

Adaptive Neighborhood Graph for LTSA Learning Algorithm without Free-Parameter

Xianlin Zou

College of Computer, Chongqing University
College of Computer, Jiaying University
Meizhou, Guangdong, China

Qingsheng Zhu

College of Computer, Chongqing University
Chongqing, China

ABSTRACT

Local Tangent Space Alignment (LTSA) algorithm is a classic local nonlinear manifold learning algorithm based on the information about local neighborhood space, i.e., local tangent space with respect to each point in dataset, which aims at finding the low-dimension intrinsic structure lie in high dimensional data space for the purpose of dimensionality reduction. In this paper, we present a novel learning algorithm, called 3N-LTSA which needs no free parameter in contrast to LTSA by using an adaptive nearest neighborhood graph. Experimental results show that 3N-LTSA algorithm without free parameter performs more practical and simple algorithm than LTSA.

General Terms

Pattern Recognition; Dimensionality Reduction;

Keywords

Natural Nearest Neighbor(3N); Adaptive Neighborhood Graph; Local Tangent Space Alignment (LTSA); Free Parameter Learning; Unsupervised learning.

1. INTRODUCTION

Recently years, many efficient manifold learning algorithms with respect to dimensionality reduction have been received widely attentions for discovering the low dimension intrinsic structure hidden in high dimensional input space and trying to preserve some invariant properties as accurately as possible between the low and high dimensional spaces, these nonlinear learning algorithms fall into two class, one is of global and the other is of local. The global learning algorithms include Isomap [1,2], C-Isomap[3] and L-Isomap[3]. The local learning algorithms include LLE[4,5], Lapacian Eigenmap(LE)[6], LTSA[7], LLTSA[8], NPE[9], SNE[10], LPP [11], RML[12], etc.. Basically, almost all of nonlinear dimensionality reduction algorithms usually concerns a foundational concept of neighborhood, because it is of central importance not only in studies of bijective map between high and low dimensional space, due to every point in low dimension embedding space has a neighborhood homeomorphic to an open set of high dimensional real space from viewpoint of topology, but also in the analysis of algorithm's robustness related to the problem of topological stability [13,14]. Indeed, all learning algorithm mentioned above, except SNE, are closely related to the information about the representation of local neighborhood structure, i.e., the choice of nearest neighbors that may be used naturally to results in a corresponding neighborhood in each data points. The first step involved in the procedures of manifold learning algorithms is always to extract and construct an efficient representation about neighborhood in high-dimensional input space that can be viewed as an infrastructure corresponding to data space, and then describing certain geometric or algebraic

properties upon this local structure, for example, LTSA [7] and LLTSA [8] algorithms require information about the local tangent space based on every neighborhood in each data point.

However, in the case of learning intrinsic structure, neighborhood in each data point used in various state-of-the-art approaches are determined by the common used concept of k-nn or ϵ -nn neighbors, but the k or ϵ appears in a form of parameter within algorithms which must be specified by user. In some toolboxes related to dimensionality reduction, the value of parameter k is selected to equal a fixed number with respect to variant testing data set. But as suggested in [7] that k should be chosen to match the sampling density, noise level and the curvature at each data points so as to extract an accurate local representations, and that it's worthy of considering variable number of neighbors that are adaptively chosen at each data point. In [15,16], an adaptive neighborhood graph is proposed for learning the infrastructure of dataset which can be easily applied to the LLE and Isomap algorithms, and resulted in two novel and no free-parameter algorithm: 3N-LLE and 3N-Isomap. The key ideal is motivated from a comprehensive conception of natural nearest neighbor selection that is irrelevant to the parameter k. Following this routine, we will apply this adaptive mechanism of choosing natural nearest neighbor to the local tangent space alignment (LTSA) algorithm to handle the problem of nonlinear dimensionality reduction. Solutions presented by original LTSA can also be interpreted as an invariant subspaces spanned by true low-dimensional representations of the observations [17], which aims at the performance analysis related to the worst-case upper bound on the angle between the estimated linear invariant subspace and the true linear invariant subspace. For the problem of face recognition, two variant versions of LTSA-based algorithms are presented recently, one is the linear local tangent space alignment (LLTSA) [8], and the other is the orthogonal discriminant linear local tangent space alignment (ODLLTSA) [18].

In this paper, we present a novel LTSA-based algorithm, called 3N-LTSA, which depends on the concept of natural nearest neighbor (3N) instead of k-nn neighbors used in original LTSA. 3N-LTSA is also an adaptive and unsupervised learning algorithm, like 3N-LLE, that does not need to specify free parameter k . It implies that the use of 3N-LTSA for the dimensionality reduction does not need any priori information about the intrinsic structure. Experimental results show that 3N-LTSA algorithm without free parameter performs more practical and simple algorithm than LTSA.

2. ADAPTIVE NEIGHBORHOOD GRAPH

2.1 Natural Nearest Neighbor

Natural nearest neighbor [15, 16] can be viewed as an

extended version to the k -nn neighbor, but it has a significant implication with respect to the true distribution of data in real world. For example, even under the assumption of uniformly distribution, there affirmatively exist a few data objects may be taken into consideration as outliers which are far from the main data objects from the distance and statistic perspectives. The key idea is originated from the real world observations that the neighbors should be accepted each other, similar to the “friendship” relations between individuals, naturally, some person have more friends whereas some person have few friends, the number of one’s friends is determined by the number of how many people are taken him or her as a friends. For general data objects, an object y is one of the neighbors of object x if and only if object x is considered as a neighbor of object y . The more the objects like object y , the more the neighbors of x should have. In particular, data points lying in sparse region should have small number of neighbors, whereas data points lying in dense region should have large number of neighbors. The relationship between neighbors should not only represent the information of the distribution of data objects, but also reveal certain mechanism of generating data, such as Poisson random process.

Finding natural nearest neighbor (3N) can be formalized as following formula, an indicator that defines a possible and compact super-bound of k [15,16]:

$$\text{sup}_k \equiv \sup_{r \in \mathbb{N}} \{r \mid (\exists r)(r \in \mathbb{N} \wedge ((\forall x)((x \in S) \rightarrow (\exists y)(y \in S \wedge y \neq x \wedge x \in NN_r(y))))))\} \quad (1)$$

where $NN_r(y)$ denotes the r -th nearest neighborhood in sense of k -nn, \mathbb{N} the set of non-zero nature numbers, and S the data set. Clearly, the supremum in the right hand of formula (1) does exist, and the fact that the supremum can be taken automatically with respect to all possible r is important because one does not need to have any priori information on r according to a process of searching r -nn in step by step way (Table 1) for all data points. In fact, sup_k indicates a situation in which all data objects within data set may be in a well state of connectivity, so we call sup_k as an indicator of saturation connection. For the sake of description simplicity, we also call $NN_r(y)$ as r -nearest neighbor path (r -NNP) of point y . Algorithm 1 provides a process of calculating the number of neighbors for every data points that conforms to the implications given in (1).

Table 1. Finding 3N and constructing 3NG or SNG for a data set S

Algorithm 1: Definition of 3NG or SNG. Input data set S . Output the indicator of saturation connectivity, the number of neighbors at each point and the neighbors within corresponding neighborhood.

1. $r=1$; for all $i \in S$, $nb(i)=0$, $ratio_nb(i)=0$, $NN_r(i)=\emptyset$.
2. For every point $i \in S$, calculates the r -th nearest neighbor of i : $nn_r(i)$; $NN_r(i) = NN_{r-1}(i) \cup \{nn_r(i)\}$.
3. For every point $i \in S$, counts the number of i occurred in all $NN_r(j)$: $nb(i)$, ($j=1, \dots, N$); if there exist some $nb(i) = 0$ then $r=r+1$ and goto step 2;
4. $\text{sup}_k = r$;
5. For all i , output: $nb(i)$, $ratio_nb(i)=nb(i)/(N \times \text{sup}_k)$, $NN_{nb(i)}(i)$ or $NN_{\text{sup}_k}(i)$.
6. Define 3NG: connecting each point i to its $nb(i)$ nearest neighbors for all data points or SNG: connecting each point

to the sup_k or multi- sup_k nearest neighbors if the connectivity of graph is not satisfied the requirements.

Computing sup_k implies a new strategy of how to automatically finding the value of k , i.e. the processes of searching k -nn at each point should be completed when points such as outliers which keep away from the main data set at least belong to one k -nearest neighborhood. The value of sup_k is considerable stable (Fig.1) for Swiss roll data with different size. In fact, sup_k is a constant for a specialized data set and equals the mean value of all $nb(i)$. It shows that the value of sup_k presents a relatively invariant quantity for the data generating model.

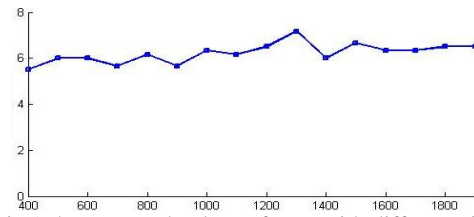


Fig.1 the averaged values of sup_k with different size for Swiss roll data in which six group’s data at each size N are sampled randomly.

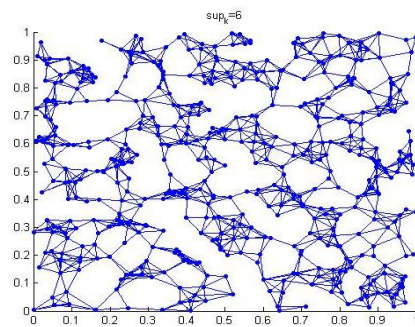


Fig.2 3NG with 500 random sampled points generated by Matlab function $\text{rand}(2,500)$, the connected indicator $\text{sup}_k = 6$.

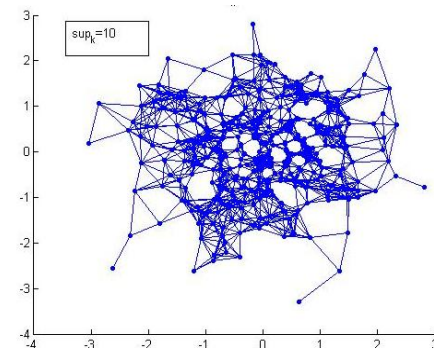


Fig.3 3NG with 400 random sampled points generated from Matlab function $\text{randn}(2,400)$, in which the connectivity indicator is of 10.

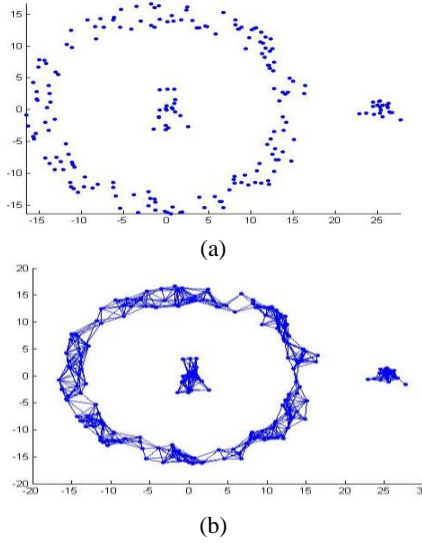


Fig.4 (a) Artificial data set, consist of 200 points include one circular plus two gaussian blobs, one inside, one outside (Jianbo Shi, 1997). (b) 3NG corresponding to the data (right), in which $sup_k=9$.

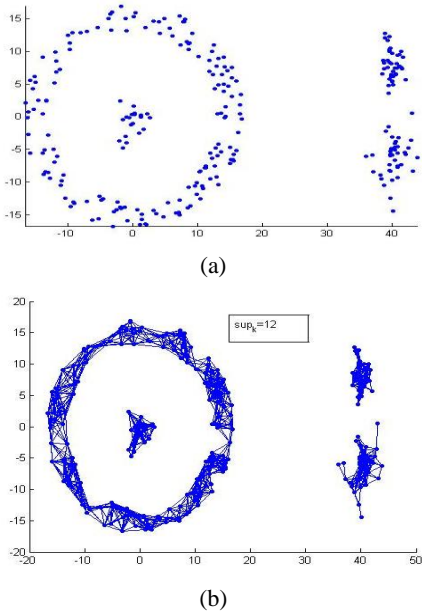


Fig.5 (a) Artificial data set, consist of 260 points include one circular plus three gaussian blobs, one inside, two outside (Jianbo Shi, 1997). (b) 3NG corresponding to the data in (a), in which $sup_k=12$.

2.2 Constructing Adaptive Neighborhood Graph

Two main ways may be taken into consideration to construct the nearest neighbor graph according to the amount of observations. One way is to use a same number of neighbors for all points in the graph for the requirement of connectivity when the amount of sample data is too small, we dub this nearest neighbor graph the saturation nearest neighbor graph (SNG). Another way is to use a variant number at each point which follows from the new concept of 3N and induces a natural nearest neighbor graph (3NG) corresponding to the distribution of a sampled data. Given data set, the procedure

of constructing an adaptive neighborhood graph is very simple. As described in algorithm (Table 1), there are two kinds of infrastructure representations relevant to any given data set, one is 3NG which can be constructed by connecting each point i to its $nb(i)$ nearest neighbors (Fig.2, Fig.3, Fig.4, Fig.5), the other is SNG which can be comprised by connecting each point i to its sup_k nearest neighbors, in such case, all points have the same number of neighbors similar to $k-nn$ graph but k is of the value of sup_k . For two standard distributions, one is of uniformly distribution (Fig.2), one is of normal distribution (Fig.3) and for the two mixed distributions (Fig.4, Fig.5), natural nearest neighbor graphs (3NG) can reveal the true information about the densities. In particular, Fig.4 (right) and Fig.5 (right) preserve the best clustering structures related to the raw data used in testing the efficiency of promising approaches, such as spectral graph clustering and normalized cuts for image segmentation ([19,20]). It shows that 3NG may be able to use to improving the performance of clustering methods.

3. ALGORITHM

3.1 Manifold Learning via Local Tangent Space Alignment (LTSA)

Consider a data set $X=[x_1, \dots, x_N]$ sampled from M^d which is an underlying nonlinear low-dimension manifold of dimension d . Furthermore, suppose M^d is embedded in the real Euclidean space R^D , where $d < D$. LTSA algorithm attempts to find a nonlinear mapping that maps the set X of N points to the low-dimensional coordinates $Y=[y_1, \dots, y_N]$ in R^d in a way of preserving local neighborhood structure through aligning local tangent space to a Jacobi matrix at each points. The fundamental model based on first order Taylor expansion is taken as following [7]:

$$f(\tilde{\tau}) = f(\tau) + J_f(\tau)(\tilde{\tau} - \tau) + O(\|\tilde{\tau} - \tau\|)$$

where $f: \Omega \subset R^d \rightarrow R^D$ is assumed smooth enough, and $f(\Omega)$ is a parameterized manifold. $J_f(\tau)$ denotes the Jacobi matrix of f at point τ .

3.2 3N-LTSA Algorithm

Given the data set $X=[x_1, \dots, x_N]$ in R^D , for each point x_i , we denote the set of its $nb(i)$ natural nearest neighbors (3N-neighbor) or sup_k nearest neighbors by a matrix $X_i=[x_{i1}, \dots, x_{inb(i)}]$ (for the second case, $nb(i)=sup_k \ i=1, \dots, N$). Note that $nb(i)$ ($i=1, \dots, N$) and the connectivity indicator sup_k can be automatically calculated according to the algorithm 1 described above. To preserve the local structure of each neighborhood X_i , like original LTSA [7] and LLTSA [8], the local linear approximation for the data points in X_i by using tangent space should be given as

$$\begin{aligned} & \arg \min_{x, \Theta, Q} \sum_{j=1}^{nb(i)} \|x_{ij} - (x + Q\theta_j)\|_2^2 \\ & = \arg \min_{\Theta, Q} \|X_i H_{nb(i)} - Q\Theta\|_2^2 \end{aligned} \quad (2)$$

where $H_{nb(i)} = I_{nb(i)} - ee^T/nb(i)$, $I_{nb(i)}$ denotes the $nb(i)$ -dimension identity matrix, e the $nb(i)$ -dimension vector of all ones, Q is an orthonormal basis matrix of the tangent space and has d columns, $\Theta = [\theta_1, \dots, \theta_{nb(i)}]$, where θ_j is the local coordinate corresponding to the basis Q . The optimal solution x in the above optimization is given by \bar{x}_i , the mean of all the x_{ij} s and the optimal Q is given by Q_i , the matrix of d left singular vectors of $X_i H_{nb(i)}$ corresponding to its d largest singular values, and Θ is given by Θ_i defined as

$$\Theta_i = Q_i^T X_i H_{nb(i)} = [\theta_1^{(i)}, \dots, \theta_{nb(i)}^{(i)}], \quad (3)$$

$$\theta_j^{(i)} = Q_i^T (x_{i_j} - \bar{x}_i).$$

Also similar to the calculations taken out in the original LTSA [7] or LLTSA [8], we can construct the global coordinates y_i , $i=1, \dots, N$, in R^d as following

$$y_{i_j} = \bar{y}_i + L_i \theta_j^{(i)} + \varepsilon_j^{(i)}, j=1, \dots, nb(i), i=1, \dots, N. \quad (4)$$

where \bar{y}_i denotes the mean value of y_{i_j} , L_i a local affine transformation matrix that needs to be determined, and $\varepsilon_j^{(i)}$ the local reconstruction error. Formula (4) can be represented in matrix form

$$Y_i H_{nb(i)} = L_i \Theta_i + E_i \quad (5)$$

where $Y_i = [y_{i_1}, \dots, y_{i_{nb(i)}}]$, $E_i = [\varepsilon_1^{(i)}, \dots, \varepsilon_{nb(i)}^{(i)}]$.

In order to preserve as much local structure information as possible in embedding space, minimizing the total local reconstruction errors E_i is required

$$\arg \min_{Y_i, L_i} \sum_i \|E_i\|_2^2 \equiv \arg \min_{Y_i, L_i} \sum_i \|Y_i H_{nb(i)} - L_i \Theta_i\|_2^2. \quad (6)$$

So, the optimal alignment matrix L_i that minimizes the local reconstruction error has the form (similar to [7])

$$L_i = Y_i H_{nb(i)} \Theta_i^+, \text{ and } E_i = Y_i H_{nb(i)} (I_{nb(i)} - \Theta_i^+ \Theta_i),$$

where Θ_i^+ is the Moore-Penrose generalized inverse of Θ_i .

Let $Y = [y_1, \dots, y_N]$ and S_i be 0-1 selection matrix such that $Y S_i = Y_i$. The objective function can be converted into the form [7, 8]

$$\arg \min_Y \sum_i \|E_i\|_F^2 = \arg \min_Y \|Y S W\|_F^2 \quad (7)$$

$$= \arg \min_Y \text{tr}(Y S W W^T S^T Y^T).$$

where $S = [S_1, \dots, S_N]$, and $W = \text{diag}(W_1, \dots, W_N)$ with $W_i = H_{nb(i)} (I_{nb(i)} - \Theta_i^+ \Theta_i)$. According to the numerical analysis in [7], W_i can also be written as $W_i = H_{nb(i)} (I_{nb(i)} - V_i V_i^T)$, where V_i presents the matrix of d right singular vectors of $X_i H_{nb(i)}$ corresponding to its d largest singular values. To uniquely determine Y , we impose the constraint $Y Y^T = I_d$. it turns out that the vector e of all ones is an eigenvector of

$$B = S W W^T S^T \quad (8)$$

corresponding to a zero eigenvalue. Therefore the optimal Y is obtained through the d eigenvectors of B corresponding to the 2nd to $d+1$ st smallest eigenvalues. We call this algorithm 3N-LTSA.

Comparison to the original LTSA algorithm, here an adaptive neighborhood based on 3N neighbor is adopted that making the data set into an automatically state in which local neighborhood structure information can be easily extracted without any assumptions and priori knowledge about the data. The rest of steps are strictly confirmed with the steps in original LTSA except the size of neighborhood may be different at each point and the dimension of tangent space upon each neighborhood.

As discussed above, 3N-LTSA algorithm can be summarized as following steps. Given a data set $X = [x_1, \dots, x_N]$.

Step 1: Constructing an adaptive neighborhood 3NG upon the data set X . For each x_i , $i=1, \dots, N$, select $nb(i)$ or sup_k nearest neighbors automatically. Indeed $nb(i)$ may be

other formats, such as $\max\{nb(i)\}$ if the amount of sampled points is relevant small.

Step 2: Extracting neighborhood feature spaces information. Compute V_i , the matrix of d singular vectors of $X_i H_{nb(i)}$ corresponding to its d largest singular values, and set $W_i = H_{nb(i)} (I_{nb(i)} - V_i V_i^T)$.

Step 3: Constructing alignment matrix. Form the matrix B by locally summing as follows:

$$B(I_i, I_i) \leftarrow B(I_i, I_i) + W_i W_i^T, i=1, \dots, N$$

with the initialization $B = 0$, where $I_i = \{i_1, \dots, i_{nb(i)}\}$ denotes the set of indices for the $nb(i)$ nearest neighbors of x_i .

Step 4: Aligning global coordinates. Compute the $d+1$ smallest eigenvectors of B and pick up the eigenvectors matrix corresponding to the 2nd to $d+1$ st smallest eigenvalues in non-descending order. Set the low dimensional coordinates as $Y = [Y_1, \dots, Y_N]$.

4. EXPERIMENTAL RESULTS

In this section, several experiments are carried out to evaluate the efficiency of the 3N-LTSA algorithm presented in this paper, and several data sets include synthetic data and real face data are used in these experiments. Three schemes based on the natural nearest neighbor are formulated to select the number of neighbors. First, the number of original 3N neighbors $nb(i)$ obtained from algorithm 1 is used directly to each point, second, the number of neighbors at each point is based both on the values of sup_k and $nb(i)$, i.e. $nb(i)$ is designed to be the value of $sup_k + |sup_k - nb(i)|$. Third, for every point i , the number of neighbors is selected to be $nb(i) - std(nb)$ if $nb(i)$ is great than sup_k , otherwise, the $nb(i) + std(nb)$ is adopted. For example, Fig.9 shows the results of 3N-LTSA by using three different schemes.

4.1 Synthetic Data Sets

Five sets of synthetic data obtained from the MANI demo are used to illustrate the low-dimensional intrinsic representations induced by 3N-LTSA (Fig.6, Fig.7, Fig.8, Fig.9, Fig.10), in which 3NG or variant SNG are used for representing the infrastructures corresponding to the several data respectively. Another three parameterized curve are also used to demonstrate the effectiveness of the 3N-LTSA (Fig.11, Fig.12, Fig.13).

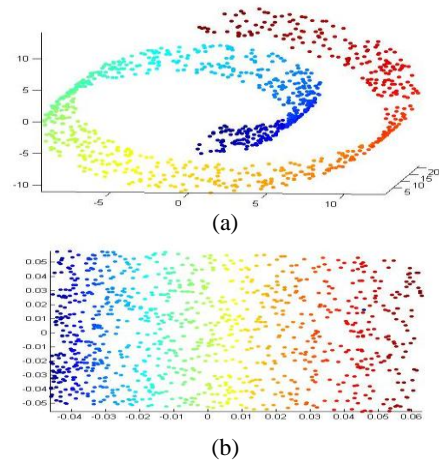


Fig.6. Swiss Roll data with 1000 points (top),The embedding (bottom) of 3N-LTSA corresponding to the left, in which, the number of neighbors at each point i is formulated to $nb(i)$, and $sup_k=8$.

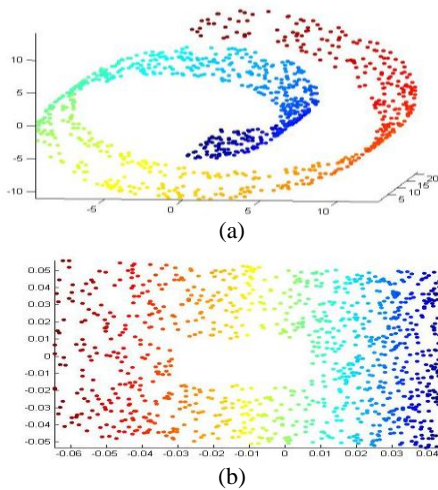


Fig.7. (a) Swiss Roll/ hole data (Donoho & Grimes) with 1000 points. (b) The embedding of 3N-LTSA corresponding to the data in (a), in which, the number of neighbors at each point i is formulated to $sup_k+abs(nb(i) -sup_k)$, and $sup_k=5$.

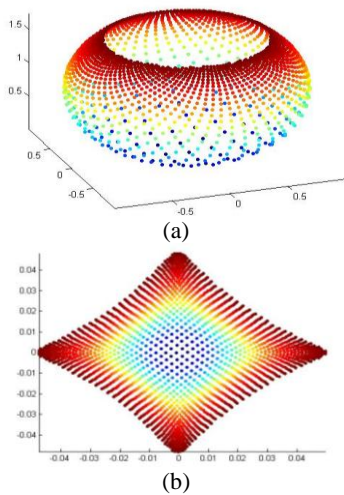


Fig.8 (a) Punctured sphere data (Saul & Roweis) with 2000 points. (b) The embedding of 3N-LTSA corresponding to the data in (a), in which, the number of neighbors at each point i is formulated to $nb(i)$, and $sup_k=5$.

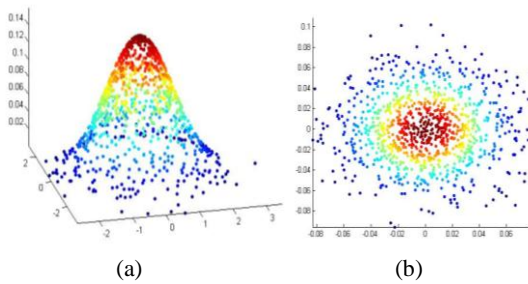


Fig.9. (a) Gaussian randomly sampled 1000 points. (b) The embedding of 3N-LTSA corresponding to the data, the number of neighbors at each point i equals $nb(i)$, the indicator $sup_k=49$.

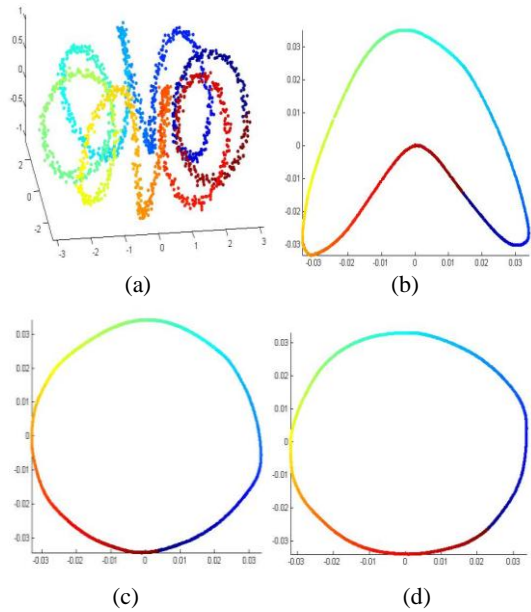


Fig.10. (a) Horoidal Helix (Coifman & Lafon) data with 1800 points. Three embeddings of 3N-LTSA corresponding to the data in (a) are illustrated in (b), (c) and (d), which corresponding to the number of neighbors at each point i , $nb(i)$, $sup_k+abs(nb(i)-sup_k)$ and if $nb(i)>sup_k$ then $nb(i)=nb(i)-std(nb)$ else $nb(i)=nb(i)+std(nb)$ respectively, the indicator $sup_k=11$.

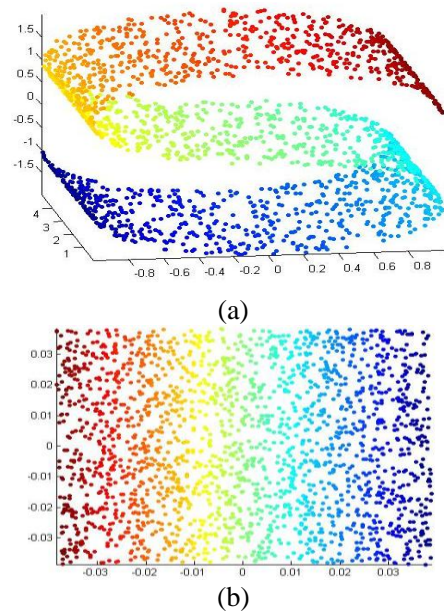


Fig.11. S-surface with 2000 points (left). The embedding of 3N-LTSA corresponding to the data (right), the number of neighbors at each point i equal $nb(i)$, the indicator $sup_k=6$.

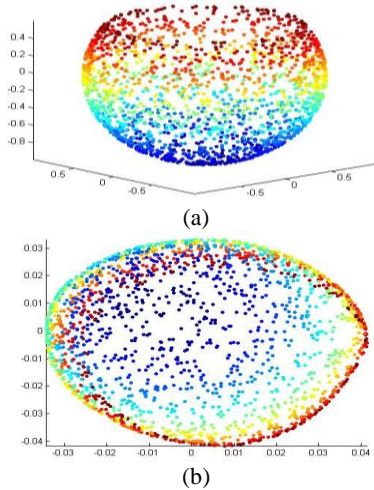


Fig.12. (a) Fish Bowl data with 2000 points. (b) The embedding of 3N-LTSA corresponding to the data, the number of neighbors at each point i equal $nb(i)$, the indicator $sup_k=7$.

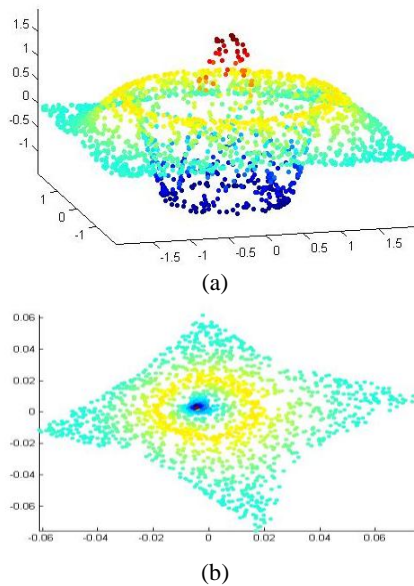


Fig.13. (a) Parameterized surface with 2000 points. (b) The embedding of 3N-LTSA corresponding to the data in (a), the number of neighbors at each point i equals to $nb(i)=sup_k+abs(nb(i)-sup_k)$, the indicator $sup_k=6$.

4.2 Face data

Isomap face data used in [1] is selected to illustrate the performance of 3N-LTSA algorithm, which contains 698 images represented as a set of 4096-dimensional vectors. Each vector represents the bright values of 64 pixels by 64 pixels image of a face in a way of rendering with different pose and lighting directions. In this experiment, a variant 3NG is used with 3N-LTSA algorithm, in which, the third scheme of selecting neighbors is applied, and the 2-D embedding is shown in Fig.14.

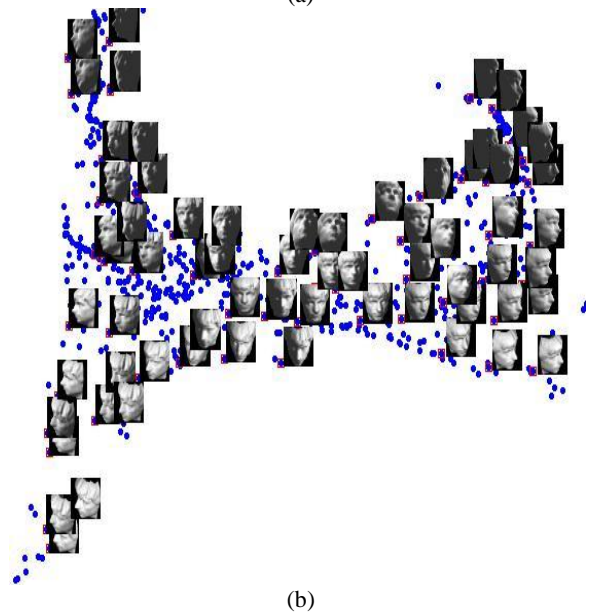
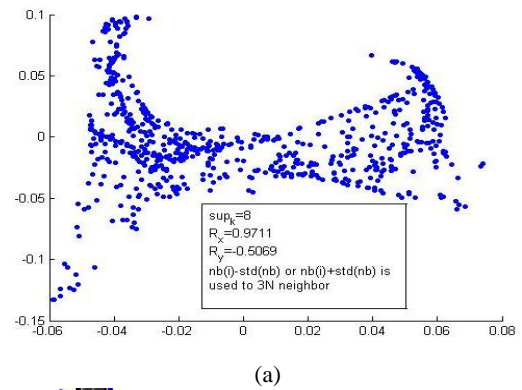


Fig.14. Isomap Face data contain 698 images. (a) The result of embedding generated by 3N-LTSA corresponding to this data, the number of neighbors at each point i are determined by following strategy: if $nb(i)>sup_k$ then $nb(i)=nb(i)-std(nb)$ else $nb(i) = nb(i) + std(nb)$, the indicator $sup_k=8$. (b) Visualization of all images in 2D space with respect to the embedding in (a).

5. CONCLUSIONS AND FUTURE WORKS

Here we propose a novel algorithm 3N-LTSA that applies a very simple and general adaptive neighborhood graph 3NG or SNG to the local manifold learning algorithm LTSA based on a novel strategy of choosing nearest neighbors, it provides a suitable and compactable representation about the various kind of data set, which is closely related to the data distributions whatever for the high or low dimensional data as shown above. Meanwhile, it is an adaptive unsupervised learning algorithm which extends the original LTSA algorithm to more broad applications, because it does not need any other information about the intrinsic structure with respect to the selection of free parameter. Many experiments show that it is a more practical and simple algorithm than LTSA.

Indeed, there are some other strategies about choosing adaptive neighborhood based on the statistics properties of $nb(i)$ and the indicator sup_k that also closely relates to the problem of learning similarity matrix in machine learning and

pattern recognition fields for the task of clustering and classification, this is a future work for us.

6. ACKNOWLEDGMENTS

This project is supported by the National Natural Science Foundation of China (61073058).

7. REFERENCES

- [1] J. Tenenbaum, V De Silva and J. C. Langford. A global geometric framework for nonlinear dimension reduction. *Science*, 290:2319–2323, 2000.
- [2] M. Berstein, V de Silva, J. Langford and J. Tenenbaum. Graph approximations to geodesics on embedded manifolds. <http://isomap.stanford.edu/BdSLT.pdf>, 2000.
- [3] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15*, Cambridge, MA:MIT Press, 2003, 705-712.
- [4] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290: 2323–2326, 2000.
- [5] L. Saul and S. Roweis, Think Globally, Fit Locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research* 4 (2003) 119-155.
- [6] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003
- [7] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, *SIAM J. Sci. Comput.* 26 (1)(2004) 313–338
- [8] T. Zhang, J. Yang, D. Zhao, X. Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing* 70(2007) 1547-1553. Letters.
- [9] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, in: *Proceedings of the 10 IEEE International Conference on Computer Vision*, Beijing, China, October 2005, pp. 1208–1213.
- [10] G.Hinton, S. Roweis. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems 15 (NIPS'02)*. pp. 857--864
- [11] X. He, P. Niyogi, Locality preserving projections, *Proceedings of Advances in Neural Information Processing Systems*. Cambridge:MIT Press, 2004: 153-160.
- [12] Tony Lin, Hongbin Zha, and Sang Uk Lee. Riemannian manifold learning for nonlinear dimensionality reduction. in *ECCV 2006*, A. Leonardis, H. Bischof, and A. Prinz (Eds.): Part I, LNCS 3951, pp. 44-55, 2006. Springer-Verlag Berlin Heidelberg 2006.
- [13] M. Balasubramanian and E. L. Schwartz, The Isomap algorithm and topological stability. *Science*, 295, 7a(2002).
- [14] J. Tenenbaum, V De Silva and J. C. Langford, The isomap algorithm and topological stability--response, *Science* 295, 7a (2002).
- [15] Xianlin Zou, Qingsheng Zhu and Yifu Jin. An adaptive neighborhood graph for LLE algorithm without free parameter. *International Journal of Computer Applications* 16(2), February 2011, ISBN: 978-93-80747-56-8, Digital Object Identifier: 10.5120/1984-2673, PP20-23.
- [16] Xianlin Zou, Qingsheng Zhu and Ruilong Yang. Natural nearest neighbor for Isomap algorithm without free-parameter. *Advance Materials Research*. Vols. 219-220(2011) pp. 994-998. doi: 10.4028
- [17] Xiaoming Huo and Andrew K. Smith, Matrix perturbation analysis of local tangent space alignment. *Linear Algebra and its Applications* 430 (2009) 732-746.
- [18] Yongzhou Li, Dayong Luo and Shaoqiang Liu, Orthogonal discriminant linear local tangent space alignment. *Neurocomputing* 72 (2009) 1319-1323.
- [19] J. Shi and J. Malik, Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.
- [20] S. Yu and J. Shi, "Multiclass Spectral Clustering," *Proc. Int'l Conf. Computer Vision*, 2003.