An Adaptive Neighborhood Graph for LLE 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

Yifu Jin
Department of Computer
Zhanjiang Normal University
Zhanjiang, Guangdong, China

ABSTRACT
Locally Linear Embedding (LLE) algorithm is the first classic nonlinear manifold learning algorithm based on the local structure information about the data set, which aims at finding the low-dimension intrinsic structure lie in high dimensional data space for the purpose of dimensionality reduction. One deficiency appeared in this algorithm is that it requires users to give a free parameter *k* which indicates the number of nearest neighbors and closely relates to the success of unfolding the true intrinsic structure. Here, we present an adaptive neighborhood graph with respect to LLE algorithm for learning an adaptive local infrastructure in order to avoid the problem of how to automatically choosing nearest neighbors existed in manifold learning by making use of a novel concept: natural nearest neighbor (3N). Experiment results show that LLE algorithm without free parameter performs more practical and simple algorithm than LLE.

General Terms
Machine Learning; Pattern Recognition; Algorithm; Dimensionality Reduction;

Keywords
Natural Nearest Neighbor; Adaptive Neighborhood Graph; LLE; Free Parameter Learning; Unsupervised learning

1. INTRODUCTION
Recently years, many efficient manifold learning algorithms with respect to dimensionality reduction have been proposed 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, such as Isomap[1,2], LLE[3,4], Lapacian Eigenmap(LE)[5], LTSA[6], NPE[7], SNE[8], LLP [9], RML[10], 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, but also in the analysis of algorithm's robustness with respect to the problem of topological stability [11,12].

LLE algorithm considered as the first and classic locally nonlinear manifold learning algorithm provided a primary approach to yield the relations between high and low dimensional representations of data points with the same locally linear relationships. This ideal leads to more other local geometry based learning algorithm as described above. Description about locally linear relationship is directly related to the size of neighborhood for each data points that eventually depends on the determination of the number *k* of nearest neighbors. In the case of learning intrinsic structure, neighborhood used in various state-of-the-art approaches is determined by the common used concept of *k-nn* or *ε-nn*, but how to find an appropriate value of *k* or *ε* is still an open issue, especially for the very high dimensional data, such as face data used in [1]. As suggested in [6] 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 thought about that it's worthy of considering variable number of neighbors that are adaptively chosen at each data point.

In this paper we describe a novel strategy to determine the number of nearest neighbors automatically for each data points, which leads to a novel concept of natural nearest neighbor (3N) in contrast to the *k-nn* or *ε-nn* neighbor, and results in an adaptive nearest neighborhood that can be easily applied to LLE algorithm for manifold learning and has no free parameter selection, it means that using LLE to reduce the dimensions of high dimensional data does not need any priori information about the intrinsic structure. Experiment results show that LLE algorithm without free parameter performs more practical and simple algorithm than LLE.

2. ALGORITHM

2.1 Determination of Natural Nearest Neighbors
The key idea is inspired 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 data objects, 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.

To demonstrate the concept of natural nearest neighbor (3N), we introduce an indicator that defines a possible and compact super-bound of *k* as following:

$$\sup_k = \sup_{r \in N}\{r \mid (\exists r)(r \in N \wedge ((\forall x)((x \in S) \rightarrow (\exists y)(y \in S \wedge y \neq x \wedge x \in NN_r(y)))))\} \quad (1)$$

where $NN_r(y)$ denotes the *r-th* nearest neighborhood in sense of *k-nn*, *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∈S, nb(i)=0， ratio_nb(i)=0, $NN_r(i)=\Phi$.
2. For every point i∈S, calculates the r^{th} nearest neighbor of i: $nn_r(i)$; $NN_r(i)= NN_r(i)\cup\{ nn_r(i)\}$.
3. For every point i∈S, counts the number of i occurred in all $NN_r(j)$:nb(i), (j=1,…,N); if there exist some nb(i) =0 then r=r+1 and goto step 2;
4. sup_k=r;
5. For all i, output: nb(i)， ratio_nb(i)=nb(i)/(N×sup_k), $NN_{nb(i)}(i)$ or $NN_{sup\_k}(i)$.
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.

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 comprised by connecting each point *i* to its *nb*(*i*) nearest neighbors (Fig.1, Fig.2, Fig.3), 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.

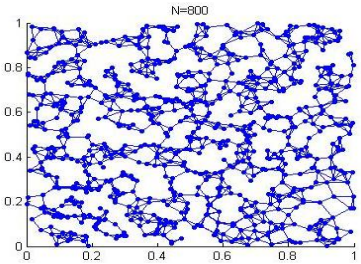

Fig.1 3NG with 800 random sampled points within region of [0,1]×[0,1]

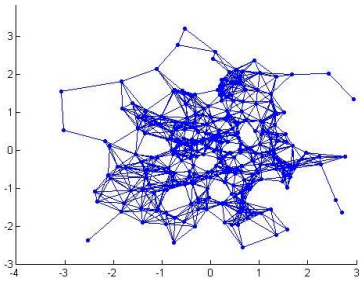
Fig.2 3NG with 300 random sampled points following normal distribution.

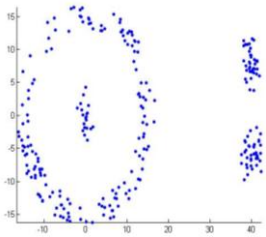
Fig.3 Artificial data points.

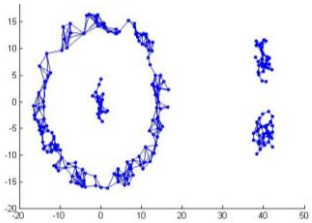
Fig.4 3NG relates to the data points in Fig.3

2.3 LLE Algorithm without Free Parameter k

Appling 3NG or SNG to the nonlinear local dimensionality reduction methods LLE [3] is very simple; the algorithm is illustrated in Table 2. In contrast to the LLE, this algorithm does not need the user to give the free parameter *k* which must be specified in LLE. We would call this algorithm as 3N-LLE, due to the use of novel concept of natural nearest neighbor (3N) that makes LLE to have more adaptability and flexibility, and leads to a more efficient neighborhood graph for unsupervised learning (Fig.3).

Table 2. Adaptive Locally Linear Embedding

Algorithm 2: **3N-LLE. Input data S={x_i|x_i∈R^D, *i=1, 2, …, N*}, and d: embedding dimension; Output: low-dimensional representations Y={y_i|y_i∈R^d, *i=1, 2, …, N* }**.

Step1. For all data points i∈S, calculating the number of neighbors nb(i)， indicator sup_k, $NN_{nb(i)}$(i) or NN_{supk}(i) by using algorithm 1.

Step2. Calculating the local geometry *W*={w_{ij}| *i=1, 2, …, N* , *j=1, 2, …, nb(i) or* sup_k} for all points as following:

$$\arg_w \min \sum_i || x_i - \sum_{j \in NN_{nb(i)}(i)} w_{ij} x_j ||^2$$

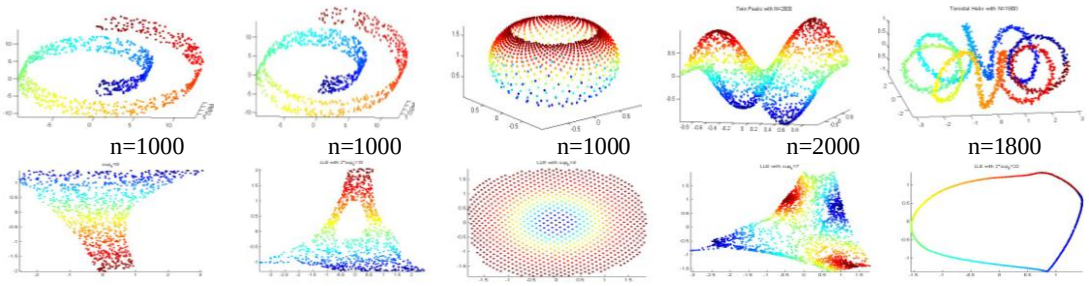

Fig.4 3N-LLE embeddings with fixed numbers of neighbors.

s.t. $\sum_{j\in NN_{nb(i)}(i)} w_{ij} = 1,\ for\ all\ i \in \{1,\cdots,N\}$ or

$$\arg_w \min \sum_i \| x_i - \sum_{j\in NN_{\sup_k}(i)} w_{ij} x_j \|^2$$

s.t. $\sum_{j\in NN_{\sup_k}(i)} w_{ij} = 1,\ for\ all\ i \in \{1,\cdots,N\}$

Step3. For all data point i∈S, calculating the low-dimensional representations Y as following:

$$\arg_Y \min \sum_i \| y_i - \sum_{j\in NN_{nb(i)}(i)} w_{ij} y_j \|^2$$

s.t. $\sum_i y_i = 0\ and\ \frac{1}{N}\sum_i y_i \otimes y_i = I$ or

$for\ all\ i \in \{1,\cdots,N\}$

$$\arg_Y \min \sum_i \| y_i - \sum_{j\in NN_{\sup_k}(i)} w_{ij} y_j \|^2$$

s.t. $\sum_i y_i = 0\ and\ \frac{1}{N}\sum_i y_i \otimes y_i = I$

$for\ all\ i \in \{1,\cdots,N\}$

3. EXPERRIMENT RESULTS

3.1 Synthetic Data Sets

Five sets of synthetic data obtained from the MANI demo (http://www.math.umn.edu/~wittman/mani/) are used to illustrate the low-dimensional intrinsic representations induced by 3N-LLE (Fig.4), in which SNG is used for represents the infrastructure corresponding to the data, i.e., each points have the same number of neighbors $\sup_k$ or multiple of $\sup_k$, and the number of neighbors automatically detected are shown under the respective embeddings

3.2 Face Data

Isomap face data used in [1] is selected to illustrate the performance of 3N-LLE algorithm, which consists of 698 images presented 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, 3NG is used within 3N-LLE algorithm, and the 2-D embedding is shown in Fig.5.

3.3 COIL-20 Data Set

COIL-20 data set (available at ftp:zen.cs.columbia.edu) is also

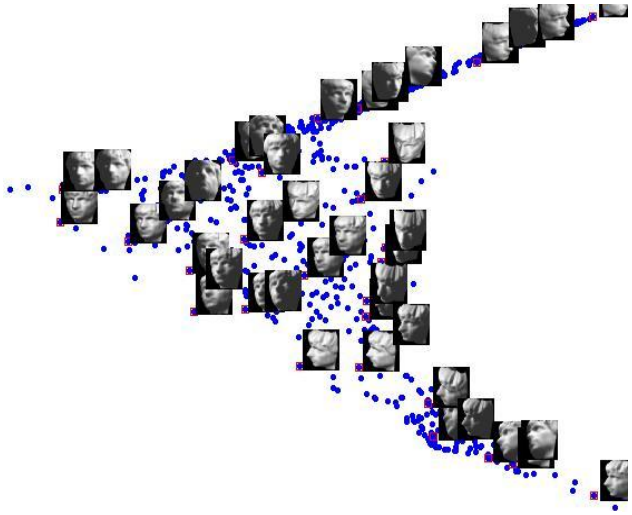

Fig.5 3N-LLE 2-D embedding with face data used in [1]

used to estimate the efficiency of unsupervised learning algorithm 3N-LLE, the 2-D embeddings of 3N-LLE with the use of SNG graph are illustrated in Fig.6 for six objects in COIL-20 data set. Images of the objects were taken at pose intervals of 5 degrees, while the object was rotated through 360 degrees with respect to a fixed camera, so 72 images were generated relevant to each object. As shown in Fig.6, the 3N-LLE's embeddings reveal much better results that conform to the mechanism of generating data.

4. CONCLUSIONS AND FUTURE WORKS

Here we present a very simple and general adaptive neighborhood graph 3NG or SNG for local manifold learning algorithm LLE 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, an adaptive unsupervised learning algorithm 3N-LLE is proposed which extends the classic LLE algorithm to be able to applied in more broad applications, because it does not need any other information about the intrinsic structure. Observing the result in Fig.4, we find that 3NG itself have certain abilities of clustering and classification, it may be considered as a

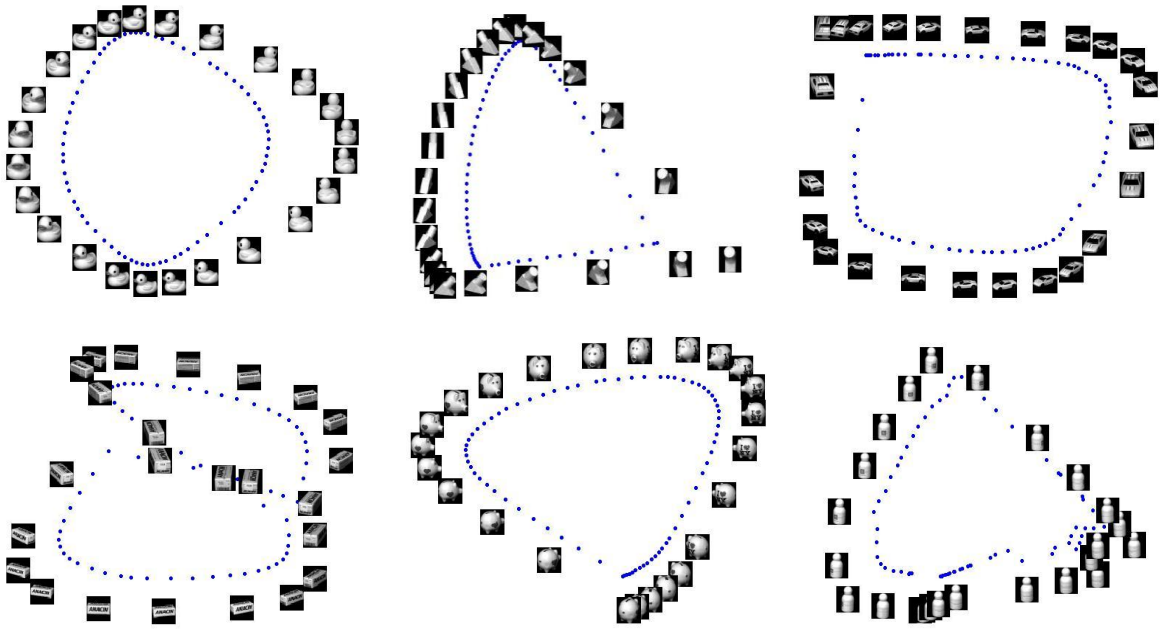

Fig.6 3N-LLE 2-D embeddings for small sample and high-dimensional images corresponding to six objects in Coil-20 data are illustrated, where SNG is applied to the 3N-LLE. There are 72 images related to each object.

promising direction to clustering analysis based on the natural nearest neighborhood graph.

5. KNOWLEDGMENTS

This project is supported by Guangdong Natural Science Fund, China (Grant No. 9151027501000039). The authors are grateful for the support of National Natural Science Foundation of China (61073058).

6. 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] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290: 2323–2326, 2000.

[4] L. Saul and S. Roweis, Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. Journal of Machine Learning Research 4 (2003) 119-155.

[5] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15(6):1373–1396, June 2003

[6] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, SIAM J. Sci. Comput. 26 (1)(2004) 313–338

[7] 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.

[8] G.Hinton, S. Roweis. Stochastic Neighbor Embedding. Advances in Neural Information Processing Systems 15 (NIPS'02). pp. 857--864

[9] X. He, P. Niyogi, Locality Preserving Projections, Proceedings of Advances in Neural Information Processing Systems. Cambridge:MIT Press, 2004: 153-160.

[10] 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.

[11] M. Balasubramanian and E. L. Schwartz, The Isomap Algorithm and Topological Stability. Science, 295, 7a(2002).

[12] J. Tenenbaum, V De Silva and J. C. Langford, The isomap algorithm and topological stability--response, Science 295, 7a (2002).