Local Neighborhood Embedding for Unsupervised Nonlinear Dimension Reduction

Liangli Zhen, PENG Xi, Dezhong Peng
Machine Intelligence Laboratory, College of Computer Science, Sichuan University, 610065, P. R. China
Email: mi.lianglizhen@gmail.com, pangsaai@gmail.com, dezhongpeng@gmail.com

***Abstract*—The construction of similarity relationship among data points plays a critical role in manifold learning. There exist two popular schemes, i.e., pairwise-distance based similarity and reconstruction coefficient based similarity. Existing works only have involved one scheme of them. These two schemes have different drawbacks. For pairwise-distance based similarity graph algorithms, they are sensitive to the noise and outliers. For reconstruction coefficient based similarity graph algorithms, they need sufficient sampled data and the neighborhood size is sensitive. This paper proposes a novel algorithm, called Local Neighborhood Embedding (LNE), which preserves pairwise-distance based similarity and reconstruction coefficient based similarity for finding the latent low dimensional structure of data. It has following three advantages: Firstly,it is insensitive to the choice of neighborhood size; Secondly, it is robust to the noise; Thirdly, It works well even in under-sampled case. Furthermore, the proposed objective function has a closed-form solution, which means it has a low computational complexity, and the experimental results illustrate that LNE has a competitive performance in dimensionality reduction.**

Index terms - Dimension reduction, manifold learning, similarity graph, unsupervised learning

I. Introduction

In many topics of machine intelligence, information retrieval and computer vision, we are confronted with very high dimensional data, and it is really a challenge for us to understand and analyze these high dimensional data. Fortunately, they are located on a low-dimensional manifold [1]–[3]. Therefore, it is important to find the essential structure hidden in the high-dimensional observation data, and get more compact representations of the original data for higher-level decision making. Following the scheme of machine learning, a variety of dimensionality reduction methods under supervised, semi-supervised, and unsupervised scenarios have been proposed in the last decades. The supervised dimensionality reduction methods include linear discriminant analysis (LDA) [4], maximum margin criterion (MMC) [5], marginal Fisher analysis (MFA) [6], etc. The semi-supervised dimensionality reduction methods include semi-supervised dimensionality reduction (SSDR) [7], semi-supervised discriminant analysis (SDA) [8], etc. The unsupervised methods include principal component analysis (PCA) [9], locality preserving projections (LPP) [10], neighbor preserving embedding (NPE) [11], etc. In this paper, we mainly focus on unsupervised scenario.

One of the most well-known unsupervised dimension reduction methods is principal component analysis (PCA), which works well when the data points lie close to a single linear subspace. However, many real data are situated on several nonlinear subspaces. To reveal the nonlinear hidden structure of high-dimensional data, some works introduced a kernel function into PCA to map the original data into a linear space, namely Kernel PCA [12] . Even though it can solve nonlinear dimensionality reduction problems, much of them do not explicitly find the latent manifold structure of the data set. In addition, Selecting kernel function is another problem in many practical application and it is difficult to solve.

In the past decade, manifold learning has become one of the most important dimension reduction methods. It aims to project the high-dimensional data points into their low-dimensional counterparts, and simultaneously preserves certain geometric properties. Many algorithms based on manifold learning have been proposed. The most famous and efficient three algorithms are Isometric feature mapping (ISOMAP) [1], Laplacian Eigenmaps (LE) [13], and Locally Linear Embedding (LLE) [2], [3]. ISOMAP embeds the neighborhood relationship of data points with geodesic distance. LE is based on computing the low dimensional representation that best preserves locality measured by heat kernel with pairwise distance. LLE computes certain linear reconstruction coefficients to maintain the local geometric properties in the manifold. We can divided them into two schemes, i.e., pairwise-distance based similarity and reconstruction coefficient based similarity for manifold learning. These two schemes have different drawbacks. For pairwise-distance based similarity graph algorithms, they are sensitive to the noise and outliers. For reconstruction coefficient based similarity graph algorithms, they need sufficient sampled data and the neighborhood size is sensitive. A small neighborhood size K may not capture sufficient information about the manifold. On the other hand, a large neighborhood size K will lead to inaccurate representation since the K nearest neighbors of a data point may include faraway inhomogeneous data. In [14], they employ the dual weighted voting method to choose the neighbors in

This work was supported by the National Natural Science Foundation of China under grants 61172180.

The corresponding author Dezhong Peng is with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, No. 24 South Section 1, Yihuan Road, Chengdu, 610065, P. R. China (Email: dezhongpeng@gmail.com).

classification task.

In this paper, we propose an algorithm, namely Local Neighborhood Embedding (LNE), which incorporates pairwise-distance based similarity with reconstruction coefficient based similarity for finding the latent low dimensional structure of data. A key assumption of LNE is that there exists a small neighborhood in which only the points that come from the same manifold lie approximately in a low-dimensional affine subspace. It is different from previous works that use only one similarity scheme. Our algorithm combines these two relationships together. It uses the pairwise distances, from a specified point to its neighbors, to weigh the neighbors' contribution to represent this point. So that it preserves pairwise distance similarity and represent coefficient similarity at the same time. Moreover, the proposed objective function for LNE has a closed-form solution, which means its computing cost is very low, and the demonstrations show that LNE has a competitive performance in dimensionality reduction.

The rest of this paper is organized as follows: Section 2 reviews three popular unsupervised dimensionality reduction algorithms, i.e., ISOMAP, LLE, and LE. Section 3 presentes our LNE algorithm. Section 4 carries out the experiments to confirm the effectiveness of LNE for dimensionality reduction. Finally, Section 5 concludes this work.

II. Related work

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ be a set of n sampled data points from $\mathbb{R}^m$ space. The dimensionality reduction algorithms are proposed to find a set of points $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \mathbb{R}^{d \times n}$, $d \ll m$, to represent the given data set $\mathbf{X}$, such that it keeps the geometrical property.

Isometric feature mapping (ISOMAP) [1] is built on classical Multidimensional Scaling (MDS) [15], but it seeks to preserve the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. Laplacian Eigenmaps (LE) [13] employs a heat kernel function to represent the pairwise distance between two points. Then, computes a low-dimensional representation of the data set by using the laplace Beltrami operator to the similarity graph. They both construct a similarity graph based on a pairwise distance. As another pioneer work of manifold learning, locally linear embedding (LLE) [3] obtains the similarity relationship by using linear reconstruction coefficients among data points. We will review these three classical manifold learning algorithms below.

A. Isometric feature mapping (ISOMAP)

1. Searching the nearest neighbors. Find the K-nearest neighbors for each point $\mathbf{x}_i$ by using the Euclidean distance.

2. Constructing the similarity matrix. Find a matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ whose entries $\mathbf{G}_{ij}$ is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbf{G}_{ij}$ equals the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_j$ is a K-nearest neighbors of $\mathbf{x}_i$, otherwise, enforcing $\mathbf{G}_{ij} = \infty$. Then, for each value of $k = 1, 2, \ldots, n$ in turn, replace all entries $\mathbf{G}_{ij}$ by $\min\{\mathbf{G}_{ij}, \mathbf{G}_{ik} + \mathbf{G}_{kj}\}$. Finally, the matrix $\mathbf{G}$ will contain the shortest path distances between all pairs of the points.

3. Embedding the high-dimensional data. Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(\mathbf{G}) = -\mathbf{HSH}/2$, where $\mathbf{S}$ is the matrix of squared distances $\{\mathbf{S}_{ij} = \mathbf{G}_{ij}^2\}$, $\mathbf{H}$ is the "centering matrix" $\{\mathbf{H}_{ij} = \delta_{ij} - 1/n\}$, $\delta_{ij} = 1$ if $i = j$, otherwise, $\delta_{ij} = 0$, and $\mathbf{v}_p^i$ be the i-th component of the p-th eigenvector. Then, set the p-th component of the d-dimensional coordinate vector y_i equal to $\sqrt{\lambda_p \mathbf{v}_p^i}$.

B. Laplacian Eigenmaps (LE)

1. Searching the nearest neighbors. Find the K-nearest neighbors for each point $\mathbf{x}_i$ by using the Euclidean distance.

2. Constructing the similarity matrix. Construct a matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ whose entries $\mathbf{G}_{ij}$ is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Enforcing $\mathbf{G}_{ij} = 0$ if $\mathbf{x}_j$ is not a K-nearest neighbors of $\mathbf{x}_i$, otherwise,
(a) Heat kernel (parameter $\sigma \in \mathbb{R}$).

$$\mathbf{W}_{ij} = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2).$$

(b) Simple-minded (no parameters ($\sigma = \infty$)).

$$\mathbf{W}_{ij} = 1.$$

This simplification avoids the need to choose σ.

3. Embedding the high-dimensional data. Find vectors $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \mathbb{R}^{d \times n}$ that minimize the following objective function,

$$\mathbf{\Phi} = \sum_{i,j} \frac{\|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \mathbf{W}_{ij}}{\sqrt{\mathbf{D}_{ii}\mathbf{D}_{jj}}} = trace(\mathbf{YLY}^T),$$

where $\mathbf{D}$ is a diagonal matrix whose entries $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, and $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$.

C. Locally Linear Embedding

1. Searching the nearest neighbors. Find the K-nearest neighbors $\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^K$ for each point $\mathbf{x}_i$ by using the Euclidean distance.

2. Constructing the similarity matrix. For each data point $\mathbf{x}_i$, calculating its reconstruction coefficients $\mathbf{C}_i^l (l = 1, 2, \ldots, K)$ over the collection of the corresponding K nearest neighbors via minimizing the following objection function,

$$\mathbf{E} = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{l=1}^{K} \mathbf{C}_i^l \mathbf{x}_i^l \right\|_2^2,$$

where $\mathbf{x}_i^l$ denotes the l-th nearest neighbors of $\mathbf{x}_i$, and $\mathbf{C}_i^l$ denotes the reconstruction coefficient of $\mathbf{x}_i$ over $\mathbf{x}_i^l$.

Construct a similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. For each point $\mathbf{x}_i$, we set $\mathbf{W}_{ij}$, the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, as the corresponding similarity in $\mathbf{C}_i^l (l = 1, 2, \ldots, K)$, which have been calculated above, if $\mathbf{x}_j$ is a K-nearest neighbors to $\mathbf{x}_i$, otherwise, set $\mathbf{W}_{ij} = 0$.

3. Mapping each high dimensional input $\mathbf{x}_i$ to a low dimensional output $\mathbf{y}_i$. This is done by finding vectors $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \mathbb{R}^{d \times n}$ that minimize the following objective function,

$$\mathbf{\Phi} = \sum_i \left\| \mathbf{y}_i - \sum_j \mathbf{W}_{ij} \mathbf{y}_j \right\|_2^2$$
$$s.t. \sum_i \mathbf{y}_i = \mathbf{0}, \frac{1}{n} \sum_i \mathbf{y}_i \mathbf{y}_i^T = \mathbf{I}.$$

III. Local Neighborhood Embedding

It is difficult to interpret the real structure of the data if its dimensionality is more than four. One approach to simplification is to assume that the data actually lie on a non-linear manifold within the higher-dimensional space. Therefore, many non-linear and linear dimension reduction methods are proposed to find the real structure of data, i.e., ISOMAP, LLE, LE, etc. In these works, a key philosophy is that the real similarity relationship among data points is invariant to the dimensionality. Thus, the construction of similarity relationship plays an essential role in manifold learning.

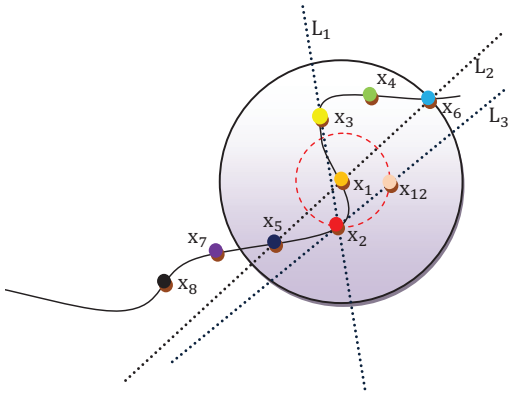

Fig. 1. Illustration of local neighborhood embedding representation.

In this section, we will describe our algorithm Local Neighborhood Embedding (LNE) which preserves pairwise-distance based similarity and reconstruction coefficient based similarity for finding the latent low dimensional structure of the data sets. Refer to Fig. 1, we sampled the data from a bend rope, and there is a outlier $\mathbf{x}_{12}$. The two points have highest similarities to $\mathbf{x}_1$ should be $\mathbf{x}_2$ and $\mathbf{x}_3$. However, the pairwise distance based similarity algorithms will give $\mathbf{x}_{12}$ a higher weight than $\mathbf{x}_2$ and $\mathbf{x}_3$, and it's same to LLE if the neighborhood size is very small (i.e., $K = 2$ here). The similarity based on reconstruction coefficient algorithms will give $\mathbf{x}_5$ and $\mathbf{x}_6$ highest scores, since $\mathbf{x}_5$ and $\mathbf{x}_6$ spanned an affine subspace across $\mathbf{x}_1$, which means that $\mathbf{x}_5$ and $\mathbf{x}_6$ can represent $\mathbf{x}_1$ perfectly. But the distances from $\mathbf{x}_5$ and $\mathbf{x}_6$ to $\mathbf{x}_1$ are much further than $\mathbf{x}_2$ and $\mathbf{x}_3$ to $\mathbf{x}_1$. Obviously, the similarity based on reconstruction coefficient algorithms ignored this important information. From above analysis, for a specified point $\mathbf{x}_i$, we should give $\mathbf{x}_j$ high similarity if $\mathbf{x}_j$ is close to $\mathbf{x}_i$ and can well represent $\mathbf{x}_i$ at the same time. $\mathbf{x}_2$ and $\mathbf{x}_3$ are close to $\mathbf{x}_1$, and the subspace spanned by $\mathbf{x}_2$ and $\mathbf{x}_3$ is also very close to $\mathbf{x}_1$. As a result, LNE will give $\mathbf{x}_2$ and $\mathbf{x}_3$ higher similarity scores than others.

A. The algorithm

The algorithmic procedure is formally stated as below:

Step 1: Searching the nearest neighbors

For each point $\mathbf{x}_i$, calculate the Euclidean distance between $\mathbf{x}_i$ and other points. Then, choose K-nearest neighbors $\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^K$ to construct a local dictionary $\mathbf{D}_i$, and the corresponding distances $\mathbf{S} = [\mathbf{s}_i^1, \mathbf{s}_i^2, \ldots, \mathbf{s}_i^K]$.

Step 2: Constructing the similarity matrix

Constructing the similarity matrix by reconstruction coefficient based similarity incorporating with pairwise similarity. LNE assumes that there exists a small neighborhood in which only the points that come from the same manifold lie approximately in a low-dimensional affine subspace. We use $\mathbf{C}_i \in \mathbb{R}^{K \times 1} (i = 1, 2, \ldots, n)$ to represent the similarities between $\mathbf{x}_i$ and its neighbors. In the local dictionary $\mathbf{D}_i$, the points which are close to $\mathbf{x}_i$ should give higher contributions than the points that are further to represent $\mathbf{x}_i$, mathematically,

$$\min_{\mathbf{C}_i} \lambda \|\mathbf{S}_i \mathbf{C}_i\|_2^2 + (1-\lambda) \|\mathbf{x}_i - \mathbf{D}_i \mathbf{C}_i\|_2^2 \quad s.t. \quad \mathbf{1}^T \mathbf{C}_i = 1, \tag{1}$$

where $\mathbf{D}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^K]$, $\mathbf{S}_i = diag(\mathbf{s}_i^1, \mathbf{s}_i^2, \ldots, \mathbf{s}_i^K)$, and $\lambda \in [0, 1]$ is a balance parameter.

The solution of the problem (1) is:

$$\mathbf{C}_i = \frac{\mathbf{M}^{-1}\mathbf{1}}{\mathbf{1}^T \mathbf{M}^{-1} \mathbf{1}},$$

where $\mathbf{M} = \lambda \mathbf{S}_i^T \mathbf{S}_i + (1-\lambda)(\mathbf{X}_i - \mathbf{D}_i)^T (\mathbf{X}_i - \mathbf{D}_i)$, and $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_i, \ldots, \mathbf{x}_i] \in \mathbb{R}^{m \times K}$.

Construct a matrix of similarities $\mathbf{W} \in \mathbb{R}^{n \times n}$. For the point $\mathbf{x}_i$, we set $\mathbf{W}_{ij}$, the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, as the corresponding similarity in $\mathbf{C}_i$, which have been calculated above, if $\mathbf{x}_j$ is a K-nearest neighbors to $\mathbf{x}_i$, otherwise, set $\mathbf{W}_{ij} = 0$.

Step 3: Embedding to the global coordinates

After getting the similarity relationship $\mathbf{W}$ among data points, as in LLE, the low dimensional representation can be achieved by preserving the $\mathbf{W}$ and minimizing the following objective function:

$$\Phi(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_j \mathbf{W}_{ij} \mathbf{y}_j\|^2 = trace(\mathbf{Y}\mathbf{Q}\mathbf{Y}^T) \quad s.t. \quad \sum_i \mathbf{y}_i = \mathbf{0}, \frac{1}{n} \sum_i \mathbf{y}_i \mathbf{y}_i^T = \mathbf{I}, \tag{2}$$

where $\mathbf{Q} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$.

The solution of the embedding objective function can be achieved by solving a sparse $n \times n$ eigenvalue problem. The output is a matrix consisted by d eigenvalue eigenvectors, which corresponding to the second to $(d+1)$-th smallest eigenvalue of the matrix $\mathbf{Q}$.

B. Theory analysis

In this subsection, we provide theoretical analysis of the LNE algorithm. For a given data point $\mathbf{x}_i$, its similarity relationship with the other points is calculated by solving the following problem,

$$\min_{\mathbf{C}_i} \ \lambda\|\mathbf{S}_i\mathbf{C}_i\|_2^2+(1-\lambda)\|\mathbf{x}_i-\mathbf{D}_i\mathbf{C}_i\|_2^2 \quad s.t. \quad \mathbf{1}^T\mathbf{C}_i=1,$$

Using Lagrangian method, we can get following objective function,

$$\begin{aligned} L &= \lambda\|\mathbf{S}_i\mathbf{C}_i\|_2^2+(1-\lambda)\|\mathbf{x}_i\mathbf{1}^T\mathbf{C}_i-\mathbf{D}_i\mathbf{C}_i\|_2^2 \\ &\quad +\theta(\mathbf{1}^T\mathbf{C}_i-1) \\ &= \lambda\|\mathbf{S}_i\mathbf{C}_i\|_2^2+(1-\lambda)\|(\mathbf{X}_i-\mathbf{D}_i)\mathbf{C}_i\|_2^2+\theta(\mathbf{1}^T\mathbf{C}_i-1), \end{aligned}$$

where θ is the Lagrangian multiplier, and $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_i, \ldots, \mathbf{x}_i] \in \mathbb{R}^{m\times K}$. Clearly,

$$\frac{\partial L}{\partial \mathbf{C}_i} = 2\mathbf{M}\mathbf{C}_i+\theta\mathbf{1},$$

where $\mathbf{M} = \lambda\mathbf{S}_i^T\mathbf{S}_i+(1-\lambda)(\mathbf{X}_i-\mathbf{D}_i)^T(\mathbf{X}_i-\mathbf{D}_i)$.

Letting $\frac{\partial L}{\mathbf{C}_i}=0$, it gives that,

$$\mathbf{C}_i = -\frac{1}{2}\theta\mathbf{M}^{-1}\mathbf{1} \tag{3}$$

Multiplying both sides of (3) by $\mathbf{1}^T$, and since $\mathbf{1}^T\mathbf{C}_i=1$, then,

$$\theta = -\frac{2}{\mathbf{1}^T\mathbf{M}^{-1}\mathbf{1}}.$$

Substituting θ into (3), the solution is then given by

$$\mathbf{C}_i = \frac{\mathbf{M}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{M}^{-1}\mathbf{1}}.$$

where $\mathbf{M} = \lambda\mathbf{S}_i^T\mathbf{S}_i+(1-\lambda)(\mathbf{X}_i-\mathbf{D}_i)^T(\mathbf{X}_i-\mathbf{D}_i)$.

IV. Experimental verification and analysis

In this section, we will compare our algorithm with other four popular dimension reduction methods, i.e., PCA, LE, ISOMAP, and LLE. Several synthetic data sets and two real image databases are used to examine the effectiveness of the competitive algorithms. For all the experiments, we use the optimization program (1), where we typically set $\lambda = 0.05, 0.2, 0.5$. If the data are well and uniformly sampled from a smooth manifold, then pairwise distance is not important, we should choose a smaller value for λ. In case the data is contaminated by noises or the neighbor size is quit large, we should give a bigger value to λ, since we want to give a heavy penalty on the faraway points which are included in the K neighbors.

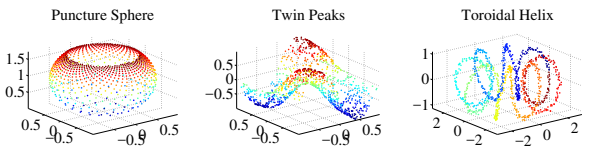

Fig. 2. Synthetic manifolds used in the experiments. (a) Puncture Sphere. (b) Twin Peaks. (c) Toroidal Helix.

A. Experiments with Synthetic Data

To test the ability of our proposed algorithm, we used three data sets having different manifolds: Puncture Sphere, Twin Peaks, and Toroidal Helix, as shown in Fig. 2. The effectiveness of tested algorithms are evaluated with respect to the following aspects: 1) the sensitivity of neighbors number K, 2) the ability to solve bend shape data, 3) anti-noise ability, and 4) the performance in the low sampled rate case. For the first three tests, 1000 points are drawn from each manifold. For the last one, 300 points from Puncture Sphere manifold are used.

(1) The performance versus the variance of the neighborhood size K

As shown in Fig. 3, we give different K values to LNE to evaluate its sensitivity to the parameter K. ISOMAP is failed to get the correct manifold, since it maps the manifold from the top into a plane. LE can get a quit satisfied result when $K = 20$, but it going to bad when $K = 10$ or $K = 30$. Similar to LE, LLE can get a good result when $K = 10$, but it works worse and worse along K become larger and larger. However, the proposed algorithm LNE, which performances well over all cases, is insensitive to K.

(2) The capability to the bend shape

Twin Peaks (Fig. 2(b)), which is generated by folding up the corners of a plane and the 2D projection should show a roughly rectangular shape with blue and red in opposite corners, is used in this experiment. We set the degree of bend (Z-Scaling, marked as ζ in Fig. 4) equals 1.0 and 2.0. The embedding result is illustrated in Fig. 4, all tested algorithms are failed to reveal real low dimensional structure on this data set, except LE and the proposed LNE. Compared with LE, the result of LNE is more similar to the real manifold which is nearly a square plane. In addition, we can see that there are some blank areas inside the result of LE.

(3) Robustness to noise

To examine the anti-noise ability of our method, a small amounts of white gaussian noise ($\sigma = 0.05, 0.1$) are added into a data set containing of 1000 points sampled from Toroidal Helix (Fig. 2(c)) whose real low dimensional structure is unraveled into a circle. As shown in the Fig. 5, we can see that only LNE achieves correct result, while PCA gets a top-down overlooking profile, and ISOMAP over emphasizes the outliers.

(4) Performance in the case of low sampled rate

To evaluate the result of LNE in the under-sampled case, we form a data set by sampling 300 points from Punctured Sphere (as shown in Fig. 2(a)). The result in Fig. 6 shows that only LNE reveal the correct manifold in case the samples are not enough, while other four algorithms produced unsatisfied results because the number of samples are too small which make them unable to properly connect to each other.

B. Experiments with Real Data

In this subsection, we compare the performance of LNE with three popular non-linear dimension reduction algo-

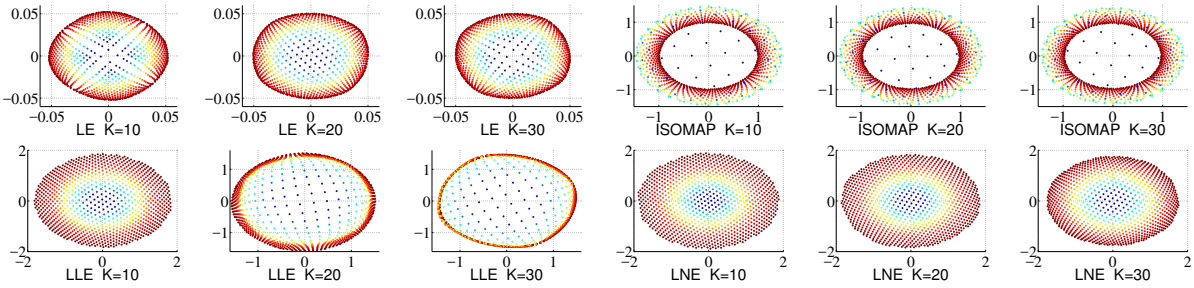

Fig. 3. Comparison of LE, ISOMAP, LLE, and LNE to the number of neighbors on the Punctured Sphere data set (1000 points).

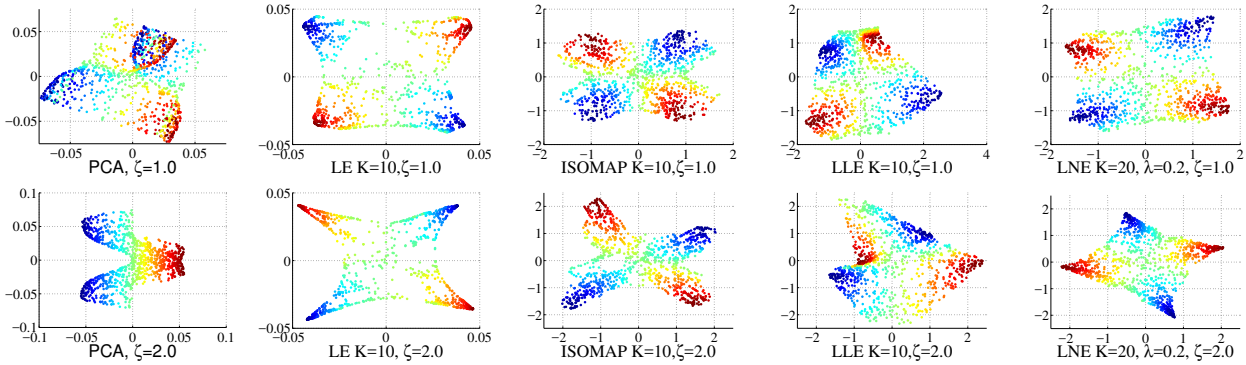

Fig. 4. Performance of PCA, LE, ISOMAP, LLE, and LNE to the degree of bend ($\zeta = 1.0, 2.0$) on the Twin Peaks data set (1000 points).

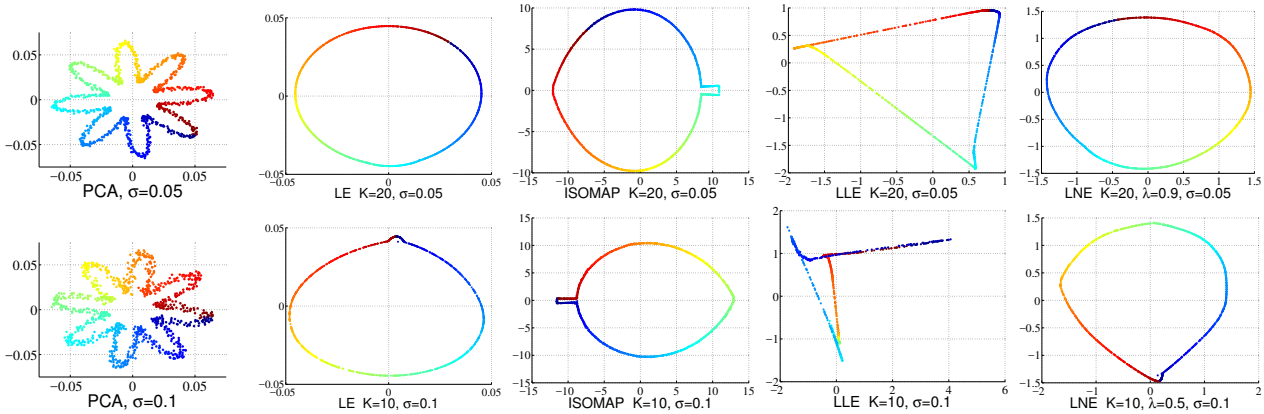

Fig. 5. Comparison the robust of PCA, LE, ISOMAP, LLE, and LNE to the white gaussian noise ($\sigma = 0.005, 0.1$) on the Toroidal Helix data set (1000 data points)

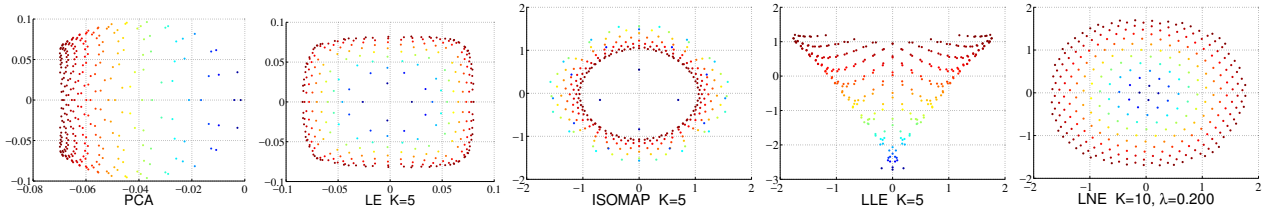

Fig. 6. Comparison of PCA, LE, ISOMAP, LLE, and LNE to the low-sampled rate case on the Punctured Sphere data set (300 data points).

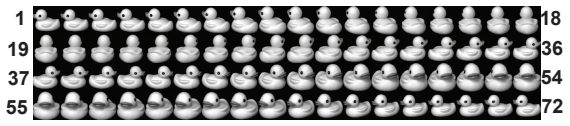

Fig. 7. The "Duck" images of COIL-20. Each image is labeled with a identification number from 1 to 72 according to its view angle.

rithms, LE, ISOMAP, and LLE, on a subset of Columbia Object Image Library (COIL-20) [16] and Extended Yale B database [17].

The "Duck" images which belongs to a subject of the COIL-20, are sequentially obtained from 72 views at the intervals of 5 degrees around a "Duck". We resized each image from 128×128 pixels to 32×32 pixels, and mark them with the identification number from 1 to 72 from left to right and then top to down, as shown in the Fig. 7. The images in the first column are marked with the identification number 1, 19, 37, and 55 corresponding to γ equals 0°, 90°, 180°, and 270° views, respectively.

Fig. 8 demonstrates the results of LE, ISOMAP, LLE and LNE when neighbors number *K* equals 5 and 8. We can see that the 2-D data representation form a closed loop, and LNE captures the image variations of both view angles and zooming effects. Specifically, the data points No. 1 ($\gamma = 0^\circ$) and No. 37 ($\gamma = 180^\circ$) are nearly symmetrically distributed with the axis, and their link is perpendicular to the link between the point No. 19 ($\gamma = 90^\circ$) and No. 55 ($\gamma = 270^\circ$). Moreover, we can see LNE captures the drastic variation between the image with $\gamma = 90^\circ$ and $\gamma = 270^\circ$ owing to the variation of the beak of the duck. Furthermore, it is easy to find that the points around No. 1 and No. 37 distributed denser than other places, due to the extra strong zooming effects of the fixed bounding box. It means that LNE discovers the hidden structures successfully. On the other hand, LLE failed to get the correct result. ISOMAP only get the images variation of view angles, but not zooming effects.

The Extended Yale B database contains 38 individuals and around 64 frontal images for each individual under different illuminations. The images of the first person are used in our experiment. Similar to [18], and each image is resized from 192×168 to 32×32.

The tested algorithms projects each face onto a 2-D plane, as shown in the Fig. 9. We can see that, as we move along the horizontal axis, the direction of the light source changes from left to right, while as we move along the vertical axis, the overall darkness of the images changes from dark to light. Moreover, the results of LNE are nearly symmetrically distributed in the 2-D plane, which reflects the real distribution of the data set. LE and ISOMAP can get quit good results as well, but LLE produces a irregular scatted distribution.

V. Conclusion

In this paper, we proposed LNE for unsupervised dimensionality reduction by integrating the pairwise distance and local linear representation together to obtain the similarity relationship among the points. The experiments in synthetic and real data demonstrate that LNE, compared with the algorithms that only use the pairwise distance or reconstruction coefficients similarities, has following advantages: 1) Insensitive to the choice of neighborhood size; 2) Robust to the noise; 3) Works well even in under-sampled case. Moreover, the proposed algorithm is computational efficient owing to its analytic solution.

There are several potential ways to improve or extend this work. LNE determines the similarity relationship in unsupervised way, the previous studies show that the label information is helpful to find the real neighborhood of data, i.e. , the semi-supervised learning way [19] [20]. It is interesting to develop the supervised or semi-supervised case for LNE.

References

[1] J. Tenenbaum, V. De Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[2] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[3] L. Saul and S. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[4] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[5] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *Neural Networks, IEEE Transactions on*, vol. 17, no. 1, pp. 157–165, 2006.

[6] D. Xu, S. Yan, D. Tao, S. Lin, and H. Zhang, "Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval," *Image Processing, IEEE Transactions on*, vol. 16, no. 11, pp. 2811–2821, 2007.

[7] D. Zhang, Z. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *SIAM Conference on Data Mining (ICDM)*, 2007, pp. 629–634.

[8] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *International Conference on Computer Vision (ICCV)*, 2007.

[9] I. Jolliffe, "Principal component analysis," *Springer Series in Statistics, Berlin: Springer, 1986*, vol. 1, 1986.

[10] X. Niyogi, "Locality preserving projections," in *Neural Information Processing Systems*, vol. 16. The MIT Press, 2003, p. 153.

[11] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *International Conference on Computer Vision (ICCV)*, 2005.

[12] B. Schlkopf, A. Smola, and K. Mller, "Kernel principal component analysis," *Artificial Neural NetworksICANN'97*, pp. 583–588, 1997.

[13] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in neural information processing systems*, vol. 14, pp. 585–591, 2001.

[14] J. Gou, T. Xiong, and Y. Kuang, "A novel weighted voting for k-nearest neighbor rule," *Journal of Computers*, vol. 6, no. 5, pp. 833–840, 2011.

[15] T. Cox and M. Cox, "Multidimensional scaling," *Chapman & Hall, London, UK*, 1994.

[16] S. Nene, S. Nayar, and H. Murase, "Columbia object image library (coil-20)," *Dept. Comput. Sci., Columbia Univ., New York.[Online] http://www. cs. columbia. edu/CAVE/coil-20. html*, 1996.

[17] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 684–698, 2005.

[18] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with ℓ^1-graph for image analysis," *Image Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 858–866, 2010.

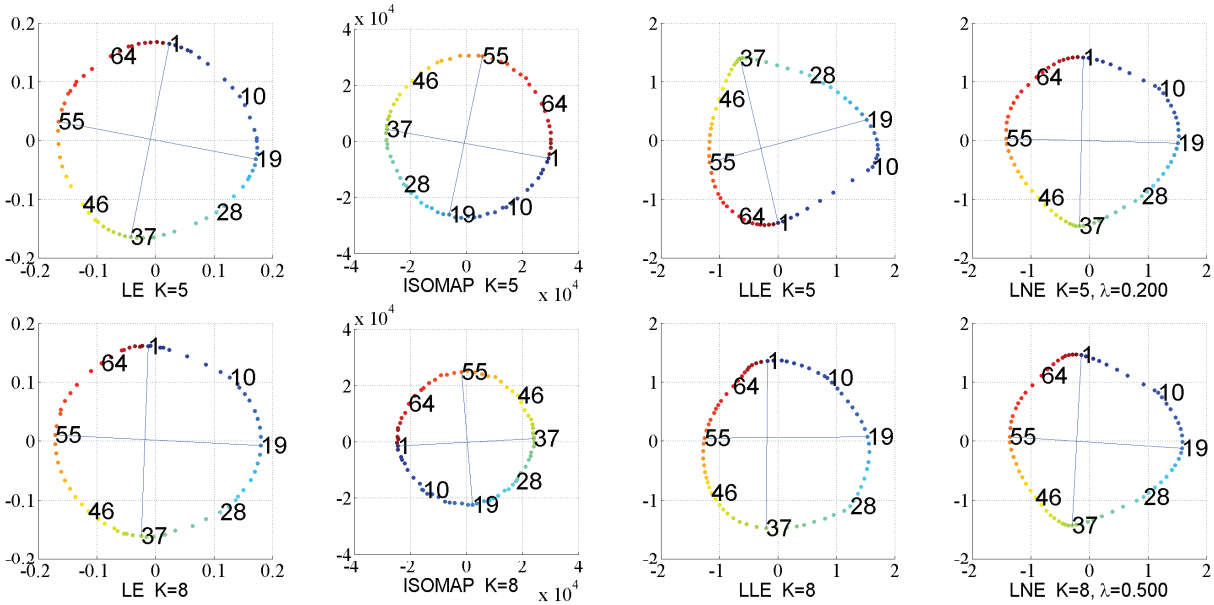

Fig. 8. Two-dimensional representations of the "Duck" images using LE, ISOMAP, LLE, and LNE with neighborhood size 5 and 8, respectively.

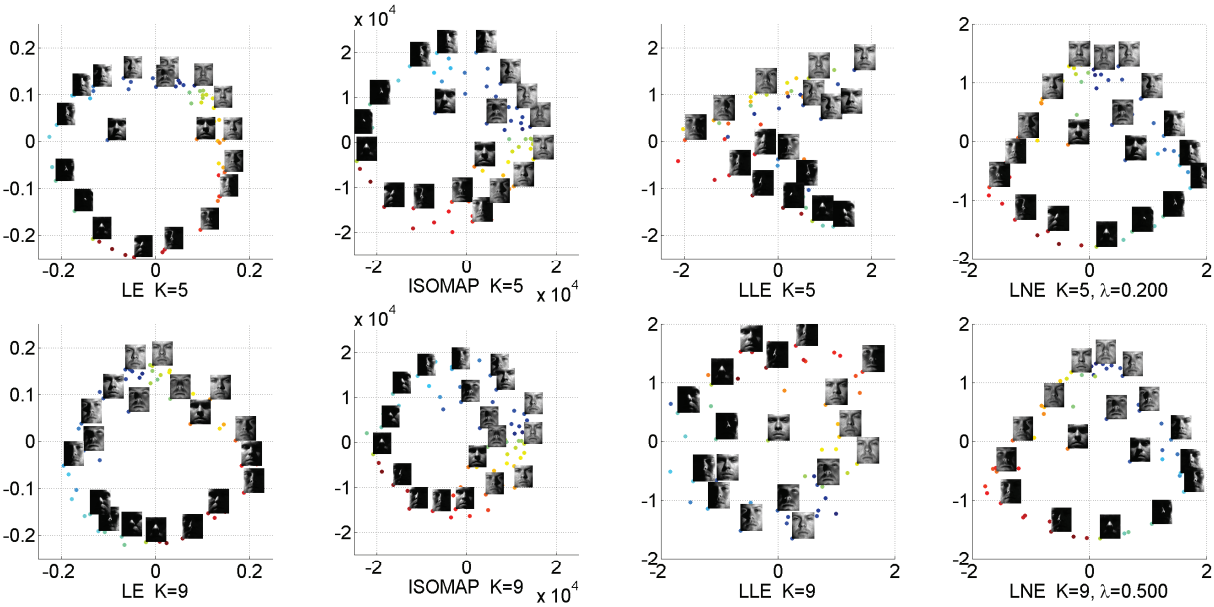

Fig. 9. Two-dimensional representations of the face images belonging to a person from Extended Yale B database using LE, ISOMAP, LLE, and LNE with neighborhood size 5 and 9, respectively.

[19] T. Guo and G. Li, "Confidence estimation for graph-based semi-supervised learning," *Journal of Software*, vol. 7, no. 6, pp. 1307–1314, 2012.

[20] X. Xu, "A new sub-topics clustering method based on semi-supervised learing," *Journal of Computers*, vol. 7, no. 10, pp. 2471–2478, 2012.