

## Research Article

# Robust Semi-Supervised Manifold Learning Algorithm for Classification

Mingxia Chen, Jing Wang , Xueqing Li, and Xiaolong Sun

*The School of Computer Science and Technology, Huaqiao University, Xiamen 361021, China*

Correspondence should be addressed to Jing Wang; [wroaring@hqu.edu.cn](mailto:wroaring@hqu.edu.cn)

Received 17 June 2017; Accepted 4 January 2018; Published 1 February 2018

Academic Editor: Nazrul Islam

Copyright © 2018 Mingxia Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the recent years, manifold learning methods have been widely used in data classification to tackle the curse of dimensionality problem, since they can discover the potential intrinsic low-dimensional structures of the high-dimensional data. Given partially labeled data, the semi-supervised manifold learning algorithms are proposed to predict the labels of the unlabeled points, taking into account label information. However, these semi-supervised manifold learning algorithms are not robust against noisy points, especially when the labeled data contain noise. In this paper, we propose a framework for robust semi-supervised manifold learning (RSSML) to address this problem. The noisy levels of the labeled points are firstly predicted, and then a regularization term is constructed to reduce the impact of labeled points containing noise. A new robust semi-supervised optimization model is proposed by adding the regularization term to the traditional semi-supervised optimization model. Numerical experiments are given to show the improvement and efficiency of RSSML on noisy data sets.

## 1. Introduction

The problem of dimensionality reduction, that is, the transformation of high-dimensional data into meaningful low-dimensional features, has arisen much interest of researchers. Recently, there have been much research efforts on developing effective and efficient manifold learning algorithms which can discover the potential intrinsic low-dimensional structures of the high-dimensional data. These algorithms included Isometric Mapping (ISOMAP) [1], Locally Linear Embedding (LLE) [2], Laplacian Eigenmaps (LE) [3], and Local Tangent Space Alignment (LTSA) [4].

The above classical manifold learning methods are all unsupervised learning algorithms; that is, they do not consider the prior information. In many applications, we can get some prior information of the input data. For example, in a classification problem, the class labels of partial data can be obtained. Considering prior information in the form of low-dimensional coordinates of certain sample points, the classical manifold learning methods can be extended to semi-supervised manifold learning methods [5]. And the semi-supervised manifold learning algorithms can yield the

low-dimensional coordinates that bear the same meaning as the prior information.

However, these unsupervised and semi-supervised methods may have a limited efficiency on real-world data, due to large noise or distortion of data. Practically, each method for dimensionality reduction requires certain assumptions on the data manifold to guarantee its expected efficiency. For example, ISOMAP needs a convex embedding domain of manifolds or a relatively uniform data distribution for estimating geodesic distance. LTSA should have neighbor sets that can approximately recover the local tangent spaces. In LLE, the local geometric structure of the manifold should be well determined via local combination of data neighborhoods.

There are some efforts on improving the original algorithms. One line is to preprocess the data set before applying the methods, without any modifications on algorithms. Smoothing the data set by weighted SVD, or equivalently, weighted PCA to reduce data noise before performing LTSA is suggested in [6]. In [7], the outliers are first detected by the histogram analysis of the neighborhood distances of each point, and the locally smoothed values of data are then computed using the linear error-in-variables (EIV) model.

A fast outlier detection method for high-dimensional data sets is proposed in [8]. It also employs a local smoothing method and introduces a weighted global function to further reduce the undesirable effect of outliers and noise on the embedding results. These algorithms can be also improved by adaptively selected neighborhoods [9]. The other line is to adjust some details of algorithms. For example, multiple local weight vectors are used to solidify the structures determined by neighborhoods in [10]. In [11], the influence of noisy points on the reconstruction is greatly reduced by solving a new local optimization model. In [12], a robust DLPP version based on L1-norm maximization is proposed. In [13], the short-circuit errors can be reduced by solving the problem of selecting the right number and position of landmarks automatically. A robust version of LTSA is proposed in [14] to further reduce the influence of noise on embedding results by endowing clean data points and noise data points with different weights into local alignment errors. In [15], an out-of-sample extension framework for a global manifold learning algorithm (ISOMAP) that uses temporal information in out-of-sample points in order to make the embedding more robust to noise and artifacts is proposed.

Although the improved manifold learning algorithms are more robust against noise than the original algorithms, few works are done on the semi-supervised algorithms [16]. In fact, the undesirable effect caused by noise is more complicated in the semi-supervised problem. Firstly, it is difficult to accurately explore the local geometric structures when the local neighborhoods contain noisy points. Secondly, the provided prior information may be inexact for noisy points. And the constructed low-dimensional coordinates using the inexact prior information may be far from the real on-manifold coordinates of the sample points. The first issue can be solved by constructing noise-free neighbor sets [7–9] or constructing robust local geometric structures of the noisy neighbor sets [10, 11, 17]. And we do not extend the topic regarding the first issue in the paper.

We focus on the second issue in the paper. We estimate the noise levels of the sample points which reflect the confidence levels in the prior information. Then we construct a new semi-supervised optimization model to reduce the undesirable effect of the inexact prior information with low confidence levels. A framework for robust semi-supervised manifold learning (RSSML) is proposed by solving the new semi-supervised optimization model.

The rest of this paper is organized as follows. In Section 2, we give a brief review of semi-supervised manifold learning. In Section 3, we show how to extend the semi-supervised manifold learning algorithms so that they can handle inexact information for noisy points. The framework for robust semi-supervised manifold learning (RSSML) is presented in the section. After that, we give numerical experiments in Section 4 to show the effectiveness of RSSML.

## 2. A Brief Review of Semi-Supervised Manifold Learning

Our work is an extension of semi-supervised manifold learning (SSML). In this section, we give a brief review

of (SSML) [5]. Assume that we are given a data set  $X = \{x_1, \dots, x_N\}$  (possibly with noise) from a  $d$ -dimensional manifold. Without loss of generality, suppose that the prior information of the first  $m$  points is known. And denote by  $\hat{T}_1 = [t_1^*, \dots, t_m^*] \in R^{d \times m}$  the constructed low-dimensional coordinates using the prior information. The goal of SSML is to calculate the unknown low-dimensional coordinates  $T_2 = [t_{m+1}, \dots, t_N] \in R^{d \times (N-m)}$ . SSML proceeds in the following steps.

*Step 1* (finding local neighborhoods). Determine the neighbor set  $\mathcal{N}_i = \{x_{i_1}, \dots, x_{i_k}\}$  for each  $x_i$  ( $i = 1, \dots, N$ ).

*Step 2* (extracting local geometry). The local geometry of the determined neighbor set can be extracted by solving the classical local optimization methods [1–4]. Take LLE as an example; the local geometry is characterized by the linear combination coefficients which can be computed by minimizing the least square optimization model:

$$\begin{aligned} \Gamma(W) &= \sum_{i=1}^N \left\| x_i - \sum_{x_j \in \mathcal{N}_i} W_{ji} x_j \right\|^2, \\ \text{s.t. } \sum_j W_{ji} &= 1. \end{aligned} \quad (1)$$

*Step 3* (constructing semi-supervised optimization model). In the unsupervised manifold learning algorithms, the global low-dimensional coordinates are calculated by solving the embedding cost functions which can preserve the extracted local geometries. For example, in LLE, the low-dimensional coordinates  $t_i$  ( $i = 1, \dots, N$ ) can be computed by minimizing the embedding cost function:

$$C(t_1, \dots, t_N) = \sum_{i=1}^N \left\| t_i - \sum_{x_j \in \mathcal{N}_i} W_{ji} t_j \right\|^2. \quad (2)$$

Different from the embedding cost functions in the unsupervised manifold learning algorithms, a regularization term concerning the prior information is added such that the low-dimensional coordinates can obey the prior information. As in semi-supervised LLE, the low-dimensional coordinates are obtained by minimizing the semi-supervised optimization model:

$$\sum_{i=1}^N \left\| t_i - \sum_{x_j \in \mathcal{N}_i} W_{ji} t_j \right\|^2 + \beta \sum_{i=1}^m \|t_i - t_i^*\|^2, \quad (3)$$

where  $\beta$  is the regularization parameter that reflects the confidence level in prior information.

Semi-supervised manifold learning has been widely used in many real-life applications such as face recognition [18], remote sensing image classification [19], object tracking [5], and data visualization [18]. Figure 1 illustrates some of the real-life applications. The available prior information and the computed embedding coordinates are different in different



FIGURE 1: Some real-life applications of semi-supervised manifold learning. (a) Face recognition, (b) remote sensing image classification, (c) object tracking, and (d) data visualization.

applications. For example, in the applications of face recognition and remote sensing image classification, the known low-dimensional coordinates are constructed according to the labels of the training data points, and SSML aims to predict the labels of the remaining data points using the known low-dimensional coordinates (see Remark 2 in the next section for the way of constructing low-dimensional coordinates and predicting the labels). In object tracking,

SSML aims to recover the real locations of the object, using the given locations of the object in certain frames. In data visualization, SSML projects the data points to 2-dimensional or 3-dimensional embedding space to discover the hidden relations of the data points, using the given 2-dimensional or 3-dimensional coordinates of the training points.

Generally, semi-supervised manifold learning algorithms work well if the data sets are well sampled from a manifold.

In the situation,  $\beta \rightarrow \inf$ , and the minimization of (3) is equivalent to

$$\begin{aligned} \min_{t_{m+1}, \dots, t_N} \quad & \sum_{i=1}^N \left\| t_i - \sum_{x_j \in \mathcal{N}_i} W_{ji} t_j \right\|^2, \\ \text{subject to} \quad & t_i = t_i^*, \\ & i = 1, \dots, m. \end{aligned} \quad (4)$$

When the data sets contain noise, the effectiveness of SSML will be significantly decreased. This is because the noise level of each sample point may be different. For the sample points containing large noise, their prior information may not be trustworthy, and the regularization parameter  $\beta$  tends to be small. For those points only containing small noise, we are confident in the provided prior information, and  $\beta$  tends to be large. A fixed  $\beta$  cannot reflect the different confidence levels in the prior information of each point. It is desirable to construct a robust semi-supervised optimization model against noise.

### 3. Robust Semi-Supervised Manifold Learning Algorithm

In this section, a framework for robust semi-supervised manifold learning algorithm (RSSML) will be proposed. Since the prior information may be inexact for the noisy points, the major problem is how to deal with inexact prior information according to the different noise levels of the sample points. Note that the noise levels of the sample points are unknown generally. It is desirable to measure the noise levels of the sample points before proposing the robust semi-supervised optimization model.

**3.1. Measure the Noise Level.** Recently, some work has been done to measure the noise levels of the points [20–23]. In this paper, we measure the noise levels by the outlier detection algorithm based on reconstruction weights (ODBRW), due to little computation cost, low parameter requirement, and high effectiveness [20]. The ODBRW method is applied only on the training points  $x_1, \dots, x_m$ , which consists of the following steps.

**Step 1** (constructing the edge point sets). Search  $k$ -nearest neighbors (KNN) of each  $x_i$ ,  $i = 1, \dots, m$ , and determine the neighbor set  $\mathcal{N}_i = \{x_{i_1}, \dots, x_{i_k}\}$  firstly. Then select the edge points from  $\mathcal{N}_i$  by

$$\begin{aligned} \mathcal{EP}(x_i) = \left\{ y \in \mathcal{N}_i \mid (x_i - x_{i_j})^T (y - x_{i_j}) \geq 0, \right. \\ \left. j = 1, \dots, k \right\}, \end{aligned} \quad (5)$$

which requires that angles between any adjacent edges  $\langle x_i, x_{i_j} \rangle$  and  $\langle y, x_{i_j} \rangle$  should be acute or right. If the angle between the adjacent edges  $\langle x_i, x_{i_j} \rangle$  and  $\langle y, x_{i_j} \rangle$  is obtuse, it means that  $x_{i_j}$  separates  $x_i$  and  $y$ . A point  $y$  is said to be an

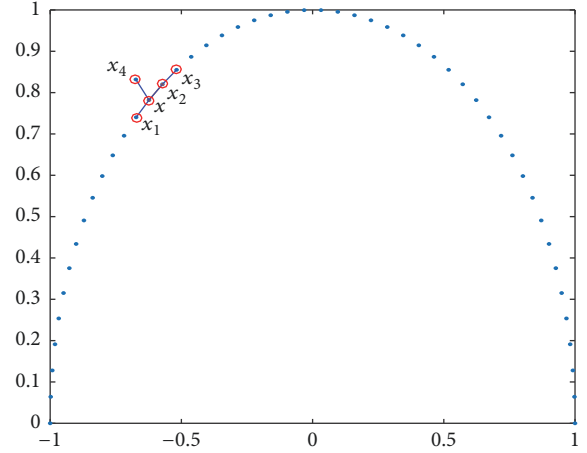


FIGURE 2: An illustrated example of the edge point set. The neighbor set of  $x$  determined by KNN is  $\{x_1, x_2, x_3, x_4\}$ . And the edge point set is  $\{x_1, x_2, x_4\}$ .  $x_3$  is not an edge point of  $x$  since  $x_2$  separates  $x$  and  $x_3$ ; that is, the angle between the adjacent edges  $\langle x, x_2 \rangle$  and  $\langle x, x_3 \rangle$  is obtuse. The edge point set is always  $\{x_1, x_2, x_4\}$  for different neighborhood size  $k$  ( $k \geq 4$ ).

edge point of  $x_i$  if there is no other neighbor  $x_{i_j}$  that separates  $x_i$  and  $y$ . The determined edge point sets are very robust on the neighborhood size  $k$ . See an illustrated example of the edge point set in Figure 2. More explanations about the edge point set can be found in [24].

**Step 2** (calculating reconstruction weights). The reconstruction weights of the edge point set  $\{y_1, \dots, y_{k_i}\}$  for  $x_i$  can be obtained by solving the least square problem

$$\begin{aligned} \min_{w_{ji}} \quad & \left\| x_i - \sum_{j=1}^{k_i} w_{ji} y_j \right\|^2, \\ \text{s.t.} \quad & \sum_{j=1}^{k_i} w_{ji} = 1. \end{aligned} \quad (6)$$

Denote  $G_i = [y_1 - x_i, \dots, y_{k_i} - x_i]$  and  $w_i = [w_{1i}, \dots, w_{k_i i}]^T$ ; the least square problem can be solved by

$$\begin{aligned} G_i^T G_i h_i &= 1_{k_i}, \\ w_i &= \frac{h_i}{1_{k_i}^T h_i}, \end{aligned} \quad (7)$$

where  $h_i, 1_{k_i} \in \mathbb{R}^{k_i}$  and  $1_{k_i}$  is a vector with all ones.

**Step 3** (measuring the noise levels). Form an  $m \times m$  matrix  $H$  by  $h_i$ ,  $i = 1, \dots, m$ ; that is,

$$\begin{aligned} H(J_i, i) &= h_i, \\ H(j, i) &= 0, \\ j &\notin J_i, \end{aligned} \quad (8)$$



where  $J_i$  is the index set of  $\mathcal{EP}(x_i)$ . The noise level of  $x_i$  can be measured by

$$r_i = \sum_{j=1}^m |H_{ij}| + \sum_{j=1}^m |H_{ji}|. \quad (9)$$

**3.2. Robust Semi-Supervised Optimization Model.** It is shown in theory that the smaller  $r_i$  is, the more likely that  $x_i$  tends to be an outlier [20]. For a small  $r_i$ , it means that the prior information on  $x_i$  is not trustworthy. Hence, we hope that the effect of the prior information on computing the embedding coordinates can be reduced. For a large  $r_i$ , the sample point  $x_i$  tends to be a clean point, and we are more confident on its prior information. The computed low-dimensional coordinates should bear similar meaning as the prior information. Based on the above analysis, we construct a new robust semi-supervised optimization model:

$$\min_{T=[t_1, \dots, t_N]} C(T) + \beta \sum_{i=1}^m \|(t_i - t_i^*) r_i\|^2. \quad (10)$$

Here  $T = [t_1, \dots, t_N] \in R^{d \times N}$  are the low-dimensional coordinates of  $x_i$ ,  $i = 1, \dots, N$ , and  $t_i^*$ ,  $i = 1, \dots, m$  are the constructed low-dimensional coordinates using the prior information of the training points  $x_i$ ,  $i = 1, \dots, m$ .  $C(T)$  is the embedding cost function on the manifold, and  $r_i$  is the measured noise level which is normalized to a real number in  $[0, 1]$ . Take LLE as an example; the robust semi-supervised optimization model is

$$\min_{T=[t_1, \dots, t_N]} \sum_{i=1}^N \left\| t_i - \sum_{x_j \in \mathcal{N}_i} W_{ji} t_j \right\|^2 + \beta \sum_{i=1}^m \|(t_i - t_i^*) r_i\|^2. \quad (11)$$

Denote  $\hat{T}_1 = [t_1^*, \dots, t_m^*] \in R^{d \times m}$  and  $T_1 = [t_1, \dots, t_m] \in R^{d \times m}$  and  $D_1 \in R^{m \times m}$  is the diagonal matrix whose elements are  $r_1, \dots, r_m$ . The optimization model (11) can be expressed as

$$\min_T \Phi(T) = \text{trace}(TMT^T) + \beta \|(T_1 - \hat{T}_1) D_1\|_F^2, \quad (12)$$

where  $M \in R^{N \times N}$  and the elements  $M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$  with  $\delta_{ij} = 1$  for  $i = j$  and  $\delta_{ij} = 0$  for  $i \neq j$ .

We can solve the optimization model by setting the gradient of (12) to be zero. Partition  $T = [T_1 \ T_2]$  with  $T_1 \in R^{d \times m}$  and  $T_2 \in R^{d \times (N-m)}$ ; it is easy to get that

$$\begin{aligned} \frac{\partial \Phi(T)}{\partial T} &= 2TM + 2\beta \left[ (T_1 - \hat{T}_1) D_1^2 \ \mathbf{0}_{d \times (N-m)} \right] \\ &= \mathbf{0}_{d \times N}. \end{aligned} \quad (13)$$

Partition

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \quad (14)$$

with  $M_{11} \in R^{m \times m}$ ,  $M_{12} \in R^{m \times (N-m)}$ ,  $M_{21} \in R^{(N-m) \times m}$ ,  $M_{22} \in R^{(N-m) \times (N-m)}$ ; the  $d$ -dimensional

embedding  $T$  can be computed by solving the following linear system of equations:

$$\begin{pmatrix} M_{11} + \beta D_1^2 & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} T_1^T \\ T_2^T \end{pmatrix} = \begin{pmatrix} \beta D_1^2 \hat{T}_1^T \\ \mathbf{0}_{(N-m) \times d} \end{pmatrix}. \quad (15)$$

Based on the above analysis, we propose a new algorithm called robust semi-supervised manifold learning (RSSML) which is summarized as follows.

#### RSSML Algorithm

**Input.** Data set  $X = [x_1, \dots, x_N]$ , low-dimensional coordinates of labeled points  $\hat{T}_1 = [t_1^*, \dots, t_m^*]$ , parameter  $\beta$ , neighborhood size  $k$ .

**Output.** The low-dimensional coordinates  $T$ .

**Step 1** (selecting local neighborhoods). Determine the neighborhood set  $N_i = [x_{i_1}, \dots, x_{i_k}]$  of each point  $x_i$ ;  $J_i = [i_1, \dots, i_k]$  is the index set of the neighbors:  $i = 1, \dots, N$ .

**Step 2** (extracting local geometry). Extract the local geometry  $W$  by some local optimization models of manifold learning. Matrix  $M$  is given by  $M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$  with  $\delta_{ij} = 1$  if  $i = j$  or 0 otherwise.

**Step 3** (calculating noise levels). Obtain the noise levels  $\{r_i\}_{i=1}^m$  for the first  $m$  data points (training points) by the ODBRW method and construct the diagonal matrix  $D_1 = \text{diag}(r_1, \dots, r_m) \in R^{m \times m}$ .

**Step 4** (embedding global coordinates). Compute the low-dimensional embedding coordinates  $T$  by solving the linear system of equations (15).

**Remark 1.** Notice that many unsupervised learning methods can be extended to their semi-supervised versions by the proposed robust semi-supervised manifold learning (RSSML) algorithm. In this paper, we explore the local geometry by solving the least squares (LS) problem of LLE. And the local geometry can be explored by other local optimization methods such as RLLPE and LTSA. We call them RSSML-LLE, RSSML-RLLPE, and RSSML-LTSA.

**Remark 2.** In a classification problem, we are given the label information  $\{l_i\}_{i=1}^m$  of  $m$  training points and  $c$  classes. Without loss of generality, assume that the first  $m$  points are labeled. The low-dimensional coordinates of labeled points  $\hat{T}_1 = [t_1^*, \dots, t_m^*] \in R^{c \times m}$  can be constructed with  $t_{ij} = 1$  if  $x_i$  has label  $l_i = j \in \{1, \dots, c\}$  and  $t_{ij} = 0$  otherwise. Then, the labels of the unlabeled points  $x_i$  ( $i = m+1, \dots, N$ ) can be estimated as  $l_i = \max_j |t_{ij}|$ .

## 4. Experiment Results

To verify the effectiveness of the proposed algorithm RSSML on real-world data, we perform experiments on CMU

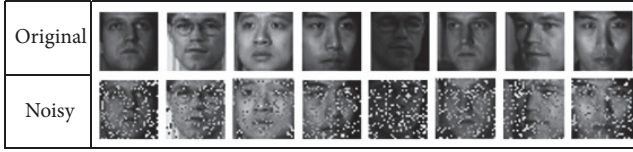


FIGURE 3: Some CMU PIE images (original CMU PIE images and noisy CMU PIE images).

PIE data set [25], Handwritten-Alpha data set [26], and HAND\_SHAPE [27]. For comparison, we apply the unsupervised methods LLE, RLLPE, and LTSA (see [2, 4, 11]), their semi-supervised versions, and the proposed robust semi-supervised versions on the above data sets. In the three real-world examples, some noisy points are also added to test the robustness of RSSML to noisy data sets.

**CMU PIE [25].** The original data set contains 11560 samples of 68 individuals in  $32 \times 32$  gray-scale image. In the experiment, we randomly selected 160 samples from 10 individuals (for a total of 1,600 samples). Some samples are plotted in Figure 3.

**Handwritten-Alpha [26].** The data set (HW-Alpha) is extracted from “binaryalphadigs” data set. It consists of 936 images of 26 handwritten alphabets. Each class has 36 images which are of size of  $20 \times 16$ .

**HAND\_SHAPE [27].** The original data set (Cambridge Hand Gesture Data) contains 9 gesture classes in  $320 \times 240$  gray-scale images, which are defined by 3 primitive hand shapes and 3 primitive motions. In this experiment, the target task is to classify different hand shapes. Therefore, the final data set is divided into five groups (see Figure 4). We randomly selected 350 points of each group to form the experiment set.

In the experiments, the parameters of these methods are set as follows. All of the manifold learning algorithms are involved with the neighborhood size parameter. The neighborhood size  $k$  is selected from 8 to 36. For the unsupervised methods LLE, RLLPE, and LTSA, the intrinsic dimension  $d \in \{10, 14, 18, 22, 26, 30\}$  is tried. The different regularization parameter  $\beta \in \{5, 50, 100, \dots, 500\}$  is tried in our robust semi-supervised version of LLE, LTSA, or RLLPE. We only report the best results.

To perform classification, the data sets are firstly projected onto low-dimensional space by these unsupervised methods, and then the Nearest Feature Line (NFL) classifier (see [28, 29]) is used on the low-dimensional embedding results for the recognition. For the semi-supervised methods, the label information of the unlabeled points can be predicted directly.

In the first experiment, some noisy points are added to the original three data sets. We randomly select 10% sample points from the data sets to generate the noisy points. For each selected image, we randomly chose 1/6 pixels from the original image and changed the pixel values from  $v$  to  $255 - v$ . Some of the noisy images are shown in Figure 3. Half of the sample points are randomly selected as training samples and the remaining are used for testing.

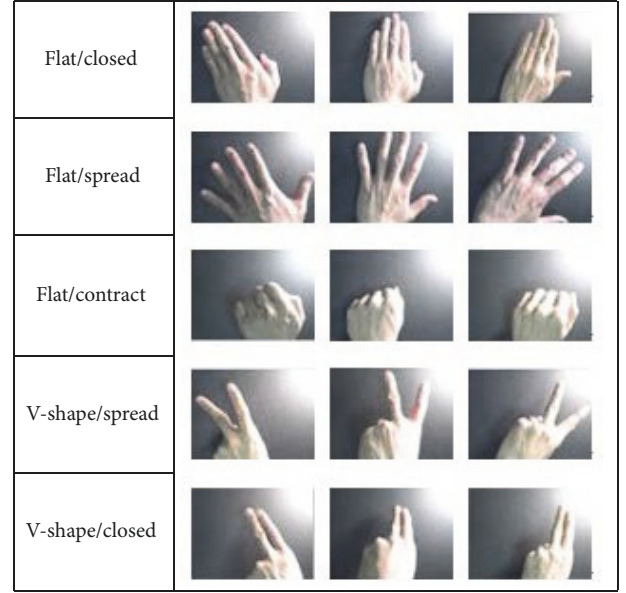


FIGURE 4: The HAND\_SHAPE images that include five kinds of hand gestures.

Table 1 lists the classification rates of the manifold learning methods on the three data sets. It is clear that the unsupervised methods are sensitive to noise, especially for LTSA. When the data sets contain noisy points, LTSA may fail to find a reasonable local tangent space of each data point. For SS-LLE, SS-RLLPE, and SS-LTSA, the classification accuracies can be improved obviously. RSSML-LLE, RSSML-RLLPE, and RSSML-LTSA outperform the semi-supervised manifold learning approaches in the experiments. It is shown that the robustness to noisy points can be improved by the proposed robust semi-supervised model.

To better compare the effectiveness of the above manifold algorithms on noisy data sets, we generate the noisy data with different densities of Gaussian and reverse noise. We randomly select 10% samples from CMU PIE data set to generate the noisy images. The noisy images are generated in two ways. One way is to randomly choose 1/6, 1/8, or 1/12 pixels of each selected image and invert the value from  $v$  to  $255 - v$ . Another way is to add the Gaussian noise to the selected image with different variances: 0.02, 0.05, or 0.1. The experimental results of the unsupervised, semi-supervised, and robust semi-supervised versions of LLE, RLLPE, and LTSA methods on the six experimental sets are shown in Table 2.

As can be seen in Table 2, SS-LLE, SS-RLLPE, and SS-LTSA are more sensitive to noisy points than RSSML (-LLE, -RLLPE, and -LTSA). Under different noise levels, RSSML methods can achieve higher classification accuracies than the other algorithms. It is further shown that RSSML can better handle the noisy points in the experiments. We notice that the classification rates of RSSML are also higher than those of the other methods on the original data set. This is because the original data set may be contaminated by noise in the sampling process. The proposed RSSML methods can also

TABLE 1: The classification rates of the 9 classification methods on three different noisy data sets (CMU PIE, HAND\_SHAPE and HW-Alpha).

METHOD	CMU PIE	HAND_SHAPE	HW-Alpha
LLE + NFL	0.6088	0.8549	0.7030
SS-LLE	0.6613	0.8777	0.7607
<i>RSSML-LLE</i>	<b>0.7763</b>	<b>0.9280</b>	<b>0.7792</b>
RLLPE + NFL	0.6263	0.8411	0.7294
SS-RLLPE	0.6375	0.8617	0.7714
<i>RSSML-RLLPE</i>	<b>0.6963</b>	<b>0.9326</b>	<b>0.7778</b>
LTSA + NFL	0.4488	0.3657	0.3205
SS-LTSA	0.8300	0.8160	0.7500
<i>RSSML-LTSA</i>	<b>0.8500</b>	<b>0.8183</b>	<b>0.7585</b>

TABLE 2: The classification rates of the 9 methods on CMU PIE data set with different noise densities.

METHOD	Reverse-noise			Gauss-noise			noiseless
	1/6	1/8	1/12	0.02	0.05	0.1	
LLE + NFL	0.6088	0.6163	0.5925	0.6213	0.6888	0.6738	0.6263
SS-LLE	0.6613	0.6763	0.7350	0.7025	0.7838	0.8450	0.7363
<i>RSSML-LLE</i>	<b>0.7763</b>	<b>0.7250</b>	<b>0.7388</b>	<b>0.7315</b>	<b>0.8300</b>	<b>0.8613</b>	<b>0.7613</b>
RLLPE + NFL	0.6263	0.6675	0.6438	0.6613	0.7263	0.7050	0.6538
SS-RLLPE	0.6375	0.6788	0.7138	0.7125	0.7488	0.7463	0.7075
<i>RSSML-RLLPE</i>	<b>0.6963</b>	<b>0.6875</b>	<b>0.7275</b>	<b>0.7513</b>	<b>0.7838</b>	<b>0.7875</b>	<b>0.7075</b>
LTSA + NFL	0.4488	0.4325	0.2825	0.2838	0.3488	0.4288	0.6900
SS-LTSA	0.8300	0.8050	0.8425	0.8525	0.8475	0.8288	0.8863
<i>RSSML-LTSA</i>	<b>0.8550</b>	<b>0.8213</b>	<b>0.8438</b>	<b>0.8763</b>	<b>0.8688</b>	<b>0.8375</b>	<b>0.9013</b>

TABLE 3: The classification rates on HAND\_SHAPE data set for the different proportions (Training : Testing).

Training : Testing	5 : 2	3 : 2	1 : 1	2 : 3	2 : 5
LLE + NFL	0.8520	0.8200	0.8594	0.8552	0.8160
SS-LLE	0.9320	0.9057	0.8777	0.8562	0.8360
<i>RSSML-LLE</i>	<b>0.9480</b>	<b>0.9343</b>	<b>0.9280</b>	<b>0.9095</b>	<b>0.8840</b>
RLLPE + NFL	0.8980	0.8443	0.8411	0.8324	0.8128
SS-RLLPE	0.9300	0.8757	0.8617	0.8590	0.8392
<i>RSSML-RLLPE</i>	<b>0.9460</b>	<b>0.9243</b>	<b>0.9326</b>	<b>0.8876</b>	<b>0.8744</b>
LTSA + NFL	0.4200	0.3285	0.3657	0.3657	0.3200
SS-LTSA	0.8580	0.7829	0.8160	0.7951	0.7432
<i>RSSML-LTSA</i>	<b>0.8640</b>	<b>0.7920</b>	<b>0.8223</b>	<b>0.8038</b>	<b>0.7488</b>

reduce the impact of the noisy points that come from the real world. It is interesting that, in some cases, the results on the noisy data sets may be better than those on the original data set. One explanation is that the linear combination coefficients of large noisy points calculated by the local least square optimization model tend to be small. When we added some large artificial noises to the original points, the effect of these points on the reconstruction can be greatly reduced. Hence, these outliers will not destroy the low-dimensional structure of the manifold.

To further quantitatively compare the performance of the above algorithms, the noisy HAND\_SHAPE data (1/6 reverse noise) are divided into five different proportions of training and testing data (2:5, 2:3, 1:1, 3:2, and 5:2). As can be seen in Table 3, our robust versions (of LLE, RLLPE, and

LTSA) have the best performance for different proportions of training and testing data. It is evident that our method can greatly reduce the classification error on the noisy data set.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by NSFC (61370006), NSF of Fujian Province (2014J01237 and 2015J01256), Program for New Century Excellent Talents in Fujian University (2012FJ-NCET-ZR01), and Program for Young and Middle-Aged

Teacher in Science and Technology Research of Huaqiao University (ZQN-PY116).

## References

- [1] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [2] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [4] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.
- [5] X. Yang, H. Fu, H. Zha, and J. Barlow, "Semi-supervised nonlinear dimensionality reduction," in *Proceedings of the International Conference on Machine Learning*, 2006.
- [6] Z. Zhang and H. Zha, *Local Linear Smoothing for Nonlinear Manifold Learning*, Department of Computer Science and Engineering, Pennsylvania State University, PA, USA, 2003, CSE-03-003.
- [7] H. Chen, G. Jiang, and K. Yoshihira, "Robust nonlinear dimensionality reduction for manifold learning," in *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, pp. 447–450, China, August 2006.
- [8] X. Xing, S. Du, and K. Wang, "Robust Hessian locally linear embedding techniques for high-dimensional data," *Algorithms*, vol. 9, no. 2, article no. 36, 2016.
- [9] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 253–265, 2012.
- [10] J. Wang and Z. Zhang, "Nonlinear embedding preserving multiple local-linearities," *Pattern Recognition*, vol. 43, no. 4, pp. 1257–1268, 2010.
- [11] J. Wang, "Real local-linearity preserving embedding," *Neurocomputing*, vol. 136, pp. 7–13, 2014.
- [12] F. Zhong, J. Zhang, and D. Li, "Discriminant locality preserving projections based on L1-norm maximization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2065–2074, 2014.
- [13] Q. Gan, F. Shen, and J. Zhao, "An extended isomap for manifold topology learning with SOINN landmarks," in *Proceedings of the 22nd International Conference on Pattern Recognition, ICPR 2014*, pp. 1579–1584, Sweden, August 2014.
- [14] Y. Zhan and J. Yin, "Robust local tangent space alignment via iterative weighted PCA," *Neurocomputing*, vol. 74, no. 11, pp. 1985–1993, 2011.
- [15] H. Dadkhahi, M. F. Duarte, and B. Marlin, "Isomap out-of-sample extension for noisy time series data," in *Proceedings of the 25th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2015*, USA, September 2015.
- [16] N. Gu, M. Fan, and D. Meng, "Robust Semi-Supervised Classification for Noisy Labels Based on Self-Paced Learning," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1806–1810, 2016.
- [17] K. Kim and J. Lee, "Nonlinear dynamic projection for noise reduction of dispersed manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2303–2309, 2014.
- [18] Y. Li, *Semi-Supervised Manifold Learning and Its Application*, Department of Computer Science and Technology, 2010, Xian University.
- [19] H. Huang, G.-F. Qin, and H.-L. Feng, "Semi-supervised manifold learning and its application to remote sensing image classification," *Guangxue Jingmi Gongcheng/Optics and Precision Engineering*, vol. 19, no. 12, pp. 3025–3033, 2011.
- [20] J. Wang, "Outlier detection approach based on reconstruction weights," *Journal of Software*, vol. 22, no. 7, pp. 1571–1579, 2011.
- [21] H. P. Kriegel, M. Schubert, and A. Zimek, "Anglebased outlier detection in high-dimensional data," in *Proc of the Intl Conf on Knowledge Discovery and Data Mining*, pp. 444–459, 2008.
- [22] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, Kluwer Academic Publishers, Norwell, MA, USA, 2005.
- [23] H. Chang and D.-Y. Yeung, "Robust locally linear embedding," *Pattern Recognition*, vol. 39, no. 6, pp. 1053–1065, 2006.
- [24] T. Lin, H. Zha, and S. U. Lee, "Riemannian Manifold Learning for Nonlinear Dimensionality Reduction," in *European Conference on Computer Vision (ECCV)*, vol. 3951, pp. 44–55, 2006.
- [25] T. Sim, S. Baker, and M. Bsat, "PIE Face Database of Carnegie Mellon University," <http://vasc.ri.cmu.edu/idb/html/face/>.
- [26] "A database for Binary Alphadigits," <http://www.cs.nyu.edu/roweis/data>.
- [27] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'07*, pp. 1–8, June 2007.
- [28] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 10, no. 2, pp. 439–443, 1999.
- [29] S. Z. Li, K. L. Chan, and C. Wang, "Performance evaluation of the nearest feature line method in image classification and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1335–1339, 2000.



