Local and Global Regressive Mapping for Manifold Learning with Out-of-Sample Extrapolation

Yi Yang[1], **Feiping Nie**[2], **Shiming Xiang**[3], **Yueting Zhuang**[1] **and Wenhua Wang**[1]
[1]College of Computer Science, Zhejiang University, Hangzhou, 310027, China
[2]Department of Computer Science and Engineering, University of Texas, Arlington, 76019, USA
[3]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

Abstract

Over the past few years, a large family of manifold learning algorithms have been proposed, and applied to various applications. While designing new manifold learning algorithms has attracted much research attention, fewer research efforts have been focused on out-of-sample extrapolation of learned manifold. In this paper, we propose a novel algorithm of manifold learning. The proposed algorithm, namely Local and Global Regressive Mapping (LGRM), employs local regression models to grasp the manifold structure. We additionally impose a global regression term as regularization to learn a model for out-of-sample data extrapolation. Based on the algorithm, we propose a new manifold learning framework. Our framework can be applied to any manifold learning algorithms to simultaneously learn the low dimensional embedding of the training data and a model which provides explicit mapping of the out-of-sample data to the learned manifold. Experiments demonstrate that the proposed framework uncover the manifold structure precisely and can be freely applied to unseen data.

Introduction & Related Works

Unsupervised dimension reduction plays an important role in many applications. Among them, manifold learning, a family of non-linear dimension reduction algorithms, has attracted much attention. During recent decade, researchers have developed various manifold learning algorithms, such as ISOMap (Tenenbaum, Silva, & Langford 2000), Local Linear Embedding (LLE) (Roweis & Saul 2000), Laplacian Eigenmap (LE) (Belkin & Niyogi 2003), Local Tangent Space Alignment (LTSA) (Zhang & Zha 2004), Local Spline Embedding (LSE) (Xiang *et al.* 2009), etc .

Manifold learning has been applied to different applications, particularly in the field of computer vision, where it has been experimentally demonstrated that linear dimension reduction methods are not capable to cope with the data sampled from non-linear manifold (Chin & Suter 2008). Suppose there are n training data $\mathcal{X} = \{x_1, ..., x_n\}$ densely sampled from smooth manifold, where $x_i \in \mathbb{R}^d$ for $1 \leq i \leq n$. Denote $\mathcal{Y} = \{y_1, ..., y_n\}$, where $y_i \in \mathbb{R}^m (m < d)$ is the low dimensional embedding of x_i. We define $Y = [y_1, ..., y_n]^T$ as the low dimensional embedding matrix. Although the motivation of manifold learning algorithm differs from one to another, the objective function of ISOMap, LLE and LE can be uniformly formulated as follows (Yan *et al.* 2005).

$$\min_{Y^T BY=I} tr(Y^T LY), \tag{1}$$

where $tr(\cdot)$ is the trace operator, B is a constraint matrix, and L is the Laplacian matrix computed according to different criterions. It is also easy to see that (1) generalizes the objective function of other manifold learning algorithms, such as LTSA. Clearly, the Laplacian matrix plays a key role in manifold learning.

Different from linear dimension reduction approaches, most of the manifold learning algorithms do not provide explicit mapping of the unseen data. As a compromise, Locality Preserving Projection (LPP) (He & Niyogi 2003) and Spectral Regression (SR) (Cai, He, & Han 2007) were proposed, which introduce linear projection matrix to LE. However, because a linear constraint is imposed, both algorithms fail in preserving the intrinsical non-linear structure of the data manifold.

Manifold learning algorithms can be described as Kernel Principal Component Analysis (KPCA) (Schölkopf, Smola, & Müller 1998) on specially constructed Gram matrices (Ham *et al.* 2004). According to the specific algorithmic procedures of manifold learning algorithms, Bengio et al. have defined a data dependent kernel matrix K for ISOMap, LLE and LE, respectively (Bengio *et al.* 2003). Given the data dependent kernel matrix K, out-of-sample data can be extrapolated by employing Nyström formula. The framework proposed in (Bengio *et al.* 2003) generalizes Landmark ISOMap (Silva & Tenenbaum 2003). Similar algorithm was also proposed in (Chin & Suter 2008) for Maximum Variance Unfolding (MVU). Note that Semi-Definite Programming is conducted in MVU. It is very time consuming and thus less practical. One limitation of this family of algorithms is that the design of data dependent kernel matrices for various manifold learning algorithms is a nontrivial task. For example, compared with LE, it is not that straightforward to define the data dependent kernel matrix for LLE (Bengio *et al.* 2003) and it still remains unclear how to define the kernel matrices for other manifold learning algorithms, i.e., LTSA.

In (Saul & Roweis 2003), a nonparametric approach was proposed for out-of-sample extrapolation of LLE. Let $x_o \in \mathbb{R}^d$ be the novel data to be extrapolated and $\mathcal{X}_o = \{x_{o1}, ..., x_{ok}\} \subset \mathcal{X}$ be a set of data which are k nearest neighbor set of x_o in $\mathbb{R}^d$. The low dimensional embedding y_o of x_o is given by $\sum_{i=1}^{k} w_i y_{oi}$, in which y_{oi} is the low dimensional embedding of x_{oi} and w_i $(1 \leq i \leq k)$ can be obtained by minimizing the following objective function.

$$\min \sum_{i=1}^{k} \|x_o - w_i x_{oi}\|^2, \quad s.t. \sum_{i}^{k} w_i = 1. \tag{2}$$

This algorithm is a general one and can be applied to any other manifold learning algorithms for out-of-sample data extrapolation. A limitation is that, as indicated in (Saul & Roweis 2003), the mapping may discontinuously change as the novel points move between different neighborhoods. Furthermore, if the novel data are off manifold, i.e., the locally linear condition is not satisfied, it is uncertain how the algorithm will behave (Chin & Suter 2008).

In this paper, we propose a new manifold learning algorithm, namely Local and Global Regressive Mapping (LGRM), which learns a novel Laplacian matrix for manifold learning. In the proposed algorithm, we additionally add a kernelized global regression regularization to learn a model, which is used to extrapolate the unseen data. Based on the algorithm, we propose a new framework, which can be applied to many other manifold learning algorithms for out-of-sample data extrapolation. An interesting observation is that several well-known unsupervised dimension reduction algorithms are special cases of our framework. We observe in the experiment that the proposed algorithm preserves the manifold structure precisely during the projection and it works well when applied to unseen data.

The rest of this paper is organized as follows. Firstly, we detail the proposed algorithm. Then, based on the new algorithm, we propose a new framework for manifold learning and discuss the connections between our algorithm and other existing algorithms. After that, we show the experimental results and give conclusions.

The Proposed Algorithm

It is crucial to exploit the local structure in manifold learning. Inspired by (Roweis & Saul 2000; Zhang & Zha 2004; Yang *et al.* 2009), we construct a local clique $\mathcal{N}_i = \{x_i, x_{i1}, ..., x_{ik-1}\}$ for each data x_i, which comprises k data, including x_i and its $k-1$ nearest neighbors. We assume that the low dimensional embedding y_p of $x_p \in \mathcal{N}_i$ can be well predicted by a local prediction function f_i, and use it to predict the low dimensional embedding of all the data in $\mathcal{N}_i$. To obtain a good local prediction function, we minimize the following *regularized local empirical loss function*(Yang *et al.* 2009):

$$\sum_{x_p \in \mathcal{N}_i} \ell(f_i(x_p), y_p) + \gamma \Omega(f_i), \tag{3}$$

where ℓ is a predefined loss function, $\Omega(f_i)$ is a regularization function measuring the smoothness of f_i and $\gamma > 0$ is a regularization parameter. Note that our objective is intrinsically different from (Wu & Schölkopf 2006) which only utilizes a local learning model to predict the cluster identification of *a single datum* in $\mathcal{N}_i$. Traditional manifold learning aims to map each datum $x_i \in \mathcal{X}$ to $y_i \in \mathcal{Y}$. Alternatively, our framework additionally learns a mapping function which can be used to cope with unseen data. We therefore propose to minimize the following objective function:

$$\min_{Y^TY=I, f_i, f} \sum_{i=1}^{n} \left(\sum_{x_j \in \mathcal{N}_i} \ell(f_i(x_j), y_j) + \gamma \Omega(f_i) \right) + \sum_{i=1}^{n} \ell\left(f(x_i), y_i\right) + \gamma \Omega(f). \tag{4}$$

Different from all of the previous manifold learning algorithms, such as ISOMap, LLE, LE and LTSA, the objective function shown in (4) not only learns the low dimensional embedding Y of the input data, but also learns a mapping function $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ for out-of-sample data extrapolation. Observing the connection between KPCA and manifold learning, we first map the data into a Hilbert space $\mathcal{H}$ and assume that there is a linear transformation between $\mathcal{H}$ and $\mathbb{R}^m$, i.e. $y_i = \phi(W)^T \phi(x_i) + b$, where $\phi(W)$ is the projection matrix from $\mathcal{H}$ to $\mathbb{R}^m$ and $b \in \mathbb{R}^m$ is bias term. The data number in $\mathcal{N}_i$ is usually small. Following (Zhang & Zha 2004), we perform local PCA to reduce the dimension of each datum in $\mathcal{N}_i$ as preprocessing to avoid overfitting. Because the local structure of manifold is linear (Roweis & Saul 2000), f_i is defined as a linear regression model, i.e. $f_i(x) = W_i^T x + b_i$, where $W_i \in \mathbb{R}^{p \times m}$ is the local projection matrix and $b_i \in \mathbb{R}^m$ is bias and $p \in [m, d)$ is the dimension of each datum after local PCA being performed. The least squares loss function gains comparable performance to other complicated loss functions, provided that appropriate regularization is added (Fung & Mangasarian 2005). We therefore propose the following objective function to simultaneously learn the low dimensional embedding Y and the mapping function $\phi(W)\phi(\cdot) + b$:

$$\min_{\phi(W), W_i, b, b_i, Y} \sum_{i=1}^{n} \sum_{x_j \in \mathcal{N}_i} \left(\left\|W_i^T x_j + b_i - y_j\right\|^2 + \gamma \left\|W_i\right\|_F^2 \right) + \mu \left[\sum_{i=1}^{n} \left\|\phi(W)^T \phi(x_i) + b - y_i\right\|^2 + \gamma \left\|\phi(W)\right\|_F^2 \right] \quad s.t. Y^T Y = I,$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix. Let $X_i = [x_i, x_{i1}, ..., x_{ik-1}] \in \mathbb{R}^{p \times k}$ $(p < d)$ be the data matrix of $\mathcal{N}_i$ after local PCA being performed and $Y_i = [y_i, y_{i1}, ..., y_{ik-1}]^T \in \mathbb{R}^{k \times m}$ be the low dimensional embedding of $\mathcal{N}_i$. The objective function can be rewritten as:

$$\min_{\phi(W), W_i, b, b_i, Y} \sum_{i=1}^{n} \left(\left\|X_i^T W_i + 1_k b_i^T - Y_i\right\|_F^2 + \gamma \left\|W_i\right\|_F^2 \right) + \mu(\left\|\phi(X)^T \phi(W) + 1_n b^T - Y\right\|_F^2 + \gamma \left\|\phi(W)\right\|_F^2) \quad s.t. Y^T Y = I, \tag{5}$$

where $1_k \in \mathbb{R}^k$ and $1_n \in \mathbb{R}^n$ are two vectors of all ones. By employing the property that $\|M\|_F^2 = tr(M^T M)$ for any matrix M, the first term of (5) can be written as:

$$\sum_{i=1}^{n} \left\{ tr \left[(X_i^T W_i + 1_k b_i^T - Y_i)^T (X_i^T W_i + 1_k b_i^T - Y_i) \right] \right.$$
$$\left. + \gamma tr(W_i^T W_i) \right\}. \quad (6)$$

By setting its derivative w.r.t. W_i and b_i to zero, we have:

$$\begin{aligned} & W_i^T X_i 1_k + k b_i - Y_i^T 1_k = 0 \\ & \Rightarrow b_i = \frac{1}{k}(Y_i^T 1_k - W_i^T X_i 1_k); \end{aligned} \quad (7)$$

$$\begin{aligned} & X_i X_i^T W_i + X_i 1_k b_i^T - X_i Y_i + \gamma W_i = 0 \\ & \Rightarrow W_i = (X_i H_k X_i^T + \gamma I)^{-1} X_i H_k Y_i, \end{aligned} \quad (8)$$

where $H_k = I - \frac{1}{k} 1_k 1_k^T$ is the local centering matrix. Substituting W_i and b_i by (7) and (8), (6) can be written as:

$$\sum_{i=1}^{n} tr(Y_i^T A_i Y_i), \quad (9)$$

where

$$A_i = H_k - H_k X_i^T (X_i H_k X_i^T + \gamma I)^{-1} X_i H_k. \quad (10)$$

Note that X_i and Y_i are selected from X and Y. Define a selection matrix $S_p \in \{0,1\}^{n \times k}$ in which $(S_p)_{ij} = 1$ if x_i is the j-th element in $\mathcal{N}_i$ and $(S_p)_{ij} = 0$ otherwise. It is easy to see that $Y_i = S_i^T Y$ and thus (9) can be rewritten as:

$$\sum_{i=1}^{n} tr(Y^T S_i A_i S_i^T Y) = tr \left[Y^T \left(\sum_{i=1}^{n} S_i A_i S_i^T \right) Y \right]. \quad (11)$$

Then (6) is reformulated as (Yang *et al.* 2009)

$$Y^T L_l Y, \quad (12)$$

where $L_l = \sum_{i=1}^{n} S_i A_i S_i^T$. Similarly, the second term of (5) can be written as:

$$tr \left\{ \left[\phi(X)^T \phi(W) + 1_n b^T - Y \right]^T \left[\phi(X)^T \phi(W) \right. \right.$$
$$\left. \left. + 1_n b^T - Y \right] \right\} + \gamma tr \left[\phi(W)^T \phi(W) \right]. \quad (13)$$

Denote $H = I - \frac{1}{n} 1_n 1_n^T$ as the global centering matrix. The same as before, we rewrite (13) as:

$$Y^T L_g Y, \quad (14)$$

where

$$L_g = H - H\phi(X)^T \left[\phi(X) H \phi(X)^T + \gamma I \right]^{-1} \phi(X) H.$$

Note that

$$\begin{aligned} & H - H\phi(X)^T \phi(X) H \left[H\phi(X)^T \phi(X) H + \gamma I \right]^{-1} \\ & = \gamma H (H\phi(X)^T \phi(X) H + \gamma I)^{-1} H. \end{aligned} \quad (15)$$

Although $\phi(X)$ can not be explicitly computed, $\phi(X)^T \phi(X)$ can be calculated by a kernel function. We suppose the dot production of x_i and x_j in $\mathcal{H}$ is given by the following kernel function:

$$K_{x_i, x_j} = (\phi(x_i) \cdot \phi(x_j)) = \phi(x_i)^T \phi(x_j), \quad (16)$$

where $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ can be any positive kernel satisfying Mercer's condition. For example, we can use the Radial Basis Function(RBF) kernel, which is defined as:

$$K_{x_i, x_j} = \exp\left(-\|x_i - x_j\|^2 / \sigma^2 \right), \quad (17)$$

where σ is a parameter. Then L_g can be computed by:

$$L_g = \gamma H (H\mathbf{K}H + \gamma I)^{-1} H, \quad (18)$$

where $\mathbf{K}$ is the kernel matrix with its element $\mathbf{K}_{ij} = K_{x_i, x_j}$. Thus, we have obtained the objective function of the proposed algorithm as follows:

$$\min_{Y^T Y = I} Y^T (L_l + \mu L_g) Y. \quad (19)$$

The low dimensional embedding Y can be obtained by eigen-decomposition of $(L_l + \mu L_g)$. Note that by setting the derivative of (13) w.r.t. W_i and b_i to zero, we have:

$$\begin{aligned} \phi(W) & = (\phi(X) H \phi(X)^T + \gamma I)^{-1} \phi(X) H Y \\ & = \phi(X) H (H\phi(X)^T X H + \gamma I)^{-1} Y \end{aligned} \quad (20)$$

$$b = \frac{1}{n} Y^T 1_n - \frac{1}{n} W^T \phi(X) 1_n = \frac{1}{n} Y^T 1_n -$$
$$\frac{1}{n} Y^T (H\phi(X)^T \phi(X) H + \gamma I)^{-1} H\phi(X)^T \phi(X) 1_n. \quad (21)$$

Therefore, given a novel data x which is out of training set, its low dimensional embedding $y = \phi(W)^T \phi(x) + b$ can be computed by:

$$y = Y^T (H\phi(X)^T \phi(X) H + \gamma I)^{-1} H\phi(X)^T \phi(x) + \frac{1}{n} Y^T 1_n$$
$$- \frac{1}{n} Y^T (H\phi(X)^T \phi(X) H + \gamma I)^{-1} H\phi(X)^T \phi(X) 1_n.$$

Denote $\mathbf{K}_x \in \mathbb{R}^n$ as a vector with its i-th element $\mathbf{K}_{xi} = (\phi(x) \cdot \phi(x_i)) = \phi(x)^T \phi(x_i)$, where $x_i \in \mathcal{X}$ is the i-th instance in training set. y can be computed by:

$$\begin{aligned} y = & Y^T (H\mathbf{K}H + \gamma I)^{-1} H\mathbf{K}_x + \frac{1}{n} Y^T 1_n \\ & - \frac{1}{n} Y^T (H\mathbf{K}H + \gamma I)^{-1} H\mathbf{K} 1_n. \end{aligned} \quad (22)$$

Discussions

A new framework for manifold learning.

The objective function of different manifold learning algorithms, such as ISOMap, LLE, LE and LTSA, can be unified by (1)(Yan *et al.* 2005). We can prove that L_l and $L_{lg} = L_l + \mu L_g$ are Laplacian matrices. Therefore our algorithm coincides with previous manifold learning algorithms. Yet, our algorithm additionally learns a model for out-of-sample data extrapolation. The proposed algorithm is a general one and can be applied to other manifold learning

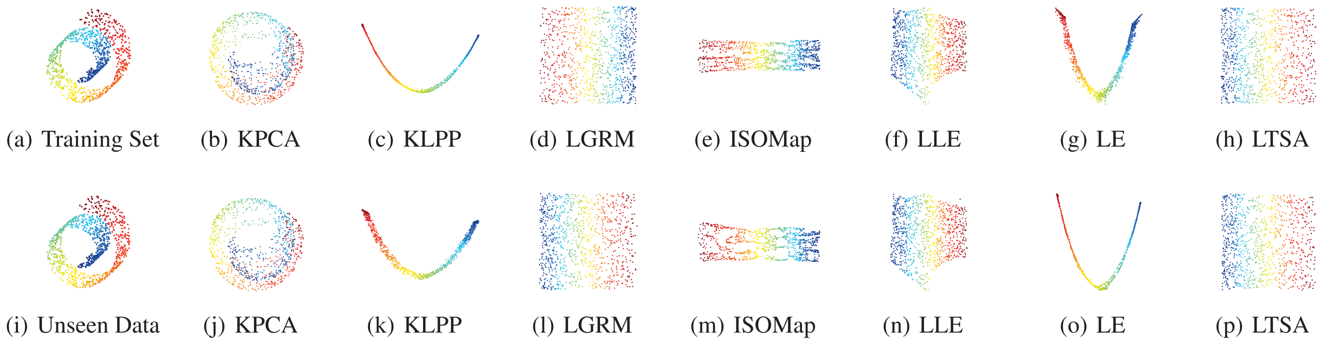


Figure 1: 2D embedding of Swiss roll. The upper line shows 1000 training data sampled from Swiss roll in $\mathbb{R}^3$ and their low dimensional embedding in $\mathbb{R}^2$. The lower line shows 1000 unseen data and their low dimensional embedding in $\mathbb{R}^2$.

algorithms for out-of-sample data extrapolation. In general, the new framework can be formulated as:

$$\min_{Y^TY=I} tr\left[Y^T(L_{loc}+\mu L_g)Y\right], \tag{23}$$

where L_{loc} can be any Laplacian matrix computed according to local structure of data manifold. For example, we can replace L_{loc} in (23) by the well-known Gaussian Laplacian matrix L_{le}, which is employed by LE (Belkin & Niyogi 2003), for manifold learning. It is easy to see that when $\mu = 0$, our framework reduces to traditional manifold learning algorithms.[1] Due to the space limitation, we omit the detailed discussion.

Connection with dimension reduction algorithms

Besides traditional manifold learning algorithms, our framework also generalizes many unsupervised dimension reduction algorithms. In this subsection, we present the connection between our framework and other representative dimension reduction algorithms by the following propositions.

Proposition 1. When $\mu \to \infty, \mu\gamma \to 0$, Locality Preserving Projection (LPP) (He & Niyogi 2003) is a special case of our proposed framework, provided that L_{loc} in (23) is Gaussian Laplacian (denoted as L_{le}), and K_{x_i,x_j} is a linear kernel function.

Proof: When $\mu \to \infty$, $\mu\gamma \to 0$ and $L_{loc} = L_{le}$, the objective function shown in (23) reduces to:

$$\min_{\phi(W)^T\phi(X)\phi(X)^T\phi(W)=I} tr(\phi(W)^T\phi(X)L_{le}\phi(X)^T\phi(W)).$$

If K_{x_i,x_j} is a linear kernel function, then $\phi(W) = W$ and $\phi(x) = x$. Therefore, the objective function turns to

$$\min_{W^TXX^TW=I} tr(W^TXL_{le}X^TW). \quad \Box \tag{24}$$

Proposition 2. When $\mu \to \infty, \mu\gamma \to 0$, Kernel Locality Preserving Projection (KLPP) (He & Niyogi 2003) is a special case of our proposed framework, provided that L_{loc} in (23) is Gaussian Laplacian.

This proposition can be similarly proved as Proposition 1.

Proposition 3. When $\mu \to 0$, Spectral Regression (SR) (Cai, He, & Han 2007) is a special case of our framework, provided that L_{loc} in (23) is Gaussian Laplacian and K_{x_i,x_j} is a linear kernel function.

Proof: When $\mu \to 0$ and $L_{loc} = L_{le}$, the objective function shown in (23) turns to a two-step approach. In the first step, it computes Y by minimizing:

$$\min_{Y^TY=I} tr(Y^TL_{le}Y). \tag{25}$$

Then, it solves the following optimization problem:

$$\min_{\phi(W),b} \left\|\phi(X)^T\phi(W) + 1_nb^T - Y\right\|_F^2 + \gamma\left\|\phi(W)^T\right\|_F^2.$$

Given that linear kernel is employed, we can see that it yields the same results of SR. □

Proposition 4. When $\mu \to 0$, Kernel Spectral Regression (KSR) is a special case of our framework, provided that L_{loc} in (23) is Gaussian Laplacian.

This proposition can be similarly proved as Proposition 3.

Experiments

In this section, we conduct extensive experiments to test the performance of the proposed algorithm. Pervious works on manifold learning (Tenenbaum, Silva, & Langford 2000; Roweis & Saul 2000; Belkin & Niyogi 2003; Zhang & Zha 2004; Chin & Suter 2008) mainly use synthetic data and some image data sets, such as teapot images (Weinberger & Saul 2006) and face expression images (Roweis & Saul 2000). Following the convention of manifold learning, we also use them in the experiments.

First, we use synthetic data to test the proposed algorithm. In this experiment, RBF kernel defined in (17) is used with $\sigma = 10$. We set $\mu = 10^{-5}$ and $\gamma = 10^{-4}$. Fig.1 shows the learning results for the data sampled from Swiss roll. The upper line shows the 1000 training data in $\mathbb{R}^3$ and the mapping results in $\mathbb{R}^2$ using different algorithms. We can see that our LGRM and LTSA perform best. LLE and ISOMap can preserve the manifold structure to certain extend, but not as accurate as LGRM and LTSA. The other algorithms, including LE, KPCA and KLPP, failed in preserving the intrinsical

[1] Note that $\min_{Y^TY=I} Y^TL_lY$ yields another new manifold learning algorithm as well.

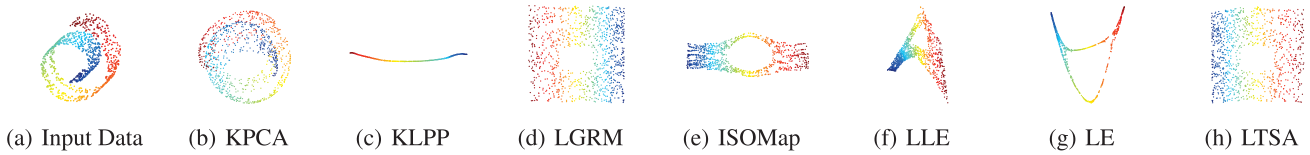

(a) Input Data (b) KPCA (c) KLPP (d) LGRM (e) ISOMap (f) LLE (g) LE (h) LTSA

Figure 2: 2D embedding of 1000 data sampled from Swiss roll with hole in it.

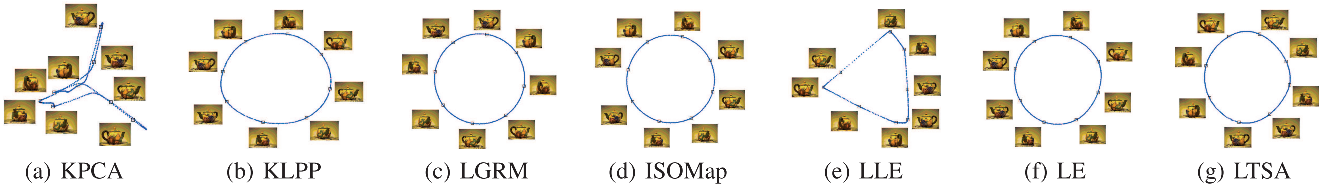

(a) KPCA (b) KLPP (c) LGRM (d) ISOMap (e) LLE (f) LE (g) LTSA

Figure 3: 2D embedding of 400 teapot images.

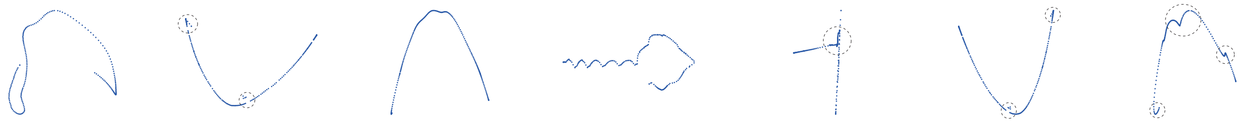
manifold structure. The lower line of Fig.1 shows another 1000 data, which are used as out-of-sample data, to test the performance of unseen data extrapolation. For the sake of out-of-sample data extrapolation, we use the algorithm proposed in (Silva & Tenenbaum 2003) for ISOMap, the algorithm proposed in (Saul & Roweis 2003) for LLE and LTSA, and the algorithm proposed in (Bengio *et al.* 2003) for LE. We can see that our algorithm works well when applied to unseen data. Fig.2 shows 1000 data sampled from Swiss roll with one hole inside. Again, we observe that our LGRM and LTSA yield the most faithful projection.

Next, we use image data to test the performance of the proposed algorithm. The teapot image set collected by Weinberger and Saul contains 400 images of teapot in full rotation (Weinberger & Saul 2006). Each image is represented by a 23028 dimensional vector. In this experiment, we similarly set $\mu = 10^{-5}$ and $\gamma = 10^{-4}$. For RBF kernel, we set $\sigma = 100$. Fig.3 shows the embedding of all the 400 images $\mathbb{R}^2$. In the figure, the blue dots represent the embedding. The embedding of each exemplar image is indicated by a marker. We can see in Fig.3 that our LGRM, ISOMap, LE and LTSA yield the most faithful projection. Compared with KPCA and LLE, KLPP gains better performance, but still worse than LGRM, ISOMap, LE and LTSA. We additionally report the mapping results for the first 200 images from teapot image set when $k = 30$. Fig.4 shows the embedding of the 200 images in $\mathbb{R}^2$, in which the incorrectly projected data are surrounded by circles. We observe in Fig.4 that although LTSA and our LGRM gain best performance in previous experiments, LGRM outperforms LTSA in this case. Therefore, we conclude that our LGRM obtains the best performance over the above 4 cases.

Lastly, we use Swiss roll and the face expression image database (Roweis & Saul 2000) to compare different algorithms in terms of out-of-sample data extrapolation. Because KPCA and KLPP are not manifold learning algorithms, we do not plot the results due to the lack of space. In this experiment, each data set $\mathcal{D}$ is divided into two subsets $\mathcal{A}$ and $\mathcal{B}$. Data in $\mathcal{A}$ are used as training data to learn the low dimensional embedding of data manifold. Data in $\mathcal{B}$ are used as testing data to test the performance of out-of-sample data extrapolation. We first use $\mathcal{D}$ to learn the low dimension embedding Y of all the data. Without loss of generality, we can write Y as $Y = [Y_{train}, Y_{test}]^T$, where Y_{train} and Y_{test} are the low dimensional embedding of training data and testing data, respectively. After that, we use $\mathcal{A}$ to train the manifold and then each datum in $\mathcal{B}$ is extrapolated into the learned manifold. Let Y'_{test} be a matrix comprising of the low dimensional embedding of all the data in testing set after being extrapolated. We define the Embedding Error as $\frac{1}{|\mathcal{B}|}\|Y_{test} - Y'_{test}\|_F^2$, in which Y_{test} and Y'_{test} are normalized to remove the scaling factor. To extrapolate testing data, we use the algorithm proposed by (Silva & Tenenbaum 2003) for ISOMap, the algorithm proposed by (Saul & Roweis 2003) for LLE and LTSA, and the algorithm proposed by (Bengio *et al.* 2003) for LE. In this experiment, we use 1000 data from Swiss Roll with 800 training data and 200 testing data, and the 1765 face expression images (Roweis & Saul 2000) with 1575 training data and 200 testing data. For each database, we randomly split it into two subsets, which is independently repeated 10 times. For face expression images, we use the same parameters used for teapot images. Fig.5 shows the average Embedding Error of different algorithms. We observe that the error rate of our algorithm is the lowest. Moreover, because our algorithm dose not need to perform KNN search or compute the shortest path between data pairs, it is faster. Compared with (Bengio *et al.* 2003), our algorithm is a more general one and can be applied to any manifold algorithms for unseen data extrapolation. While the algorithm proposed in (Saul & Roweis 2003) can be applied to other manifold algorithms, the limitation is that the mapping may discontinuously change as novel points move between different neighborhoods (Saul & Roweis 2003), and it remains unclear how the algorithm will behave if the novel points are far from training data.

Conclusions

There are two main contributions in this paper. First, we propose a new Laplacian matrix, i.e., L_l defined in (12), for manifold learning, which can be easily extended to many



(a) KPCA (b) KLPP (c) LGRM (d) ISOMap (e) LLE (f) LE (g) LTSA

Figure 4: 2D embedding of the first 200 teapot images. The incorrectly projected data are indicated by circles.

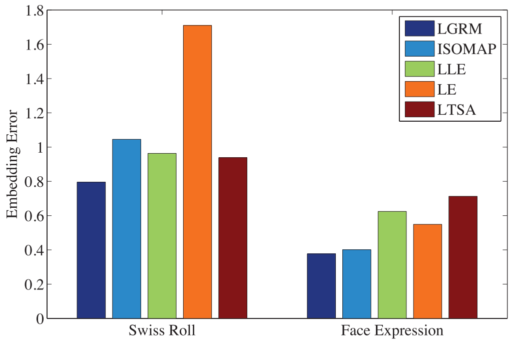


Figure 5: Average embedding error comparison.

other algorithms, such as semisupervised subspace learning, classification, feature selection, etc. Second, we propose a new framework for manifold learning. The new framework simultaneously learns the low dimensional embedding of the input data and a model for unseen data extrapolation of the learned manifold. Connection between our framework and other algorithms has been discussed. One appealing feature of our framework is that it does not only generalize the existing manifold learning algorithms, but also generalizes several other dimension reduction algorithms. Experiments show that the proposed algorithm preserves the manifold structure precisely and can effectively map unseen data to the learned manifold.

Acknowledgments

This work is supported by 973 Program (No.2010CB327905), NSFC(No.90920303), the National High Technology Research and Development Program of China (2006AA010107). Shiming Xiang is supported by NSFC (No.60975037).

References

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

Bengio, Y.; Paiement, J.-F.; Vincent, P.; Delalleau, O.; Roux, N. L.; and Ouimet, M. 2003. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*.

Cai, D.; He, X.; and Han, J. 2007. Spectral regression for efficient regularized subspace learning. In *ICCV*, 1–8.

Chin, T., and Suter, D. 2008. Out-of-sample extrapolation of learned manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(9):1547–1556.

Fung, G., and Mangasarian, O. L. 2005. Multicategory proximal support vector machine classifiers. *Machine Learning* 59(1-2):77–97.

Ham, J.; Lee, D. D.; Mika, S.; and Schölkopf, B. 2004. A kernel view of the dimensionality reduction of manifolds. In *ICML*.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *NIPS*.

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

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

Schölkopf, B.; Smola, A.; and Müller, K. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 26(1):1299–1319.

Silva, V., and Tenenbaum, J. B. 2003. Global versus local methods in nonlinear dimensionality reduction. In *NIPS*, 705–712.

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

Weinberger, K. Q., and Saul, L. K. 2006. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision* 70(1):77–90.

Wu, M., and Schölkopf, B. 2006. A local learning approach for clustering. In *NIPS*, 1529–1536.

Xiang, S.; Nie, F.; Zhang, C.; and Zhang, C. 2009. Nonlinear dimensionality reduction with local spline embedding. *IEEE Transactions on Knowledge and Data Engineering* 21(9):1285–1298.

Yan, S.; Xu, D.; Zhang, B.; and Zhang, H. 2005. Graph embedding: A general framework for dimensionality reduction. In *CVPR*, 830–837.

Yang, Y.; Xu, D.; Nie, F.; Luo, J.; and Zhuang, Y. 2009. Ranking with local regression and global alignment for cross media retrieval. In *ACM Multimedia*.

Zhang, Z., and Zha, H. 2004. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing* 10:313–338.