Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 603025, 11 pages
http://dx.doi.org/10.1155/2014/603025

Research Article

Dimension Reduction Using Samples' Inner Structure Based Graph for Face Recognition

Bin Li,[1] Wei Pang,[2] Yuhao Liu,[1] Xiangchun Yu,[1] Anan Du,[1] Yecheng Zhang,[1] and Zhezhou Yu[1]

[1] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2] *School of Natural and Computing Sciences, University of Aberdeen, Aberdeen AB24 3UE, UK*

Correspondence should be addressed to Zhezhou Yu; yuzz@jlu.edu.cn

Received 31 March 2014; Accepted 5 June 2014; Published 3 July 2014

Academic Editor: Zexuan Zhu

Graph construction plays a vital role in improving the performance of graph-based dimension reduction (DR) algorithms. In this paper, we propose a novel graph construction method, and we name the graph constructed from such method as samples' inner structure based graph (SISG). Instead of determining the k-nearest neighbors of each sample by calculating the Euclidean distance between vectorized sample pairs, our new method employs the newly defined sample similarities to calculate the neighbors of each sample, and the newly defined sample similarities are based on the samples' inner structure information. The SISG not only reveals the inner structure information of the original sample matrix, but also avoids predefining the parameter k as used in the k-nearest neighbor method. In order to demonstrate the effectiveness of SISG, we apply it to an unsupervised DR algorithm, locality preserving projection (LPP). Experimental results on several benchmark face databases verify the feasibility and effectiveness of SISG.

1. Introduction

Dimensionality reduction (DR) [1–4] has been intensively used as an effective approach to analyze high-dimensional data, especially face images. In particular, graph-based DR receives more and more attention recently in the fields of pattern recognition and machine learning. It is stated that most existing DR methods [5–11] actually fall into the graph embedding framework [12]. In graph embedding algorithms graph construction plays a vital role, because graph is an effective tool to reveal the structure information hidden in the original data. So it is worthwhile to study graph construction [13–17] and develop novel construction approaches to construct more reasonable graphs for graph-based DR methods. Jebara et al. [14] presented the so-called b-matching graph, which is an alternative approach to the traditional k-nearest neighbor graph. The authors in [15, 17] focused on developing a way of combining different graphs so that a better graph will be given a heavier weight. However, we point out that the traditional graph construction method suffers from the following two issues.

(1) The k-nearest neighbors of each sample are based on the Euclidean distance between every two vectorized samples. However, samples' inner structure information is not taken into consideration by the traditional graph construction method, and such information can be utilized to construct better graphs for dimension reduction algorithms.

(2) The same neighbor parameter k (or ε) [18, 19] has to be predefined for all samples before constructing graphs. This may cause the difficulty of parameter selection and it is not reasonable to set the same parameter value for all samples.

To mitigate the shortcomings of the traditional graph construction method, in this paper we present a samples' inner structure based graph construction method, and we name the graph constructed from such method as samples' inner structure based graph (SISG). In this new method we first use the column similarity to determine the nearest neighbors of each column for sample matrices. Then we use the sample similarity measured by the number of nearest neighbor columns between sample pairs to determine the nearest neighbors of each sample. This strategy not only avoids the stiff criterion (predefining the same parameter k for all samples) as used in the traditional graphs but also utilizes every sample's inner structure information. We summarize the favorable and attractive characteristics of SISG as follows.

(1) SISG preserves samples' intrinsic features by using the inner structure information of sample matrices to construct graph.

(2) SISG uses the newly defined sample similarities to calculate the neighbors of each sample. This strategy avoids predefining the neighbor parameter k (or ε) in traditional graph construction methods.

(3) The edge weights of SISG are determined by the sample similarities between sample pairs. If the sample similarity between two samples is high, the edge weights between these two samples will be big. This means the greater the sample similarity between two samples is, the more important the corresponding edge is in the graph.

(4) Both the weighted adjacency matrix and the adjacency matrix of SISG are generally asymmetric. This characteristic may be more reasonable for capturing the relationship among samples.

(5) The construction method of SISG is very general. It can be applied to many graph-based DR algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews traditional graph construction and locality preserving projection (LPP). Section 3 firstly presents SISG's construction method and then applies SISG to LPP. In Section 4 we perform a series of the experiments to evaluate the feasibility and effectiveness of SISG. This is followed by the conclusions made in Section 5.

2. Related Work

2.1. Traditional Graph Construction. Let $A = \{A_1, \ldots, A_N\}$ be a set of N sample matrices, which are taken from an $(n \times m)$ dimensional image space. Then, the original samples were transformed into their vectorial forms, and we denote these vectors by $X = \{x_1, \ldots, x_N\}$, $x_i \in R^D$. The weighted graph can be denoted by $G = \{V, E, W\}$, where V corresponds to the vectors in the set X, E is the set of edges, each of which is between one sample pair, and the matrix W contains weight values of the edges among sample pairs. The construction process of the graph G normally consists of two steps.

The first step is the construction of edges, which includes the two categories of ε-neighborhood and k-nearest neighbor [12, 19].

ε-Neighborhood: x_i and x_j are connected if $\|x_i - x_j\| < \varepsilon$, where $\|\cdot\|$ is the Euclidean distance in R^D and ε is a local threshold parameter.

k-Nearest neighbor: x_i and x_j are connected if x_i is one of the k-nearest neighbors of x_j or x_j is one of the k-nearest neighbors of X_i.

The second step is the calculation of the weight value for each edge. The weight value of W_{ij} can be calculated by the following two ways [12]:

(1) heat kernel

$$W_{ij} = \begin{cases} \exp\left(-\dfrac{\left\|x_i - x_j\right\|^2}{t}\right), & \text{if } x_i \text{ and } x_j \text{ are neighbors,} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where t is the width parameter in the heat kernel;

(2) simple minded

$$W_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are neighbors,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

2.2. Locality Preserving Projection. The aim of locality preserving projection (LPP) [20] is to map the sample set in high-dimensional space into the low-dimensional one, in which the local manifold structures of high-dimensional space are preserved. This means if the original points x_i and x_j in the high-dimensional space are neighbors, the corresponding points y_i and y_j in the projected low-dimensional space are also neighbors. Suppose the projection from the high-dimensional space to the low-dimensional one is $y_i = \alpha^T x_i$, where α is the projection vector; the objective function of LPP is given in [20]

$$\sum_{ij} \left\| y_i - y_j \right\|^2 w_{ij} = \sum_{ij} \left\| \alpha^T x_i - \alpha^T x_j \right\|^2 = 2\alpha^T X L X^T \alpha. \tag{3}$$

In (3) the weighted adjacency matrix W can be computed by either (1) or (2). The matrix D is a diagonal one $D_{ii} = \sum_j w_{ij}$, and $L = D - W$. The constraint $\alpha^T X D X^T \alpha = 1$ was added so that the arbitrary scaling in the embedding can be removed. The minimization problem in (3) thus becomes

$$\alpha^* = \arg \min_{\alpha^T X D X^T \alpha = 1} \alpha^T X L X^T \alpha. \tag{4}$$

LPP can be solved by the generalized eigenvector approach [20]:

$$X L X^T \alpha = \lambda X D X^T \alpha. \tag{5}$$

3. Samples' Inner Structure Based Graph (SISG)

As discussed in Section 1, the traditional graph construction method suffers from two main issues: the stiff criterion problem and the loss of inner structure information of samples.

```
**INPUT**: Sample set $A = \{A_1, \ldots, A_i, \ldots, A_N\}$, where $A_i$ is a sample matrix.
**OUTPUT**: The weighted adjacency matrix $W^{\mathrm{SISG}}$ of SISG
**Algorithm**:
for $i = 1$ to $N$ ($N$ is the number of the training samples)
    for $j = 1$ to $N$
        for $l = 1$ to $m$ ($m$ is the column number of sample matrix)
            if $\exp\{-C(A_i^l, A_j^l)/t\} > (1/n)\sum_{k=1}^{N}\exp\{-C(A_i^l, A_k^l)/t\}$
                $G_{ij}^l = 1$;
            else
                $G_{ij}^l = 0$;
            end if
        end for
    end for
    for $j = 1$ to $N$
        $S_{ij} = \sum_{l=1}^{m} G_{ij}^l$;
        if $S_{ij} > \|S_{i\cdot}\|_1 / \|S_{i\cdot}\|_0$
            $W_{ij}^{\mathrm{SISG}} = S_{ij} \cdot \left[-\exp\left(\frac{\|x_i - x_j\|^2}{t}\right)\right]$;
            ($X_i$ is the vectorization representation of $A_i$)
        else
            $W_{ij}^{\mathrm{SISG}} = 0$;
        end if
    end for
end for
```

Algorithm 1: Construction of SISG.

To overcome these limitations to some extent, in this section we first present a new approach to graph construction, and graphs constructed by this new approach are called samples' inner structure based graph (SISG); then we incorporate SISG into LPP, which forms a new algorithm called SISG-LPP.

3.1. The Construction of SISG. Given a set of N sample matrices $A = \{A_1, \ldots, A_N\}$, which is taken from an ($n \times m$) dimensional image space, we let $X = \{x_1, \ldots, x_N\}$, $x_i \in R^D$, denote the vector pattern of image set A, and let l ($l = 1, 2, \ldots, m$, and m is the maximum column number of the sample matrix) be the column number of each sample (Algorithm 1). A_i^l denotes the lth column of sample matrix A_i. SISG can be denoted by SISG $= \{V, E, S, W^{\mathrm{SISG}}\}$, where V corresponds to the vectors in the set X, E is the adjacency matrix of SISG which denotes the edge set between the sample pairs, S is the sample similarity matrix of SISG which denotes sample similarities between sample pairs, and W^{SISG} is the weighted adjacency matrix which denotes the weight values of the edges between sample pairs. There are two steps to build a SISG, as detailed below.

Step 1. For the lth columns of all the samples, we calculate the nearest neighbors of each column.

Definition 1 (column similarity). Column similarity is calculated by the column similarity function: $\exp\{-C(A_i^l, A_j^l)/t\}$, where $C(A_i^l, A_j^l) = \|A_i^l - A_j^l\| / \sum_{k=1}^{N}\|A_i^l - A_k^l\|$ and t is the width parameter in the heat kernel. Let G^l denote the adjacency matrix of all samples' lth columns:

$$G_{ij}^l = \begin{cases} 1, & \text{if } \exp\left\{-\dfrac{C\left(A_i^l, A_j^l\right)}{t}\right\} > \dfrac{1}{n}\displaystyle\sum_{k=1}^{N}\exp\left\{-\dfrac{C\left(A_i^l, A_k^l\right)}{t}\right\}, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The meaning of (6) is as follows: for the lth column of the sample matrix i (A_i^l), if the column similarity between this column and A_j^l is greater than the mean of column similarities between A_i^l and all other samples' lth column, A_j^l will become a neighbor of A_i^l, and we place an edge between A_i^l and A_j^l; that is, $G_{ij}^l = 1$.

Figure 1 shows the 3 nearest column neighbors of A_i^l, and the black rectangular boxes in the sample matrices represent the lth column of each sample.

Step 2. Determine every sample's neighbors by the sample similarity between sample pairs, and calculate the weight value for each neighbor pair .

In this step, the original samples are transformed into their forms representation, and we denote these vectors by $X = \{x_1, \ldots, x_N\}$, $x_i \in R^D$.

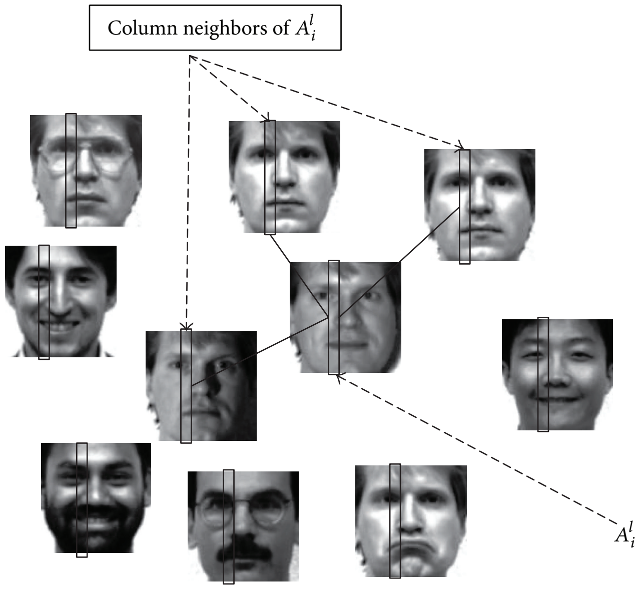


Figure 1: The 3 nearest neighbors of Sample A_i's lth column.

Definition 2 (sample similarity). Sample similarity is determined by the number of column neighbors between sample pairs.

We let S_{ij} denote the number of column neighbors of sample x_i for x_j. So, $S_{ij} = \sum_{l=1}^{m} G_{ij}^l$ (m is the maximum column number of a sample matrix) means the number of column neighbors of sample matrix A_i for sample matrix A_j. It is noted that S_{ij} is normally not equal to S_{ji}, and this characteristic is simply shown in Section 3.2. Consequently, the sample similarity of sample x_i for x_j is described by S_{ij}.

When deciding which samples can be the nearest neighbors of sample x_i, we only consider those samples whose sample similarity with sample x_i is nonzero. This is because if there are no nearest neighbor columns between two sample matrices, these two sample matrices will not be similar at all, and thus they cannot become neighbors. The weighted adjacency matrix W^{SISG} of the SISG is constructed according to the following equation:

$$W_{ij}^{\mathrm{SISG}} = \begin{cases} S_{ij} \cdot \left[- \exp\left(\dfrac{\left\| x_i - x_j \right\|^2}{t} \right) \right], & \text{if } S_{ij} > \dfrac{\|S_{i\cdot}\|_1}{\|S_{i\cdot}\|_0}, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $S_{i\cdot}$ is a vector and $S_{i\cdot} = [S_{ij}]$, $j = 1, \ldots, N-1$. $S_{i\cdot}$ denotes the sample similarity of sample x_i for all other samples. $\|\cdot\|_0$ represents the L^0-norm, which is the number of nonzero entries in a vector. So $\|S_{i\cdot}\|_0$ denotes the number of nonzero elements in vector $S_{i\cdot}$. $\|\cdot\|_1$ represents the L^1-norm, which is a linear combination of the absolute values of all entries in a vector. So $\|S_{i\cdot}\|_1$ is the sum of all entries in vector $S_{i\cdot}$.

The meaning of (7) is as follows: if the sample similarity between samples x_i and x_j is greater than the mean of sample similarities between x_i and all samples, then x_j becomes a neighbor of x_i, and we put an edge between them; that is, $E_{ij} = 1$.

The weight value of the edge between x_i and x_j is the heat kernel multiplied by the sample similarity between samples x_i and x_j. By doing this, the greater the sample similarity between samples x_i and x_j, the more important the corresponding edge in W^{SISG}. The meaning of W_{ij}^{SISG} is as follows: the weight value between x_i and x_j is proportional to their sample similarity and inversely proportional to their Euclidean distance.

3.2. Characteristic of SISG. Both the weighted adjacency matrix W^{SISG} and adjacency matrix E are asymmetric in most situations, and the symmetric ones are only special cases. This characteristic is more reasonable for effectively capturing and fitting the relationship among samples.

For A_i^l, the mean of column similarities between this column and all other samples' lth column is calculated as follows:

$$M_i^l = \frac{1}{n} \sum_{k=1}^{N} \exp\left\{ - \frac{C\left(A_i^l, A_k^l\right)}{t} \right\}. \tag{8}$$

For A_j^l, the mean of column similarities between this column and all other samples' lth column is calculated as follows:

$$M_j^l = \frac{1}{n} \sum_{k=1}^{N} \exp\left\{ - \frac{C\left(A_j^l, A_k^l\right)}{t} \right\}. \tag{9}$$

(1) The following situations may arise when calculating the column similarity.

(1.1) if

$$\begin{gathered} \exp\left\{ - \frac{C\left(A_i^l, A_j^l\right)}{t} \right\} > M_i^l, \\ \exp\left\{ - \frac{C\left(A_j^l, A_i^l\right)}{t} \right\} > M_j^l, \end{gathered} \tag{10}$$

A_i^l and A_j^l become column neighbors of each other.

(1.2) if

$$\begin{gathered} \exp\left\{ - \frac{C\left(A_i^l, A_j^l\right)}{t} \right\} \le M_i^l, \\ \exp\left\{ - \frac{C\left(A_j^l, A_i^l\right)}{t} \right\} \le M_j^l, \end{gathered} \tag{11}$$

A_i^l is not a column neighbor of A_j^l, and vice versa.

(1.3) if

$$\exp\left\{-\frac{C\left(A_i^l, A_j^l\right)}{t}\right\} > M_i^l,$$
$$\text{while } \exp\left\{-\frac{C\left(A_j^l, A_i^l\right)}{t}\right\} \leq M_j^l, \tag{12}$$

A_j^l is a column neighbor of A_i^l while A_i^l is not a column neighbor of A_j^l.

(2) The following situations may arise when calculating the column similarity.

(2.1) if

$$S_{ij} > \frac{\|S_{i\cdot}\|_1}{\|S_{i\cdot}\|_0}, \qquad S_{ji} > \frac{\|S_{j\cdot}\|_1}{\|S_{j\cdot}\|_0}, \tag{13}$$

sample A_i and sample A_j become neighbors of each other.

(2.2) if

$$S_{ij} \leq \frac{\|S_{i\cdot}\|_1}{\|S_{i\cdot}\|_0}, \qquad S_{ji} \leq \frac{\|S_{j\cdot}\|_1}{\|S_{j\cdot}\|_0}, \tag{14}$$

sample A_i is not a neighbor of sample A_j, and vice versa.

(2.3) if

$$S_{ij} > \frac{\|S_{i\cdot}\|_1}{\|S_{i\cdot}\|_0}, \qquad S_{ji} \leq \frac{\|S_{j\cdot}\|_1}{\|S_{j\cdot}\|_0}, \tag{15}$$

sample A_j is a neighbor of sample A_i while sample A_i is not a neighbor of sample A_j.

If all the columns of A_i (A_i^l, $l = 1, 2 \ldots, m$) and all the columns of A_j (A_j^l, $l = 1, 2 \ldots, m$) meet (1.1) or (1.2), S_{ij} will equal S_{ji}. Furthermore, if (2.1) or (2.2) can always be met for arbitrary A_i and A_j ($i, j = 1, \ldots, n$), $W_{ij}^{\text{SISG}} = W_{ji}^{\text{SISG}}$. This is because the edge weights of SISG depend on the sample similarity of each sample. In this case, both the weighted adjacency matrix W^{SISG} and the adjacency matrix E are symmetric.

If the condition in (1.3) is met, S_{ij} will not be equal to S_{ji}. Furthermore, if (2.1) or (2.2) can always be met for arbitrary A_i and A_j ($i, j = 1, \ldots, n$), the adjacency matrix E will be symmetric, but W_{ij}^{SISG} will not be equal to W_{ji}^{SISG} and the weighted adjacency matrix W^{SISG} will be asymmetric.

Apart from the two cases discussed in the above two paragraphs, both the weighted adjacency matrix W^{SISG} and adjacency matrix E are normally asymmetric.

We should notice that for all the columns of A_i (A_i^l, $l = 1, 2 \ldots, m$) and all the columns of A_j they must meet (1.1) or (1.2) at the same time, but (1.1) can never be met alone. Because M_i^l is the mean of column similarities, it is impossible that all the column similarities between any two columns A_i^l and A_j^l are both greater than M_i^l. For similar reasons, (1.2), (2.1), and (2.2) can never be met alone either.

From what has been discussed above, we can see that both the weighted adjacency matrix W^{SISG} and adjacency matrix E are generally asymmetric, and symmetric ones are just their special cases, which are very rare situations.

3.3. SISG-LPP. The construction method of SISG is very general. So SISG can be used in many graph-based dimensionality reduction algorithms. In this subsection, we use SISG in state-of-the-art unsupervised DR algorithm, locality preserving projection (LPP), to develop a new DR algorithm called SISG-LPP.

Similar to LPP, the goal of SISG-LPP is preserving the local manifold structures in high-dimensional space. Given a set of N samples $X = \{x_1, \ldots, x_N\}$, $x_i \in R^D$, in high-dimensional space, we try to find a transformation matrix which can map these N points to a set of points $y_1, \ldots, y_N$ in low-dimensional space. Assuming that the projection is $y = \alpha^T x$, where α is the projection vector, the objective function of SISG-LPP is given in

$$\begin{aligned}
&\sum_{ij} \left\| y_i - y_j \right\|^2 W^{\text{SISG}} \\
&\quad = \sum_{ij} \left(y_i^2 + y_j^2 - 2 y_i y_i \right) W^{\text{SISG}} \\
&\quad = \sum_{ij} y_i^2 W_{ij}^{\text{SISG}} + \sum_{ij} y_j^2 W_{ij}^{\text{SISG}} - \sum_{ij} y_i y_j W_{ij}^{\text{SISG}} \\
&\qquad - \sum_{ij} y_j y_i W_{ij}^{\text{SISG}} \\
&\quad = \sum_{ij} y_i^2 D_{ii} + \sum_{ij} y_i^2 D'_{jj} - \sum_{ij} y_i y_j W_{ij}^{\text{SISG}} - \sum_{ij} y_j y_i W_{ij}^{\text{SISG}} \\
&\quad = \alpha^T X \left(\left(D_{ii} + D'_{jj} \right) - \left(W_{ij}^{\text{SISG}} + \left(W_{ij}^{\text{SISG}} \right)^T \right) \right) X^T \alpha \\
&\quad = \alpha^T X \left(\overline{D} - \overline{W} \right) X^T \alpha \\
&\quad = \alpha^T X \overline{L} X^T \alpha.
\end{aligned} \tag{16}$$

In the above, W^{SISG} is an asymmetric matrix, but $\overline{W} = W_{ij}^{\text{SISG}} + (W_{ij}^{\text{SISG}})^T$ is a symmetric matrix. Let D_{ii} and D'_{jj} denote the diagonal matrices; the entries of D_{ii} are column sums of W^{SISG}, and the entries of D'_{jj} are column sums of $(W_{ij}^{\text{SISG}})^T$. $\overline{D} = D_{ii} + D'_{jj}$ is the diagonal matrix whose entries are column sums of $\overline{W}$. So, $\overline{L} = \overline{D} - \overline{W}$ is the Laplacian matrix. Because $\overline{D}$ provides a measure on the “importance” of the data points, we impose the following constraint:

$$\alpha^T X \left(D_{ii} + D'_{jj} \right) X^T \alpha = \alpha^T X \overline{D} X^T \alpha = 1. \tag{17}$$

Thus, the optimization objective of SISG-LPP is

$$\alpha^* = \arg \min_{\alpha^T X \overline{D} X^T \alpha = 1} \alpha^T X \overline{L} X^T \alpha. \tag{18}$$

The solutions of (18) can be obtained by solving the generalized eigenvalue decomposition problem [21]. That is to say, the projection vectors of (18) are actually the eigenvectors which correspond to the first l smallest eigenvalues of $X\overline{L}X^T\alpha = \lambda X\overline{D}X^T\alpha$ [22].

4. Experiments

In order to intuitively illustrate the construction process and the properties of SISG, we created an experiment to elucidate structure changes of SISG during the construction process. In addition, the experiment was also designed to show the differences between SISG and the k-nearest neighbor graph.

In order to show the visualization effect of SISG, we compared the visualization effect of SISG-LPP, LPP [20], PCA [5], and NPE [11] on the ORL database [23].

To investigate the influence of parameters on the classification performance of learning algorithms, we designed an experiment to show the sensitivity of LPP to the neighbor parameter k.

In order to test and evaluate the effectiveness of SISG and SISG-LPP, we conducted a series of face recognition experiments on three well-known databases.

4.1. Experiment for the Structure of SISG. In this experiment, we hope to demonstrate the structure changes of SISG during the construction process. By comparing the structure of k-nearest neighbor graph and that of SISG, we will be able to illustrate differences between them. This experiment was conducted on the well-known ORL database [23]. The ORL database contains 400 images from 40 different persons (ten for each person). All images are gray scale and the size of each image is 112 × 92 pixels. Figure 6(a) shows 10 sample face images for one person in ORL.

First, we design a dataset containing ten images selected from the ORL database. Among all these images, six of them belong to the same person (in our dataset, images 2, 3, 4, 6, 7, and 9 belong to the same person) and the rest were selected at random. Then, we visualize the sample similarity matrix (S_{ij}) of SISG, the adjacency matrix (E) of SISG, and the traditional k-nearest neighbor graph for the dataset, as shown in Figures 2, 3(a), and 3(b). Finally, we compare the adjacency matrix of SISG and the k-nearest neighbor graph.

In Figure 2, those numbers without parentheses display the sample similarity between sample pairs. Sample similarity is described by the number of column neighbors between sample pairs. The value of Row 2 and Column 6 of Figure 2 is 16, and the value of Row 6 and Column 2 is 19, which, respectively, means that 16 columns of Image A_6 become column neighbors of Image A_2 and 19 columns of Image A_2 become column neighbors of Image A_6. After calculating the sample similarities, we determine each nearest neighbor according to sample similarity. When determining the nearest neighbors of each sample, we do not consider such sample pairs which are not similar to each other at all. For example, the value of Row 1 and Column 3 of Figure 2 is zero, which means Image A_3 is impossible to become a neighbor of Image A_1.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1)	0	3	0	0	25	0	4	14	8	8
(2)	0	0	12	14	1	16	13	2	13	1
(3)	1	12	0	27	2	2	9	3	2	3
(4)	1	14	27	0	2	6	20	2	2	5
(5)	13	6	0	1	0	7	4	6	15	14
(6)	1	19	6	8	2	0	7	0	19	7
(7)	2	17	11	24	3	6	0	0	3	3
(8)	15	8	8	7	9	7	8	0	0	14
(9)	0	22	3	9	7	22	5	1	0	6
(10)	5	4	0	1	16	8	2	15	9	0

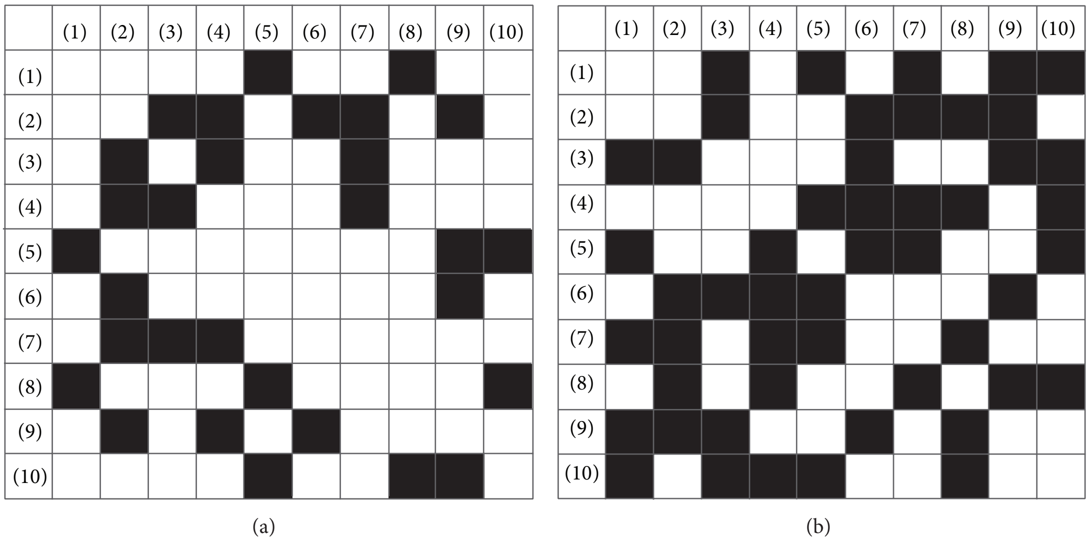
FIGURE 2: The sample similarity matrix of SISG.

Black squares in Figures 3(a) and 3(b) indicate the two samples which are connected by them are neighbors. From Figures 3(a) and 3(b) we can see the following.

(1) The k-nearest neighbor graph is symmetric while the SISG is asymmetric. For example, we can observe from (a) that Image A_9 is a neighbor of Image A_5 but Image A_5 is not a neighbor of Image A_9.

(2) SISG can more accurately reflect the relationship between samples. From (a) we can see that images which became neighbors generally belong to the same person. For example, from the second row of (a) we can see that Images A_3, A_4, A_6, A_7, and A_9 are neighbors of Image A_2, and from the third row of (a) we also can see that Images A_2, A_4, and A_7 are neighbors of A_3. We know that, in our dataset, Images A_2, A_3, A_4, A_6, A_7, and A_9 belong to the same person, so this example shows that the SISG makes similar samples become neighbors.

(3) The k-nearest neighbor graph does not very successfully reflect the relationship between samples. We also take Image A_2 as an example; from the second row of (b) we can observe that Images A_3, A_6, A_7, A_8, and A_9 are neighbors of Image A_2, but in fact, Images A_8 and A_2 do not belong to the same person. From the third row of (b) we can also see that Images A_1, A_2, A_6, A_9, and A_{10} are neighbors of Image A_3, but in fact, Images A_1, A_3, and A_{10} do not belong to the same person.



	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1)					■			■		
(2)			■	■		■	■		■	
(3)		■		■			■			
(4)		■	■				■			
(5)	■								■	■
(6)		■							■	
(7)		■	■	■						
(8)	■				■					■
(9)		■		■		■				
(10)					■			■	■	

(a)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1)			■		■		■		■	■
(2)			■			■	■	■	■	
(3)	■	■				■			■	■
(4)					■	■	■	■		■
(5)	■			■		■	■			■
(6)		■	■	■	■				■	
(7)	■	■		■	■			■		
(8)		■		■			■		■	■
(9)	■	■	■			■		■		
(10)	■		■	■	■			■		

(b)

FIGURE 3: Visualization of the adjacency matrix of SISG (a) and the k-nearest neighbor (k = 5) graph (b).

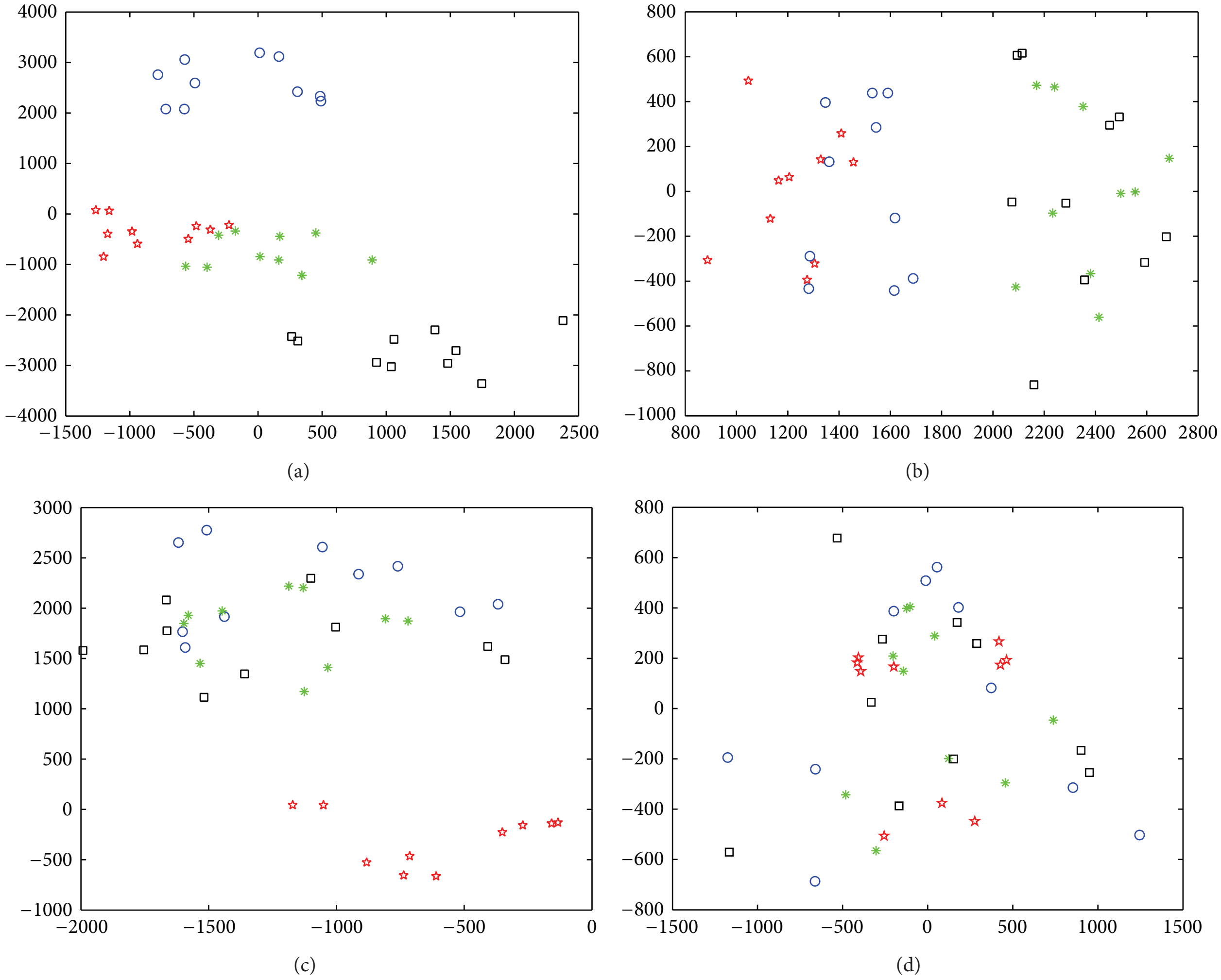


FIGURE 4: The 2-dimensional visualization of the four algorithms on the ORL database. (a) SISG-LPP, (b) LPP, (c) NPE, and (d) PCA.

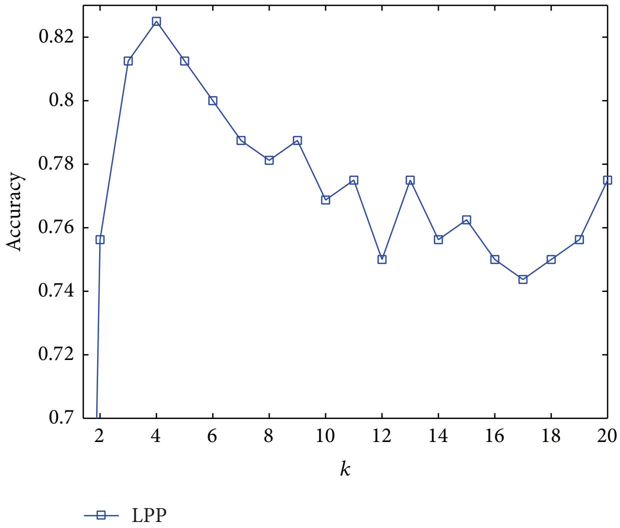
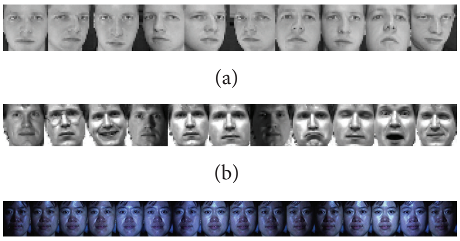


Figure 5: Influence of the parameter k on the classification performance of LPP.



(a)

(b)

(c)

Figure 6: Sample face images from one person. (a) ORL, (b) YALE, and (c) the Illumination subset of PIE.

4.2. Face Manifold Visualization. In this experiment, we compare the visualization effect of SISG-LPP, LPP, PCA, and NPE. We randomly selected 4 people from the ORL database and 10 samples from each person. Then we mapped all these samples to the 2-dimensional subspace using these algorithms. From Figure 4(a) we can see that SISG-LPP more effectively separates the 4-class samples in its 2-dimensional reduction subspace. In contrast, in the subspaces of LPP (Figure 4(b)) and NPE (Figure 4(c)), the samples are not very well separated, and more than half of the samples overlapped. From Figure 4(d) we can see that in the subspace of PCA, the samples are basically entangled together.

4.3. Parameter Sensitivity of LPP. Since SISG does not have the neighbor parameter k, in this experiment, we only investigate the sensitivity of LPP to the neighbor parameter k. During this experiment, the set of images selected from face databases were partitioned into different sample collections. We use G_m/P_n to indicate that for each person in the face database m images were selected at random for training and the remaining n images were employed for testing. We conducted this experiment on G_6/P_4 in the ORL database. From Figure 5 we can see that LPP is very sensitive to the neighbor parameter k. In contrast, SISG-LPP does not have neighbor parameter k, so it is much less sensitive to the parameter than LPP.

4.4. Face Recognition. To evaluate the proposed SISG and SISG-LPP algorithm, we compare the performance of SISG-LPP with LPP on three face databases. Here, we adopt the benchmark face databases ORL [23], YALE [24], and the subset of CMU PIE [25] to conduct experiments on face recognition. There are faces of 15 people in the YALE database, and each person has 11 face images with size 100 × 100. The CMU PIE database contains 41,368 images from 68 people, and the word PIE means Pose, Illumination, and Expression. The size of images in the PIE database is 217×178. In this research, we use the Illumination subset of the CMU PIE database to conduct our experiment; we select 16 different images per person from the Illumination subset. Figures 6(a), 6(b), and 6(c) show part of the face images in ORL, YALE, and the Illumination subset of PIE, respectively.

As described above, we use G_m/P_n to indicate that m images from each person are randomly selected as the training data and the remaining n images are used for testing. For each division with G_m/P_n, 50 random splits are generated and the final performance of the algorithm being tested is obtained by averaging the results of 50 classification accuracy values. The neighbor parameter k for LPP is set to $m - 1$.

Firstly, for each person we select m ($m = 5, 6$) images from the ORL database and YALE database, respectively. Four divisions are considered: G_5/P_5, G_6/P_4 on the ORL database as well as G_5/P_6, G_6/P_5 on the YALE database; 50 random splits are generated and the final results of the four divisions are obtained by taking the mean of the 50 classification accuracy values. The accuracy values versus the numbers of reduced dimensions are shown by Figure 7.

From Figures 7(a) and 7(b) we can see that SISG-LPP outperforms LPP for all divisions, and from Figure 7(b) we can observe that SISG-LPP significantly outperforms LPP when the number of the reduced dimensions is relatively low. From Figures 7(c) and 7(d) we can find that SISG-LPP outperforms LPP for most of the situations.

The mean accuracy values of LPP and SISG-LPP on ORL, YALE, and the Illumination subset of PIE database are listed in Tables 1~3, respectively.

From the results shown in Tables 1~3, one can find the following.

(1) The overall accuracy values of all algorithms are improved at various degrees when the number of training samples is increased.

(2) From the results shown in Tables 1 and 2, one can see that the recognition accuracy values of SISG-LPP are much higher than that of LPP when the number of training samples is relatively small. For instance, the accuracy of SISG-LPP on the division G_3/P_7 of the ORL database is 15.69% higher than that of LPP, while the accuracy of SISG-LPP on the division G_8/P_2 is only 1.33% higher than that of LPP.

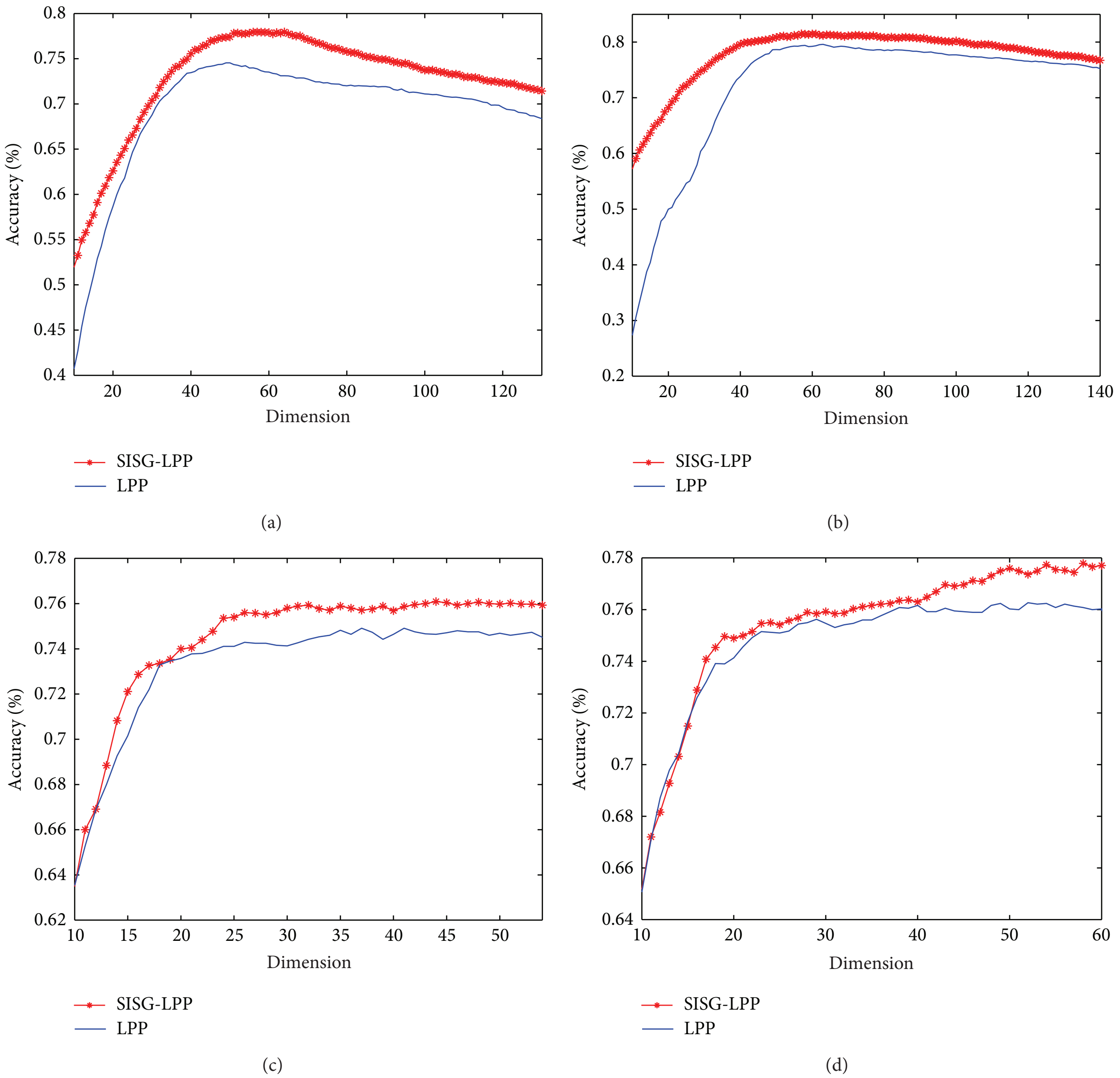


Figure 7: The recognition accuracy values of SISG-LPP and LPP versus the dimensions. (a) G_5/P_5 of the ORL database, (b) G_6/P_4 of the ORL database, (c) G_5/P_6 of the YALE database, and (d) G_6/P_5 of the YALE database.

Table 1: Mean accuracy values of LPP and SISG-LPP on the ORL database. The numbers in parentheses are the corresponding feature dimensions with the best results after dimensionality reduction.

	G_2/P_8	G_3/P_7	G_4/P_6	G_5/P_5	G_6/P_4	G_7/P_3	G_8/P_2
LPP	55.53 (55)	59.41 (95)	76.42 (46)	82.70 (32)	85.63 (59)	88.13 (45)	90.40 (37)
SISG-LPP	65.88 (61)	75.10 (55)	79.88 (48)	83.87 (52)	85.99 (67)	89.70 (63)	91.73 (57)

Table 2: Mean accuracy values of LPP and SISG-LPP on the YALE database. The numbers in parentheses are the corresponding feature dimensions with the best results after dimensionality reduction.

	G_2/P_9	G_3/P_8	G_4/P_7	G_5/P_6	G_6/P_5	G_7/P_4	G_8/P_3
LPP	61.36 (25)	67.42 (32)	74.67 (37)	77.20 (34)	78.51 (17)	79.37 (46)	81.33 (69)
SISG-LPP	66.93 (25)	71.63 (35)	75.43 (45)	78.36 (26)	79.33 (47)	80.20 (55)	82.40 (61)

Table 3: Mean accuracy values of LPP and SISG-LPP on the Illumination subset of PIE database. The numbers in parentheses are the corresponding feature dimensions with the best results after dimensionality reduction.

	G_2/P_{14}	G_3/P_{13}	G_4/P_{12}	G_5/P_{11}	G_6/P_{10}	G_7/P_9	G_8/P_8
LPP	81.88 (88)	91.91 (110)	95.88 (134)	98.31 (155)	99.17 (169)	99.58 (184)	99.90 (195)
SISG-LPP	80.75 (90)	92.98 (122)	96.32 (145)	98.55 (154)	99.33 (168)	99.62 (182)	99.94 (199)

(3) SISG-LPP outperforms LPP in all divisions with G_m/P_n on the ORL and YALE databases. SISG-LPP outperforms LPP for most of the divisions with G_m/P_n on the Illumination subset of PIE database.

5. Conclusions

In this paper, we present a new graph construction method, and we name the graph constructed by this method as samples' inner structure based graph (SISG). Unlike the traditional graph construction method, SISG avoids predefining neighbor parameter k (or ε). Moreover, SISG can also well preserve intrinsic features of samples by using samples' inner structure information to construct graph. Both the weighted adjacency matrix and the adjacency matrix of SISG are generally asymmetric, which may be more reasonable for capturing the relationships among samples. For the sake of proving that the construction method of SISG is very general, we incorporated it into a state-of-the-art DR algorithm, locality preserving projection (LPP), and thus developed a novel DR algorithm SISG-LPP. Finally, several experiments are conducted on three well-known face databases. Experimental results verified the effectiveness and feasibility of the SISG and SISG-LPP algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by (1) the Ph.D. Programs Foundation of Ministry of Education of China under Grant (no. 20120061110045) and (2) the Natural Science Foundation of Jilin Province of China under Grant (no. 201115022).

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] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacian faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[4] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholköpf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, vol. 16, pp. 321–328, The MIT Press, 2004.

[5] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 1989.

[6] M. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, June 1991.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[8] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional datawith application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[9] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[10] J. B. Tenenbaum, "Mapping a manifold of pe rceptual observations," *Advances in Neural Information Processing Systems*, pp. 682–688, 1998.

[11] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 1208–1213, Beijing, China, October 2005.

[12] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[13] M. Maier, U. von Luxburg, and M. Hein, "Influence of graph construction on graph-based clustering measures," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 1025–1032, December 2008.

[14] T. Jebara, J. Wang, and S. F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pp. 441–448, ACM, Montreal, Canada, June 2009.

[15] S. I. Daitch, J. A. Kelner, and D. A. Spielman, "Fitting a graph to vector data," in *Proceedings of the 26th International Conference on Machine Learning (ACM '09)*, pp. 201–208, Montreal, Canada, June 2009.

[16] M. Hein, J. Audibert, and U. von Luxburg, "Graph laplacians and their convergence on random neighborhood graphs," *Journal of Machine Learning Research*, vol. 8, pp. 1325–1368, 2007.

[17] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph laplacians for semi-supervised learning," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS '05)*, pp. 67–74, December 2005.

[18] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[19] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of the*

Neural Information Processing Systems (NIPS '01), vol. 14, pp. 585–591, 2001.

[20] X. F. He and P. Niyogi, "Locality preserving projections," in *Proceedings of the Neural Information Processing Systems (NIPS '04)*, vol. 6, pp. 1059–1071, 2004.

[21] F. Chung, *Spectral Graph Theory*, Cbms Regional Conference Series in Mathematics, no. 92, American Mathematical Society, 1997.

[22] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[23] ORL database: Cambridge University Computer Laboratory, 2002, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[24] D. Cai, X. He, and J. Han, "SRDA: an efficient algorithm for large scale discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 1–12, 2007.

[25] The PIE database, http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261.