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
{hongch,dyyeung}@cs.ust.hk

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 embed it onto a low-dimensional embedding space, using tractable linear algebraic techniques that are easy to implement and are not prone to local minima. Unfortunately, these NLDR methods are not robust to outliers in the data, yet so far very little has been done to address the outlier 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 a method for outlier detection and then modify the original LLE algorithm to become a robust version called robust locally linear embedding (RLLE). Experimental results on both synthetic and real-world data show that RLLE is very robust even in the presence of outliers.

Keywords: nonlinear dimensionality reduction, manifold learning, locally linear embedding, principal component analysis, robust statistics

1 Introduction

Dimensionality reduction is concerned with the problem of mapping data 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) [17] and multidimensional scaling (MDS) [6] 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) [27, 28], locally linear embedding (LLE) [24, 25], and Laplacian eigenmap [1, 2] 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 [13] 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 to outliers in the data. Although some extensions have been proposed to the original methods [3, 5, 9, 11, 12, 19, 23, 25, 26, 30, 31, 32], 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) [26], which combines the subspace mixture modeling approach with LLE. A recent work by de Ridder and Duin [10] 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 [22]. 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 [33] 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 [14] studied the outlier problem and proposed a method for robustifying LLE. 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 hence they 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 a method for outlier detection. 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 examples. Robust locally linear embedding (RLLE), which is based on robust PCA techniques, is then proposed in Section 4. 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 $\mathbb{R}^D$. 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 $\mathbb{R}^d$. Let us denote the set of $N$ points in the embedding space $\mathbb{R}^d$ by $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$.

The LLE algorithm [25] 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 $\mathcal{Y}$ 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 $\mathbf{x}_i$:

$$ E_i = \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} \mathbf{x}_j\|^2, \quad (1) $$

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 $\mathbf{x}_j \notin \mathcal{N}_i$. Apparently, minimizing $E_i$ subject to the constraints is a constrained least squares problem.

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

3 Sensitivity of Locally Linear Embedding to Outliers

In this section, we will show through examples how the LLE results are affected by outliers in the data. We use three artificial data sets that are commonly used by other researchers: Swiss roll (Figure 2),
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 [16].

Our robust version of LLE, or RLLE, first performs local robust PCA [7, 8] 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

PCA is among the most popular techniques for multivariate data analysis. In particular, it is a very useful technique for linear dimensionality reduction. To facilitate our subsequent discussions on a robust version of PCA, we first summarize in this subsection the main points of conventional PCA. We will
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Swiss roll</th>
<th>S curve</th>
<th>Helix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensionality of data space ($D$)</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Dimensionality of embedding space ($d$)</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Number of nearest neighbors ($K$)</td>
<td>15</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Number of clean data points</td>
<td>1500</td>
<td>1500</td>
<td>500</td>
</tr>
<tr>
<td>Number (percentage) of outlier points</td>
<td>75(5%)</td>
<td>150(10%)</td>
<td>75(15%)</td>
</tr>
<tr>
<td>Minimum distance</td>
<td>2</td>
<td>0.2</td>
<td>0.3</td>
</tr>
</tbody>
</table>

present it in the context of performing local PCA on the data points in $X$.

As in step 1(a) of the LLE algorithm, $K$ nearest neighbors of each data point $x_i$ are identified. Let $i_1, i_2, \ldots, i_K$ denote the indices of the $K$ neighbors of $x_i$, and hence $x_{i_1}, x_{i_2}, \ldots, x_{i_K}$ denote the $K$ neighbors in $X$. We define the $D \times K$ matrix $X = [x_{i_1}, x_{i_2}, \ldots, x_{i_K}]$. Ideally, if $x_i$ 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 $x_{ij}$ can be linearly projected onto the $d$-dimensional subspace with coordinate vector $z_j = B^T(x_{ij} - d) \in \mathbb{R}^d$, where $d \in \mathbb{R}^D$ is a displacement vector and $B = [b_1, b_2, \ldots, b_d] \in \mathbb{R}^{D \times d}$ is a rotation matrix with $b_j^T b_k = \delta_{jk}$ for $1 \leq j, k \leq K$, i.e., $b_j$’s are orthonormal basis vectors. The low-dimensional image $z_j$ of $x_{ij}$ is represented as $x_{ij} = d + Bz_j = d + BB^T(x_{ij} - d)$ in the original space $\mathbb{R}^D$. Let the difference between $x_{ij}$ and $\hat{x}_{ij}$ be denoted as $\varepsilon_j = x_{ij} - \hat{x}_{ij}$. Standard PCA seeks to find the least squares estimates of $d$ and $B$ by minimizing

$$E_{\text{pca}} = \sum_{j=1}^{K} \|\varepsilon_j\|^2$$

where $Z = [z_1, z_2, \ldots, z_K] \in \mathbb{R}^{d \times K}$ and $\| \cdot \|_F$ denotes the Frobenius norm of a matrix.

The optimization problem has two parts. The first part, which minimizes $E_{\text{pca}}$ with respect to $d$, is an unconstrained optimization problem with the following least squares estimate of $d$

$$d = \frac{1}{K} \sum_{j=1}^{K} x_{ij},$$

which is equal to the sample mean $\mu$ of the $K$ neighbors. The second part of the optimization problem, which minimizes $E_{\text{pca}}$ with respect to $b_1, b_2, \ldots, b_d$ subject to $b_j^T b_k = \delta_{jk}$ for $1 \leq j, k \leq K$, is a constrained optimization problem which can be solved using Lagrange multipliers. The least squares estimates of $b_1, b_2, \ldots, b_d$ are the (orthonormal) eigenvectors of the sample covariance matrix of the $K$ neighbors

$$S = \frac{1}{K} \sum_{j=1}^{K} (x_{ij} - \mu)(x_{ij} - \mu)^T$$

corresponding to the $d$ largest eigenvalues.$^1$

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 to outliers in the data.

---

$^1$Strictly speaking, the unbiased estimator of the covariance matrix should be $\frac{1}{K} \sum_{j=1}^{K} (x_{ij} - \mu)(x_{ij} - \mu)^T$. However, this does not change the results of the eigenvalue problem. In fact, we may even perform eigendecomposition on the scatter matrix $\sum_{j=1}^{K} (x_{ij} - \mu)(x_{ij} - \mu)^T$ to obtain the same results.
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

\[
E_{\text{r-pca}} = \sum_{j=1}^{K} a_j \|e_j\|^2
\]

with respect to \( \mathbf{d}, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \), subject to \( \mathbf{b}_k^T \mathbf{b}_k = \delta_{jk} \) for \( 1 \leq j, k \leq d \). It can easily be shown that (4) and (5) will become

\[
\begin{align*}
\mathbf{d}_\mathcal{A} &= \frac{\sum_{j=1}^{K} a_j \mathbf{x}_{ij}}{\sum_{j=1}^{K} a_j} = \mu_\mathcal{A} \\
\mathbf{S}_\mathcal{A} &= \frac{1}{K} \sum_{j=1}^{K} a_j (\mathbf{x}_{ij} - \mu_\mathcal{A})(\mathbf{x}_{ij} - \mu_\mathcal{A})^T,
\end{align*}
\]

which are the weighted sample mean vector and weighted sample covariance matrix, respectively. 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 \( \mathbf{x}_{ij} \) has a large error norm \( \|e_j\| \), 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 \( e_j \)'s. We also note that \( e_j \)'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) [20], 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 \( \varepsilon_{j}^{(t-1)} = \mathbf{x}_{ij} - \mathbf{d}^{(t-1)} - \mathbf{B}^{(t-1)}(\mathbf{b}_k^{(t-1)})^T(\mathbf{x}_{ij} - \mathbf{d}^{(t-1)}) \), \( 1 \leq j \leq K \);

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

   (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 \( e_j = \|e_j\| \):

\[ a_j = a(e_j) = a(\|e_j\|). \]

Following the ideas of Huber [15], we replace the least squares estimator by a robust estimator that minimizes

\[ E_\rho = \sum_{j=1}^{K} \rho(e_j) = \sum_{j=1}^{K} \rho(\|e_j\|), \]

where \( \rho(\cdot) \) is some convex function.\(^2\) Using the Huber function

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

for some parameter \( c > 0 \), the weight function can be defined as

\[ a(e) = \frac{\psi(e)}{e} = \frac{\rho'(e)}{e} = \begin{cases} 
\frac{1}{2} & |e| \leq c \\
\frac{c}{|e|} & |e| > c
\end{cases} \]

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., \( c = \frac{1}{2K} \sum_{j=1}^{K} e_j \).

\(^2\) A function \( f(x) \) is said to be convex if for any two points \( x_1 \) and \( x_2 \), we have \( f(\frac{x_1 + x_2}{2}) \leq \frac{1}{2} f(x_1) + f(x_2) \).
4.4 RLLE Algorithm

After the IRLS procedure converges to give the weighted least squares estimates of $d$ and $B$, each neighbor $x_j$ 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 $x_i$. For all points not in the neighborhood of $x_i$, their weights are set to 0. After performing robust PCA for all points in $X$, a total reliability score $s_i$ is obtained for each point by summing up the normalized weight values from all robust PCA runs. The total reliability score can then be used as a criterion for outlier detection. The smaller the value of $s_i$ for a point $x_i$, the more likely it is that $x_i$ is an outlier.

We use a threshold $\alpha > 0$ for outlier detection, so that $x_i$ is detected as an outlier if $s_i < \alpha$.

Standard LLE can then be performed on all data points except the outliers detected. To preserve the integrity of the data, we do not remove the detected outliers from the data set. Instead, we project them onto the embedding space together with the data points on the manifold. However, the mapping is computed based on the clean data points only, so that the embedding is not affected by the outliers. Let $\mathcal{X}_o$ denote the set of outliers detected and $\mathcal{X}_d = \mathcal{X} \setminus \mathcal{X}_o$ denote the set of remaining data points. The RLLE algorithm can be described as follows:

1. For each data point $x_i \in \mathcal{X}_d$:
   (a) Find the set $\mathcal{N}_i^d \subset \mathcal{X}_d$ of $K$ nearest neighbors of $x_i$.
   (b) Compute the reconstruction weights of the neighbors that minimize the error of reconstructing $x_i$.

2. Compute the low-dimensional embedding $\mathcal{Y}_d$ for $\mathcal{X}_d$ that best preserves the local geometry by minimizing the following cost function:

$$\Phi' = \sum_{x_i \in \mathcal{X}_d} \| y_i - \sum_{x_j \in \mathcal{N}_i^d} w_{ij} y_j \|^2.$$

3. For each data point $x_i \in \mathcal{X}_o$:
   (a) Find the set $\mathcal{N}_i^o \subset \mathcal{X}_d$ of $K$ nearest neighbors of $x_i$.

   (b) Compute the reconstruction weights of the neighbors that minimize the error of reconstructing $x_i$.

4. Compute the low-dimensional embedding $\mathcal{Y}_o$ for $\mathcal{X}_o$ using the reconstruction weights and $\mathcal{Y}_d$:

$$y_i = \sum_{x_j \in \mathcal{N}_i^o} w_{ij} y_j.$$

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 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. 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. However, the irregular mapping discovered by LLE shows that it cannot preserve the neighborhood relationship well.

5.2 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.\(^3\) The images used in our experiments are rotated texture images of four different orientations or rotation angles. We divide each of the original images ($512 \times 512$) into 841 partially overlapping blocks of size $64 \times 64$. Thus the resulting wood texture data set contains a total of 3,364 images each of 4,096 ($= 64 \times 64$) dimensions.

Figure 5(a) shows the result when LLE embeds the clean data onto $\mathbb{R}^2$. As we can see, texture images

\(^3\)http://sipi.usc.edu/services/database/
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 horizontal dark line to each image. These artificially created noisy images act as outliers in the subsequent experiments. Some examples of these noisy images are shown in Figure 6.

LLE and RLLE are then applied to the wood texture data set with noisy images added. Figure 5(b) and Figure 5(c) show the embedding results of LLE and RLLE. It is easy to see that RLLE is superior to LLE in preserving the separation between clusters and the data distribution within each cluster.

In order to compare the embedding performance of LLE and RLLE more accurately and quantitatively, we apply support vector machines (SVM) [29] with Gaussian kernel to the noisy data set obtained by LLE and RLLE. For comparison, we also include results using ordinary PCA. We perform experiments in both the inductive and transductive settings [29]. The classification results based on 5-fold cross validation are shown in Table 2. As we can see, applying SVM in the embedding spaces obtained by LLE and RLLE is significantly better than that by PCA. Moreover, RLLE gives better classification results than LLE.

The above experiments show that RLLE is more robust to 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 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.

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 7 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. 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 apparent. Besides the ROC curves, we also compare the two outlier detection methods in terms of the signal-to-noise ratios (SNR), which show the consistent comparison results.

5.3 Wood/Knot Texture Images from University of Oulu Databases

In all the previous experiments, noisy data are artificially created and added to the original data sets. While the results clearly show the efficacy of RLLE for both low-dimensional toy data and high-dimensional real-world data, we would like to further
Table 2: SVM classification results on the noisy wood texture data set in the embedding spaces obtained by PCA, LLE, and RLLE.

<table>
<thead>
<tr>
<th>SVM input space</th>
<th>Dimensionality</th>
<th>Classification rate (inductive setting)</th>
<th>Classification rate (transductive setting)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding space by PCA</td>
<td>2</td>
<td>35.12%</td>
<td>35.27%</td>
</tr>
<tr>
<td>Embedding space by PCA</td>
<td>3</td>
<td>45.94%</td>
<td>47.57%</td>
</tr>
<tr>
<td>Embedding space by LLE</td>
<td>2</td>
<td>95.83%</td>
<td>96.43%</td>
</tr>
<tr>
<td>Embedding space by LLE</td>
<td>3</td>
<td>96.74%</td>
<td>97.98%</td>
</tr>
<tr>
<td>Embedding space by RLLE</td>
<td>2</td>
<td>98.12%</td>
<td>98.43%</td>
</tr>
<tr>
<td>Embedding space by RLLE</td>
<td>3</td>
<td>98.40%</td>
<td>99.67%</td>
</tr>
</tbody>
</table>

demonstrate the effectiveness of RLLE using some real-world data with naturally occurring noise. In this subsection, we will apply RLLE to the wood surface inspection problem, which aims at detecting as well as classifying different types of wood and knot textures. Comparison of several linear and nonlinear dimensionality reduction problems, including PCA, MDS, LLE, Isomap, generative topographic mapping (GTM) [4], and self-organizing map (SOM) [18], for this problem has recently been studied by Niskanen and Silvén [21].

We use the image databases from the University of Oulu for the following experiments. Some wood images from the wood image database are divided into small partially overlapping square regions, from which we randomly select 1,000 sound wood images. From the knot image database, we select and crop the images into the same size as the sound wood images. Knot images from five typical classes of wood defects are selected, containing 56 dry knots, 170 sound knots, 29 encased knots, 20 leaf knots, and 50 edge knots. Sample images from the six different classes are shown in Figure 8. For each of the 1,325 32×32 RGB wood/knot images, values from different color channels are concatenated into one vector with $D = 3,072$.

Figures 9 show the results when embedding the data set onto an embedding space in $\mathbb{R}^2$ using LLE and RLLE, respectively. It is clear that RLLE is superior to LLE in preserving the manifold structures of different texture classes and in separating the different classes from each other.

![Sample wood/knot texture images](http://www.ee.oulu.fi/research/imag/WOOD/)

![LLE/RLLE applied to the wood/knot texture data set.](http://www.ee.oulu.fi/research/imag/KNOTS/)

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
Table 3: SVM classification results on the wood/knot texture data set in the embedding spaces obtained by PCA, LLE, and RLLE.

<table>
<thead>
<tr>
<th>SVM input space</th>
<th>Dimensionality</th>
<th>Classification rate (inductive setting)</th>
<th>Classification rate (transductive setting)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding space by PCA</td>
<td>2</td>
<td>75.82%</td>
<td>83.46%</td>
</tr>
<tr>
<td>Embedding space by PCA</td>
<td>3</td>
<td>80.50%</td>
<td>86.57%</td>
</tr>
<tr>
<td>Embedding space by LLE</td>
<td>2</td>
<td>85.98%</td>
<td>86.67%</td>
</tr>
<tr>
<td>Embedding space by LLE</td>
<td>3</td>
<td>86.83%</td>
<td>87.28%</td>
</tr>
<tr>
<td>Embedding space by RLLE</td>
<td>2</td>
<td>87.75%</td>
<td>89.50%</td>
</tr>
<tr>
<td>Embedding space by RLLE</td>
<td>3</td>
<td>88.26%</td>
<td>90.40%</td>
</tr>
</tbody>
</table>

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). Outliers are then detected and excluded from the procedure of computing the optimal low-dimensional embedding, thus reducing the influence of the outliers on the embedding found. Experimental results on both synthetic and real-world data show the efficacy of RLLE. Besides for embedding applications, the SVM experiments show that RLLE may be used as a preprocessing scheme for improving the subsequent classification results.

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 $\alpha$ automatically and embedding the clean data points and outliers simultaneously in one step by incorporating the reliability scores. 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 authors would like to thank members of the AI Group at the Department of Computer Science of the Hong Kong University of Science and Technology for insightful comments on an earlier version of this paper. Special thanks are due to James Kwok, Ivor Tsang, Nevin Zhang, and Zhihua Zhang.

References


