Int J Software Informatics, Volume 7, Issue 3 (2013), pp.435–451 Tel: +86-10-62661040
International Journal of Software and Informatics, ISSN 1673-7288 http://www.ijsi.org

Email: ijsi@iscas.ac.cn, ijsi2007@gmail.com

Locally Regressive Projections

Lijun Zhang

(Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI 48824, USA)

Abstract We propose a novel linear dimensionality reduction algorithm, namely Locally Regressive Projections (LRP). To capture the local discriminative structure, for each data point, a local patch consisting of this point and its neighbors is constructed. LRP assumes that the low dimensional representations of points in each patch can be well estimated by a locally fitted regression function. Specifically, we train a linear function for each patch via ridge regression, and use its *fitting error* to measure how well the new representations can respect the local structure. The optimal projections are thus obtained by minimizing the summation of the *fitting errors* over all the local patches. LRP can be performed under either supervised or unsupervised settings. Our theoretical analysis reveals the connections between LRP and the classical methods such as PCA and LDA. Experiments on face recognition and clustering demonstrate the effectiveness of our proposed method.
Key words: dimensionality reduction; local learning; locally regressive projections; ridge regression

Zhang LJ. Locally regressive projections. ***Int J Software Informatics*, Vol.7, No.3 (2013): 435–451. http://www.ijsi.org/1673-7288/7/i164.htm**

1 Introduction

High dimensional data sets are common in various domains such as engineering, biology, and economics. However, the performance of many learning algorithms degrades rapidly as the dimensionality increases, which is referred to as the *curse of dimensionality*[14]. As a result, dimensionality reduction becomes an essential data preprocessing technique for finding meaningful low-dimensional structures hidden in the original high-dimensional space[10]. Two of the most well known algorithms are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

Principal Component Analysis (PCA)[3] reduces the dimensionality of the input data by finding a few orthogonal linear projections such that the variance of the projected data is maximized. It turns out that these projections are the leading eigenvectors of the covariance matrix of the data, which are called principal components. Linear Discriminant Analysis (LDA)[8] is a supervised dimensionality reduction method which seeks directions that are optimal for discrimination. It aims to map points of different classes far from each other, while keeping points from the same class close. Both PCA and LDA consider only the global Euclidean structure, and fail to respect the local geometric structure.

Instead of focusing on global structure, local approaches for dimensionality reduction try to preserve the local geometry of the data space[9]. Typical local

Corresponding author: Lijun Zhang, Email: zhanglij@msu.edu
Received 2012-11-04; Revised 2013-07-14; Accepted 2013-08-31.

approaches include Locally Linear Embedding (LLE)[18], Laplacian Eigenmap (LE)[4], Neighborhood Preserving Embedding (NPE)[11], and Locality Preserving Projections (LPP)[12]. These methods have succeeded in recovering the intrinsic geometric structure of a broad class of nonlinear data manifolds. Besides, it has been shown that all of those algorithms can be reformulated in a general graph embedding framework, and their differences lie in the way of describing the local geometry[15,26].

Among those approaches, Locally Linear Embedding (LLE)[18] is one typical local learning method which characterizes the local geometry of the data space by linear coefficients that reconstruct each point from its neighbors. It then assumes that the embedding of each point can also be reconstructed from its neighbors' embeddings with the same coefficients. Recently, Wu[24] et al. proposed another local method named Local Learning Projections (LLP). Similar to LLE, LLP assumes that the projection value of each point can be estimated based on its neighbors and their projected coordinates. The difference is that LLP trains a kernel machine at each point and each projection to do the estimation. Both LLE and LLP aim to minimize the estimation errors of all the data points, and the estimation error of each point is counted once in their objection functions. Thus, different points are treated equally in LLE and LLP. However, in real word applications, data points are usually not uniformly distributed, and thus it is more reasonable to assign large weights to points sampled from dense regions.

Inspired by recent developments in local learning[18,24,25], a novel linear dimensionality reduction algorithm, called Locally Regressive Projections (LRP), is proposed in this paper. LRP is fundamentally built upon the idea of local linear regression, which is recently applied to ranking[25] and coclustering[27]. For the purpose of discovering the local discriminative structure, we define a local patch for each data point as the set containing this point and its neighbors. LRP assumes that for each local patch in the data space, the low dimensional representations of points belonging to it can be well estimated by a locally fitted function. Specifically, we adopt ridge regression to learn a locally linear function for each patch, using point belonging to this patch as the training data. Then, the *fitting error* of the local function provides a natural measurement for the projection performance. The objective function of LRP is thus defined as the summation of the *fitting errors* over all the local patches. And the optimal projections are obtained by minimizing this summation, which can be solved efficiently via eigenvalue decomposition. Since the local patch can be constructed according to the label information, LRP can be easily extended by incorporating prior knowledge.

One important property of LRP is that it is adaptive to the underlying data density. Because the fitting error of ridge regression contains the estimation errors of all the training points, LRP actually minimizes the estimation errors of all the points in each patch. Since points sampled from dense regions would appear in more local patches, these points will receive higher weights than those from sparse regions. In this way, LRP can model the local discriminative structure more accurately than LLE and LLP.

The rest of the paper is organized as follows. In Section 2, we give a brief review of several related work. Our proposed dimensionality reduction algorithm LRP is introduced in Section 3. Discussions with related methods are given in Section 4. Experiments are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2 Related Work

The major notations used in this paper are summarized in Table 1.

Table 1 List of notations used in this paper

$\mathbf{x}_i \in \mathbb{R}^n$	the i-th data point
$X \in \mathbb{R}^{n\times m}$	the data matrix consisting of $\mathbf{x}_i$'s, i.e. $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$
$\mathbf{y}_i \in \mathbb{R}^p$	the p-dimensional new representation of $\mathbf{x}_i$
$Y \in \mathbb{R}^{p\times m}$	the data matrix consisting of $\mathbf{y}_i$'s, i.e. $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_m]$
$\mathbf{y}^l \in \mathbb{R}^m$	the l-th column of Y^T, i.e. $Y^T = [\mathbf{y}^1, \cdots, \mathbf{y}^p]$
$P \in \mathbb{R}^{n\times p}$	the projection matrix, i.e. $Y = P^T X$
$\mathbf{p}_l \in \mathbb{R}^n$	the l-th column of P, i.e. $P = [\mathbf{p}_1, \cdots, \mathbf{p}_p]$ and $\mathbf{y}^l = X^T \mathbf{p}_l$
$\mathcal{N}_i^-$	the local patch consisting of neighboring points of $\mathbf{x}_i$
n_i^-	the number of points in $\mathcal{N}_i^-$, i.e. $\lvert\mathcal{N}_i^-\rvert$
$\mathcal{N}_i$	the local patch consisting of $\mathbf{x}_i$ and its neighboring points
n_i	the number of points in $\mathcal{N}_i$, i.e. $\lvert\mathcal{N}_i\rvert$
$\mathbf{1}_k$	the k-dimensional constant vectors of all ones
I	the identity matrix
Π_k	the $k \times k$ centering matrix, i.e. $I - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^T$
$\mathbf{e}_k$	the k-th unit vector

2.1 The linear dimensionality reduction problem

Given a set of m points $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\} \subseteq \mathbb{R}^n$, linear dimensionality reduction looks for a projection matrix $P = [\mathbf{p}_1, \cdots, \mathbf{p}_p] \in \mathbb{R}^{n\times p}$ which maps these m points to a set of points $\{\mathbf{y}_1, \cdots, \mathbf{y}_m\} \subseteq \mathbb{R}^p$, that capture the content in the original data, according to some criterion[10]. Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$ and $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_m]$, then $Y = P^T X$.

2.2 Classical methods

2.2.1 Principal component analysis (PCA)

PCA[3,8] can be defined in terms of the orthogonal projections which maximize the variance in the projected subspace. The optimization problem of PCA can be formularized as:

$$\begin{aligned} &\max_P \operatorname{Tr}\big(P^T C P\big) \\ &\text{s.t.} \quad P^T P = I \end{aligned} \tag{1}$$

where C is the data covariance matrix. Let $\bar{\mathbf{x}}$ be the sample mean, $\mathbf{1}_m$ be the m-dimensional constant vectors of all ones, and $\Pi_m = I - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ be the centering matrix. The data covariance matrix C is defined as:

$$C = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{m} X \Pi_m X^T \tag{2}$$

The optimal solution P^* of (1) is given by the eigenvectors of C associated with the largest eigenvalues.

2.2.2 *Linear discriminant analysis (LDA)*

LDA[8] tries to minimize the within-class variance and maximize the between-class variance simultaneously. Suppose the given m points belong to c classes. Let $\mathbf{m}$ be the total sample mean vector, r_i be the number of samples in the i-th class, $\mathbf{m}^i$ be the sample mean vector of the i-th class, and $\mathbf{x}_j^i$ be the j-th sample in the i-th class. One of the most common objective functions of LDA is as follows:

$$\max_P Tr\big((P^T S_W P)^{-1}(P^T S_B P)\big) \tag{3}$$

where S_W and S_B are the *within-class scatter matrix* and *between-class scatter matrix*, respectively. They are defined as:

$$S_W = \sum_{i=1}^{c} \left(\sum_{j=1}^{r_i} \left(\mathbf{x}_j^i - \mathbf{m}^i\right) \left(\mathbf{x}_j^i - \mathbf{m}^i\right)^T \right) \tag{4}$$

$$S_B = \sum_{i=1}^{c} r_i (\mathbf{m}^i - \mathbf{m})(\mathbf{m}^i - \mathbf{m})^T \tag{5}$$

The projections of LDA are computed by solving the following generalized eigenvalue problem:

$$S_B \boldsymbol{\alpha} = \gamma S_W \boldsymbol{\alpha} \tag{6}$$

The optimal projections correspond to the eigenvectors associated with the largest eigenvalues. Since the rank of S_B is bounded above by $c-1$, LDA is unable to find more than $c-1$ projection vectors.

2.3 *Local learning based methods*

2.3.1 *Locally linear embedding (LLE)*

Locally Linear Embedding (LLE)[18] characterizes the local geometric structure by linear coefficients that reconstruct each data point from its neighbors. Let $\mathcal{N}_i^-$ be local patch consisting of neighboring points of $\mathbf{x}_i$, not including $\mathbf{x}_i$ itself. The optimal reconstruction coefficients are found by solving the following optimization problem:

$$\begin{aligned} &\min_W \textstyle\sum_{i=1}^m \|\mathbf{x}_i - \sum_{j=1}^m W_{ij}\mathbf{x}_j\|^2 \\ &s.t.\ \textstyle\sum_{j=1}^m W_{ij} = 1,\ i = 1, \cdots, m \\ &\qquad W_{ij} = 0 \ \text{ if } \ \mathbf{x}_j \notin \mathcal{N}_i^- \end{aligned} \tag{7}$$

where the variable is the matrix $W \in \mathbb{R}^{m \times m}$, and W_{ij} summarizes the contribution of the j-th data point to the i-th reconstruction. LLE assumes that the low dimensional embedding of each point can also be reconstructed from its neighbors' embeddings with the same coefficients. For each data point $\mathbf{x}_i$, the following function is used to reconstruct its embedding:

$$\mathbf{f}_i(\mathbf{x}_i) = \sum_{j=1}^{m} W_{ij}\mathbf{y}_j \tag{8}$$

LLE aims to minimize the reconstruction errors of all the data samples, and its objective function is given by

$$\sum_{i=1}^{m}\|\mathbf{y}_i-\mathbf{f}_i(\mathbf{x}_i)\|^2=\sum_{i=1}^{m}\|\mathbf{y}_i-\sum_{j=1}^{m}W_{ij}\mathbf{y}_j\|^2=\|Y^T-WY^T\|^2 \tag{9}$$

As can be seen from Eq. (9), the reconstruction error of each point is counted once, thus different points receive the same weight in LLE.

Different from PCA and LDA, LLE is a nonlinear dimensionality reduction method, and can't apply to unseen data samples. To address the out-of-sample problem, the linearization of LLE is discussed in Ref. [11].

2.1.1 Local learning projections (LLP)

LLP[24] assumes that the projection value of each point can be estimated based on its neighbors and their projected coordinates. At each point $\mathbf{x}_i$ and each projection l, LLP fits a *Kernel Machine* $f_i^l(\mathbf{x})$ using $\{\mathbf{x}_j, y_j^l\}_{\mathbf{x}_j\in\mathcal{N}_i^-}$ as the training data. Denote the size of $\mathcal{N}_i^-$ as n_i^-. The function $f_i^l(\mathbf{x}_i)$ is fitted via kernel ridge regression[20], and we obtain

$$f_i^l(\mathbf{x}_i)=(\mathbf{k}_i^-)^T(K_i^-+\lambda I)^{-1}\mathbf{y}_i^l \tag{10}$$

where $\mathbf{k}_i^-\in\mathbb{R}^{n_i^-}$ is the vector $[K(\mathbf{x}_i,\mathbf{x}_j)]^T$ for $\mathbf{x}_j\in\mathcal{N}_i^-$, $K_i^-\in\mathbb{R}^{n_i^-\times n_i^-}$ is the local kernel matrix over $\mathcal{N}_i^-$, and $\mathbf{y}_i^l\in\mathbb{R}^{n_i^-}$ is the vector $[y_j^l]^T$ for $\mathbf{x}_j\in\mathcal{N}_i^-$.

Let $\boldsymbol{\alpha}_i^T=(\mathbf{k}_i^-)^T(K_i^-+\lambda I)^{-1}$. The objective function of LLP is defined as the summation of the estimation errors of all the points:

$$\sum_{l=1}^{p}\sum_{i=1}^{m}\left(y_i^l-f_i^l(\mathbf{x}_i)\right)^2=\sum_{l=1}^{p}\sum_{i=1}^{m}(y_i^l-\boldsymbol{\alpha}_i^T\mathbf{y}_i^l)^2 \tag{11}$$

The estimation error of each point is also counted once in Eq. (11), so the importance of each point is the same in LLP.

2.4 More recent progresses

As an extension of PCA, Dirichlet Component Analysis (DCA)[23] is proposed to handle the compositional data (positive constant-sum real vectors). DCA attempts to find the optimal projection that maximizes the estimated Dirichlet precision on the projected data, thus reducing the compositional data to a lower dimensionality such that the components are de-correlated as much as possible. In Ref. [28], Worst-case Linear Discriminant Analysis (WLDA) is developed by defining new between-class and within-class scatter measures. WLDA adopts the worst-case view and is more suitable for applications such as classification.

In Ref. [19], Structure Preserving Embedding (SPE) is proposed for embedding graphs in a low-dimensional Euclidean space such that the global topological properties of the input graph are preserved. SPE is formulated as a semi-definite program constrained by a set of linear inequalities which captures the connectivity structure of the graph. Instead of focusing the structure of the data space, Local Minima Embedding (LME)[16] tries to find a low-dimensional embedding that preserves the local minima structure of a given objective function. The embedding of LME is useful for visualizing and understanding the relation between the original variables that create local minima. Stochastic neighbour embedding (SNE)[13] is a famous dimensionality reduction method which aims to optimally preserve neighborhood identity. In [7], a new method named Elastic Embedding (EE) is proposed, which reveals the relationship between Laplacian Eigenmap[4] and SNE.

3 Locally Regressive Projections (LRP)

In local dimensionality reduction approaches, points belonging to dense regions will impact many more points than others. However, existing local learning methods such as LLE and LLP ignore this difference and assign the same weight to all the points. In the following, we introduce our density adaptive algorithm: Locally Regressive Projections.

3.1 The objective

Since (local) linear regression has been widely studied in other problems, the mathematical formulations in this subsection are similar to some other work[1,25,27]. The key difference is that here local linear regression is applied to dimensionality reduction.

For each data point $\mathbf{x}_i$, we define the local patch $\mathcal{N}_i$ be the set containing $\mathbf{x}_i$ and its neighboring points, with the size n_i. And we define $\mathcal{A}_i = \{j \mid \mathbf{x}_j \in \mathcal{N}_i\}$ be the set containing the indices of samples in $\mathcal{N}_i$. Let $X_i \in \mathbb{R}^{n \times n_i}$ be the local data matrix consisting of samples in $\mathcal{N}_i$, that is, $X_i = [\mathbf{x}_j]$ for $j \in \mathcal{A}_i$. Let $Y_i \in \mathbb{R}^{p \times n_i}$ be the data matrix containing the new representations of points in $\mathcal{N}_i$, that is, $Y_i = [\mathbf{y}_j]$ for $j \in \mathcal{A}_i$. Since matrix Y_i is a part of Y, we can construct a selection matrix $S_i \in \{0,1\}^{m \times n_i}$ for each Y_i such that

$$Y_i = Y S_i \tag{12}$$

S_i is constructed as follows: $S_i = [\mathbf{e}_j]$ for $j \in \mathcal{A}_i$, where $\mathbf{e}_j$ is a m-dimensional vector whose j-th element is one and all other elements are zero.

For each patch $\mathcal{N}_i$, we consider fitting a multi-output linear function

$$\mathbf{f}_i(\mathbf{x}) = W_i^T \mathbf{x} + \mathbf{b}_i \tag{13}$$

to approximate the new representations of points in $\mathcal{N}_i$. In this linear function, $W_i \in \mathbb{R}^{n \times p}$ is the coefficient matrix, and $\mathbf{b}_i \in \mathbb{R}^p$ is the intercept. Using ridge regression[14], the local function is obtained by solving the following problem:

$$\min_{W_i, \mathbf{b}_i} \frac{1}{n_i} \|Y_i - W_i^T X_i - \mathbf{b}_i \mathbf{1}_{n_i}^T\|_F^2 + \lambda \|W_i\|_F^2 \tag{14}$$

where the penalty term $\lambda\|W_i\|_F^2$ is introduced to avoid overfitting. Taking the first order partial derivatives of Eq. (14) with respective to W_i, $\mathbf{b}_i$ and requiring them to be zero, we get the optimal W_i^* and $\mathbf{b}_i^*$:

$$W_i^* = (X_i \Pi_{n_i} X_i^T + n_i \lambda I)^{-1} X_i \Pi_{n_i} Y_i^T \tag{15}$$

$$\mathbf{b}_i^* = \frac{1}{n_i}\big(Y_i - (W_i^*)^T X_i\big)\mathbf{1}_{n_i} \tag{16}$$

where I is the identity matrix and $\Pi_{n_i} = I - \frac{1}{n_i}\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T$ is the centering matrix. For the sake of brevity, we drop the subscript n_i from the centering matrix Π_{n_i} when the dimension can be easily inferred from the context.

Let J_i denote the *fitting error* of the local function $\mathbf{f}_i(\mathbf{x})$, which is given by the minimum of Eq. (14):

$$\begin{aligned}
&J_i\\
=&\frac{1}{n_i}\|Y_i - (W_i^*)^T X_i - \mathbf{b}_i^*\mathbf{1}_{n_i}^T\|_F^2 + \lambda\|W_i^*\|_F^2\\
=&\frac{1}{n_i}\|Y_i - (W_i^*)^T X_i - \frac{1}{n_i}\big(Y_i - (W_i^*)^T X_i\big)\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T\|_F^2 + \lambda\|W_i^*\|_F^2\\
=&\frac{1}{n_i}\|\big(Y_i - (W_i^*)^T X_i\big)\Pi\|_F^2 + \lambda\|W_i^*\|_F^2\\
=&\frac{1}{n_i}\|Y_i\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)\|_F^2 + \lambda\|(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi Y_i^T\|_F^2\\
=&\frac{1}{n_i}\mathrm{Tr}\Big(Y_i\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)^2 Y_i^T\Big)\\
&+\lambda\mathrm{Tr}\big(Y_i\Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-2}X_i\Pi Y_i^T\big)\\
=&\frac{1}{n_i}\mathrm{Tr}\ \big(Y_i\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)Y_i^T\big)
\end{aligned}\tag{17}$$

In the above derivations, we have used the fact that the centering matrix is idempotent, so that $\Pi = \Pi^k$ for $k = 1, 2, \cdots$. For each local patch $\mathcal{N}_i$, we define

$$L_i = \frac{1}{n_i}\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)\tag{18}$$

which characterizes the local discriminative structure of $\mathcal{N}_i$. The formulation of L_i in Eq. (18) involves the inverse of one $n \times n$ matrix, which is computationally expensive when the dimensionality is high. Using the Woodbury-Morrison formula[21], L_i can be reformulated as[1]:

$$\begin{aligned}
&\frac{1}{n_i}\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)\\
=&\frac{1}{n_i}\Pi\big(I - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)\Pi\\
=&\frac{1}{n_i}\Pi\big(I - I\Pi X_i^T(X_i\Pi I\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi I\big)\Pi\\
=&\frac{1}{n_i}\Pi(I + \frac{1}{n_i\lambda}\Pi X_i^T X_i\Pi)^{-1}\Pi\\
=&\lambda\Pi(n_i\lambda I + \Pi X_i^T X_i\Pi)^{-1}\Pi
\end{aligned}\tag{19}$$

The above equation needs to compute the inverse of one $n_i \times n_i$ matrix, which is quite efficient, since the size of the local patch is usually very small.

The fitting error J_i treats Y_i as the variable, and measures how well do the new representations respect the local discriminative structure. Thus, good representations should give rise to minimal fitting errors. The objective function of LRP is naturally defined as the summation of the fitting errors over all the local patches $\{\mathcal{N}_i\}_{i=1}^m$:

$$\sum_{i=1}^m J_i = \sum_{i=1}^m \mathrm{Tr}(Y_i L_i Y_i^T) = \sum_{i=1}^m \mathrm{Tr}(Y S_i L_i S_i^T Y) = \mathrm{Tr}\Big(Y\big(\sum_{i=1}^m S_i L_i S_i^T\big)Y^T\Big) \tag{20}$$

Because our goal is to learn a projection matrix P such that $Y = P^T X$, the objective function in terms of P is

$$\mathrm{Tr}\Big(P^T X\big(\sum_{i=1}^m S_i L_i S_i^T\big)X^T P\Big) \tag{21}$$

Finally, we have

Definition 1. Locally Regressive Projections (LRP):

$$\begin{aligned} &\min_P \mathrm{Tr}(P^T X L X^T P) \\ &\text{s.t. } P^T X X^T P = I \\ &\qquad L = \textstyle\sum_{i=1}^m S_i L_i S_i^T \end{aligned} \tag{22}$$

The constraint $P^T X X^T P = I$ is added to remove the arbitrary scaling factor in the projection. The matrix L is similar to the Laplacian matrix in Ref. [4], as indicated by the following theorem.

Theorem 3.1. L is a positive semi-definite matrix, and $\mathbf{1}$ is its eigenvector with eigenvalue 0.

Proof: From Eq. (19), it is obvious that the matrix L_i is positive semi-definite[1]. Thus, the matrix $L = \sum_{i=1}^m S_i L_i S_i^T$ is also positive semi-definite. Following Eq. (18), we can conclude that

$$L\mathbf{1} = \sum_{i=1}^m S_i L_i S_i^T \mathbf{1} = \sum_{i=1}^m S_i L_i \mathbf{1} = \mathbf{0}, \tag{23}$$

where we use the fact that $S_i^T \mathbf{1} = \mathbf{1}$ and $\Pi\mathbf{1} = \mathbf{0}$. So, $\mathbf{1}$ is an eigenvector of L with eigenvalue 0. □

Following the Rayleigh-Ritz theorem[17], we know that the optimal P^* that minimizes Eq. (22) is given by the smallest eigenvectors of the following generalized eigenvalue problem:

$$X L X^T \boldsymbol{\alpha} = \gamma X X^T \boldsymbol{\alpha} \tag{24}$$

Let $\{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_p\} \subset \mathbb{R}^n$ be the smallest eigenvectors of Eq. (24) ordered according to their eigenvalues, $\lambda_1 \leqslant \cdots \leqslant \lambda_p$. Then, P^* is given by

$$P^* = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_p] \tag{25}$$

3.2 The algorithm

In summary, the algorithm of LRP is stated below:

1. **Local relationship construction**: The neighbors of each data point can be found using the following two ways[4]:

(a) ϵ-neighborhoods. $\mathbf{x}_j$ is the neighbor of $\mathbf{x}_i$, if $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$.

(b) k nearest neighbors. $\mathbf{x}_j$ is the neighbor of $\mathbf{x}_i$, if $\mathbf{x}_j$ is among the k nearest points of $\mathbf{x}_i$.

Note that, if we are facing *supervised dimensionality reduction problem*, we can make use of the label information by requiring that neighboring points must belong to the same class.

Then, we calculate the matrix $L = \sum_{i=1}^{m}(S_i L_i S_i^T)$, where L_i is given by Eq. (18) and Eq. (19).

2. **Data centering and PCA projection**: The mean of $\mathbf{x}$ is removed from each $\mathbf{x}_i$:

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \tag{26}$$

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}, \text{ for } 1 \leqslant i \leqslant m \tag{27}$$

Then, we project each data point $\hat{\mathbf{x}}_i$ into the PCA subspace by throwing away the smallest principal components. We denote the projection matrix of PCA by P_{PCA}. Through data centering, the trivial solution $P^T X = \mathbf{1}_m$ is removed[5]. The role of PCA is to make the matrix XX^T positive definite, which is necessary in solving the generalized eigenvalue problem (24)[5]. We use $\widehat{X}_{PCA}$ denote the data matrix after this step.

3. **Calculating the Projection matrix**: Compute the eigenvectors for the following generalized eigenvalue problem:

$$\widehat{X}_{PCA} L \widehat{X}_{PCA}^T \boldsymbol{\alpha} = \lambda \widehat{X}_{PCA} \widehat{X}_{PCA}^T \boldsymbol{\alpha} \tag{28}$$

Denote the projection matrix resulting from solving the eigenvectors from Eq. (28) as P_{LRP}.

4. **Projection**: The final projection matrix P is given by $P = P_{PCA} P_{LRP}$. Given a test point $\mathbf{x}_i$, its projection value $\mathbf{y}_i$ is

$$\mathbf{y}_i = P^T(\mathbf{x}_i - \bar{\mathbf{x}}) = P_{LRP}^T P_{PCA}^T (\mathbf{x}_i - \bar{\mathbf{x}}) \tag{29}$$

3.3 Complexity analysis of LRP

Supposing we use the k nearest neighbors way to construct the local patches, the computational complexity of LRP is dominated by the following steps:

- Find the k nearest neighbors of each point, and compute L
  - $O(m^2 n)$ is used to calculate the pairwise distances between the m samples, and $O(m^2 \log m)$ is used for finding k-nearest neighbors of all the m samples.
  - $O(m(nk^2 + k^3))$ is used to calculate L_i according to Eq. (19) for all the m samples.

– PCA Projection

  – The complexity is $O(mn^2 + n^3)$.

– Solve the generalized eigenvalue problem (28)

  – The complexity is at most $O(m^2n + mn^2 + n^3)$.

In summary, the total cost of LRP is $O\big(m^2 \log m + mn(k^2 + m + n) + mk^3 + n^3\big)$.

4 Theoretical Analysis

4.1 Connections to LLE and LLP

LLE, LLP and LRP are all local learning algorithms for dimensionality reduction. The major difference of the three methods lies in their objective function. LLE and LLP minimize the summation of the estimation (or reconstruction) errors of all the data points. Thus, all the points are treated equally in LLE and LLP. On the other hand, LRP minimize the summation of the fitting errors over all the local patches. Since the fitting error at one patch contains the estimation errors of all the points in this patch, and points sampled from dense regions will appear in more patches than others, LRP actually assigns higher weights to points in dense regions. We expect in this way, LRP can model the local geometry structure more accurately.

One interesting property of LRP is that it is equivalent to PCA and LDA in some extreme cases. And we show this fact in the following.

4.2 Connection to PCA

Theorem 4.1. When $n_i = m$, LRP is equivalent to PCA.

Proof: Using the fact that $X_i = XS_i$ holds for all i, and after some mathematical derivations we have

$$\begin{aligned} XLX^T &= \sum_{i=1}^{m}(XS_iL_iS_i^TX^T) = \sum_{i=1}^{m} X_iL_iX_i^T \\ &= \sum_{i=1}^{m} \frac{1}{n_i} X_i\big(\Pi - \Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi\big)X_i^T \\ &= \sum_{i=1}^{m} \frac{1}{n_i}\big(X_i\Pi X_i^T - X_i\Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}X_i\Pi X_i^T\big) \\ &= \sum_{i=1}^{m} \frac{1}{n_i}\big(X_i\Pi X_i^T - X_i\Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}(X_i\Pi X_i^T + n_i\lambda I - n_i\lambda I)\big) \\ &= \lambda\sum_{i=1}^{m}\big(X_i\Pi X_i^T(X_i\Pi X_i^T + n_i\lambda I)^{-1}\big) \end{aligned} \tag{30}$$

$n_i = m$ means that the entire data set is treated as a neighborhood. So $X_i = X$ for all i, and Eq. (30) becomes:

$$XLX^T = m\lambda\big(X\Pi X^T(X\Pi X^T + m\lambda I)^{-1}\big) = XX^T(\frac{1}{m\lambda}X\Pi X^T + I)^{-1} \tag{31}$$

Since data centering is one preprocessing step in our algorithm, we can assume that data has been centered for simplicity. Then, the data covariance matrix C becomes:

$$C = \frac{1}{m} X\Pi X^T = \frac{1}{m} XX^T \tag{32}$$

Substituting Eq. (31) and Eq. (32) into Eq. (24), we have

$$mC(\frac{1}{\lambda}C + I)^{-1}\boldsymbol{\alpha} = \gamma mC\boldsymbol{\alpha} \Leftrightarrow (\frac{1}{\lambda}C + I)^{-1}\boldsymbol{\alpha} = \gamma\boldsymbol{\alpha} \Leftrightarrow (\frac{1}{\lambda}C + I)\boldsymbol{\alpha} = \frac{1}{\gamma}\boldsymbol{\alpha} \tag{33}$$

Thus, the optimal projections of LRP is given by the largest eigenvectors of $(\frac{1}{\lambda}C + I)$. Since the eigenvectors of $(\frac{1}{\lambda}C + I)$ and C are the same, and with the same orders, LRP is equivalent to PCA when $n_i = m$. □

4.3 Connection to LDA

Theorem 4.2. In supervised dimensionality reduction case, if all the points of the same class are treated as neighbors, then LRP converges to LDA as $\lambda \to \infty$.

Proof: Suppose the given m points are centered and belong to c classes. Let r_i denote the the number of samples in the i-th class, and X^i be the data matrix consisting of samples in the i-th class. Then Eq. (30) becomes:

$$\begin{aligned} & XLX^T \\ =& \sum_{i=1}^{c} r_i\lambda\Big(X^i\Pi_{r_i}(X^i)^T\big(X^i\Pi_{r_i}(X^i)^T + r_i\lambda I\big)^{-1}\Big) \\ =& \sum_{i=1}^{c}\left(X^i\Pi_{r_i}(X^i)^T\left(\frac{1}{r_i\lambda}X^i\Pi_{r_i}(X^i)^T + I\right)^{-1}\right) \end{aligned} \tag{34}$$

We have

$$\lim_{\lambda\to\infty} XLX^T = \sum_{i=1}^{c}\big(X^i\Pi_{r_i}(X^i)^T\big) \tag{35}$$

In LDA, it is easy to check that the following relationships hold for the within-class scatter matrix and between-class scatter matrix:

$$S_W = \sum_{i=1}^{c}\big(X^i\Pi_{r_i}(X^i)^T\big) \tag{36}$$

$$S_W + S_B = X\Pi_m X^T \tag{37}$$

Thus, the solution of LDA is also given by the smallest eigenvectors of the following generalized eigenvalue problem[6,15]:

$$\sum_{i=1}^{c} X^i\Pi_{r_i}(X^i)^T\boldsymbol{\alpha} = \gamma X\Pi_m X^T\boldsymbol{\alpha} \tag{38}$$

Following Eq. (4.35), we can conclude that as $\lambda \to \infty$, the eigenproblem (24) of LRP converges to the eigenproblem (38) of LDA.

In practice, λ is much smaller than ∞, so LRP can project points into a subspace whose dimensionality exceeds $c - 1$. □

5 Experiments

In this section, we evaluate the performance of our proposed LRP algorithm for both supervised dimensionality reduction (face recognition) and unsupervised dimensionality reduction (face clustering). To demonstrate the effectiveness of our proposed algorithm, we evaluate and compare five dimensionality reduction methods:

- **Principal Component Analysis (PCA)**[3];
- **Linear Discriminant Analysis (LDA)**[8];
- **Neighborhood Preserving Embedding (NPE)**[11], which is a linear approximation to Locally Linear Embedding (LLE)[18];
- **Local Learning Projections (LLP)**[24];
- **Locally Regressive Projections (LRP)**, which is the method proposed in this paper.

Two face images databases are used in the experiments: the CMU PIE face database and the Extended Yale-B face database. The PIE face database contains 41,368 images of 68 people. The face images were captured under 13 different Poses, 43 different Illumination conditions, and with 4 different Expressions. We choose the frontal pose (C27), which has 3329 face images. The Extended Yale-B face database contains 16128 images of 38 human subjects under 9 poses and 64 illumination conditions. We also choose the frontal pose, thus leaving us with 2414 images in total.

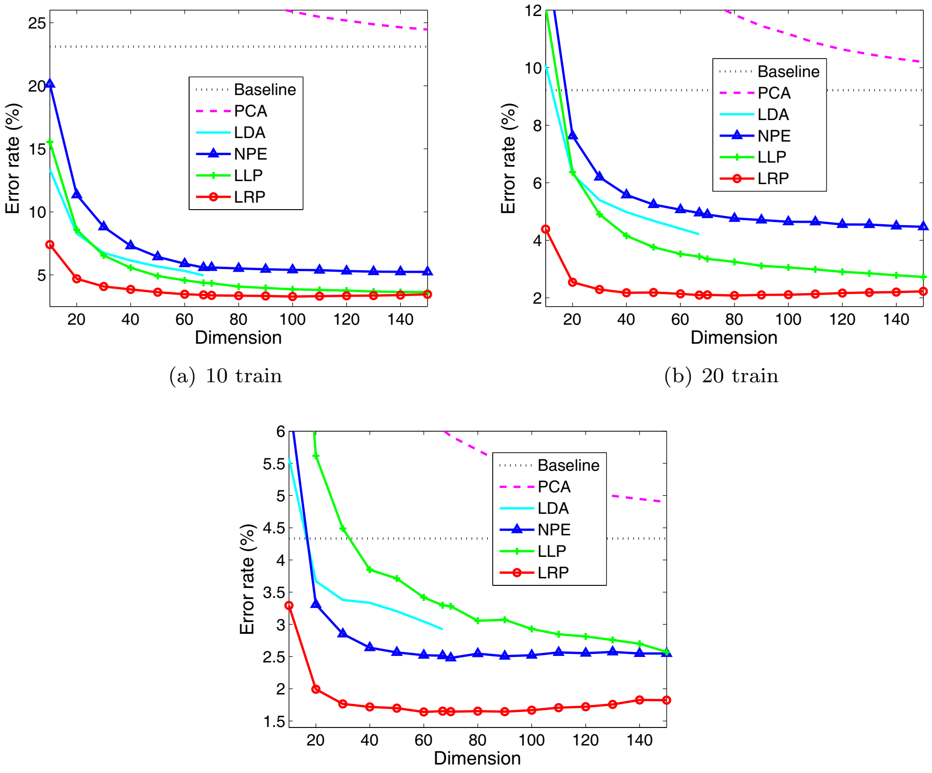
All the face images are manually aligned and resized to 32×32 pixels. So, each image is represented as a 1024-dimensional vector. In our experiments we pre-process the data by normalizing each face vector to unit length. We apply the k nearest neighbors way to finding the local patches for NPE, LLP and LRP. k is empirically set to 5 for all the methods. The parameter λ in LLP and LRP is set to 1.

5.1 Face recognition

For face recognition, we compare our algorithm with PCA[2], LDA[2], NPE[11], and LLP[24]. Classification in the original 1024-dimensional space is referred to as **Baseline**.

For each database, r images per class are randomly selected as training samples, and the rest are used for testing. The training samples are used to learn the projection matrix $P \in \mathbb{R}^{n \times p}$ for each method. For PCA, NPE, LLP and LRP, the dimension of the subspace (i.e., p) varies from 1 to 150. For LDA, p varies from 1 to $c-1$, where c is the number of classes. To make use of the label information, in NPE, LLP, and LRP, we require that the neighboring points belong to the same class. Both the training and testing data are mapped into a low-dimensional subspace by the learned matrix P. Then, 1-nearest neighbor (1-nn) classifier is used to classify the testing data. For each given r, 20 training/testing splits are randomly generated and the average testing error over these splits is used to evaluate the classification performance.

We show the error rate versus the dimension for each algorithm on the PIE and Yale-B databases in Figs. 1 and 2, respectively. We can see that the performance of these algorithms varies with the number of dimensions. Tables 2 and Table 3 show the best results together with the standard deviations obtained by these algorithms, while the numbers in parentheses denote the optimal number of dimensions. As can be seen, our LRP algorithm outperforms all the other algorithms on the entire range. And the error rate of LRP decreases much faster than other algorithms as the dimension increases.



(a) 10 train

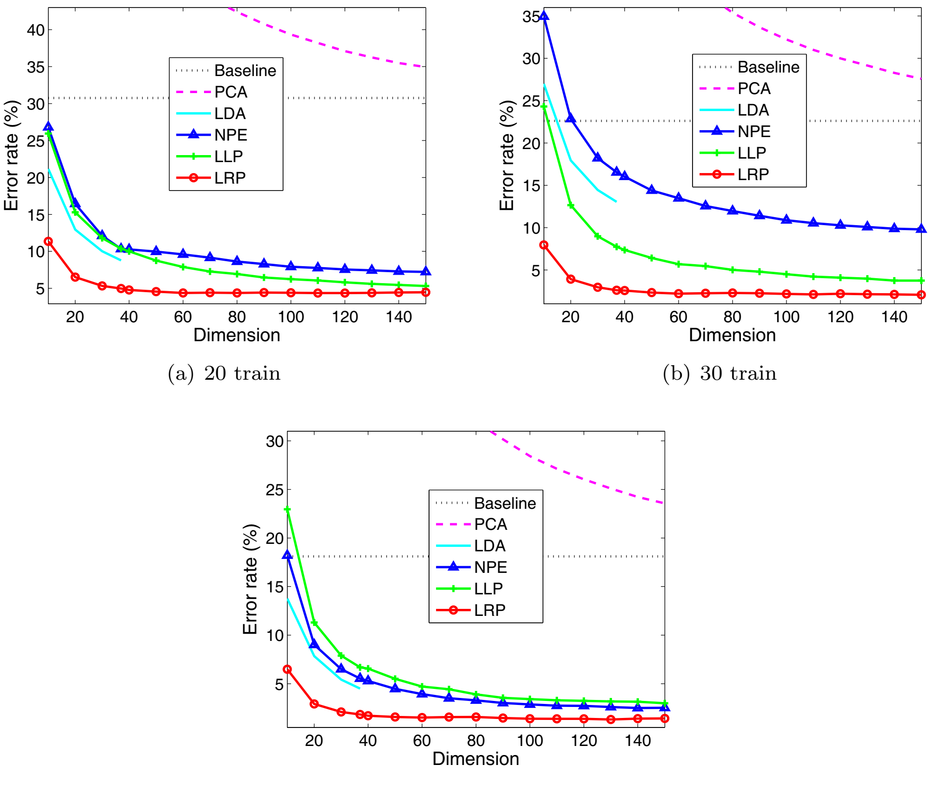
(b) 20 train

(c) 30 train

Figure 1. Error rate vs. dimensionality reduction on CMU PIE database.

Table 2 Recognition error rate of different algorithms on the CMU PIE database (mean±std-dev%)

Method	10 train	20 train	30 train
Baseline	23.1 ± 0.89 (1024)	9.22 ± 0.54 (1024)	4.33 ± 0.51 (1024)
PCA	24.5 ± 0.89 (150)	10.2 ± 0.57 (150)	4.9 ± 0.51 (150)
LDA	4.96 ± 0.40 (67)	4.21 ± 0.45 (67)	2.92 ± 0.45 (67)
NPE	5.23 ± 0.35 (147)	4.44 ± 0.59 (147)	2.44 ± 0.38 (63)
LLP	3.64 ± 0.43 (146)	2.72 ± 0.32 (150)	2.57 ± 0.42 (150)
LRP	**3.29** ± **0.37** (100)	**2.05** ± **0.33** (85)	**1.6** ± **0.31** (78)



(a) 20 train

(b) 30 train

(c) 40 train

Figure 2. Error rate versus dimension on Yale-B database.

Table 3 Recognition error rate of different algorithms on the Yale-B database (mean±std- dev%).

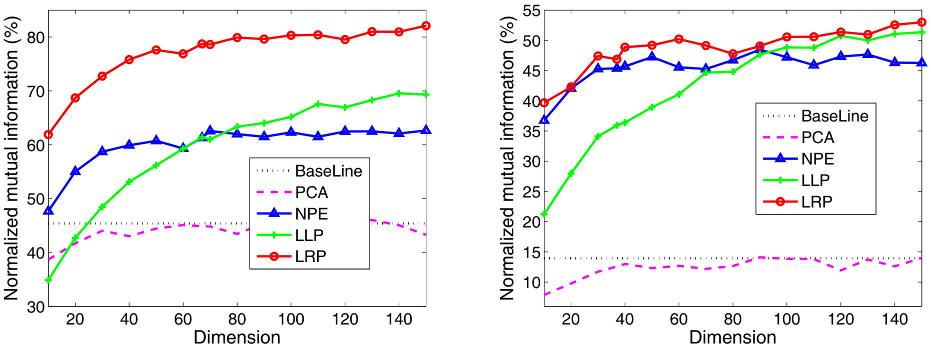
Method	20 train	30 train	40 train
Baseline	30.8 ± 1.2 (1024)	22.6 ± 0.98 (1024)	18.1 ± 1.1 (1024)
PCA	34.9 ± 1.1 (150)	27.6 ± 1.0 (150)	23.6 ± 1.0 (150)
LDA	8.78 ± 0.87 (37)	13.0 ± 1.3 (37)	4.50 ± 0.72 (37)
NPE	7.2 ± 0.77 (146)	9.74 ± 0.96 (148)	2.47 ± 0.41 (139)
LLP	5.3 ± 0.55 (144)	3.72 ± 0.59 (146)	2.99 ± 0.67 (150)
LRP	**4.28 ± 0.65** (114)	**2.06 ± 0.42** (149)	**1.32 ± 0.51** (130)

5.2 Face clustering

Face clustering is unsupervised and we compare our algorithm with PCA, NPE, and LLP. In our experiments, the entire database in used to learn a projection matrix P, and all the points are mapped into a low dimensional subspace. The low dimensional representations are centered and normalized to unit length in the projected subspace before clustering. We use k-means as our clustering algorithm. The result of k-means in the original feature space is referred to as **Baseline**.

Since k-means algorithm can only find local minimum, and is sensitive to initial points. So in each case we apply it 10 times with different start points and the best result in terms of the objective function of k-means is recorded. The clustering performance is evaluated by comparing the obtained label of each image with that provided by the ground truth. The normalized mutual information metric ($\overline{MI}$) is used to measure the clustering performance[5]. $\overline{MI}$ ranges from 0 to 1. It equals 1 if two sets of clusters are identical, and equals 0 if two sets are independent.

Figure 3 plots the normalized mutual information versus the dimension for the Baseline, PCA, NPE, LLP, and LRP on the two databases. We observe that our proposed LRP outperforms other algorithms on both data sets. The performance of LLP is bad when the dimension is low, but it increases quickly as the dimension increases.



(a) The CMU PIE database.

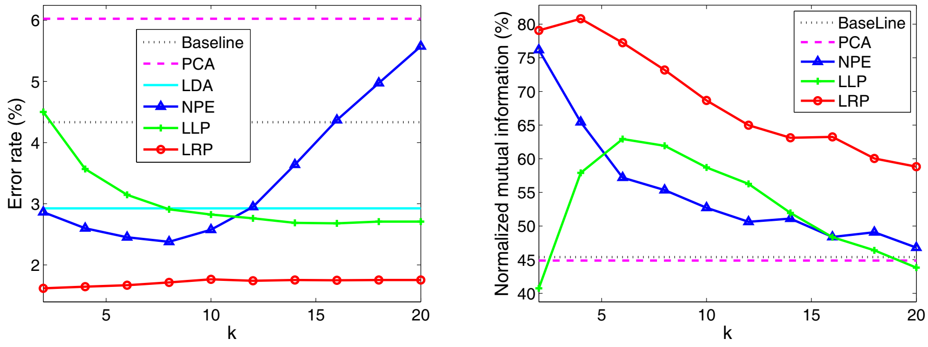
(b) The Yale-B database.

Figure 3. Normalized mutual information versus dimension on the CMU PIE and Yale-B databases.

5.3 *Parameter selection*

The size of the local patch is an essential parameter of all the local learning methods. In the previous experiments, we construct the local patch by finding the k nearest neighbors of each points and k is set to 5. In the following, we examine the impact of k on the performance of NPE, LLP and LRP. For brevity, we just show the results on the CMU PIE database, and the results on the Yale-B database is similar. Since the CMU PIE database contains 68 classes, we fix the dimension of the subspace at 67.

For face recognition, we select $r = 30$ images per class as training samples, and report the average testing error over 20 training/testing splits. The experiment of face clustering uses the whole database. Figure 4 shows the results of face recognition and clustering versus k on the CMU PIE database. The performance of LRP for face recognition is not sensitive to k. Specifically, the error rate of LRP only increases a little as k increases from 2 to 20. That is probably because in the case of face recognition, the label information is used to find the k nearest neighbors. On the other hand, the performance of LRP for face clustering changes noticeably as k varies. Nevertheless, our LRP outperforms all the other methods on the whole range of k for both face recognition and clustering. And the best size of the local patch is around 5.



(a) Results of face recognition.

(b) Results of face clustering.

Figure 4. The impact of the number of nearest neighbors k on the CMU PIE database.

6 Conclusions

This paper proposes a novel linear dimensionality reduction algorithm called Locally Regressive Projections (LRP). Unlike previous methods, we assumes the low dimensional representation of each point as well as its neighbors can be well estimated using a locally fitted function. The optimal projections are found by minimizing the summation of the fitting errors of all the local functions. Theoretical analysis reveals that there are close connections between our proposed LRP algorithm and the canonical methods PCA and LDA. Experimental results on two standard databases show that our algorithm can significantly improve the performance of both supervised face recognition and unsupervised face clustering.

References

[1] Bach F, Harchaoui Z. Diffrac: a discriminative and flexible framework for clustering. Advances in Neural Information Processing Systems, 2008, 20: 49–56.

[2] Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. on Pattern Analysis and Machine Intelligence, Jul. 1997, 19(7): 711–720.

[3] Bishop CM. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2007.

[4] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems, 2002, 14: 585–591.

[5] Cai D, He X, Han J. Document clustering using locality preserving indexing. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(12): 1624–1637.

[6] Cai D, He X, Han J. Srda: An efficient algorithm for large-scale discriminant analysis. IEEE Trans. on Knowledge and Data Engineering, Jan. 2008, 20(1): 1–12.

[7] Carreira-Perpiñán MÁ. The elastic embedding algorithm for dimensionality reduction. Proc. of the 27th International Conference on Machine Learning. 2010. 167–174.

[8] Duda RO, Hart PE, Stork DG. Pattern Classification. Wiley-Interscience Publication, 2000.

[9] De Silva V, Tenenbaum JB. Global versus local methods in nonlinear dimensionality reduction. Advances in Neural Information Processing Systems, 2003, 15: 705–712.

[10] Fodor IK. A survey of dimension reduction techniques[Technical Report]. Center for Applied Scientific Computing. Lawrence Livermore National Laboratory. Jun. 2002.

[11] He X, Cai D, Yan S, Zhang HJ. Neighborhood preserving embedding. Proc. of the 10th IEEE International Conference on Computer Vision, Oct. 2005, 2: 1208–1213.

[12] He X, Niyogi P. Locality preserving projections. Advances in Neural Information Processing Systems, 2004, 16: 153–160.

[13] Hinton G, Roweis S. Stochastic neighbor embedding. Advances in Neural Information Processing Systems, 2003, 15: 857–864.

[14] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York. 2009.

[15] He X, Yan S, Hu Y, Niyogi P, Zhang HJ. Face recognition using laplacianfaces. IEEE Trans. on Pattern Analysis and Machine Intelligence, Mar. 2005, 27(3): 328–340.

[16] Kim M, De la Torre F. Local minima embedding. Proc. of the 27 th International Conference on Machine Learning. 2010. 527–534.

[17] Lütkepohl H. Handbook of Matrices. John Wiley & Sons Inc, 1996.

[18] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science, Dec. 2000, 290(5500): 2323–2326.

[19] Shaw B, Jebara T. Structure preserving embedding. Proc. of the 26th Annual International Conference on Machine Learning. 2009. 937–944.

[20] Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis. Cambridge University Press. New York, NY, USA. 2004.

[21] Strang G. Introduction to Linear Algebra(3rd Edition). Wellesley-Cambridge Press, 2003.

[22] Turk MA, Pentland AP. Face recognition using eigenfaces. Proc. of 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1991. 586–591.

[23] Wang HY, Yang Q, Qin H, Zha H. Dirichlet component analysis: feature extraction for compositional data. Proc. of the 25th International Conference on Machine Learning. 2008. 1128–1135.

[24] Wu MR, Yu K, Yu SP, Schölkopf B. Local learning projections. Proc. of the 24th International Conference on Machine Learning. 2007. 1039–1046.

[25] Yang Y, Xu D, Nie FP, Luo JB, Zhuang YT. Ranking with local regression and global alignment for cross media retrieval. Proc. of the 17th Annual ACM International Conference on Multimedia. 2009. 175–184.

[26] Yan SC, Xu D, Zhang BY, Zhang HJ. Graph embedding: A general framework for dimensionality reduction. Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005. 830–837.

[27] Zhang LJ, Chen C, Bu JJ, Chen ZG, Cai D, Han JW. Locally discriminative coclustering. IEEE Trans. on Knowledge and Data Engineering, 2012, 24(6): 1025–1035.

[28] Zhang Y, Yeung DY. Worst-case linear discriminant analysis. Advances in Neural Information Processing Systems, 2011, 23: 2568–2576.