Journal of Zhejiang Unirsy SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Image feature optimization based on nonlinear dimensionality reduction*

Rong ZHU[1,2,3], Min YAO[1]

([1]School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)
([2]School of Information Engineering, Jiaxing University, Jiaxing 314001, China)
([3]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)
E-mail: {zr, myao}@zju.edu.cn

Received May 26, 2009; Revision accepted Aug. 10, 2009; Crosschecked Nov. 11, 2009

Abstract: Image feature optimization is an important means to deal with high-dimensional image data in image semantic understanding and its applications. We formulate image feature optimization as the establishment of a mapping between high- and low-dimensional space via a five-tuple model. Nonlinear dimensionality reduction based on manifold learning provides a feasible way for solving such a problem. We propose a novel globular neighborhood based locally linear embedding (GNLLE) algorithm using neighborhood update and an incremental neighbor search scheme, which not only can handle sparse datasets but also has strong anti-noise capability and good topological stability. Given that the distance measure adopted in nonlinear dimensionality reduction is usually based on pairwise similarity calculation, we also present a globular neighborhood and path clustering based locally linear embedding (GNPCLLE) algorithm based on path-based clustering. Due to its full consideration of correlations between image data, GNPCLLE can eliminate the distortion of the overall topological structure within the dataset on the manifold. Experimental results on two image sets show the effectiveness and efficiency of the proposed algorithms.

Key words: Image feature optimization, Nonlinear dimensionality reduction, Manifold learning, Locally linear embedding (LLE)
doi:10.1631/jzus.A0920310 **Document code:** A **CLC number:** TP391

INTRODUCTION

Image feature optimization is aimed to mine useful features from a set of basic elements or numeric values that are generally used to describe the characteristics of image data. In other words, image feature optimization is a process for refining a few ‘small quantity and high quality’ features from the raw ones generated by feature extraction, which means that feature dimensionalities should be reduced as much as possible without losing the critical characteristics of image data. Image semantic understanding and its applications, such as image object recognition, image clustering and classification, and semantic-based image retrieval, usually involve large high-dimensional datasets. Nowadays, with the rapid development of network technology and fast decline of the cost of storage devices, a large number of digital multimedia resources are available on the Internet. Images in the network environment (i.e., Web images) are preferable due to their easy sharing and distribution properties. Different from ordinary images where little information is provided, there exists a lot of additional contextual information on Web pages like surrounding text and links (Hua *et al.*, 2005). To bridge the semantic gap (Smeulders *et al.*, 2000; Dorai and Venkatesh, 2003) generated by the differences between computer representation and human perception, both low-level visual features (e.g., color, texture, shape, and spatial relationship) and high-level text contents (e.g., metadata, keyword, and phrase) should be fully exploited and utilized, which will result in a very high feature dimensionality (e.g.,

* Project (No. 2008AA01Z132) supported by the National High-Tech Research and Development Program of China

100, or even 1000) in some applications. However, most researchers engaged in image semantic understanding have paid more attention to constructing the mappings between visual features and semantic concepts or building effective learning machines, whereas the contributions of feature optimization for high-dimensional data have often been ignored and traditional approaches and techniques are often not scalable enough for use in Web images to handle the vast data amount. Consequently, for minimizing the effect of 'dimensionality curse' (Bellman, 1961), developing an efficient image feature optimization approach to improve system performance, reduce execution complexity and ease the burden on learners (e.g., classifiers) is not a trivial task (Datta *et al.*, 2008).

Wang *et al.*(2003) presented a new method to optimize feature extraction for target recognition, which is based on wavelet theory in view that wavelet transform has perfect local performance and can automatically adjust the sampling frequency along with the changes of signal frequency components. Krishnapuram *et al.*(2004) developed a Bayesian generalization of the support vector machine (SVM), which identifies the optimal nonlinear classifier and selects the optimal set of image features via the optimization of Bayesian likelihood functions. Zhang and Izquierdo (2006) proposed a multi-feature optimization method for object-based image classification, where the best linear combination of visual descriptors can be found and the metrics of these descriptors are optimized by analyzing the underlying patterns of low-level visual primitives. Different from the above literature, in this paper we formulate image feature optimization by a five-tuple model, i.e., establishing a mapping from a high-dimensional space to a feature space with low dimensions. Hence, it is feasible to perform the optimization based on various dimensionality reduction approaches.

Principal component analysis (PCA) (Jolliffe, 1986; Turk and Pentand, 1991) and linear discriminant analysis (LDA) (Sweis and Weng, 1996; Belhumeur *et al.*, 1997) are two well-known linear dimensionality reduction approaches, both of which are eigenvector approaches designed to establish the linear variability model for a high-dimensional dataset. These two approaches are simple to implement but they cannot deal with the nonlinear dataset in real-world applications. Meanwhile, for eliminating the nonlinear redundancies within features, the above linear approaches can be transformed to nonlinear versions by applying a certain kernelization technique, i.e., kernel PCA (KPCA) or kernel LDA (KLDA). Manifold learning is a recently developed technique for nonlinear dimensionality reduction. Seung and Lee (2000) pointed out that data in a high-dimensional space can be mapped into a manifold with low dimensions. This low-dimensional manifold embodies the relations between image data and has a lower intrinsic dimensionality. Theoretically, based on such an assumption, if able to find the potential topological data structure, we will acquire useful information (e.g., data differences) hidden in the manifold without analyzing the data in the high-dimensional space; this not only reduces the data dimensionality but also effectively avoids the possibility of 'dimensionality curse'. Therefore, nonlinear dimensionality reduction based on manifold learning provides a new solution to image feature optimization.

So far many nonlinear dimensionality reduction approaches have been proposed, such as locally linear embedding (LLE) (Roweis and Saul, 2000; Saul and Roweis, 2003), Laplacian eigenmap (LE) (Belkin and Niyogi, 2001; 2003), isometric mapping (ISOMAP) (Tenenbaum *et al.*, 2000), and multi-level mahalanobies-based dimensionality reduction (MMDR) (Jin *et al.*, 2003). Yin (2007) gave a review of nonlinear dimensionality reduction approaches and their variants in the recent years. Among them, LLE is a representative algorithm and is widely used owing to its strong abilities of shape preserving mapping (i.e., two nearby data in a high-dimensional space maintain their relations when they are mapped into a manifold with low dimensions) and good computational performance. Based on the assumption of local linearity in the nonlinear manifold, LLE first constitutes the local coordinates with the least constructed cost and then maps them into a global one. As is well known, in most cases, local information may be more important than global information for image semantic understanding and its applications (Wang *et al.*, 2001; Jeon *et al.*, 2003). Many recent studies have shown that LLE is suitable for the dimensionality reduction of image data (Wu *et al.*, 2004; Xu *et al.*, 2004;

Abusham *et al.*, 2005; Yao and Tao, 2005; Li *et al.*, 2006; Yang *et al.*, 2007).

Although LLE is a powerful algorithm in dimensionality reduction, some limitations greatly restrict its applicability, e.g., high sensibility to noise and incapacity to deal with sparse datasets. Among many newly proposed nonlinear dimensionality reduction approaches based on LLE, four groups can be roughly identified: (1) Modify the distance calculation formula. Wang *et al.*(2006) proposed a coherence measurement by increasing the distances between the data in densely distributed regions and decreasing the distances in sparsely distributed regions. But it is still unsuitable for handling sparse datasets. (2) Transform an unsupervised approach to a supervised or semi-supervised one. de Ridder *et al.*(2003) presented a supervised locally linear embedding (SLLE) algorithm, where category information is added to modify the calculation of distance measure. Although keeping the distances between the data in the same class and ensures the feasibility of subsequent supervised classification, SLLE enlarges the distances between the data from different classes, which may result in a distortion of the intrinsic structure of the data. (3) Define an embedding function using the kernel technique. Cao and Ye (2007) proposed a local project linear embedding (LPLE) algorithm by designing a global embedding function to reconstruct the eigenvectors of local neighbors. In LPLE, the low-dimensional manifold can be easily obtained via a cost matrix, but it is hard to choose a reasonable kernel. (4) Unify linear and nonlinear dimensionality reduction approaches. Chang *et al.*(2004) put forward a merging strategy—firstly, for furthest preserving nonlinear structures, the data are mapped into a transitional space via LLE, and then another dimensionality reduction is performed using LDA. This strategy can effectively reduce the loss of useful information caused by loss of too much data variance in LLE, but may lead to an increase in the computational complexity due to the selection of an appropriate dimensionality for the transitional space.

In this paper, we propose a novel locally linear embedding algorithm, the globular neighborhood based locally linear embedding algorithm (GNLLE for short), based on neighborhood update and an incremental neighbor search scheme. Since the irregular neighborhood identified by the neighbor number is replaced by the globular neighborhood identified by the globular radius, GNLLE not only has the capability to deal with sparse datasets but also is less sensitive to noise and more topologically stable. As the distortion of the overall topological structure within the dataset on the manifold is usually caused by pairwise similarity calculation and it may be more serious when handling a dataset with curved surfaces, we present an improved algorithm, the globular neighborhood and path clustering based locally linear embedding algorithm (GNPCLLE for short). With path-based clustering, GNPCLLE can reselect the nearest neighbors of the current data and update the globular neighborhood obtained by GNLLE. Since data correlation is applied in GNPCLLE, data connections can be truly acquired through the mediating paths of intermediate data rather than certain high mutual similarities between the data, resulting in a psychophysically plausible definition of distance measure. Experimental results showed that the proposed algorithms outperform some existing ones and are both efficient for feature optimization in image semantic understanding and its applications.

IMAGE FEATURE OPTIMIZATION

Image feature optimization can be formulated by a five-tuple model:

$$\mathrm{FO} = (X, D, \delta, d, Y), \tag{1}$$

where D and d ($<<D$) denote the dimensionalities of the high- and low-dimensional space (i.e., manifold), respectively. X is an input image set, composed of N D-dimensional real-valued vectors in the high-dimensional space $\mathbb{R}^D$; i.e., $X=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \mid \boldsymbol{x}_i=[x_i^1, x_i^2, \ldots, x_i^D]^{\mathrm{T}}, i=1, 2, \ldots, N\}$. Y is an output image set consisting of N d-dimensional real-valued vectors in the low-dimensional space $\mathbb{R}^d$; i.e., $Y=\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N \mid \boldsymbol{y}_i=[y_i^1, y_i^2, \ldots, y_i^d]^{\mathrm{T}}, i=1, 2, \ldots, N\}$. $\delta: X \mapsto Y$ denotes a mapping and is the core of FO. Since X is a nonlinear image set, δ can be defined as a one-to-one mapping. Thus, image feature optimization formulated by FO can be viewed as establishing a mapping from $\mathbb{R}^D$ to $\mathbb{R}^d$. Obviously, only the optimized image features can be taken as the input data for subsequent learning machines.

LOCALLY LINEAR EMBEDDING ALGORITHM

Outline of LLE

The LLE algorithm proposed by Roweis and Saul (2000) attempts to find the hidden topological structure between two data by mapping the data from a high-dimensional space into a low-dimensional embedding manifold while preserving their local similarities. The assumption that the obtained embedding manifold is locally linear guarantees the shape preserving mapping of the data. The basic principle of LLE is to utilize linear structure to approximately characterize the parts of nonlinear structure. Thus, nonlinear dimensionality reduction can be simplified into local linear dimensionality reduction; i.e., each data in the high-dimensional space can be expressed by a linear representation of its nearest neighbors and then it can be reconstructed on the low-dimensional embedding manifold by minimizing a cost function. A summary of LLE is illustrated in Fig.1.

As shown in Fig.1, LLE consists of three steps: (1) Select k nearest neighbors for each current data $\boldsymbol{x}_i$ (i=1, 2, …, N) and regard these nearest neighbors as contributions to the reconstruction of $\boldsymbol{x}_i$ in the embedding manifold. (2) Compute the local linear reconstruction weights w_{ij} (j=1, 2, …, k) of the linear representation that best reconstructs $\boldsymbol{x}_i$ using the k nearest neighbors of $\boldsymbol{x}_i$. (3) Reconstruct $\boldsymbol{x}_i$ (expressed by $\boldsymbol{y}_i$ on the manifold) via w_{ij} to map the high-dimensional input X into the low-dimensional output Y.

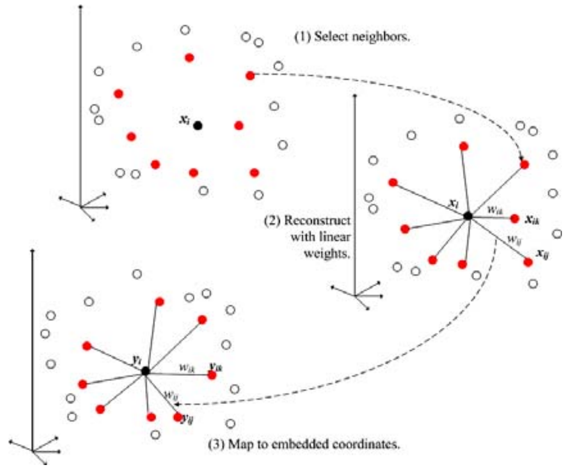


Fig.1 A summary of the locally linear embedding algorithm (Roweis and Saul, 2000)

Experimental analysis

To observe how the neighbor number k affects the performance of LLE, we designed an experiment on the Twin Peaks dataset. The dataset is composed of samples randomly chosen from 3D Twin Peaks surfaces, where the relations between the samples are depicted by colors. The performance of LLE can be visually seen through the aggregation extent of samples with the same color. In the experiment, the samples were mapped from a 3D space into a 2D space.

According to the experimental results (Fig.2), some conclusions can be drawn as follows:

1. The parameter k determines the performance of LLE to some extent. When k is small, many nearest neighbors that contribute to the reconstruction of the

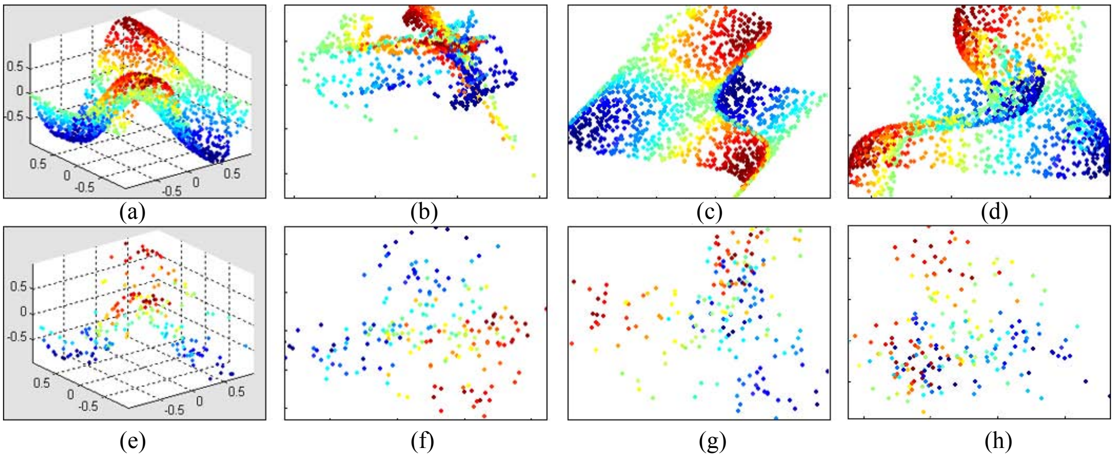


Fig.2 Performance of the locally linear embedding algorithm on the Twin Peaks dataset

(a) Distribution map when N=2000; (b) k=8; (c) k=15; (d) k=30; (e) Distribution map when N=200; (f) k=8; (g) k=15; (h) k=30. N=2000 for (b)~(d); N=200 for (f)~(h). N denotes the data number; k denotes the neighbor number current data will be lost and LLE may fail to maintain the local topological structure of the samples in the low-dimensional space (Fig.2b), whereas if k is large, some outliers (may be caused by noise) will be treated as the neighbors of the current data, which may confuse neighbor relations between the samples (Fig.2d). Thus, only at an appropriate k, will LLE succeed in revealing the underlying embedding manifold and fully reflecting the aggregation ability of similar data (Fig.2c).

2. The dense or sparse degree of data distribution affects the performance of LLE. We can infer that the samples' distribution in the high-dimensional space is denser; LLE obtains its best result with a smaller k and enters into a stable status earlier. In the low-dimensional space, the samples' distribution is sparser; LLE obtains its best result with a bigger k and enters into its stable status later. In sum, the denser the dataset is, the better the LLE performance will be, and vice versa. For example, when the sparse degree of the dataset is high (e.g., N=200), most samples cross or overlap with each other, which may easily damage the data intrinsic structure (Figs.2f~2h).

Compared with other nonlinear dimensionality reduction approaches, LLE has only one free parameter and thus has a simple implementation. However, in the network environment, the sparse distribution of the data generated by huge image data is inevitable. Even if the image set is complete enough, its high dimensionality will lead to an unfeasible calculation. Additionally, some noise or external disturbances may be absorbed wrongly during feature extraction. Consequently, the assumption that the data distribution on the embedding manifold is relatively even cannot be usually satisfied in real-world applications. It is obvious that LLE is more sensitive to the parameter k especially when handling the dataset with an uneven distribution.

A NOVEL LOCALLY LINEAR EMBEDDING BASED ON OPTIMIZED NEIGHBORHOOD

Construction of globular neighborhood

In this paper, we propose a novel nonlinear dimensionality reduction algorithm named globular neighborhood based locally linear embedding (GNLLE). That is, inspired by geometric intuition, we present a new locally linear embedding based on LLE using neighborhood update and an incremental neighbor search scheme. The main idea is to use a regular neighborhood constructed by a radius instead of the irregular neighborhood constructed by a neighbor number in LLE and to search for the candidate data within the globular neighborhood based on radius increment, and then these selected data are regarded as the nearest neighbors of the current data $\boldsymbol{x}_i$ (located at the core of the globular neighborhood). Consistent with the principle of LLE, in GNLLE, the closer the data is to $\boldsymbol{x}_i$ on the high-dimensional embedding manifold, the more opportunity the data have to be chosen as the neighbor of $\boldsymbol{x}_i$. From another point of view, the smaller the distance between the neighbor and $\boldsymbol{x}_i$ is, the bigger contribution the neighbor makes to the reconstruction of $\boldsymbol{x}_i$ on the embedding manifold. Therefore, GNLLE not only meets the demand of shape preserving mapping but also has the aggregation ability of the similar data. Before describing the details of GNLLE, we first give the processes of both neighborhood construction and neighbor selection in LLE from a geometry view.

Suppose that N points are distributed in a D-dimensional Euclidean space E^D (for simplification, each point denotes one data in X). Take $\boldsymbol{x}_i$ as a reference and calculate the distance $d_{\boldsymbol{x}_{ij}\boldsymbol{x}_i}$ between $\boldsymbol{x}_{ij}$ ($\boldsymbol{x}_{ij}$ denotes the neighbors of $\boldsymbol{x}_i$) and $\boldsymbol{x}_i$ (i, j=1, 2, …, N), and then a symmetric distance matrix $\boldsymbol{D}$ can be established as follows:

$$\boldsymbol{D}_{N\times N}=\begin{bmatrix} d_{\boldsymbol{x}_{11}\boldsymbol{x}_1} & d_{\boldsymbol{x}_{21}\boldsymbol{x}_2} & \cdots & d_{\boldsymbol{x}_{N1}\boldsymbol{x}_N} \\ d_{\boldsymbol{x}_{12}\boldsymbol{x}_1} & d_{\boldsymbol{x}_{22}\boldsymbol{x}_2} & \cdots & d_{\boldsymbol{x}_{N2}\boldsymbol{x}_N} \\ \vdots & \vdots & & \vdots \\ d_{\boldsymbol{x}_{1N}\boldsymbol{x}_1} & d_{\boldsymbol{x}_{2N}\boldsymbol{x}_2} & \cdots & d_{\boldsymbol{x}_{NN}\boldsymbol{x}_N} \end{bmatrix}. \quad (2)$$

For selecting k (k<N) nearest neighbors for $\boldsymbol{x}_i$ conveniently, the matrix $\boldsymbol{D}$ needs sorting in ascending order by column, i.e., $d'_{\boldsymbol{x}_{i1}\boldsymbol{x}_i} \le d'_{\boldsymbol{x}_{i2}\boldsymbol{x}_i} \le \cdots \le d'_{\boldsymbol{x}_{iN}\boldsymbol{x}_i}$, and then the 2nd to ($k$+1)th elements in each column are used to establish a new matrix $\boldsymbol{D}'$ for each point and its nearest neighbors:

$$\boldsymbol{D}'_{k\times N}=\begin{bmatrix} d'_{\boldsymbol{x}_{11}\boldsymbol{x}_1} & d'_{\boldsymbol{x}_{21}\boldsymbol{x}_2} & \cdots & d'_{\boldsymbol{x}_{N1}\boldsymbol{x}_N} \\ d'_{\boldsymbol{x}_{12}\boldsymbol{x}_1} & d'_{\boldsymbol{x}_{22}\boldsymbol{x}_2} & \cdots & d'_{\boldsymbol{x}_{N2}\boldsymbol{x}_N} \\ \vdots & \vdots & & \vdots \\ d'_{\boldsymbol{x}_{1k}\boldsymbol{x}_1} & d'_{\boldsymbol{x}_{2k}\boldsymbol{x}_2} & \cdots & d'_{\boldsymbol{x}_{Nk}\boldsymbol{x}_N} \end{bmatrix}. \quad (3)$$

Based on the above analysis, inspired by geometric intuition, we present GNLLE via neighborhood optimization. Suppose that N points are distributed in a D-dimensional Euclidean space E^D. Take $\boldsymbol{x}_i$ as the globular core and construct a globe G based on the radius r, and then G is the globular neighborhood of the current data $\boldsymbol{x}_i$. If there exist p_i ($p_i<N$) points within G, such p_i points can be regarded as the nearest neighbors of $\boldsymbol{x}_i$ and then the distance $d_{\boldsymbol{x}_{ij}\boldsymbol{x}_i}$ ($\leq r$) between $\boldsymbol{x}_{ij}$ (j=1, 2, …, p_i) and $\boldsymbol{x}_i$ should be computed. Since the neighbor number of each point within its globular neighborhood is different, the distance matrix cannot be directly established as in LLE. We use N column vectors to express them:

$$\begin{bmatrix} d''_{\boldsymbol{x}_{11}\boldsymbol{x}_1} \\ d''_{\boldsymbol{x}_{12}\boldsymbol{x}_1} \\ \vdots \\ d''_{\boldsymbol{x}_{1p_1}\boldsymbol{x}_1} \end{bmatrix}, \begin{bmatrix} d''_{\boldsymbol{x}_{21}\boldsymbol{x}_2} \\ d''_{\boldsymbol{x}_{22}\boldsymbol{x}_2} \\ \vdots \\ d''_{\boldsymbol{x}_{2p_2}\boldsymbol{x}_2} \end{bmatrix}, \ldots, \begin{bmatrix} d''_{\boldsymbol{x}_{N1}\boldsymbol{x}_N} \\ d''_{\boldsymbol{x}_{N2}\boldsymbol{x}_N} \\ \vdots \\ d''_{\boldsymbol{x}_{Np_N}\boldsymbol{x}_N} \end{bmatrix}. \quad (4)$$

In order to obtain a simple and efficient computation, we define the distance $d_{\boldsymbol{x}_{ij}\boldsymbol{x}_i}$ (i,j=1, 2, …, N, $i\neq j$) using L_2 distance measure (de Juan and Bodenheimer, 2004) as follows:

$$d_{\boldsymbol{x}_{ij}\boldsymbol{x}_i}=\mathrm{dist}(\boldsymbol{x}_{ij},\boldsymbol{x}_i)=\left\|\boldsymbol{x}_{ij}-\boldsymbol{x}_i\right\|_{L_2}. \quad (5)$$

For viewing the different ideas of neighborhood construction and neighbor selection in GNLLE and LLE more intuitively, theses steps in the two algorithms are illustrated in a 2D Euclidean space.

As shown in Fig.3, the data distribution around $\boldsymbol{x}_i$ (in the circle on the right) is denser than around $\boldsymbol{x}_{i'}$ (i'=1, 2, …, N, $i'\neq i$) (in the circle on the left); therefore, if the neighborhood is constructed using the neighbor number (e.g., k=6), the neighborhood range of $\boldsymbol{x}_i$ is smaller than that of $\boldsymbol{x}_{i'}$ (Fig.3b). But if the neighborhood is identified by a globular radius (e.g., r=0.8), none of the nearest neighborhoods of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ are effected by the dense or sparse degree of the distribution (Fig.3a). Obviously, due to a fixed neighbor number in LLE, some outliers may be regarded as the nearest neighbors and added into the neighborhood mistakenly. While the neighbor selection in GNLLE relies only on the distance between the candidate data and the globular core (denotes the current data), the neighbor number is therefore variable, which effectively alleviates its sensitivity to noise.

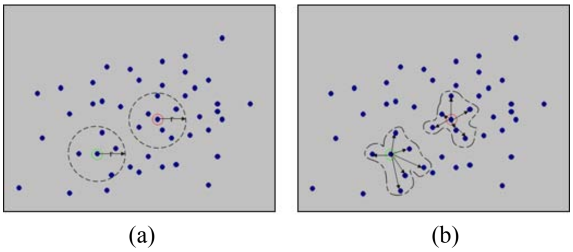

(a) (b)

Fig.3 The neighborhood construction and neighbor selection in (a) GNLLE and (b) LLE

The points in the circles denote the current data

It is worth noting that since a significant performance degradation may occur in LLE due to small changes of the parameter k, we adopt in the implementation of GNLLE an incremental search scheme for optimization; i.e., a radius increment Δr is adopted to search for the nearest neighbors rather than directly rely on the value of the globular radius r. Specifically, each search step starts from an initial radius $r_{\min}$ (which is the shortest distance between the current data and its neighbors) and then expands outwards constantly with Δr until it exceeds the globe identified by r. Such a scheme has the following advantages: (1) A precise similarity measurement can be obtained via the continuous changes of Δr; (2) Since the value of Δr is relatively small, the process of neighbor searching can also be viewed as an approximate sorting process, which can greatly reduce the computational complexity of GNLLE.

GNLLE algorithm

The GNLLE algorithm can be summarized as follows:

Step 1: For each data $\boldsymbol{x}_i$ in X, construct its globular neighborhood via a globular radius r and compute the distance between the candidate data $\boldsymbol{x}_{ij}$

(within the globular neighborhood) and the current data $\boldsymbol{x}_i$ (Eq.(5)). If there exist p_i data that satisfy the condition $d_{x_{ij},x_i} < r$, these data are chosen as the nearest neighbors of $\boldsymbol{x}_i$.

Step 2: Calculate the local linear reconstruction weight w_{ij} using the p_i nearest neighbors of $\boldsymbol{x}_i$. The reconstruction error function can be defined as

$$\begin{aligned}\varepsilon_i &= \left|\boldsymbol{x}_i - \sum_{j=1}^{p_i} w_{ij}\boldsymbol{x}_{ij}\right|^2 = \left|\sum_{j=1}^{p_i} w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_{ij})\right|^2 \\ &= \sum_{j=1}^{p_i}\sum_{l=1}^{p_i} w_{ij}w_{il}\boldsymbol{Q}^i, \qquad i=1, 2, ..., N,\end{aligned} \tag{6}$$

where w_{ij} is the local linear reconstruction weight between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$. $\boldsymbol{Q}^i$, a symmetric and semi-positive covariance matrix, can be written as

$$\boldsymbol{Q}^i = (\boldsymbol{x}_i - \boldsymbol{x}_{ij})(\boldsymbol{x}_i - \boldsymbol{x}_{il}), \tag{7}$$

where $\boldsymbol{x}_{ij}$ and $\boldsymbol{x}_{il}$ are two different neighbors of $\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_{ij} \neq \boldsymbol{x}_{il}$ (j, $l \leq p_i$, $j \neq l$). Thus, the reconstruction error function can be minimized as

$$\varepsilon(\boldsymbol{W}) = \arg\min \sum_{i=1}^{N} \left|\boldsymbol{x}_i - \sum_{j=1}^{p_i} w_{ij}\boldsymbol{x}_{ij}\right|^2, \tag{8}$$

where w_{ij} is subject to: $\sum_{j=1}^{p_i} w_{ij} = 1$ if $\boldsymbol{x}_{ij} \in \mathrm{gnei}(\boldsymbol{x}_i)$; $w_{ij}=0$ if $\boldsymbol{x}_{ij} \notin \mathrm{gnei}(\boldsymbol{x}_i)$ (gnei($\boldsymbol{x}_i$) denotes the globular neighborhood of $\boldsymbol{x}_i$). Calculate the value of w_{ij} based on the Lagrange multiplier and then the local reconstruction weight matrix $\boldsymbol{W}$ (=$[w_{ij}]_{N\times N}$) can be obtained. $\boldsymbol{W}$ is invariant to rotation, translation, and scale.

Step 3: Compute the low-dimensional embedding Y for X via the local reconstruction weight matrix $\boldsymbol{W}$ and the p_i nearest neighbors of each current data. For realizing shape preserving mapping, minimize the embedding cost function as follows:

$$\varepsilon(Y) = \arg\min \sum_{i=1}^{N} \left|\boldsymbol{y}_i - \sum_{j=1}^{p_i} w_{ij}\boldsymbol{y}_{ij}\right|^2, \tag{9}$$

where $\boldsymbol{y}_i$ denotes the mapping of $\boldsymbol{x}_i$ on the low-dimensional embedding manifold and $\boldsymbol{y}_{ij}$ (j=1, 2, ..., p_i) denote the nearest neighbors of $\boldsymbol{y}_i$. Y is subject to two constraints: $\sum_{i=1}^{N} \boldsymbol{y}_i = \boldsymbol{0}$ and $\sum_{i=1}^{N} \boldsymbol{y}_i \boldsymbol{y}_{ij}^{\mathrm{T}} / N = \boldsymbol{I}$. Then, Eq.(9) can be rewritten in the matrix form:

$$\varepsilon(Y) = \arg\min \sum_{i=1}^{N}\sum_{j=1}^{N} m_{ij}\boldsymbol{y}_i^{\mathrm{T}}\boldsymbol{y}_{ij}, \tag{10}$$

where $\boldsymbol{M}$ (=$[m_{ij}]_{N\times N}$) is a sparse, symmetric and positive semi-definite cost matrix, given by $\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W})$. At last, based on the Rayleigh-Ritz theorem, Eq.(10) is performed by finding the eigenvectors with the smallest (nonzero) eigenvalues of the cost matrix $\boldsymbol{M}$. The flow of the GNLLE algorithm is as follows:

Algorithm 1 Globular neighborhood-based locally linear embedding (GNLLE)

```
Input: $X$ is an image set in the high-dimensional space, $X=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N \mid \boldsymbol{x}_i \in \mathbb{R}^D, i=1, 2, ..., N\}$; $r$ is a globular radius; $\Delta r$
    is a radius increment; $d$ ($<<D$) is the dimensionality of
    the low-dimensional embedding manifold.
Output: $Y=\{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N \mid \boldsymbol{y}_i \in \mathbb{R}^d, i=1, 2, ..., N\}$.
Call a function to construct globular neighborhood:
    (Nei)=CalculateNeighborhood($X$, $r$, $\Delta r$, $d$);
For each $\boldsymbol{x}_i \in X$ do {
  Calculate the reconstruction weight $w_{ij}$;
  Establish the weight matrix $\boldsymbol{W}$;        // Eqs.(6)~(8)}
According to $\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W})$, compute the cost matrix $\boldsymbol{M}$;
Find eigenvectors corresponding to the $d$ (1: $d$+1) eigen-
    values of $\boldsymbol{M}$, and then compose $Y$;   // Eqs.(9) and (10)
Return $Y=\{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N\}$;
Function (Nei)=CalculateNeighborhood($X$, $r$, $\Delta r$, $d$)
Input: omitted.
Output: Nei={gnei($\boldsymbol{x}_i$) | $i$=1, 2, …, $N$}.
Compute distance matrix $\boldsymbol{D}$: $\boldsymbol{D}$=dist($X$, $X$);       // Eq.(5)
Sort $\boldsymbol{D}$ in ascending order by column, and then the results are
    stored in matrix $\boldsymbol{D}'$ and their corresponding indexes are
    stored in matrix $\boldsymbol{T}$; i.e., ($\boldsymbol{D}'$, $\boldsymbol{T}$)=sort($\boldsymbol{D}$);
For each $\boldsymbol{x}_i \in X$ do {
  $r_{\min}$=min($\boldsymbol{x}_{ik}$, $\boldsymbol{x}_i$), $k$=1, 2, ..., $N$, $k \neq i$;
  $r'=r_{\min}$;       // $r'$ is the current radius
  $p_i$=1;       // $p_i$ is the neighbor number of $\boldsymbol{x}_i$
  While $r'<r$ do {
    If $\boldsymbol{D}'(\boldsymbol{x}_{ij}, \boldsymbol{x}_i) \leq r'$ then {
      $p_i=p_i+1$;
      Put $\boldsymbol{x}_{ij}$ into the neighborhood of $\boldsymbol{x}_i$:gnei($\boldsymbol{x}_i$)=$\boldsymbol{T}(\boldsymbol{x}_{ij}, \boldsymbol{x}_i)$;}
    Else { $r'=r'+\Delta r$; }} }
Return Nei={gnei($\boldsymbol{x}_1$), gnei($\boldsymbol{x}_2$), …, gnei($\boldsymbol{x}_N$); }
```

In terms of the computational complexity of the GNLLE algorithm, we compute only the additional complexity generated by the construction of globular neighborhood. Since the function CalculateNeighborhood() includes a dual loop, we calculate its time consumption as follows:

$$\lim_{N\to\infty}\left(\sum_{i=1}^{N}\sum_{r_{\min}}^{r}1\Big/N\right)=\lim_{N\to\infty}\frac{N(r-r_{\min})}{N}=r-r_{\min}. \quad (11)$$

Therefore, the GNLLE algorithm has an additional computational complexity of $O(N)$.

Performance comparison between GNLLE and LLE

To validate the effectiveness of the proposed GNLLE, we designed two experiments to compare the performance between GNLLE and LLE. The Swiss Roll dataset, which is commonly used and has good comparability for different dimensionality reduction algorithms, was used in the experiments.

Experiment 1 The performance comparison between GNLLE and LLE on a dense dataset (N=2000)

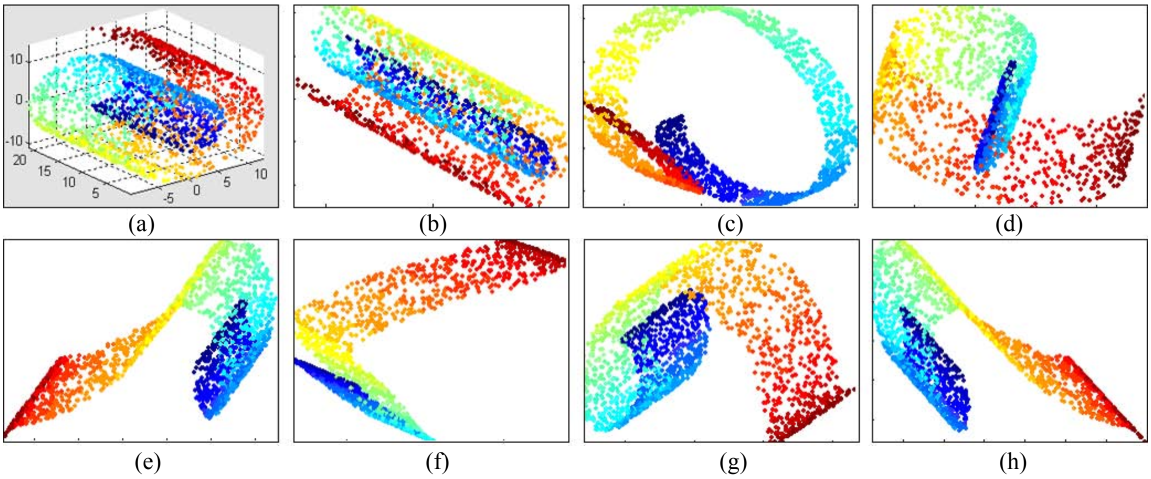
Fig.4 shows some results of GNLLE and LLE on a dense dataset. LLE can distinguish different samples and preserve their local topological structure, although when k=12, its result is slightly inferior to those when k=13 and 14 (Figs.4b~4d). Figs.4e~4h illustrate the performance of GNLLE when r=$r_{\min}$+2.8, $r_{\min}$+3.0, $r_{\min}$+3.2, and $r_{\min}$+3.4, respectively, indicating that GNLLE not only has the capability of distinguishing the samples correctly but also embodies the aggregation ability of the similar samples on the embedding manifold. Therefore, GNLLE can obtain better performance on dense datasets, just as LLE.

Experiment 2 The performance comparison between GNLLE and LLE on a sparse dataset (N=200)

Fig.5 presents some results of GNLLE and LLE on a sparse dataset. Figs.5b~5d show that LLE cannot correctly distinguish different samples when k=12, 13 and 14. But in GNLLE, with the increase of the radius r (r=$r_{\min}$+19, $r_{\min}$+21, $r_{\min}$+23, and $r_{\min}$+25), the samples of the same color can be grouped into a homogeneous region on the embedding manifold to some extent (Figs.5e~5h). Thus, GNLLE can be used for dimensionality reduction on sparse datasets and outperforms LLE in this regard.

Some conclusions can be drawn as follows:

1. Different relationships between the number of the nearest neighbors and the size of the dataset. In LLE, since a fixed number of samples are added into the neighborhood of the current data with the increase of k (which is proportional to the size of the dataset; i.e., if k increases by 1, the total number of the added samples is equal to the sample number in the dataset), the samples that have no contribution to the reconstruction of the current data on the embedding manifold may be blindly regarded as the nearest neighbors, which will easily lead to a failure. However, in GNLLE, with the increase of Δr, only the samples within the globular neighborhood can be viewed as the nearest neighbors of the current data, so the



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig.4 The performance comparison between GNLLE and LLE (N=2000)

(a) Distribution map; (b) k=12; (c) k=13; (d) k=14; (e) r=$r_{\min}$+2.8; (f) r=$r_{\min}$+3.0; (g) r=$r_{\min}$+3.2; (h) r=$r_{\min}$+3.4. N, the data number; k, the neighbor number; r, the globular radius; $r_{\min}$, the shortest distance. (b)~(d) LLE, (e)~(h) GNLLE

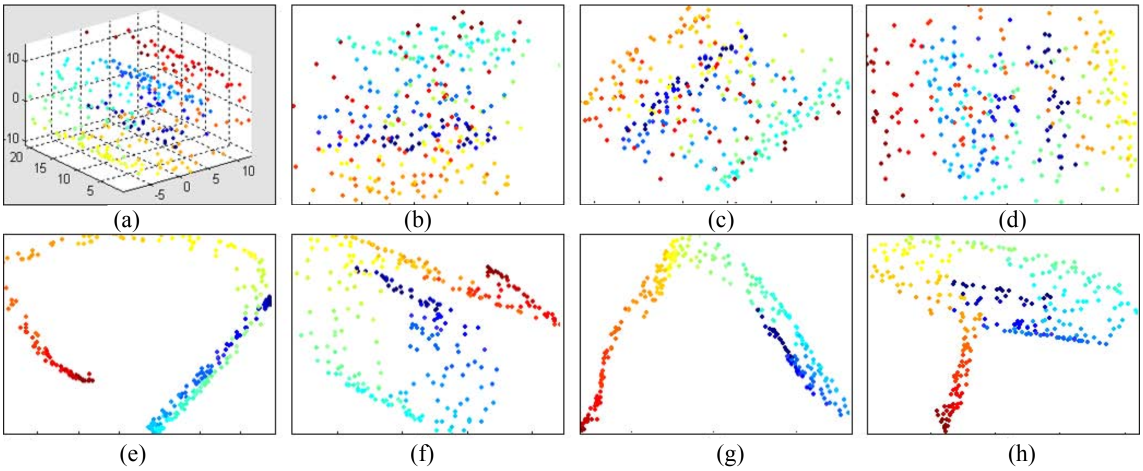


Fig.5 The performance comparison between GNLLE and LLE (N=200)

(a) Distribution map; (b) k=12; (c) k=13; (d) k=14; (e) r=r_{min}+19; (f) r=r_{min}+21; (g) r=r_{min}+23; (h) r=r_{min}+25. N, the data number; k, the neighbor number; r, the globular radius; r_{min}, the shortest distance. (b)~(d) LLE, (e)~(h) GNLLE

samples (especially some outliers) far away from the current data will be excluded (i.e., the number of the added samples has no direct relationships with the size of the dataset). Therefore, compared with LLE, GNLLE has a stronger anti-noise ability.

2. Different stabilities of preserving the local topological structure of the dataset. In LLE, since k is a fixed parameter, its small changes may cause performance degradation. However, in GNLLE, r is an adjustable parameter; i.e., its value can be decreased when dealing with dense datasets, and increased when handling sparse datasets. Therefore, compared with the strong parameter k in LLE, the parameter r in GNLLE is a weak one. Moreover, different from the neighbor selection in LLE, GNLLE does not blindly absorb the candidate data when searching the nearest neighbors within the globular neighborhood, so it embraces a better stability of preserving the local topological structure of the dataset.

GNLLE optimizes the processes of both neighborhood construction and neighbor selection based on a geometric principle, which is intuitive and easy to understand, and overcomes the shortcomings of being sensitive to noise and incapable of handling sparse datasets in LLE to some extent. GNLLE can better maintain certain neighbor relationships between adjacent data, but cannot alleviate the distortion of the overall topological structure of the dataset (i.e., two faraway data in the high-dimensional space are close to each other on the corresponding low-dimensional embedding manifold) in the nonlinear dimensionality reduction approaches based on locally linear embedding (the distortion may be more serious on sparse datasets). The reasons can be summarized as follows:

1. To preserve the local topological structure of the dataset, the size of the neighborhood for a sparse dataset is larger than that for a dense dataset. Those data that have no contribution to the reconstruction of the current data are not treated as the nearest neighbors in GNLLE; nevertheless, when dealing with a sparse dataset, for avoiding the loss of too much useful information, the radius of the globular neighborhood should be enlarged. This may distort the overall topological structure of the dataset.

2. In many cases, the intrinsic structure of the dataset with curved surfaces bears folding or bending in the high-dimensional space. The spacing between two different curved surfaces is very small, but the distance between two data from different curved surfaces is the smallest and these two data may be added into the same neighborhood during neighbor selection; this will lead to the distortion of the overall topological structure when reconstructing the current data on the embedding manifold (the distortion may be more serious on sparse datasets).

The distortion attributed to the first reason can be reduced through choosing a suitable globular radius based on experience and continuous practice. However, the distortion attributed to the second reason is related to the pairwise similarity calculation adopted by GNLLE; it also exists in most nonlinear dimensionality reduction approaches, such as LLE and ISOMAP. Since such similarity calculation is not completely consistent with human perception, it cannot shorten the semantic gap existing in image semantic understanding and its applications. As a result, for eliminating the distortion of the whole data manifold, the nearest neighbors identified by GNLLE should be reselected and the obtained neighborhood should be updated as well.

AN IMPROVED GNLLE ALGORITHM USING PATH CLUSTERING

Reselection of the nearest neighbor

Based on the assumption of local linearity on a nonlinear embedding manifold, being the same as LLE, GNLLE obtains a dimensionality reduction mapping of maintaining the local configuration between two adjacent data but not the overall topological structure of the dataset. In order to obtain a more accurate semantic similarity required for image semantic understanding and its applications, GNLLE should be developed to keep the whole data manifold completely. Path-based clustering, a psychophysically plausible similarity measure proposed by Fischer *et al.*(2001), had been applied in texture image clustering (Fischer *et al.*, 2001; Fischer and Buchmann, 2003). This method is based on empirical observation. The correlations between data can be described by local homogeneity or connectivity (some mediating 'small edge elements'). Enlightened by path-based clustering, we propose an improved GNLLE algorithm, the globular neighborhood and path clustering based locally linear embedding algorithm (GNPCLLE). GNPCLLE can reselect the nearest neighbors (which are identified by GNLLE) and further update the globular neighborhood by establishing the correlation matrix and redefining the distance measure.

Suppose that N points are distributed in a D-dimensional Euclidean space E^D, and P is a set of all paths between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$, $P(\boldsymbol{x}_{ij}, \boldsymbol{x}_i)=\{p_1(\boldsymbol{x}_{ij}, \boldsymbol{x}_i), p_2(\boldsymbol{x}_{ij}, \boldsymbol{x}_i), \ldots, p_l(\boldsymbol{x}_{ij}, \boldsymbol{x}_i)\}$. Assume that there exist q_k small edge elements on each path $p_k(\boldsymbol{x}_{ij}, \boldsymbol{x}_i)$ (k=1, 2, …, l) which are regarded as the nearest neighbors of the current data on the low-dimensional embedding manifold. We formulate the data correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$ as follows:

$$s(\boldsymbol{x}_{ij}, \boldsymbol{x}_i) = \min\{\max_{1\le e\le q_1+1} d_e, \max_{1\le e\le q_2+1} d_e, \ldots, \max_{1\le e\le q_k+1} d_e, \ldots, \max_{1\le e\le q_l+1} d_e\}, \tag{12}$$

and its abbreviated form is

$$s(\boldsymbol{x}_{ij}, \boldsymbol{x}_i) = \min_{k=1,2,\ldots,l}\{\max_{1\le e\le q_l+1} d_e\}, \tag{13}$$

where e denotes each edge on one path (which is composed of q_k small edge elements) between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$, and d_e represents the weight corresponding to e (equal to the distance between its two endpoints). That is, first judge the weights of all edges on each path between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$ and take the maximum weight as the data correlation of this path, and then according to the principle that the data correlation between two points is inversely proportional to the distance between them, the minimum data correlation in all paths is chosen as the data correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$ (i.e., the smaller the $s(\boldsymbol{x}_{ij},\boldsymbol{x}_i)$ is, the larger the data correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$ will be). Thus, the correlation matrix of the dataset can be defined as follows:

$$\boldsymbol{S}_{N\times N} = \begin{bmatrix} s(\boldsymbol{x}_{11}, \boldsymbol{x}_1) & s(\boldsymbol{x}_{21}, \boldsymbol{x}_2) & \cdots & s(\boldsymbol{x}_{N1}, \boldsymbol{x}_N) \\ s(\boldsymbol{x}_{12}, \boldsymbol{x}_1) & s(\boldsymbol{x}_{22}, \boldsymbol{x}_2) & \cdots & s(\boldsymbol{x}_{N2}, \boldsymbol{x}_N) \\ \vdots & \vdots & & \vdots \\ s(\boldsymbol{x}_{1N}, \boldsymbol{x}_1) & s(\boldsymbol{x}_{2N}, \boldsymbol{x}_2) & \cdots & s(\boldsymbol{x}_{NN}, \boldsymbol{x}_N) \end{bmatrix}. \tag{14}$$

As $s(\boldsymbol{x}_{ij},\boldsymbol{x}_i)=s(\boldsymbol{x}_i,\boldsymbol{x}_{ij})$ and $s(\boldsymbol{x}_{ii},\boldsymbol{x}_i)=0$, the correlation matrix $\boldsymbol{S}$ is symmetric.

It will take much time to calculate the data correlations on all possible paths between two data. One optional scheme to reduce the computational complexity is that, first using GNLLE to obtain the candidate neighbors identified by the globular neighborhood, then computing the data correlations on the paths (which are a subset of all possible paths) connected by these candidate neighbors that are regarded as small edge elements, and finally updating the nearest neighbors of the current data based on their data correlations.

GNPCLLE algorithm

To implement GNPCLLE, we insert one critical step between steps 1 and 2 in GNLLE. In this added step, the correlations between the current data and the other data on the paths (which are composed of the candidate neighbors) are calculated (Eq.(13)), and the nearest neighbors of the current data are reselected according to their correlations between each other. The flow of the GNPCLLE algorithm is listed below:

Algorithm 2 Globular neighborhood and path clustering based locally linear embedding (GNPCLLE)

```
Input: $X$ is an image set in the high-dimensional space, $X=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \mid \boldsymbol{x}_i \in \mathbb{R}^D, i=1, 2, ..., N\}$; $r$ is a globular radius; $\Delta r$
    is a radius increment; $d$ ($<<D$) is the dimensionality of the
    low-dimensional embedding manifold.
Output: $Y=\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N \mid \boldsymbol{y}_i \in \mathbb{R}^d, i=1, 2, ..., N\}$.
1  Call a function to construct globular neighborhood:
     (Nei)=CalculateNeighborhood($X$, $r$, $\Delta r$, $d$);
     // the same as in GNLLE
2  Call a function to reselect nearest neighbors:
     (Nei′)=UpdateNeighbor(Nei);
...   // lines 3~7 are the same as lines 2~6 in GNLLE
8  Return $Y=\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$;
Function   (Nei′)=UpdateNeighbor(Nei)
Input: Nei is a set includes the candidate neighbors, i.e.,
       Nei={gnei($\boldsymbol{x}_1$), gnei($\boldsymbol{x}_2$), …, gnei($\boldsymbol{x}_N$)}.
Output: Nei′={gnei′($\boldsymbol{x}_i$) | $i$=1, 2, …, $N$}.
9   For each $\boldsymbol{x}_i \in X$ do {
10    For each $\boldsymbol{x}_{ij} \in$ gnei($\boldsymbol{x}_i$) do {
11      For each $\boldsymbol{x}_{ijk} \in$ gnei($\boldsymbol{x}_{ij}$) do {
12        $d_1$=dist($\boldsymbol{x}_i$, $\boldsymbol{x}_{il}$);   // $\boldsymbol{x}_{il}$ is the farthest neighbor of $\boldsymbol{x}_i$
13        $d_2$=dist($\boldsymbol{x}_{ijk}$, $\boldsymbol{x}_{ij}$);
14        If $d_2<d_1$ then flag=1; else flag=0;
15        If $\boldsymbol{x}_{ijk} \notin$ gnei($\boldsymbol{x}_i$) then flag1=1; else flag1=0;
16        If $\boldsymbol{x}_{ijk} \neq \boldsymbol{x}_i$ then flag2=1; else flag2=0;
17        If flag==1&&flag1==1&&flag2==1 then {
18          Delete $\boldsymbol{x}_{il}$ from gnei($\boldsymbol{x}_i$);
19          Insert $\boldsymbol{x}_{ijk}$ into gnei($\boldsymbol{x}_i$); }
20        Else {take the next neighbor from gnei($\boldsymbol{x}_i$); }}}
21      gnei′($\boldsymbol{x}_i$)=gnei($\boldsymbol{x}_i$); }
22 Return Nei′={gnei′($\boldsymbol{x}_1$), gnei′($\boldsymbol{x}_2$), …, gnei′($\boldsymbol{x}_N$); }
```

In terms of the computational complexity of the GNPCLLE algorithm, we compute only the additional complexity generated by the calculation of data correlation and the reselection of nearest neighbors. Since UpdateNeighbor() includes a triple loop, we calculate the time consumption as follows:

$$\lim_{N\to\infty}\left(\sum_{i=1}^{N}\sum_{j=1}^{p_i}\sum_{k=1}^{p_{ij}}1\Big/N\right)=\lim_{N\to\infty}\frac{Np_ip_{ij}}{N}=p_ip_{ij}, \quad (15)$$

where p_i and p_{ij} denote the neighbor numbers of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$, respectively (p_i, $p_{ij}<<N$). Therefore, the GNPCLLE algorithm has an additional computational complexity of $O(N)$.

Performance comparison among GNPCLLE, GNLLE, PCLLE and LLE

To validate the effectiveness of GNPCLLE, we designed one experiment to compare the performance among GNPCLLE, GNLLE, PCLLE (an improved LLE algorithm based on path-based clustering) and LLE. The Swiss Roll dataset was used in the experiment.

Since the topological structure of the Swiss Roll dataset bears some folding or bending (like spirals), the curved surfaces may be close to each other on the embedding manifold. If we apply the distance measure based on pairwise similarity calculation, some data from different curved surfaces will be assigned into the same neighborhood mistakenly. Fig.6a shows the result of GNLLE before using data correlation to reselect the nearest neighbors. We can see that a distortion of the whole data manifold occurred. Fig.6b gives the result of GNPCLLE after using data correlation. It is obvious that GNPCLLE not only correctly distinguished the data but also well preserved the overall topological structure of the dataset. Figs.6c and 6d present the results of PCLLE and LLE after and before applying data correlation to reselect the nearest neighbors, and similar results to GNPCLLE

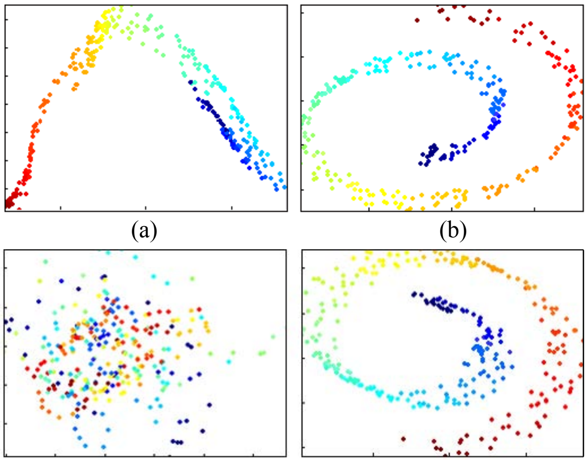

(a) (b) (c) (d)

Fig.6 The performance comparison among (a) GNLLE, (b) GNPCLLE, (c) LLE, and (d) PCLLE (N=200)

and GNLLE were obtained. In addition, from the performance comparison between GNPCLLE and PCLLE (Figs.6b and 6d), a stronger aggregation ability can be achieved by improving GNLLE than directly improving LLE.

Some conclusions can be drawn as follows:

1. Using data correlation to reselect the nearest neighbors can preserve the overall topological structure of the dataset completely. Since in the calculation of data correction the coordinates of the data on curved surfaces are considered, the overall topological structure of the dataset can be effectively maintained. By using path-based clustering similarity calculation, the reselection of the candidate neighbors recognized by GNLLE avoids topological structure distortion in GNLLE, and results in good expansibility. Data correlation can be utilized to improve the other nonlinear dimensionality reduction algorithms based on pairwise similarity calculation (e.g., PCLLE is an improved algorithm based on LLE).

2. Using data correlation to reselect the nearest neighbors can improve the performance of LLE on sparse datasets to some extent. Applying path-based clustering to GNPCLLE enables gathering the homogenous data on the embedding manifold, and thus optimizes the irregular neighborhood constructed by the nearest neighbors in LLE and improve LLE's performance on sparse datasets as well.

As discussed above, in GNLLE, maintaining the overall topological structure of the dataset is not fully considered when the high-dimensional dataset is mapped into a manifold with a lower dimension, thus inevitably causing distortion of the whole data manifold. We introduce path-based clustering into GNLLE and present GNPCLLE to reselect the candidate neighbors within the globular neighborhood, via calculating the correlations between the data and further revealing the true relationships of the data on the embedding manifold, even for those datasets with mapping distortions. Note that GNPCLLE has similarities with LSML (locally smooth manifold learning) proposed by Dollár *et al.*(2006; 2007). For example, both of them aim to recover the manifold structure, find the manifold through the selection of nearest neighbor and the definition of graph-based distance measure, and are robust algorithms, able to obtain satisfying results on the dataset with mapping distortions to some degree. However, there still exist great differences between GNPCLLE and LSML, mainly in the following aspects:

1. GNPCLLE and LSML differ in motivation and potential applicability. GNPCLLE can be grouped as a spectral embedding method (Weinberger and Saul, 2004), where manifold learning is viewed as finding a structure preserving embedding. But in LSML, manifold learning is regarded as traversing the low-dimensional manifold. Furthermore, these two algorithms focus mainly on different application areas: GNPCLLE emphasizes image understanding and its applications, such as image object recognition, image clustering and classification, and semantic-based image retrieval; LSML concentrates on image and video processing, such as tangent distance estimation, video compression, and motion transfer.

2. GNPCLLE and LSML exploit similar ideas in neighbor selection (in GNPCLLE, the nearest neighbors of the current data are recognized by a globular radius; in LSML, besides the *k*-nearest-neighbor, a threshold (e.g., a maxDist value) is preset to restrict the searching region of the nearest neighbors), but are based on different mathematical principles. The former identifies the nearest neighbors via constructing a globular neighborhood from geometric intuition and the variation of the data can be revealed by data correlation, while the latter uses a wrapping function to generate the nearest neighbors and to capture the change modes of data.

3. GNPCLLE and LSML share different schemes for distance measure calculation. Specifically, considering that a high-dimensional dataset may have special shapes, for maintaining the whole data manifold, these two algorithms do not compute the distance between two adjacent data simply on an Euclidean distance. Path-based clustering is introduced into GNPCLLE and the geodesic distance is used in LSML. Different from GNPCLLE, LSML is still an approach based on pariwise similarity calculation (which prefers dense datasets) (Hofmann and Buhmann, 1997); although the overall topological structure of the dataset can be maintained, LSML may fail to find the correct data patterns on the low-dimensional manifold for a sparse dataset, especially when dealing with the dataset with spirals, circles or a tube like data distribution. In contrast, GNPCLLE can

extract the hidden topological structure from the dataset on the manifold in a robust way, even for the dataset with a sparse distribution or noise. Consequently, in real-world applications (especially in the network environment), despite the complexity and diversification of Web images, GNPCLLE can achieve satisfying results.

We designed an experiment for comparing GNPCLLE with LSML and ISOMAP that apply the geodesic distance. As shown in Fig.7, GNPCLLE outperformed LSML and ISOMAP when handling a sparse dataset with noise (the performance comparison between LSML and ISOMAP was analyzed by Dollár *et al.*(2006; 2007)).

Note that GNPCLLE is not robust enough to be well suited for optimizing image features under all conditions. That is, we use path-based clustering to do image feature optimization for the high-dimensional dataset with curved surfaces, so the performance of GNPCLLE is probably inferior to that of GNLLE or LLE when dealing with the dataset with an even distribution or without a heavy mapping distortion. Meanwhile, due to a longer computation time for data correlation calculation, GNPCLLE may not outperform GNLLE or LLE in terms of both efficiency and precision. Additionally, for a very noisy dataset, two data may be connected by unsuitable small edges, causing the results of GNPCLLE to be unreliable.

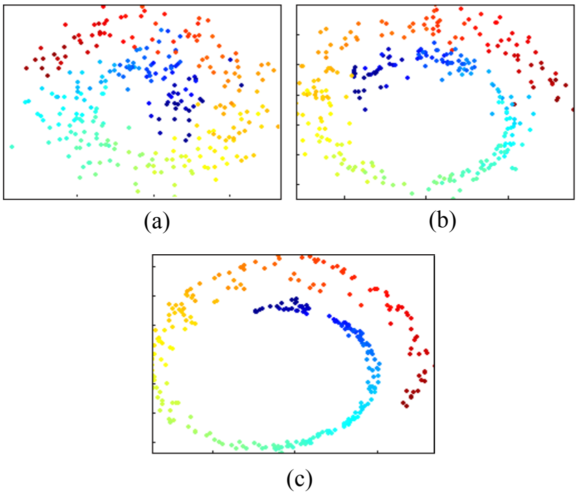


Fig.7 The performance comparison among (a) ISOMAP, (b) LSML, and (c) GNPCLLE (*N*=200, random Gaussian noise with μ=0 and σ=0.5)

APPLICATION EXAMPLES

To further verify the feasibility and effectiveness of the proposed GNLLE and GNPCLLE, we designed some experiments to compare our algorithms with PCA and LLE. Hardware platform: T2350 1.86 GHz CPU and 2 GB main memory; software tools: mixed programming of VC++6.0 and MATLAB 7.0.

Experiment 3 (Natural image clustering) Natural image clustering is unsupervised learning. It is aimed at grouping the image data in an image set into different semantic categories according to the clustering principle that the extra-class difference is as large as possible while the intra-class difference is as small as possible. The image set used here includes 539 images downloaded from the Internet, which belong to three semantic categories (Fig.8 gives some image samples). The numbers of images in the categories Beach (C1), Lawn (C2) and Sunrise/Sunset (C3) are 133, 186 and 220, respectively. Since the sizes of the original images are different, we performed a normalization preprocessing before feature extraction (the size of the normalized image is 126×189 or 189×126). After analyzing the characteristics of these natural images, we extracted a 40-dimensional raw feature vector for each image data, which includes color histogram (24), wavelet texture (3), shape invariant moment (7), and edge histogram (6) (note that for speeding up the execution, we did not extract text features from the related Web pages). Since the concept of 'semantic categories' has some fuzziness, we adopted the fuzzy c-means clustering (FCM) algorithm (Dunn, 1973; Bezdek, 1981) to realize a fuzzy clustering; i.e., the images are grouped into the corresponding categories according to their membership degrees. To quantitatively evaluate the implementation results of the algorithms, the accuracy (AC) is used to evaluate the clustering performance and defined as follows:

$$\mathrm{AC}=P(i)\,/\,T(i), \tag{16}$$

where $T(i)$ denotes the total number of the images belonging to the ith category, and $P(i)$ denotes the number of the images correctly grouped into the ith category. Table 1 and Fig.9 show the experimental results.

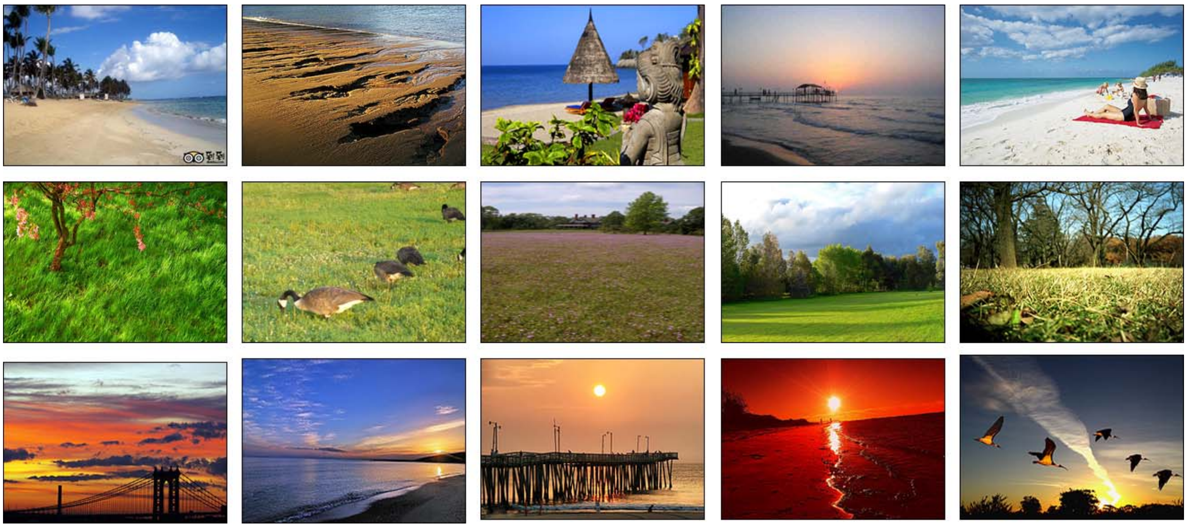


Fig.8 Some image samples (each row corresponds to one semantic category and they are Beach, Lawn, and Sunrise/Sunset from top to bottom)

Table 1 Comparison of clustering performance among four algorithms when mapping into 2D~7D space

Algorithm	Category	Accuracy						
		2D	3D	4D	5D	6D	7D	Average
PCA	C1	0.7368	0.9700	0.9625	0.9549	0.9624	0.9549	0.9236
	C2	0.5591	0.6398	0.6935	0.7312	0.7796	0.7688	0.6953
	C3	0.7955	0.8773	0.9000	0.8864	0.8545	0.7864	0.8500
	Average	0.6971	0.8290	0.8520	0.8575	0.8655	0.8367	0.8230
LLE	C1	0.6917	0.8045	0.7744	0.8045	0.8421	0.7594	0.7794
	C2	0.8763	0.8118	0.8333	0.8387	0.8011	0.8763	**0.8396**
	C3	0.9318	0.9420	0.9369	0.9227	0.9318	0.9107	0.9293
	Average	0.8333	0.8528	0.8482	0.8553	0.8583	0.8488	0.8494
GNLLE	C1	0.7444	0.8421	0.8271	0.8120	0.8571	0.9173	0.8333
	C2	0.8548	0.8011	0.8441	0.8387	0.7957	0.7419	0.8127
	C3	0.9409	0.9455	0.9318	0.9227	0.9273	0.9273	**0.9326**
	Average	0.8467	0.8629	0.8677	0.8578	0.8600	0.8622	0.8595
GNPCLLE	C1	0.9774	0.9774	0.9850	0.9950	0.9950	0.9950	**0.9875**
	C2	0.8871	0.7849	0.7849	0.7957	0.7957	0.7957	0.8073
	C3	0.7727	0.8955	0.8955	0.8000	0.8000	0.8000	0.8273
	Average	0.8791	0.8859	0.8885	0.8636	0.8636	0.8636	**0.8740**

C1: Beach; C2: Lawn; C3: Sunrise/Sunset. The bold numbers are the best results

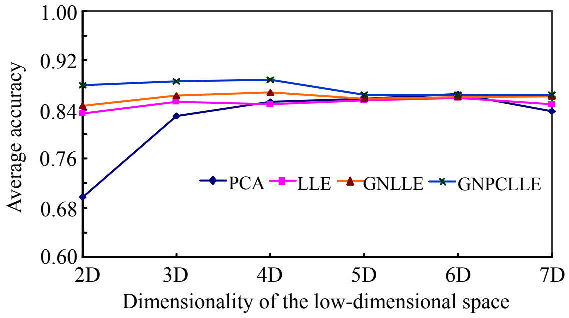


Fig.9 Comparison of the average accuracy among four algorithms when mapping into 2D~7D space

As can be seen from Table 1, for clustering three semantic categories, using GNPCLLE to optimize the image feature resulted in the highest average AC (0.8740) while PCA had the lowest one (0.8230), and the average AC of GNLLE and LLE was 0.8595 and 0.8494, respectively. It can be observed that the four algorithms had different performance on different categories. For example, for Beach (C1), GNPCLLE obtained a high accuracy (0.9875) but failed to outperform GNLLE and LLE on the clustering of Lawn (C2) and Sunrise/Sunset (C3). GNLLE obtained the best result when clustering Sunrise/Sunset (0.9326) whereas it was slightly inferior to LLE when clustering Lawn (0.8127 vs. 0.8396). GNPCLLE and PCA had good results for Beach (0.9875 and 0.9236, respectively), but GNLLE and LLE had good results for Sunrise/Sunset (0.9326 and 0.9293, respectively). None of the four algorithms achieved a high AC on the clustering of Lawn.

The reasons for such results may be as follows: among the three semantic categories, Breach achieved a relatively satisfying result due to the fact that the images belonging to this category have distinct visual characteristics, such as white cloud, blue sky and sea water. However, the Lawn images have the diversity of color and texture features, so the average clustering accuracy of this category was lower than those of the other two categories. In addition, many images in the image set have parts that are visually similar, such as the regions corresponding to sky, water, tree, and sand, so these images are easily grouped into the wrong category.

Fig.9 shows the comparison of clustering performance among four algorithms when mapping into a low-dimensional space with different dimensionalities. Obviously, GNPCLLE is superior to GNLLE and LLE for every dimensionality (from 2D to 7D), and the lower the dimensionality is, the wider the disparity range will be. During the changes of feature dimensionality, PCA shows a significant decrease (especially when the features are mapped into the 2D space) whereas GNPCLLE, GNLLE and LLE have no critical fluctuations. Therefore, compared with the linear ones, nonlinear dimensionality reduction approaches can better optimize the feature vectors that need to be reduced sharply; this is suitable for dealing with image data in the network environment (e.g., Web images).

Experiment 4 (Erotic image recognition) Erotic image recognition is essentially a binary classification problem. The image set used here includes 1284 erotic images (including Caucasians, Asians, and Blacks) and 1800 normal images (involving figures, landscapes, buildings, animals, and other topics) downloaded from the Internet. After preprocessing, an 86-dimensional raw feature vector was extracted from each image data (Jiang, 2007), including the number and the ratio of the skin color regions based on a skin mask image (6), chromaticity moment (6), color moment (9), color correlogram (4), wavelet texture (26), shape invariant moment (7), and edge invariant moment (28). To remove data redundancies in the raw features, we performed an optimization process before using a learning machine to classify these images. Herein, we used the approach proposed in Xu *et al.*(2004) to determine the dimensionality of the low-dimensional space; i.e., $\sum_{j=1}^{l}\lambda_j \Big/ \sum_{j=1}^{k}\lambda_j \geq 0.95$, where λ_i is the eigenvalue of the covariance matrix $\boldsymbol{Q}^i$ in Eq.(7). The dimensionality of the optimized feature vectors was set to 12 (the frontal 12 main components have already contained about 99.7% of the raw feature vector information).

Support vector machine (SVM) is a type of machine learning method based on statistical learning theory and Vapnik-Chervonenkis (VC) dimension theory (Vapnik, 1995). SVM pursues structural risk minimization instead of empirical risk minimization, which can guarantee the generalization ability of learning machines. Designed by finding an optimal hyperplane to solve the two-class classification problems, it has been used for various applications under the limited samples in recent years. Our work is to distinguish the erotic images from the normal ones, and hence SVM is an ideal classifier. After a comparison among several SVMs, we chose the sequential minimal optimization algorithm (SMO) (Platt, 1999) for training the SVM due to its fast implementation. The radial basis function was used here as the SVM kernel function. We designed three tests and each time the images were randomly divided into a training set (including 1164 erotic and 1600 normal images) and a test set (including 120 erotic and 200 normal images). The accurate rate (ACR) and recall rate (RER) are used as performance evaluations and are defined as follows (Jiang, 2007):

$$\mathrm{ACR} = (P_{\mathrm{nu}} + P_{\mathrm{no}}) / (T_{\mathrm{nu}} + T_{\mathrm{no}}), \tag{17}$$

$$\mathrm{RER} = P_{\mathrm{nu}} / T_{\mathrm{nu}}, \tag{18}$$

where P_{nu} and P_{no} denote the numbers of the correctly recognized erotic images and normal ones, respectively, and T_{nu} and T_{no} denote the total numbers of the erotic images and the normal ones in the test set, respectively. Table 2 shows the experimental results.

Table 2 Comparison of the recognition performance of SVM after image feature optimization using four algorithms

Test	Algorithm	ACR	RER	Time (s)
1	PCA	0.6656	0.5833	3.186
	LLE	0.8313	0.8500	6.188
	GNLLE	0.8718	**0.9250**	11.323
	GNPCLLE	**0.8875**	0.8750	16.580
2	PCA	0.7063	0.6167	4.216
	LLE	0.8250	0.8500	7.879
	GNLLE	0.8844	**0.9417**	12.957
	GNPCLLE	**0.8938**	0.9250	16.953
3	PCA	0.6906	0.6750	3.063
	LLE	0.8594	0.8917	6.167
	GNLLE	0.8875	**0.9417**	11.309
	GNPCLLE	**0.9031**	0.9250	16.237
Average	PCA	0.6875	0.6250	3.488
	LLE	0.8386	0.8639	6.745
	GNLLE	0.8894	**0.9361**	11.863
	GNPCLLE	**0.8948**	0.9083	16.590

ACR: accurate rate; RER: recall rate. The bold numbers are the best results

As shown in Table 2, except PCA, the other three algorithms (GNPCLLE, GNLLE, and LLE) all obtained high RERs. The reason may be that an obvious difference between erotic images and normal ones is mostly based on the bare degree of human skin. During feature extraction, the features of skin color and texture are extracted, which are related to the distinct characteristics of erotic images, so satisfying RERs can be obtained. Among the four algorithms, the optimization of image feature based on GNLLE obtained the highest RER (0.9361), the following were GNPCLLE (0.9083) and LLE (0.8639), and PCA was the poorest (0.6250), which further confirms the assumption that the data distribution of Web images is nonlinear. On the other hand, due to the fact that the contents of normal images in the image set are diversified (e.g., for the close-up face images, some areas' color is easily confused with skin color, and some objects' texture is also similar to skin texture), the recognition of normal images is more challenging than that of erotic images. Therefore, compared with the average RERs, the average ACRs that GNPCLLE, GNLLE and LLE obtained were relatively low.

As for PCA, the execution was fast, but both RER and ACR were small. Since GNLLE needs to construct globular neighborhood, the average execution time was 11.863 s (including the feature optimization based on GNLLE, training and testing the images via SVM). GNPCLLE needs an additional time of 3.717 s to optimize the globular neighborhood and reselect the nearest neighbors. Although the average execution time of GNLLE (11.863 s) was longer than that of LLE (6.745 s) (but shorter than that of GNPCLLE (16.590 s)), from the view of composite consideration, GNLLE was more effective than the other algorithms in terms of both recognition accuracy and execution speed. GNPCLLE achieved the highest ACR (0.8948), but its average RER was lower than that of GNLLE on the recognition of the erotic images (0.9083 vs. 0.9361); this may be caused by the intrinsic topological structure of the image set.

CONCLUSION

In this paper, we propose two novel locally linear embedding algorithms, GNLLE and GNPCLLE, to do image feature optimization for image semantic understanding and its applications. Experimental results showed that our algorithms not only inherit the characteristics of LLE in preserving the local neighbor relationships between adjacent data during dimensionality reduction, but also well reveal the overall topological structure within image data on the low-dimensional embedding manifold. Moreover, these two algorithms both have good aggregation abilities on sparse datasets and strong anti-noise capabilities, which can effectively enhance the performance of image clustering and recognition, especially for Web images whose feature dimensionality should be reduced sharply. Therefore, nonlinear dimensionality reduction approaches based on manifold learning provides a feasible way for image feature optimization.

However, there still exist some limitations in GNLLE and GNPCLLE. For example, in GNLLE, although the globular radius is a flexible parameter, its value may affect the results of dimensionality reduction. An optimum value relies on much experience and practice in most cases. GNPCLLE may suffer from unstable performance due to the distortion extent of the topological data structure when mapping the image data from a high-dimensional space into a low-dimensional manifold. For promoting the efficiency of GNPCLLE, we need to further mining the hidden topological structure within the dataset, explore the extension degree of two adjacent data, and seek the change tendency of the manifold. As a result, fully exploiting and utilizing the prior knowledge of the image data in practical applications can significantly improve the performance of the proposed algorithms. In addition, both algorithms need an extra cost for the construction of globular neighborhood and the selection of nearest neighbors, which results in a longer execution time compared with LLE, and thus how to speed up the implementations of the two algorithms should also receive enough attention. Meanwhile, since both GNLLE and GNPCLLE are unsupervised learning algorithms, to extend them to semi-supervised or supervised learning ones and to improve their generalization abilities are promising work in future.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for valuable comments and useful suggestions, and Piotr Dollar, and Vincent Rabaud for providing us with the LSML algorithm.

References

Abusham, E.E., Ngo, D., Teoh, A., 2005. Fusion of locally linear embedding and principal component analysis for face recognition (FLLEPCA). *LNCS*, **3687**:326-333. [doi:10.1007/11552499]

Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(7):711-720. [doi:10.1109/34.598228]

Belkin, M., Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neur. Inf. Process. Syst.*, **14**:585-591.

Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neur. Comput.*, **15**(6):1373-1396. [doi:10.1162/089976603321780317]

Bellman, R., 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press, New Jersey.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA.

Cao, S.M., Ye, S.W., 2007. A better scaled local line embedding algorithm. *Comput. Simul.*, **24**(5):87-90 (in Chinese).

Chang, J.P., Shen, H.X., Zhou, Z.H., 2004. Unified locally linear embedding and linear discriminant analysis algorithm (ULLELDA) for face recognition. *LNCS*, **3338**: 296-304. [doi:10.1007/b104239]

Datta, R., Joshi, D., Li, J., Wang, J., 2008. Image retrieval: ideas, influences, and treads of the new age. *ACM Trans. Comput. Surv.*, **40**(2):5-60. [doi:10.1145/1348246.1348248]

de Juan, C., Bodenheimer, B., 2004. Cartoon Textures. Proc. Eurographics, ACM SIGGRAPH Symp. on Computer Animation, p.267-276.

de Ridder, D., Kouropteva, O., Okum, O., Pietikäinen, M., Duin, R.P.W., 2003. Supervised locally linear embedding. *LNCS*, **2714**:333-341. [doi:10.1007/3-540-44989-2]

Dollár, P., Rabaud, V., Belongie, S., 2006. Learning to Traverse Image Manifolds. Proc. 12th Annual Conf. on Neural Information Processing Systems. Available from http://vision.ucsd.edu/~pdollar/research/papers/DollarRabaudBelongieNIPS06manifold.pdf [Accessed on Nov. 23, 2009].

Dollár, P., Rabaud, V., Belongie, S., 2007. Non-isometric Manifold Learning: Analysis and an Algorithm. Proc. 24th Int. Conf. on Machine Learning, p.241-248. [doi:10.1145/1273496.1273527]

Dorai, C., Venkatesh, S., 2003. Bridging the semantic gap with computational media aesthetics. *IEEE Multimedia*, **10**(2): 15-17. [doi:10.1109/MMUL.2003.1195157]

Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybern. & Syst.*, **3**(3):32-57. [doi:10.1080/01969727308546046]

Fischer, B., Buchmann, J.M., 2003. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(4):513-518. [doi:10.1109/TPAMI.2003.1190577]

Fischer, B., Zöller, T., Buhmann, J.M., 2001. Path based pairwise data clustering with application to texture segmentation. *LNCS*, **2134**:235-250. [doi:10.1007/3-540-44745-8]

Hofmann, T., Buhmann, J.M., 1997. Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(1):1-14. [doi:10.1109/34.566806]

Hua, Z.G., Wang, X.J., Liu, Q.S., Lu, H.Q., 2005. Semantic Knowledge Extraction and Annotation for Web Images. Proc. 13th Annual ACM Int. Conf. on Multimedia, p.467-470. [doi:10.1145/1101149.1101253]

Jeon, J., Lavrenko, V., Manmatha, R., 2003. Automatic Image Annotation and Retrieval Using Cross-media Relevance Models. Proc. 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.119-126. [doi:10.1145/860435.860459]

Jiang, Z.W., 2007. Research on Content-based Web Image Filter Technology. PhD Thesis, Zhejiang University, Hangzhou, China (in Chinese).

Jin, H., Ooi, B.C., Shen, H.T., Yu, C., Zhou, A.Y., 2003. An Adaptive and Efficient Dimensionality Reduction Algorithm for High-dimensional Indexing. Proc. IEEE 19th Int. Conf. on Data Engineering, p.87-98.

Jolliffe, I.T., 1986. Principal Component Analysis. Springer-Verlag, New York.

Krishnapuram, B., Carin, L., Hartemink, A.J., 2004. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *J. Comput. Biol.*, **11**(2-3):227-242. [doi:10.1089/1066527041410463]

Li, H., Du, S.D., Lu, F., Gao, D.T., 2006. Feature extraction and image reconstruction of video sequence based on nonlinear dimensionality reduction algorithms. *Pattern Recogn. Artif. Intell.*, **19**(5):646-651 (in Chinese).

Platt, J., 1999. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft Research, USA.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**(5500): 2323-2326. [doi:10.1126/science.290.5500.2323]

Saul, L.K., Roweis, S.T., 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.*, **4**(2):119-155. [doi:10.1162/153244304322972667]

Seung, H.S., Lee, D., 2000. The manifold ways of perception. *Science*, **290**(5500):2268-2269. [doi:10.1126/science.290.5500.2268]

Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(12):1349-1380. [doi:10.1109/34.895972]

Sweis, D.L., Weng, J., 1996. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(3):831-836. [doi:10.1109/34.531802]

Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500):2319-2323. [doi:10.1126/science.290.5500.2319]

Turk, M.A., Pentand, A.P., 1991. Face Recognition Using Eigenfaces. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.586-591. [doi:10.1109/CVPR.1991.139758]

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York.

Wang, H., Tian, N., Zhang, X.M., Feng, X.A., Zhao, N., 2003. Alternate Feature Optimization for 3-Class Underwater Target Recognition Based on SVM Classifiers. Proc. IEEE Int. Conf. on Neural Network and Signal Processing, p.144-148. [doi:10.1109/ICNNSP.2003.1279232]

Wang, H.Y., Zheng, J., Yao, Z.A., Li, L., 2006. Application of dimension reduction on using improved LLE based on clustering. *J. Comput. Res. Dev.*, **43**(8):1485-1490 (in Chinese). [doi:10.1360/crad20060826]

Wang, J., Li, J., Wiederhold, G., 2001. Simplicity: semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**(9):947-963. [doi:10.1109/34.955109]

Weinberger, K.Q., Saul, L.K., 2004. Unsupervised Learning of Image Manifolds by Semidefinite Programming. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, **2**:988-995. [doi:10.1109/CVPR.2004.1315272]

Wu, Y.M., Chan, K.L., Wang, L., 2004. Face Recognition Based on Discriminative Manifold Learning. Proc. 17th Int. Conf. on Pattern Recognition, p.171-174. [doi:10.1109/ICPR.2004.1333731]

Xu, Z.J., Yang, J., Wang, M., 2004. A new nonlinear dimensionality reduction for color image. *J. Shanghai Jiao Tong Univ.*, **38**(12):2063-2072 (in Chinese).

Yang, X.M., Wu, W., He, X.H., Chen, M., Xue, L., 2007. Handwritten numeral recognition based on manifold learning. *J. Optoelectron. Las.*, **18**(12):1478-1481 (in Chinese).

Yao, L.Q., Tao, Q., 2005. One kind of manifold learning method for classification. *Pattern Recogn. Artif. Intell.*, **5**:541-545 (in Chinese).

Yin, H.J., 2007. Nonlinear dimensionality reduction and data visualization: a review. *Int. J. Autom. Comput.*, **4**(3): 294-303. [doi:10.1007/s11633-007-0294-y]

Zhang, Q.N., Izquierdo, E., 2006. A Multi-feature Optimization Approach to Object-based Image Classification. Proc. 5th Int. Conf. on Image and Video Retrieval, p.310-319. [doi:10.1007/11788034_32]