more points such as 40000 points from the data set, the matlab will be out of memory. However, with our method, this error will not occur. Because our method takes up less memory.

IV. CONCLUSION

In this paper, we propose a novel manifold learning algorithm that can do dimensionality reduction. Compared to other algorithm, the outstanding advantage of our algorithm is that it needs less memory and running time, which is helpful to process the big data. Moreover, our algorithm can decide the target dimension automatically. Besides we propose a new method for the manifold generation which is an important step in manifold learning.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 61375064, 61373001 and 61321491), Foundation of Jiangsu NSF (Grant No. BK20131279).

REFERENCES

[1] Bishop C M. Pattern Recognition and Machine Learning (Information Science and Statistics)[M]. Springer-Verlag New York, Inc. 2006.
[2] Leeuw J D, Heiser W. 13 Theory of multidimensional scaling[J]. Handbook of Statistics, 2015, 2(82):285-316.
[3] Debruyne M, Verdonck T. Robust kernel principal component analysis and classification[J]. Advances in Data Analysis & Classification, 2015, 4(2-3):151-167.
[4] Tenenbaum J B, De Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5500): 2319-2323.
[5] Gan Q, Shen F, Zhao J. An Extended Isomap for Manifold Topology Learning with SOINN Landmarks[C]// International Conference on Pattern Recognition. IEEE Computer Society, 2014:1579-1584.
[6] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding.[J]. Science, 2000, 290(5500):2323-6.
[7] Balasubramanian M, Schwartz E L. The isomap algorithm and topological stability.[J]. Science, 2002, 295(5552):7-7.
[8] Furao S, Hasegawa O. An incremental network for on-line unsupervised classification and topology learning[J]. Neural Networks, 2006, 19(1): 90-106.
[9] 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.