Robust Locally Linear Embedding

Hong Chang Dit-Yan Yeung

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Corresponding author: Dit-Yan Yeung, dyyeung@cs.ust.hk, +852-2358-1477 (fax)

Abstract

In the past few years, some nonlinear dimensionality reduction (NLDR) or nonlinear manifold learning methods have aroused a great deal of interest in the machine learning community. These methods are promising in that they can automatically discover the low-dimensional nonlinear manifold in a high-dimensional data space and then embed the data points into a low-dimensional embedding space, using tractable linear algebraic techniques that are easy to implement and are not prone to local minima. Despite their appealing properties, these NLDR methods are not robust against outliers in the data, yet so far very little has been done to address the robustness problem. In this paper, we address this problem in the context of an NLDR method called locally linear embedding (LLE). Based on robust estimation techniques, we propose an approach to make LLE more robust. We refer to this approach as robust locally linear embedding (RLLE). We also present several specific methods for realizing this general RLLE approach. Experimental results on both synthetic and real-world data show that RLLE is very robust against outliers.

Key words: nonlinear dimensionality reduction, manifold learning, locally linear embedding, principal component analysis, outlier, robust statistics, M-estimation, handwritten digit, wood texture

1 Introduction

Dimensionality reduction is concerned with the problem of mapping data points that lie on or near a low-dimensional manifold in a high-dimensional data space to a low-dimensional embedding space. Traditional techniques such as principal component analysis (PCA) and multidimensional scaling (MDS) have been extensively used for linear dimensionality reduction. However, these methods are inadequate for embedding nonlinear manifolds.

In recent years, some newly proposed methods such as isometric feature mapping (Isomap) [1], locally linear embedding (LLE) [2,3], and Laplacian eigenmap [4,5] have aroused a great deal of interest in nonlinear dimensionality reduction (NLDR) or nonlinear manifold learning problems. Unlike previously proposed NLDR methods such as autoassociative neural networks which require complex optimization techniques, these new NLDR methods enjoy the primary advantages of PCA and MDS in that they still make use of simple linear algebraic techniques that are easy to implement and are not prone to local minima.

Despite the appealing properties of these new NLDR methods, they are not robust against outliers in the data. Although some extensions have been proposed to the original methods [6–12,3,13–15],very little has yet been done to address the outlier problem. Among the extensions proposed is an interesting extension of LLE proposed by Teh and Roweis, called locally linear coordination (LLC) [13], which combines the subspace mixture modeling approach with LLE. A recent work by de Ridder and Franc [16] attempted to address the outlier problem by proposing a robust version of LLC based on a recent development in the statistics community called mixtures of t-distributions. However, although the robust version of LLC is less sensitive to outliers than LLC, the authors found that it is still more sensitive to outliers than ordinary LLE. Zhang and Zha [17] proposed a preprocessing method for outlier removal and noise reduction before NLDR is performed. It is based on a weighted version of PCA. However, the method for determining the weights is heuristic in nature without formal justification. More recently, Hadid and Pietikäinen [18] studied the outlier problem and proposed a method to make LLE more robust. However, their method is also heuristic in nature. Moreover, their method is based on the assumption that all outliers are very far away from the data on the manifold and they themselves form distinct connected components in the neighborhood graph. Hence the outliers have no effect on the reconstruction of the manifold data points. Apparently, this assumption is not always true for many real-world applications.

In this paper, we address the outlier problem in the context of LLE. Based on robust PCA techniques, we propose an approach to make LLE more robust. The rest of this paper is organized as follows. In Section 2, we first give a quick review of the LLE algorithm. In Section 3, the sensitivity of LLE to outliers is illustrated through some examples based on synthetic data. A new approach called robust locally linear embedding (RLLE) is then presented in Section 4 together with several specific realizations of the approach. Section 5 shows some experimental results to demonstrate the effectiveness of RLLE in the presence of outliers. Some concluding remarks are given in Section 6.

2 Locally Linear Embedding

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of N points in a high-dimensional data space $\mathcal{R}^D$. The data points are assumed to lie on or near a nonlinear manifold of intrinsic dimensionality $d < D$ (typically $d \ll D$). Provided that sufficient data are available by sampling well from the manifold, the goal of LLE is to find a low-dimensional embedding of $\mathcal{X}$ by mapping the D-dimensional data into a single global coordinate system in $\mathcal{R}^d$. Let us denote the corresponding set of N points in the embedding space $\mathcal{R}^d$ by $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$.

The LLE algorithm [3] can be summarized as follows:

(1) For each data point $\mathbf{x}_i \in \mathcal{X}$:
   (a) Find the set $\mathcal{N}_i$ of K nearest neighbors of $\mathbf{x}_i$.
   (b) Compute the reconstruction weights of the neighbors that minimize the error of reconstructing $\mathbf{x}_i$.
(2) Compute the low-dimensional embedding $\mathcal{Y}$ for $\mathcal{X}$ that best preserves the local geometry represented by the reconstruction weights.

Step (1)(a) is typically done by using Euclidean distance to define neighborhood, although more sophisticated criteria may also be used.

Based on the K nearest neighbors identified, step (1)(b) seeks to find the best reconstruction weights. Optimality is achieved by minimizing the local reconstruction error for $\mathbf{x}_i$

$$\mathcal{E}_i = \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}\mathbf{x}_j\|^2, \tag{1}$$

which is the squared distance between $\mathbf{x}_i$ and its reconstruction, subject to the constraints $\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} = 1$ and $w_{ij} = 0$ for any $\mathbf{x}_j \notin \mathcal{N}_i$. Minimizing $\mathcal{E}_i$ subject to the constraints is a constrained least squares problem. After repeating steps (1)(a) and (1)(b) for all N data points in $\mathcal{X}$, the reconstruction weights obtained form a weight matrix $\mathbf{W} = [w_{ij}]_{N \times N}$.

Step (2) of the LLE algorithm is to compute the best low-dimensional embedding $\mathcal{Y}$ based on the weight matrix $\mathbf{W}$ obtained. This corresponds to minimizing the following cost function:

$$\Phi = \sum_{i=1}^{N} \|\mathbf{y}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}\mathbf{y}_j\|^2, \tag{2}$$

subject to the constraints $\sum_{i=1}^{N} \mathbf{y}_i = \mathbf{0}$ and $\frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i\mathbf{y}_i^T = \mathbf{I}$, where $\mathbf{0}$ is a column vector of zeros and $\mathbf{I}$ is an identity matrix. Note the similarity of this equation to (1). Based on $\mathbf{W}$, we can define a sparse, symmetric, and positive semidefinite matrix $\mathbf{M}$ as follows:

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

Note that (2) can be expressed in the quadratic form, $\Phi = \sum_{i,j} M_{ij}\mathbf{y}_i^T\mathbf{y}_j$, based on $\mathbf{M} = [M_{ij}]_{N \times N}$. By the Rayleigh-Ritz theorem [19], minimizing (2) with respect to the $\mathbf{y}_i$'s in $\mathcal{Y}$ can be done by finding the eigenvectors with the smallest (nonzero) eigenvalues.

Figure 1 shows how LLE works in finding the low-dimensional embedding of the S curve manifold from $\mathcal{R}^3$ to $\mathcal{R}^2$.

3 Sensitivity of Locally Linear Embedding to Outliers

In this section, we will show through examples how the LLE results can be affected by outliers in the data. We use three artificial data sets that have been commonly used by other researchers: Swiss roll (Figure 2), S curve (Figure 3), and helix (Figure 4). For each data set, uniformly distributed random noise points that are at least at a certain distance from the data points on the manifold are added as outliers. Table 1 shows the parameter settings used in these experiments. The parameters include the dimensionality of the data space D, the dimensionality of the embedding space d (i.e., intrinsic dimensionality of the nonlinear manifold), the number of nearest neighbors K, the number of clean data points on the manifold, the number of outlier points, and the minimum distance between randomly generated outliers and data points on the manifold.

As we can see from subfigures (b) of Figures 2–4, LLE cannot preserve well the local geometry of the data manifolds in the embedding space when there are outliers in the data. In fact, in the presence of outliers, the K nearest neighbors of a (clean) data point on the manifold may no longer lie on a locally linear patch of the manifold, leading to a small bias to the reconstruction. As for an outlier point, its neighborhood is typically much larger than that of a clean data point. As a result, the estimated reconstruction weights of its neighbors cannot reflect well the local geometry of the manifold in the embedding space, leading to a large bias to the embedding result. To make LLE more robust against outliers, we believe it is crucial to be able to identify the outliers and reduce their influence on the embedding result. In the next section, we present an approach to its realization based on robust statistics.

4 Robust Locally Linear Embedding

The main idea of robust statistics is to devise statistical procedures that reduce the influence of distributional deviations and hence become insensitive to them. This follows the notion of distributional robustness from Huber [20].

Our robust version of LLE, or RLLE, first performs local robust PCA [21] on the data points in $\mathcal{X}$. The robust PCA algorithm is based on weighted PCA. It gives us a measure on how likely each data point comes from the underlying data manifold. Outliers can then be identified and their influence is reduced in the subsequent LLE learning procedure. The major modifications of RLLE to the original LLE algorithm are discussed below.

4.1 Principal Component Analysis

As in step (1)(a) of the LLE algorithm, K nearest neighbors of each data point $\mathbf{x}_i$ are identified. Let $i_1, i_2, \ldots, i_K$ denote their indices, and hence $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_K}$ denote the K neighbors in $\mathcal{X}$. We define a $D \times K$ matrix $\mathbf{X} = [\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_K}]$. Ideally, if $\mathbf{x}_i$ lies on the manifold, we expect its K nearest neighbors to lie on a locally linear patch of the manifold as well. Let us assume the dimensionality of this locally linear subspace be d. Each neighbor $\mathbf{x}_{i_j}$ can be linearly projected onto the d-dimensional subspace with coordinate vector $\mathbf{z}_j = \mathbf{B}^T(\mathbf{x}_{i_j} - \mathbf{d}) \in \mathcal{R}^d$, where $\mathbf{d} \in \mathcal{R}^D$ is a displacement vector and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d] \in \mathcal{R}^{D \times d}$ is a rotation matrix with $\mathbf{b}_j^T \mathbf{b}_k = \delta_{jk}$ for $1 \leq j, k \leq K$, i.e., $\mathbf{b}_j$'s are orthonormal basis vectors. The low-dimensional image $\mathbf{z}_j$ of $\mathbf{x}_{i_j}$ is represented as $\hat{\mathbf{x}}_{i_j} = \mathbf{d} + \mathbf{B}\mathbf{z}_j = \mathbf{d} + \mathbf{B}\mathbf{B}^T(\mathbf{x}_{i_j} - \mathbf{d})$ in the original space $\mathcal{R}^D$. Let the difference between $\mathbf{x}_{i_j}$ and $\hat{\mathbf{x}}_{i_j}$ be denoted as $\boldsymbol{\varepsilon}_j = \mathbf{x}_{i_j} - \hat{\mathbf{x}}_{i_j}$. Standard PCA seeks to find the least squares estimates of $\mathbf{d}$ and $\mathbf{B}$ by minimizing

$$E_{pca} = \sum_{j=1}^{K} \|\boldsymbol{\varepsilon}_j\|^2 = \|\mathbf{X} - \mathbf{d}\mathbf{1}^T - \mathbf{B}\mathbf{Z}\|_F^2 \tag{3}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K] \in \mathcal{R}^{d \times K}$ and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

PCA seeks to construct the rank-d subspace approximation to the D-dimensional data that is optimal in the least squares sense. Like other least squares estimation techniques, PCA is not robust against outliers in the data.

4.2 Weighted Principal Component Analysis

Instead of using the standard optimization criterion in (3), we modify it to a weighted squared error criterion. Given a set of nonnegative weights $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$ for the K neighbors, the optimization problem becomes minimizing the total weighted squared error

$$E_{rpca} = \sum_{j=1}^{K} a_j \|\boldsymbol{\varepsilon}_j\|^2 \tag{4}$$

with respect to $\mathbf{d}, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$, subject to $\mathbf{b}_j^T \mathbf{b}_k = \delta_{jk}$ for $1 \leq j, k \leq d$. It can easily be shown that the least squares estimate of $\mathbf{d}$ is equal to the weighted sample mean vector

$$\mathbf{d}_{\mathcal{A}} = \frac{\sum_{j=1}^{K} a_j \mathbf{x}_{i_j}}{\sum_{j=1}^{K} a_j} \; = \; \boldsymbol{\mu}_{\mathcal{A}}. \tag{5}$$

The least squares estimates of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$ are the (orthonormal) eigenvectors of

$$\mathbf{S}_{\mathcal{A}} = \frac{1}{K} \sum_{j=1}^{K} a_j (\mathbf{x}_{i_j} - \boldsymbol{\mu}_{\mathcal{A}})(\mathbf{x}_{i_j} - \boldsymbol{\mu}_{\mathcal{A}})^T, \tag{6}$$

which is weighted sample covariance matrix of the K neighbors. This corresponds to a weighted version of PCA. To reduce the influence of possible outliers among the K neighbors, we would like to set $\mathcal{A}$ such that outliers get small weight values. In other words, if a data point $\mathbf{x}_{i_j}$ has a large error norm $\|\boldsymbol{\varepsilon}_j\|$, we would like to set a_j small. Robust estimation methods can help to set the appropriate weights, making weighted PCA a robust version of PCA.

4.3 Robust Principal Component Analysis

The above solution for weighted PCA assumes that $\mathcal{A}$ is fixed and is already known. For weighted PCA to work as a robust PCA algorithm, we noted above that we want $\mathcal{A}$ to depend on the $\boldsymbol{\varepsilon}_j$'s. We also note that $\boldsymbol{\varepsilon}_j$'s depend on $\mathbf{d}$ and $\mathbf{B}$, which in turn depend on $\mathcal{A}$. Because of this cyclic dependency, we use an iterative procedure to find the solution by starting from some initial estimates. This iterative procedure, called *iteratively reweighted least squares* (IRLS) [22], can be summarized as follows:

(1) Use standard PCA to find the initial least squares estimates of $\mathbf{d}$ and $\mathbf{B}$, denoted $\mathbf{d}^{(0)}$ and $\mathbf{B}^{(0)}$. Set $t = 0$.

(2) Repeat the following steps:

   (a) $t = t + 1$;

   (b) Compute $\boldsymbol{\varepsilon}_j^{(t-1)} = \mathbf{x}_{i_j} - \mathbf{d}^{(t-1)} - \mathbf{B}^{(t-1)}(\mathbf{B}^{(t-1)})^T(\mathbf{x}_{i_j} - \mathbf{d}^{(t-1)})$, $1 \leq j \leq K$;

   (c) Compute $a_j^{(t-1)} = a(\|\boldsymbol{\varepsilon}_j^{(t-1)}\|)$, $1 \leq j \leq K$;

   (d) Compute the weighted least squares estimates $\mathbf{d}^{(t)}$ and $\mathbf{B}^{(t)}$ by performing weighted PCA on $\mathbf{X}$ based on the weight set $\mathcal{A}^{(t-1)}$.

   Until $\mathbf{d}^{(t)}$ and $\mathbf{B}^{(t)}$ do not change too much from $\mathbf{d}^{(t-1)}$ and $\mathbf{B}^{(t-1)}$.

Here we assume that $a(\cdot)$ is some weight function that determines the weight a_j from the corresponding error norm or error residual $e_j = \|\boldsymbol{\varepsilon}_j\|$:

$$a_j = a(e_j) = a(\|\boldsymbol{\varepsilon}_j\|).$$

Following the ideas of Huber [23], we replace the least squares estimator by a robust estimator that minimizes

$$E_\rho = \sum_{j=1}^{K} \rho(e_j) = \sum_{j=1}^{K} \rho(\|\boldsymbol{\varepsilon}_j\|),$$

where $\rho(\cdot)$ is some convex function.Using the Huber function

$$\rho(e) = \begin{cases} \frac{1}{2}e^2 & |e| \le c \\ c(|e| - \frac{1}{2}c) & |e| > c \end{cases}$$

for some parameter $c > 0$, the weight function can be defined as

$$a(e) = \frac{\psi(e)}{e} = \frac{\rho'(e)}{e} = \begin{cases} 1 & |e| \le c \\ \frac{c}{|e|} & |e| > c \end{cases}$$

where $\psi(\cdot)$ is called the influence function which is the first derivative of $\rho(\cdot)$. This weight function allows the IRLS procedure to perform M-estimation for robust PCA. In our experiments, we set c to be half of the mean error residual of the K nearest neighbors, i.e., $c = \frac{1}{2K}\sum_{j=1}^{K} e_j$.

4.4 RLLE Algorithm

4.4.1 Reliability Scores

After the IRLS procedure converges to give the weighted least squares estimates of $\mathbf{d}$ and $\mathbf{B}$, each neighbor $\mathbf{x}_{i_j}$ has an associated weight value a_j. A normalized weight value a_j^* is then computed as $a_j^* = a_j / \sum_{k=1}^{K} a_k$. This normalized weight value can serve as a reliability measure for each neighbor of point $\mathbf{x}_i$. For all points not in the neighborhood of $\mathbf{x}_i$, their weights are set to 0. After performing robust PCA for all points in $\mathcal{X}$, a total reliability score s_i

is obtained for each point by summing up the normalized weight values from all robust PCA runs. The smaller the value of the total reliability score s_i for a point $\mathbf{x}_i$, the more likely it is that $\mathbf{x}_i$ is an outlier. The reliability scores can be used in different ways to reduce the influence of the outliers on the embedding result. We will describe some specific methods below for realizing this idea.

Note that the normalized weight values a_j^*'s are analogous to the posterior probabilities of the hidden variables in the expectation-maximization (EM) algorithm for mixture models. The reliability scores are similar to the so-called "responsibilities" used in the generative models for handwriting recognition [24].

4.4.2 Weighted Embedding with Reliability Scores

To preserve the integrity of the data, all data points including the clean data points and the outliers are projected into the embedding space. Embedding is achieved by minimizing a weighted version of the cost function in (2) with the reliability scores serving as weights.

Let $\mathcal{X}_d$ denote the set of clean data points identified based on the reliability scores, i.e., a point $\mathbf{x}_i$ is in $\mathcal{X}_d$ if and only if $s_i \geq \alpha$ for some threshold $\alpha > 0$. The RLLE algorithm can be described as follows:

(1) For each data point $\mathbf{x}_i \in \mathcal{X}$:
   (a) Find the set $\mathcal{N}_i \subset \mathcal{X}_d$ of K nearest neighbors of $\mathbf{x}_i$.
   (b) Compute the reconstruction weights of the neighbors that minimize the error of reconstructing $\mathbf{x}_i$.
(2) Compute the low-dimensional embedding $\mathcal{Y}$ for $\mathcal{X}$ that best preserves the local geometry by minimizing the following weighted cost function:

$$\Phi^s = \sum_{i=1}^{N} s_i \|\mathbf{y}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}\mathbf{y}_j\|^2.$$

Step (2) of the RLLE algorithm above can be seen as a generalization of that for standard LLE. We can express the weighted cost function in a quadratic form, $\Phi^s = \sum_{i,j} M_{ij}^s \mathbf{y}_i^T \mathbf{y}_j$, where $\mathbf{M}^s = [M_{ij}^s]_{N\times N}$ is a sparse, symmetric, and positive semidefinite matrix defined as:

$$\mathbf{M}^s = \mathbf{S}(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W}) = \mathbf{S}\mathbf{M},$$

with $\mathbf{S} = \mathrm{diag}(s_1^2, s_2^2, \ldots, s_N^2)$. By the Rayleign-Ritz theorem [19], minimizing Φ^s with respect to the $\mathbf{y}_i$'s in $\mathcal{Y}$ can be done by finding the $d+1$ eigenvectors of $\mathbf{M}^s$ with the smallest $d+1$ eigenvalues, which is equivalent to solving the generalized eigenvalue problem of $\mathbf{M}\mathbf{v} = \lambda\mathbf{S}^{-1}\mathbf{v}$.[1]

4.4.3 Variants of RLLE Algorithm

The RLLE algorithm described above is one realization of the RLLE approach based on robust statistics. However, this is not the only possibility. Two variants of the RLLE algorithm can be found in a longer version of this paper [25].

4.4.4 Analysis of RLLE Algorithm

The RLLE algorithm makes LLE more robust from two aspects. In the first step of the algorithm, the probability of choosing outliers as neighbors is reduced so that the reconstruction weights reflect more accurately the local geometry of the manifold. In the second step, the undesirable effect of outliers on the embedding result is further reduced by incorporating the reliability scores as weights into the cost function.

One property of the reliability scores is that the mean score $\bar{s}$ over all N points is equal to one, i.e., $\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i = 1$. Therefore, the threshold α used to identify the clean data points can be considered as $\alpha\bar{s}$, which is a certain fraction of the mean reliability score. This makes it easier to set the value of α. Figure 5 shows the cumulative distributions of reliability scores for the S curve, Swiss roll and helix data sets, where the x-axis indicates the reliability scores and the y-axis shows the frequency counts. The α values used in our experiments are shown as vertical dashed lines. As long as α is neither too small nor too large, the performance is not very sensitive to its exact setting.

[1] To ensure that $\mathbf{S}^{-1}$ exists, all s_i's have to be positive. We set s_i to some small positive value ϵ if the reliability score is zero.

Although we make use of the reliability scores to reduce the influence of outliers on the embedding performance of RLLE, the primary objective of the algorithm is not for outlier detection. However, the second variant of the algorithm shows that we can explicitly perform outlier detection if we so wish. The detected outliers may be embedded separately or removed, depending on the application at hand. Unlike the use of the threshold α in the RLLE algorithm, however, outlier detection is more sensitive to the setting of α. In Section 5.3, we will present some experimental results to demonstrate the outlier detection ability of a variant of the RLLE algorithm.

5 Experiments

5.1 Synthetic Data

We apply RLLE to the three artificial data sets described in Section 3 with parameter settings of the experiments depicted in Table 1. The parameter α is set to 0.5 for the Swiss roll and S curve data sets and 0.8 for the helix data set. The embedding results of RLLE are shown in subfigures (c) of Figures 2–4 for comparison with those of LLE in subfigures (b).

RLLE leads to significant improvement in embedding performance over standard LLE for all three data sets. For the Swiss roll and S curve data sets which are embedded from $\mathcal{R}^3$ to $\mathcal{R}^2$, the embedding performance can be seen best by taking into account the coloring of the data points in the plots. It can be seen that the embedding results obtained by RLLE vary the color more smoothly, showing that the local geometry of the data manifolds can be preserved better even when there are outliers in the data. As for the helix data set, RLLE results in a very smooth curve, indicating a direct correspondence between the indices of the data points and their coordinates in the embedding space. However, the irregular mapping discovered by LLE shows that it cannot preserve the neighborhood relationship well.

Analogous to finding the breakdown point of a robust estimator, we further study how varying the noise level influences the embedding performance of the RLLE algorithm. As expected, the embedding performance of RLLE gets worse as the noise level increases. Figure 6 shows the embedding results for the S curve data set as the number of outliers increases gradually from 250 to 300.

We compare our RLLE method with a related method proposed by Zhang and Zha [17]. Their local linear smoothing method tries to detect and remove outliers and hence can be used as a preprocessing step before ordinary manifold learning is performed. Figure 7 shows some results comparing Zhang and Zha's method with our method on the noisy helix data set shown in Figure 7(a). Subfigure (b) shows the data points after applying their smoothing method and (c) shows the 1-dimensional embedding after applying standard LLE to the smoothed data set in (b). The result of RLLE, as shown in Figure 7(d), is apparently better. This may be due to the somewhat heuristic way of their method in setting the weights for weighted PCA in the local smoothing step. Moreover, since their method has to make a hard decision on each data point, some outliers and noise points may not be successfully removed and hence the performance of the subsequent embedding step based on standard LLE may be impaired by them.

To evaluate the performance of RLLE on different types of outliers, we perform more experiments on the helix data set contaminated by noise other than the uniform noise used in the previous experiments. Specifically, we use Gaussian noise which is another common noise type. Figure 8(a) shows the helix data set contaminated by Gaussian noise points. From the results shown in subfigures (b)–(d), we can see that Zhang and Zha's local smoothing method can handle Gaussian noise a little better than uniform noise, while RLLE can give results comparable to the local smoothing method.

5.2 Handwritten Digits from MNIST Database

To illustrate the effectiveness of RLLE on high-dimensional real-world data, we perform experiments on handwritten digits from the well-known MNIST database. [2] The digits in the database have been size-normalized and centered to 28×28 gray-level images, so the dimensionality of the digit space is 784. In

[2] `http://yann.lecun.com/exdb/mnist/`

our experiments, we randomly choose 1,000 images of digit "8" from a total of 60,000 digit images in the MNIST training set.

Figure 9 shows the result of using LLE to embed the data set onto $\mathcal{R}^2$. We traverse along two paths within the projected manifold. The first path is the horizontal (blue) path and the second path is the vertical (green) path. Both start from the arrow-pointed images. As we can see, the first dimension appears to describe the slant of the digits, while the second dimension describes the change of digit width. The slant and width changes along the paths are quite smooth, showing that LLE can preserve the local geometry well when projecting the digits onto $\mathcal{R}^2$.

In order to add some outliers to the original digit data set, we randomly select 50 (5%) digits and change the gray-level values of 20 randomly chosen pixel locations for each digit by inverting each value.[3] Some of the noisy images generated are shown in Figure 10. These noisy images serve as outliers in the following experiments.

After adding outliers to the original data set, LLE and RLLE are applied to obtain low-dimensional embeddings in $\mathcal{R}^2$. The number of nearest neighbors K is equal to 10 and the parameter α of RLLE is set to 0.5. Figures 11 and 12 show the results of LLE and RLLE, respectively. It is easy to see that RLLE is superior to LLE in revealing the continuous changes in slant and width of the digits in $\mathcal{R}^2$.

5.3 Wood Texture Images from USC-SIPI Database

Besides handwritten digit images, we also study real-world wood texture images obtained from the USC-SIPI image database.[4] The images used in our experiments are rotated texture images of four different orientations or rotation angles (0°, 60°, 120°, and 90°). The original images are of size 512×512 captured using a digital camera. We divide each of the original images into 841 ($= 29 \times 29$) partially overlapping blocks of size 64×64. Thus the resulting wood texture data set contains a total of 3,364 images each of 4,096 ($= 64 \times 64$)

[3] A gray-level value v is inverted by replacing it with $255 - v$.

[4] `http://sipi.usc.edu/services/database/`

dimensions. Figure 13 shows 10 examples of each orientation class.

Figure 14(a) shows the result when LLE embeds the clean data onto $\mathcal{R}^2$. As we can see, texture images of four different orientations are generally well separated in the embedding space. Then, we randomly select 200 images (50 images for each class) and add a knot to each image. The knot is randomly selected from five small knot images extracted from real wood images. The location of the knot is randomly determined but its orientation follows that of the wood texture. These artificially created noisy images act as outliers in the subsequent experiments. Some examples of these noisy images are shown in Figure 15.

LLE and RLLE are then applied to the wood texture data set with noisy images added. As before, the number of nearest neighbors K is equal to 10 and the parameter α of RLLE is set to 0.5. Figure 14(b) and Figure 14(c) show the embedding results of LLE and RLLE. It is easy to see that RLLE outperforms LLE in preserving the separation between clusters and the data distribution within each cluster.

The above experiments show that RLLE is more robust against outliers in the data than LLE. This is likely due to the ability of RLLE, which is based on a robust PCA (RPCA) algorithm, in detecting the outliers and reducing their undesirable effect on the embedding performance. To verify this, we further perform some experiments to assess the outlier detection ability of RLLE. We compare it with a simple outlier detection method, which is based on the straightforward idea that a data point is more likely to be an outlier if the size of the neighborhood containing a certain number of nearest neighbors is large. More specifically, for each data point, we compute the size of its neighborhood just large enough to cover the 10 nearest neighbors. The larger the neighborhood size, the more likely the data point is an outlier. We conduct the following experiments using this simple neighborhood-based method as well as RPCA based on reliability scores for outlier detection.

We use the true positive (TP) rate and false positive (FP) rate as performance measures. The TP rate measures the chance that an outlier is correctly detected, while the FP rate measures the chance that a clean data point is incorrectly detected as an outlier. Figure 16 shows the receiver operating characteristic (ROC) curves comparing the RPCA outlier detection method and the simple neighborhood-based outlier detection method for both the S curve data set and the wood texture data set. Since a larger area under curve (AUC) implies better performance, we can see that the RPCA outlier detection method is significantly better for the S curve data set. RPCA is also better for the wood texture data set although the difference is not as significant. Besides the ROC curves, we also compare the two outlier detection methods in terms of the signal-to-noise ratio (SNR). The SNR is defined as $20 \log \mathrm{TN}/\mathrm{FN}$ (in dB), where TN (true negative) and FN (false negative) denote the number of true clean data points (signal) and the number of true outliers (noise), respectively, after outlier detection and removal. The results are shown in Figure 17 and Tables 2 and 3. We can see that the performance difference between the two methods is larger when more points are detected as outliers and removed. While the simple neighborhood-based method is as good as the RPCA method in removing trivial outliers, it is not as effective in removing the less obvious ones accurately when we intend to remove more outliers from the data (and hence have to include the less obvious ones as well).

6 Concluding Remarks

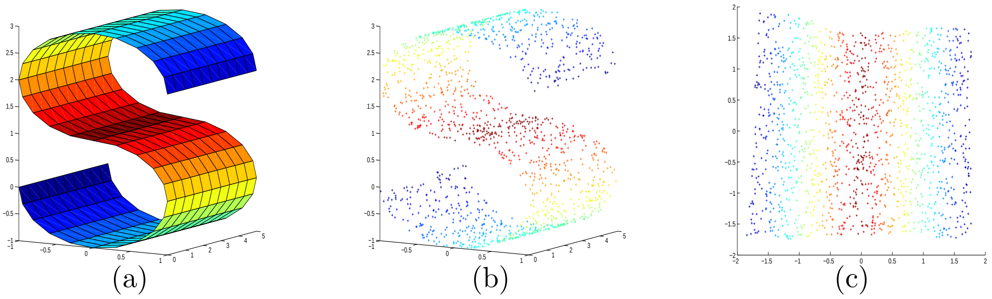
In this paper, we have proposed a robust version of LLE, called RLLE, that is very robust even in the presence of outliers. RLLE first performs local robust PCA on the data points in the manifold using a weighted PCA algorithm. A reliability score is then obtained for each data point to indicate how likely it is a clean data point (i.e., non-outlier). The reliability scores are then used to constrain the locally linear fitting procedure and generalize the subsequent embedding procedure by incorporating the reliability scores as weights into the cost function. The undesirable effect of outliers on the embedding result can thus be largely reduced. Experimental results on both synthetic and real-world data show the efficacy of RLLE.

Despite the robustness of RLLE against outliers, it should be pointed out that the computational requirement of it is significantly higher than that of LLE. The bottleneck lies in the computation of the weights a_i's by RPCA. Since the IRLS procedure has to be executed for each data point, multiple iterations are usually needed. As for the subsequent embedding procedure, its computational demand is comparable to that of LLE, which is much lower. Our future work will try to introduce approximations to speed up the RPCA procedure.

It should be remarked that the same ideas proposed in this paper for LLE may also be extended to make other NLDR methods, such as Isomap, more robust. This is a potential direction for future research. Other possible research directions include improvements of the current RLLE algorithm, such as determining the parameter α automatically and reducing the computational complexity. On the application side, we will consider more real-world applications, within and beyond areas in computer vision, image processing, and text analysis, that can benefit from the robustness of NLDR algorithms.

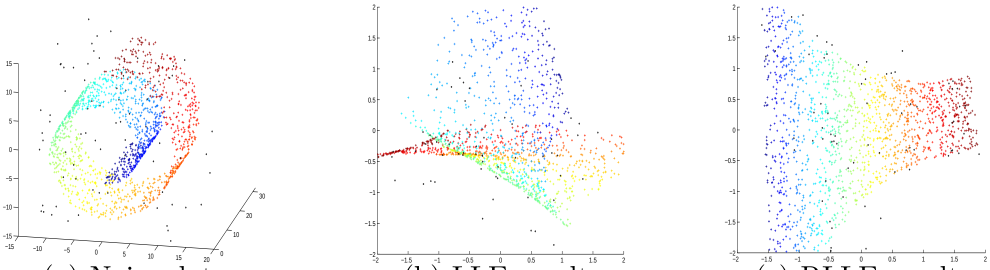
Acknowledgments

The research described in this paper has been supported by two research grants, CA03/04.EG01 (which is part of HKBU2/03/C) and HKUST6174/04E, from the Research Grants Council of the Hong Kong Special Administrative Region, China.



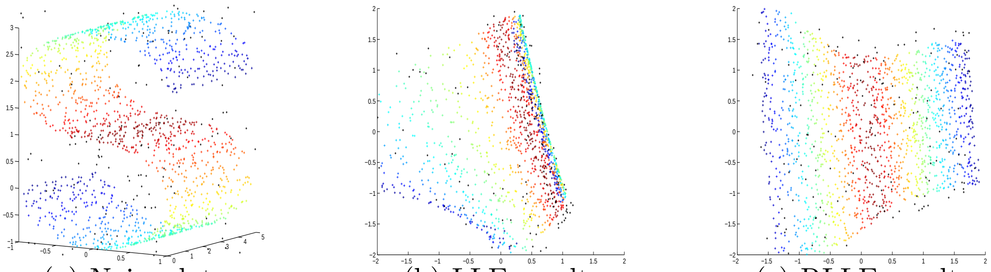
(a) (b) (c)

Fig. 1. LLE applied to the S curve data set. (a) S curve manifold in $\mathcal{R}^3$; (b) 1,500 data points randomly sampled from the manifold; (c) LLE result for embedding space in $\mathcal{R}^2$. Nearest neighbors ($K = 15$) are determined based on Euclidean distance.



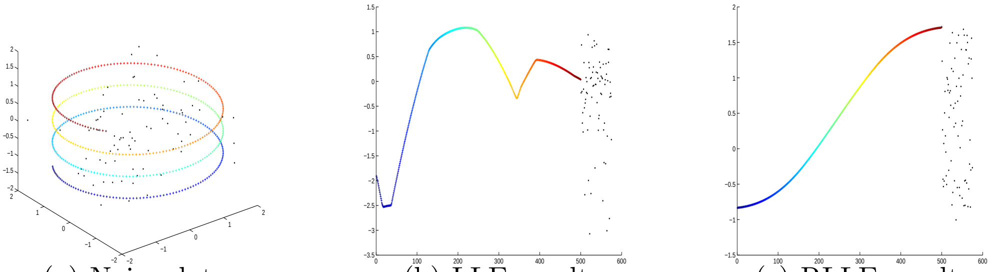
(a) Noisy data (b) LLE result (c) RLLE result

Fig. 2. LLE/RLLE applied to the noisy Swiss roll data set.



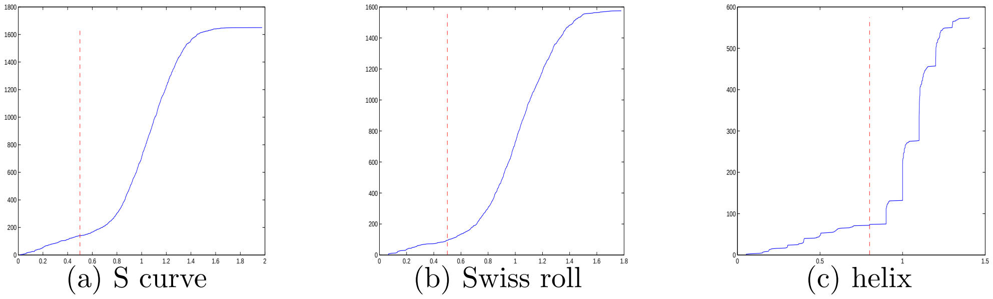
(a) Noisy data (b) LLE result (c) RLLE result

Fig. 3. LLE/RLLE applied to the noisy S curve data set.



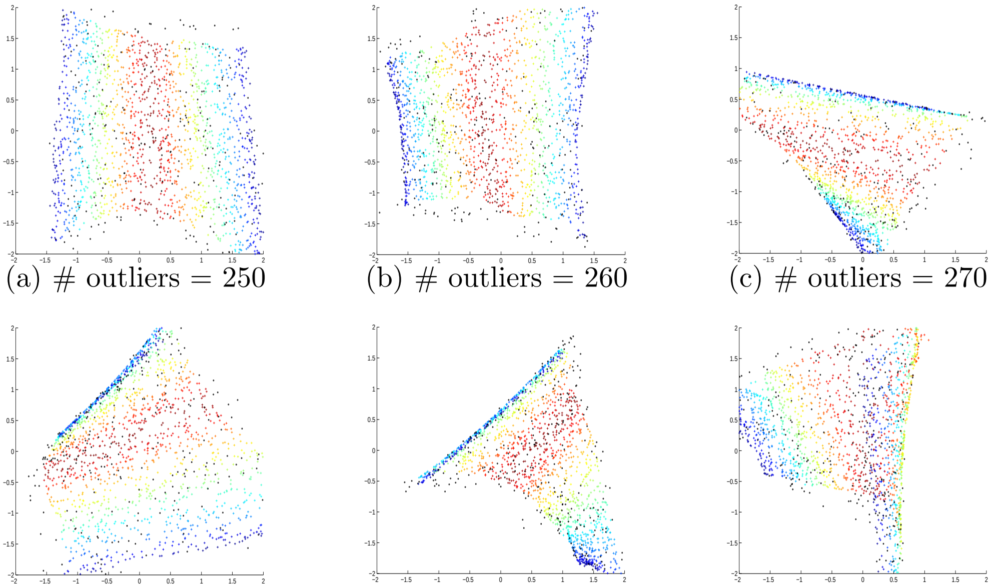
(a) Noisy data (b) LLE result (c) RLLE result

Fig. 4. LLE/RLLE applied to the noisy helix data set. The x-axis in (b) and (c) shows the indices of the data points (including outliers) and the y-axis shows their coordinates in the embedding space $\mathcal{R}^1$.



(a) S curve (b) Swiss roll (c) helix

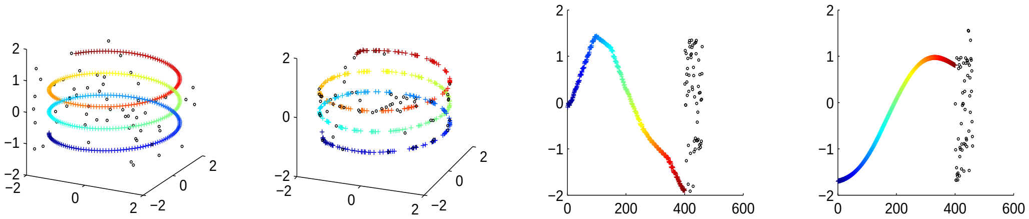
Fig. 5. Cumulative distributions of reliability scores for the S curve, Swiss roll and helix data sets.



(a) # outliers = 250 (b) # outliers = 260 (c) # outliers = 270

(d) # outliers = 280 (e) # outliers = 290 (f) # outliers = 300

Fig. 6. RLLE applied to the noisy S curve data set with different noise levels.



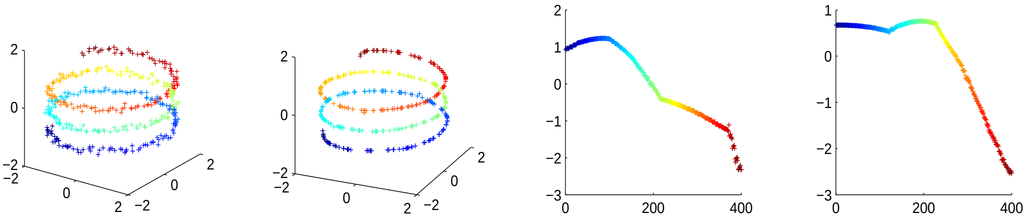
(a) Noisy data

(b) smoothing result

(c) LLE after smoothing

(d) RLLE result

Fig. 7. Zhang and Zha's method/RLLE applied to the noisy helix data set.



(a) Noisy data

(b) smoothing result

(c) LLE after smoothing

(d) RLLE result

Fig. 8. Zhang and Zha's method/RLLE applied to the helix data set contaminated by Gaussian noise.

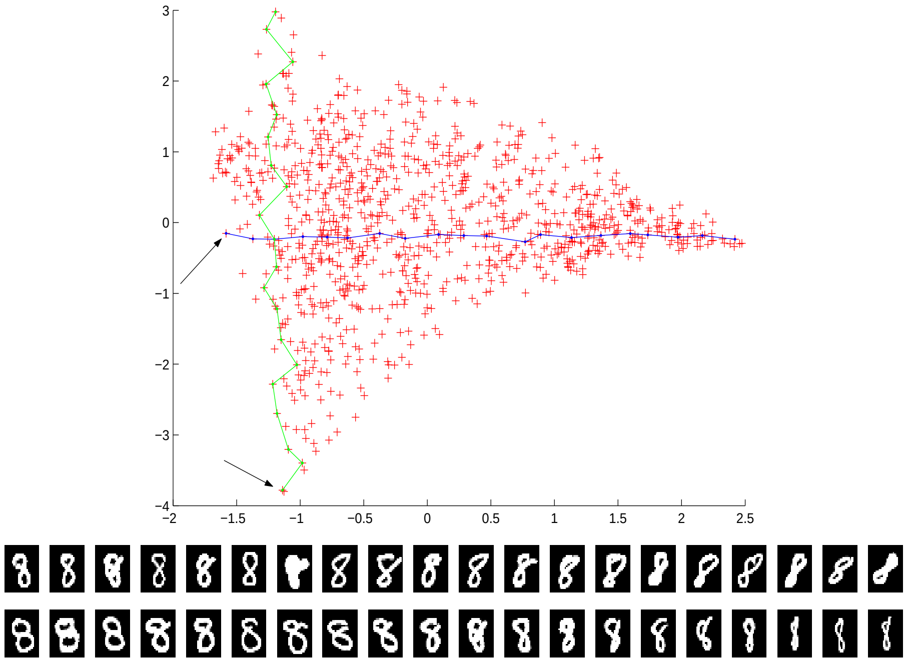


Fig. 9. LLE on handwritten digit “8”. Top: embedding of digit images onto $\mathcal{R}^2$. Bottom: images corresponding to points along the paths linked by solid lines.

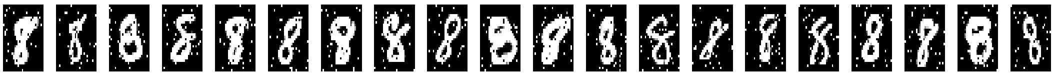


Fig. 10. Some noisy images generated to serve as outliers.

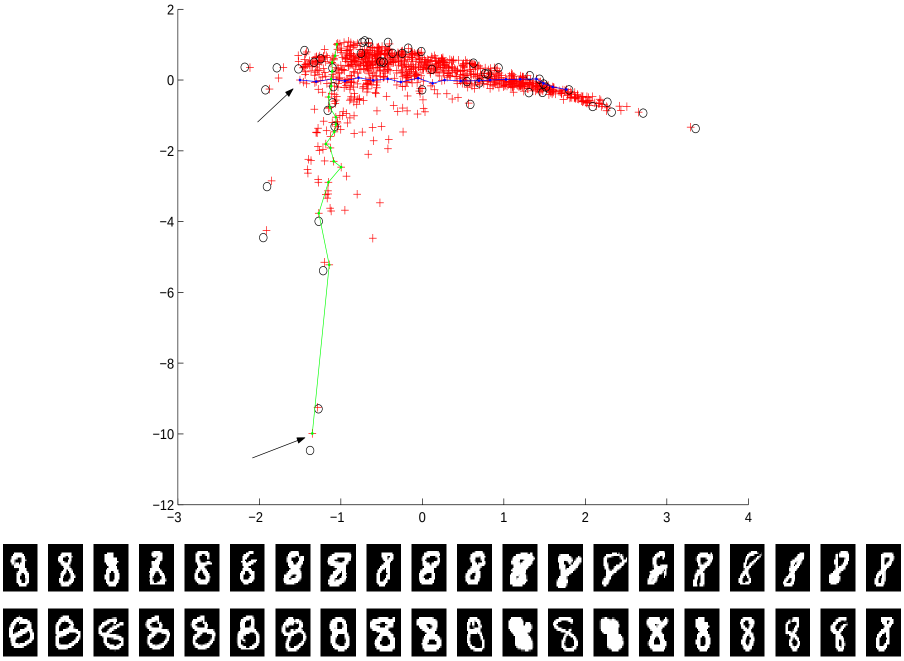


Fig. 11. LLE on handwritten digit “8” with generated outliers. Top: embedding of digit images onto $\mathcal{R}^2$ (the black circles represent the outliers). Bottom: images corresponding to points along the paths linked by solid lines.

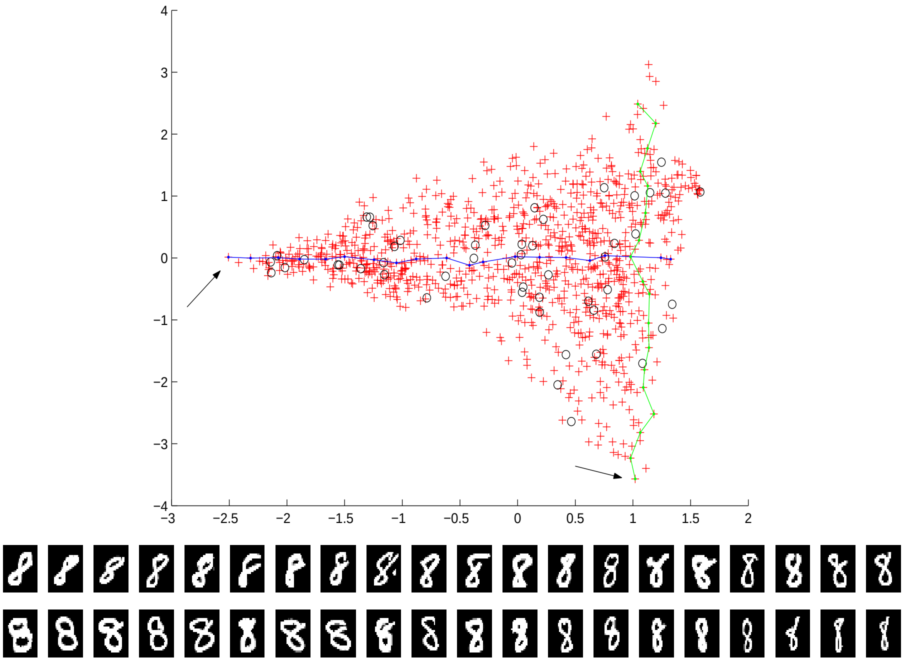


Fig. 12. RLLE on handwritten digit “8” with generated outliers. Top: embedding of digit images onto $\mathcal{R}^2$ (the black circles represent the outliers). Bottom: images corresponding to points along the paths linked by solid lines.

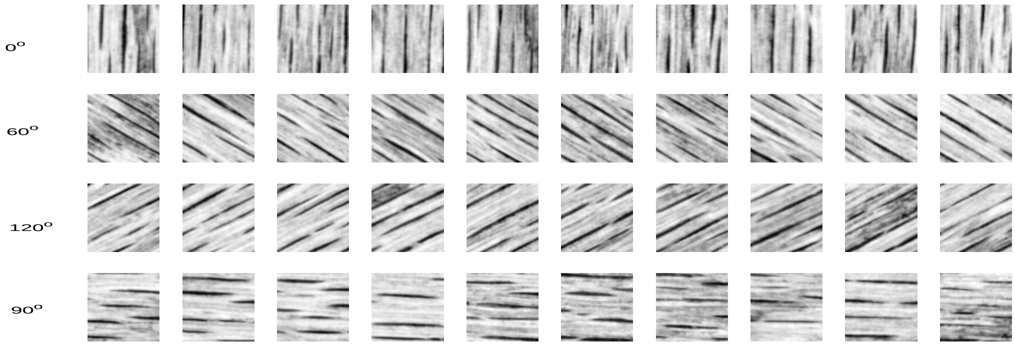
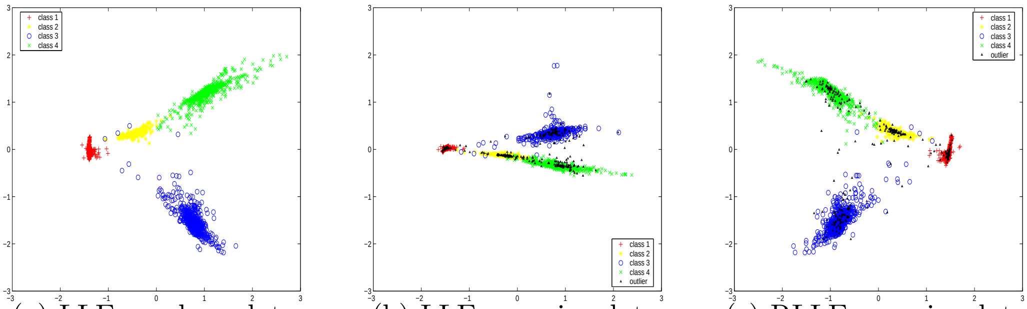


Fig. 13. Wood texture images of four different orientations.



(a) LLE on clean data (b) LLE on noisy data (c) RLLE on noisy data

Fig. 14. LLE/RLLE applied to the wood texture data set.

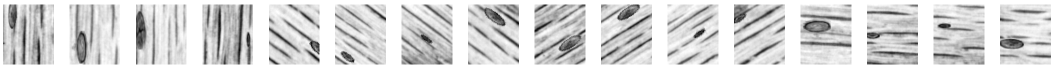
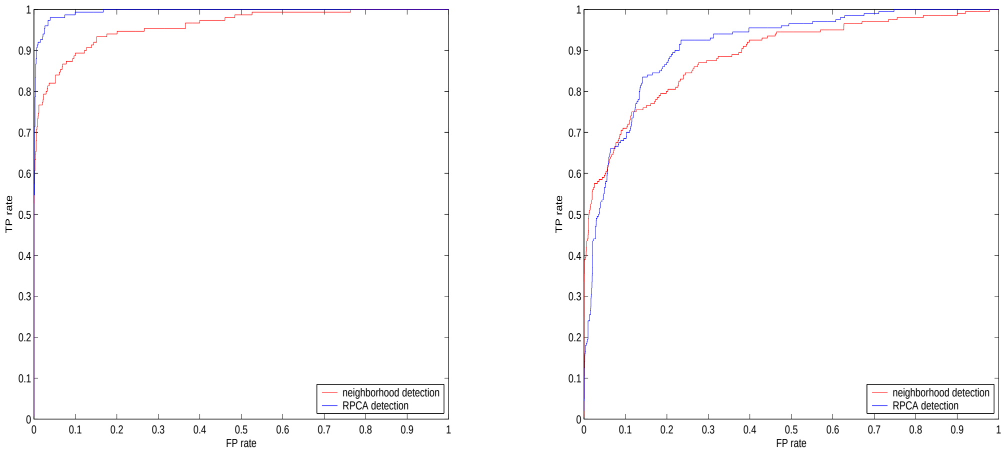


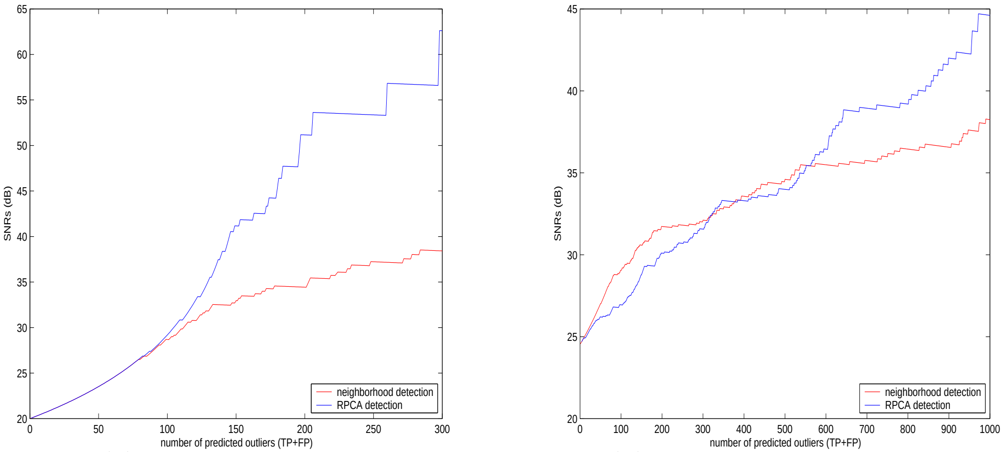
Fig. 15. Some noisy wood texture images generated to serve as outliers.



(a) S curve data set

(b) Wood texture data set

Fig. 16. ROC curves for the S curve and wood texture data sets.



(a) S curve data set

(b) Wood texture data set

Fig. 17. SNR curves for the S curve and wood texture data sets.

Table 1
Parameter settings of the LLE/RLLE experiments reported in Figures 2–4. The percentage of outlier points is computed with respect to the number of clean data points.

Parameter	Swiss roll	S curve	Helix
Dimensionality of data space (D)	3	3	3
Dimensionality of embedding space (d)	2	2	1
Number of nearest neighbors (K)	15	15	10
Number of clean data points	1500	1500	500
Number (percentage) of outlier points	75(5%)	150(10%)	75(15%)
Minimum distance	2	0.2	0.3

Table 2
SNRs (in dB) of the denoised S curve data set using the neighborhood-based and RPCA outlier detection methods.

# predicted outliers (TP+FP)	Neighborhood-based	RPCA
50	23.5218	23.5218
100	28.6856	29.1902
150	32.9583	41.1674
200	34.4368	51.7622
250	37.2288	53.3615

Table 3
SNRs (in dB) of the denoised wood texture data set using the neighborhood-based and RPCA outlier detection methods.

# predicted outliers (TP+FP)	Neighborhood-based	RPCA
200	31.7264	30.0977
400	33.5662	33.2973
600	35.4888	36.4386
800	36.4453	39.1935
1000	38.2455	44.8958

References

[1] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[2] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

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

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, USA, 2002.

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

[6] M. Belkin and P. Niyogi. Using manifold structure for partially labelled classification. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 929–936. MIT Press, Cambridge, MA, USA, 2003.

[7] M. Brand. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 547–552, Acapulco, Mexico, 9–15 August 2003.

[8] D. de Ridder and R.P.W. Duin. Locally linear embedding for classification. Technical Report PH-2002-01, Pattern Recognition Group, Department of Imaging Science and Technology, Delft University of Technology, Delft, The Netherlands, 2002.

[9] V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, USA, 2003.

[10] Dennis DeCoste. Visualizing Mercer kernel feature spaces via kernelized locally-linear embeddings. In *Proceedings of the Eighth International Conference on Neural Information Processing*, Shanghai, China, 14–18 November 2001.

[11] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. Pietikäinen. Beyond locally linear embedding algorithm. Technical Report MVG-01-2002, Department of Electrical and Information Engineering, University of Oulu, Oulu, Finland, September 2002.

[12] M. Polito and P. Perona. Grouping and dimensionality reduction by locally linear embedding. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1255–1262. MIT Press, Cambridge, MA, USA, 2002.

[13] Y.W. Teh and S. Roweis. Automatic alignment of local representations. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 841–848. MIT Press, Cambridge, MA, USA, 2003.

[14] J.J. Verbeek, N. Vlassis, and B. Kröse. Fast nonlinear dimensionality reduction with topology preserving networks. In *Proceedings of the Tenth European Symposium on Artificial Neural Networks*, pages 193–198, Bruges, Belgium, 24–26 April 2002.

[15] H. Zha and Z. Zhang. Isometric embedding and continuum Isomap. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 864–871, Washington, DC, USA, 21–24 August 2003.

[16] D. de Ridder and V. Franc. Robust manifold learning. Technical Report CTU-CMP-2003-08, Department of Cybernetics, Czech Technical University, Prague, Czech Republic, April 2003.

[17] Z. Zhang and H. Zha. Local linear smoothing for nonlinear manifold learning. Technical Report CSE-03-003, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, 2003.

[18] A. Hadid and M. Pietikäinen. Efficient locally linear embeddings of imperfect manifolds. In *Proceedings of the Third International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 188–201, Leipzig, Germany, 5–7 July 2003.

[19] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 1990.

[20] P.J. Huber. *Robust Statistical Procedures.* SIAM, Philadelphia, PA, USA, 1977.

[21] F. De la Torre and M.J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1/2):117–142, 2003.

[22] P.W. Holland and R.E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics: Theory and Methods*, A6(9):813–827, 1977.

[23] P.J. Huber. Robust regression: asymptotics, conjectures, and Monte Carlo. *Annals of Statistics*, 1(5):799–821, 1973.

[24] M. Revow, C.K.I. Williams, and G.E. Hinton. Using generative models for handwritten digit recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):592–606, 1996.

[25] H. Chang and D.Y. Yeung. Robust locally linear embedding. Technical Report HKUST-CS05-12, Hong Kong University of Science and Technology, Department of Computer Science, Clear Water Bay, Kowloon, Hong Kong, July 2005. `ftp://ftp.cs.ust.hk/pub/techreport/05/tr05-12.ps.gz`.

About the Author – DIT-YAN YEUNG received his B.Eng. degree in Electrical Engineering and M.Phil. degree in Computer Science from the University of Hong Kong, and his Ph.D. degree in Computer Science from the University of Southern California in Los Angeles. He was an Assistant Professor at the Illinois Institute of Technology in Chicago before he joined the Department of Computer Science at the Hong Kong University of Science and Technology, where he is currently an Associate Professor. His current research interests are in machine learning and pattern recognition.

About the Author – HONG CHANG received her Bachelor degree and M.Phil. degree in Computer Science from Hebei University of Technology and Tianjin University, China, respectively. She is currently a PhD student in the Department of Computer Science at the Hong Kong University of Science and Technology. Her main research interests include semi-supervised learning, nonlinear dimensionality reduction and related applications.