Symmetric low-rank preserving projections for subspace learning☆

Jie Chen, Hua Mao*, Haixian Zhang, Zhang Yi

Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, PR China

Keywords:
Low-rank representation
Manifold regularization
Sparsity regularization
Subspace learning
Dimensionality reduction

A B S T R A C T

Graph construction plays an important role in graph-oriented subspace learning. However, most existing approaches cannot simultaneously consider the global and local structures of high-dimensional data. In order to solve this deficiency, we propose a symmetric low-rank preserving projection (SLPP) framework incorporating a symmetric constraint and a local regularization into low-rank representation learning for subspace learning. Under this framework, SLPP-M is incorporated with manifold regularization as its local regularization while SLPP-S uses sparsity regularization. Besides characterizing the global structure of high-dimensional data by a symmetric low-rank representation, both SLPP-M and SLPP-S effectively exploit the local manifold and geometric structure by incorporating manifold and sparsity regularization, respectively. The similarity matrix is successfully learned by solving the nuclear-norm minimization optimization problem. Combined with graph embedding techniques, a transformation matrix effectively preserves the low-dimensional structure features of high-dimensional data. In order to facilitate classification by exploiting available labels of training samples, we also develop a supervised version of SLPP-M and SLPP-S under the SLPP framework, named S-SLPP-M and S-SLPP-S, respectively. Experimental results in face, handwriting and object recognition applications demonstrate the efficiency of the proposed algorithm for subspace learning.

1. Introduction

In many areas of machine learning, a vast amount of valuable high-dimensional data is available, such as images, videos, text and documents, as observations are usually embedded in a high-dimensional space, which is commonly referred to as the curse of dimensionality [1]. This imposes a great burden on the analysis of high-dimensional data, e.g., classification and clustering tasks [2–8]. Simultaneously, the high dimensionality of data increases the computational complexity of algorithms owing to the effect of noise (e.g., occlusions, and illumination variations) and the ambient space dimension of the training samples. As shown in Fig. 1, a number of data points are approximately drawn from a union of three subspaces, which usually refers to the ambient space. This has naturally led to the challenging problem of high-dimensional data analysis, with the aim of exploiting and capturing the underlying structure of the high-dimensional data.

In the literature, there is ample evidence showing that high-dimensional data often exhibit low-dimensional structures in the ambient space [9,10]. There has also been substantially increased interest in subspace learning, primarily motivated by the development of a number of linear and nonlinear dimensionality reduction techniques [11–15]. For efficient analysis of high-dimensional data, these techniques project the high-dimensional data into low-dimensional spaces while preserving the intrinsic features of the data. Such techniques can be employed in feature extraction, data visualization and compression, and as an effective preprocessing step in many important machine learning methods [16–19].

Subspace learning has been studied extensively for several decades [20,21]. Generally, these methods can be divided into two categories, linear and nonlinear methods, according to different assumptions regarding the underlying structure of the high-dimensional data. As linear subspace learning algorithms, principal component analysis (PCA) [11], which maximizes the sample variance to preserve the global Euclidean structure, linear discriminant analysis (LDA) [12], which maximizes the ratio of the inter- and intra-class scatters to find projection directions, locality preserving projections (LPP) [22], and their numerous variants have shown their capabilities in many application domains [23–26]. Although these methods work well if the errors follow a Gaussian distribution, they ignore the local manifold structure of data. By linearly approximating the eigenfunctions of the Laplace Beltrami operator on the manifold, LPP preserves its local relationships and uncovers the essential manifold structure of the data. By combin-

☆ This work was supported by the National Science Foundation of China (Grant Nos. 61602329, 61502322, 61432012 and U1435213) and Sichuan Science and Technology Program (Grant No. 2017JY0258).

* Corresponding author.
E-mail address: huamao@scu.edu.cn (H. Mao).

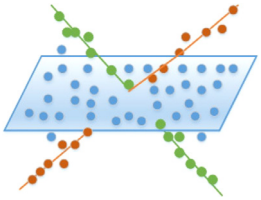

Fig. 1. Three subspaces in $\mathbf{R}^3$ with a number of data points approximately in each subspace.

ing with kernel-based techniques, nonlinear mapping can be implemented, such as kernel PCA and kernel LDA [27,28]. However, most kernel-mapping functions do not explicitly consider the underlying manifold structure.

A variety of learning algorithms have been developed to discover the underlying nonlinear structure, i.e., embedded manifold of the data, such as isometric feature mapping (ISOMAP) [29], locally linear embedding (LLE) [30], Laplacian Eigenmap (LE) [31], and neighborhood preserving embedding (NPE) [14]. These techniques usually first construct a similarity matrix, where each element measures the similarity of a pair of data points. There are several choices for the similarity strategy representing the manifold geometry approximately, such as the local neighborhood relationship in LLE, Gaussian similarity from Euclidean distance in LE, k nearest neighbors in NPE, and the geodetic distance between any pair of data points in ISOMAP. These methods typically employ different local similarity strategies to preserve local neighborhood information. However, estimating the correct size of the neighborhood, which is closely related to the intrinsic dimension of the manifold, remains an open question.

Yan et al. [32] proposed the graph embedding framework for dimensionality reduction, under which most existing graph-oriented subspace learning methods can be unified. This general framework emphasizes the importance of constructing a similarity matrix, and presents a novel formulation of graph embedding. Following the idea of this framework, some studies on sparsity and rank minimization theory have recently been proposed to study subspace learning [33–36]. Cheng et al. [33] proposed an l_1-graph learning method, which uses the sparsest representation of the data points solved by l_1-norm minimization techniques, to define a similarity matrix combined with the embedding program of NPE for subspace learning. However, the lack of having a global structural constraint limits its availability in practice. Rank minimization techniques have been proposed to alleviate these problems [37–40]. The fact that the regularization terms of l_1-norm and l_*-norm often needs hundreds of iterations before convergence may lead to computationally impracticable for the computational complexity of these methods. Fortunately, a great quantity of nonconvex optimization techniques have been proposed to alleviate this problem, e.g., nonconvex low-rank regularizers [38,40]. Low-rank representation (LRR)-based algorithms can capture globally linear structures of data by solving the convex or nonconvex optimization problem of nuclear-norm minimization instead of rank minimization. However, LRR ignores the local structure of high-dimensional data. Moreover, the lack of a symmetric constraint on the low-rank representation means that its ability to characterize the relationship of a pair of data points is often limited in practical applications. In addition, most existing LRR-oriented algorithms do not consider taking advantage of the class label information of training samples, which often provides discriminative information that facilitates the subsequent classification task.

Motivated by recent advancements in low-rank representation and manifold learning, in this paper we propose a symmetric low-rank preserving projection (SLPP) framework, which incorporates a symmetric constraint and a local regularization into low-rank representation for subspace learning. Specifically, SLPP first derives the symmetric low-rank representation coefficients of high-dimensional data, i.e., a similarity matrix, which can be efficiently calculated by solving the nuclear-norm minimization optimization problem. Then, the similarity matrix is used to construct an affinity graph. Using the graph embedding framework, the affinity graph is combined with the notion of NPE to search for a transformation matrix for dimensionality reduction. Under this framework, SLPP-M is incorporated with manifold regularization as its local regularization while SLPP-S uses sparsity regularization. Both SLPP-M and SLPP-S simultaneously consider the global structural constraint on the low-rank representation as well as the local structure of high-dimensional data on the local regularization. The incorporation of the symmetric low-rank representation with manifold or sparsity regularization enriches the relationship of high-dimensional data for robust subspace learning. In addition, with training sample label information available, we further develop a supervised version of SLPP-M and SLPP-S under the SLPP framework, named S-SLPP-M and S-SLPP-S, respectively. In contrast to most existing subspace learning algorithms, our proposed method simultaneously considers two cases of labeled and unlabeled training samples for subspace learning.

The proposed framework has the following advantages:

(1) It successfully learns a symmetric similarity matrix. The symmetric similarity matrix characterizes the global structure of high-dimensional data by inheriting the advantages of the symmetric low-rank representation. Besides, it also effectively reveals/uncovers the local intrinsic structure of high-dimensional data using manifold or sparsity regularization.
(2) Combining the graph embedding framework with the similarity matrix, it obtains a transformation matrix, which can be employed to effectively preserve the low-dimensional structure features of high-dimensional data for subspace learning.
(3) Using the symmetric low-rank representation model, it exhibits general learning ability of subspace learning in a supervised and an unsupervised manner.
(4) Compared with several popular dimensionality reduction methods, our experimental results on benchmark databases demonstrate that the proposed method realizes competitive performance, especially when a number of training sample labels are available.

The rest of the paper is organized as follows. In Section 2, we provide a brief overview of related work on subspace learning and rank minimization. Section 3 provides a detailed description of the proposed SLPP framework for subspace learning. The experimental results on benchmark databases are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related work

Here we briefly review some work closely related to the proposed method for the sake of clarity. First, we provide a review of the graph embedding framework for subspace learning in Section 2.1. Then, some work on low-rank representation techniques is reviewed in Section 2.2.

2.1. Graph embedding framework

Consider the data matrix, $X = [x_1, x_2, \ldots, x_n] \in \mathbf{R}^{m\times n}$, each column of which represents a training sample. For supervised learning problems, these training samples are assumed to belong to N classes. Without loss of generality, we assume a class label for each

sample, $l_i \in \{1, 2, \ldots, N\}$. The general purpose of subspace learning is to learn a transformation matrix, $P \in \mathbf{R}^{m \times d}$, which transforms the samples from the original high-dimensional space to a low-dimensional subspace. Given a test sample, $x \in \mathbf{R}^{m \times n}$, the desired low-dimensional representation can be obtained as

$$y = P^T x \in \mathbf{R}^d,$$

where $d \ll m$.

Let $G = \{V, E\}$ be an undirected weighted graph with the corresponding adjacency matrix W, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices, $E = \{e_{ij} | i, j \in V\}$ is the set of edges, and $W = \{w_{ij} | i, j \in V\}$ measures the similarity of a pair of vertices, i.e., vertices i and j. The adjacency matrix W can be constructed using a variety of criteria, such as Euclidean distance or local neighborhood relationship. The Laplacian matrix L is defined as

$$L = D - W, \tag{1}$$

where $D = diag(d_1, d_2, \ldots, d_n)$ is a diagonal matrix with its diagonal element defined as $d_i = \sum_{j=1}^{n} w_{ij}$. Moreover, a transformation matrix P can be obtained by solving the following minimization problem [22]:

$$\arg\min_P tr(P^T X L X^T P) \qquad s.t. \qquad P^T X D X^T P = I. \tag{2}$$

Specifically, the transformation matrix P can be given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$XLX^T P = \lambda XDX^T P, \tag{3}$$

where P consists of the r projection vectors corresponding to the r smallest eigenvalues, i.e., $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_r$.

According to graph embedding theory, a number of dimensionality reduction algorithms can be reformulated within the graph embedding framework [32]. First, we give a brief review of the LLE algorithm [30]. LLE constructs locally linear structures at each point x_i by reconstructing x_i only from its neighbors. The optimal combination weights are calculated by solving the constrained least squares problem

$$\min \left\| x_i - \sum_{j \in N_i} w_{ji} x_j \right\| \qquad s.t. \qquad \sum_{j \in N_i} w_{ji} = 1 \tag{4}$$

where N_i is an indices set composed of its selected neighbors. Then LLE computes the best low-dimensional embedding P based on the weight matrix W by minimizing the following cost function:

$$\min_{P=[p_1, p_2, \ldots, p_N]} \left\| p_i - \sum_{j \in N_i} w_{ji} p_j \right\|^2 \qquad s.t. \qquad PP^T = I. \tag{5}$$

In the following, we provide a review of NPE, which is closely related to the proposed method for subspace learning. NPE is a linear extension of the LLE algorithm, which also preserves the local neighborhood structure on the data manifold. The reconstruction errors in NPE are measured by minimizing the following cost function:

$$\phi(W) = \sum_i \left\| P^T x_i - \sum_j W_{ij} P^T x_j \right\|^2, \tag{6}$$

where P is a transformation matrix. By certain algebraic formulations, the objective function of NPE can be formulated as

$$\min_P \left\| P^T X - P^T X W \right\|_F^2 \qquad s.t. \qquad P^T X X^T P = I. \tag{7}$$

This optimization problem can be solved with a generalized eigenvalue decomposition approach,

$$XMX^T p = \lambda XX^T p, \tag{8}$$

where $M = (I - W)^T (I - W)$ and is an eigenvector. The transformation matrix $P = [p_0, p_1, \ldots, p_{d-1}]$ is composed of the d eigenvectors corresponding to the d smallest negative eigenvalues, i.e., $\lambda_0 \le \lambda_1 \le \cdots \le \lambda_{d-1}$.

2.2. Low-rank representation techniques

The low-rank representation is one of the most effective techniques to measure the relationship of high-dimensional data [37]. LRR seeks the lowest-rank solution by solving the following regularized rank minimization problem:

$$\min_{Z,E} rank(Z) + \lambda \|E\|_l \qquad s.t. \qquad X = AZ + E, \tag{9}$$

where $A = [a_1, a_2, \ldots, a_n] \in \mathbf{R}^{d \times n}$ is a basis, $\lambda > 0$ is a parameter to balance the effects of the low-rank term and error term, and $\|\cdot\|_l$ indicates a certain regularization strategy for characterizing various corruptions. As is known, Problem (9) is non-convex. By virtue of the nuclear norm as a common surrogate for the rank function under certain conditions, Problem (9) can be rewritten as the following convex optimization:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_l \qquad s.t. \qquad X = AZ + E, \tag{10}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix, i.e., the sum of its singular values. The optimal solution Z^* to Problem (10) can be obtained by the inexact augmented Lagrange multipliers (ALM) method. Then, we further construct the affinity matrix $|Z^*| + |Z^*|^T$, employed in the graph embedding framework, for subspace learning.

To avoid the post-processing symmetrization step, LRRSC imposes a symmetric constraint, which guarantees weight consistency of each pair of data points, on the representation coefficients [41]. The optimization problem of LRRSC is formulated as

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_l \quad s.t. \quad X = XZ + E, Z = Z^T, \tag{11}$$

where the original data X is considered as the dictionary. The above optimization problem is efficiently solved in [41]. The optimal solution Z^* to Problem (11) leads to highly correlated data points of subspaces being represented together.

3. Symmetric low-rank preserving projections

In this section, we propose a symmetric low-rank preserving projection (SLPP) framework for subspace learning. This framework adopts a new similarity criterion that characterizes the adjacency relationship of point pairs. In particular, it incorporates a symmetric constraint and a local regularization into low-rank representation learning to learn a symmetric similarity matrix. Under this framework, SLPP-M is incorporated with manifold regularization as its local regularization and SLPP-S uses sparsity regularization. Then SLPP uses the similarity matrix to effectively obtain a transformation matrix for dimension reduction of high-dimensional data under the graph embedding framework. Moreover, we take full advantage of training sample labels to develop a supervised version of SLPP-M and SLPP-S under the SLPP framework, named S-SLPP-M and S-SLPP-S, respectively.

3.1. Symmetric low-rank representation with manifold regularization

For a given data matrix X, each column can be represented by the linear combination of the basis, where each column is a sample. The representation coefficients of the data matrix X, which that is regarded as a similarity matrix, can be employed to measure the relationship between each pair of samples. We consider a general model of data representation:

$$\min f(Z) \quad s.t. \quad X = AZ + E, \tag{12}$$

where $f(Z)$ is a matrix function (e.g., $\|Z\|_0$ or $\|Z\|_*$, etc), and E is an error term. The optimal solution Z^* is a special representation of data X, which is closely related to the matrix function, with respect to basis X. We denote the data matrix $A = \{a_1, a_2, \ldots, a_n\} \in \mathbf{R}^{m\times n}$, each column of which represents a corrected sample, i.e., $a_i = Xz_i$.

The fact that the local manifold structure of high-dimensional data can be effectively modeled using a nearest neighbor graph, has been demonstrated in recent studies of spectral graph and manifold learning theory [42–44]. If any two data points a_i and a_j recovered from x_i and x_j, respectively, are close in the intrinsic geometry of the data distribution, then the representations of these two points, namely, z_i and z_j with respect to the same basis X, are close to each other. For larger similarity between a_i and a_j, the distance between z_i and z_j should be smaller to minimize the following objective function:

$$\begin{aligned}\Re(Z) &= \frac{1}{2}\sum_{i,j=1}^{n}\left\|z_i - z_j\right\|^2 W_{ij}\\ &= \sum_{i=1}^{n} z_i^T z_i D_{ii} - \sum_{i,j=1}^{n} z_i^T z_j W_{ij}\\ &= tr\left(Z^T DZ\right) - tr\left(Z^T WZ\right)\\ &= tr\left(Z^T LZ\right),\end{aligned} \tag{13}$$

where $tr(\cdot)$ is the trace operator, and W can be constructed using the Euclidean distance of data matrix X. This is called the manifold regularization term, which is expected to maintain the local manifold structure of the data.

High-dimensional data often lie close to a low-dimensional subspace of the ambient space. Although LRR exhibits the excellent ability of capturing the global structure of high-dimensional data, it does not consider the local manifold structure of high-dimensional data. To do so, we consider the following convex optimization problem seeking a symmetric low-rank representation Z:

$$\begin{aligned}\min_{Z}\quad & \|Z\|_* + \lambda\|X - AZ\|_l + \frac{\beta}{2}tr\left(Z^T LZ\right)\\ s.t.\quad & Z = Z^T,\end{aligned} \tag{14}$$

where β and λ are used to balance the effects of the low-rank representation item, the error item, and the manifold regularization item. Inclusion of the symmetric low-rank representation with manifold regularization enriches the relationship of high-dimensional data by simultaneously considering their global geometric and local manifold structure.

To obtain the graph Laplacian L, we must first calculate a weight matrix W, where the definition of L is given in Section 2.1. In general, weight matrix W can be obtained using Gaussian similarity from the Euclidean distance of the original data X. However, the observations are often corrupted by noise. Hence, various recovery and completion techniques are adopted for different types of noise. For example, we can use an alternative low-rank matrix, recovered from the original data X using LRR [37], instead of the original data X, to calculate weight matrix W.

3.2. Symmetric low-rank representation with sparsity regularization

Sparse representation has been recognized as one of extremely successful techniques for representation of data [33]. Each sample is represented as a linear combination of a small number of other samples in sparse representation. Intuitively, the sparsity of the weight matrix can be measured by the l_0-norm. Because l_0-norm minimization is an NP-hard problem, the l_1-norm convex optimization provides itself as a surrogate for l_0-norm minimization. In fact, the equivalence of l_0-norm and l_1-norm minimizations can be proved under certain conditions [45]. Hence, l_1-norm minimization is widely employed in sparse representation, which is an important way to improve the generalization capability of the data representation to design the weight matrix straightforwardly. Sparse representation explicitly captures local geometrical structure of high-dimensional data with a very small number of non-zero elements. To characterize the local geometrical structure of the data, we replace the manifold regularization of Problem (14) using a sparse penalty on symmetric low-rank representation as follows.

$$\begin{aligned}\min_{Z}\quad & \|Z\|_* + \gamma\|Z\|_1 + \lambda\|X - AZ\|_l\\ s.t.\quad & Z = Z^T,\end{aligned} \tag{15}$$

where $\lambda > 0$ is a sparsity parameter.

By enhancing sparsity of the symmetric low-rank representation, the sparsity of the weight matrix Z can naturally preserve the local geometric structure of the data. Thus, the global and local structures of high-dimensional data can be characterized in weight matrix by incorporating a symmetric constraint and sparsity regularization into low-rank representation learning.

3.3. Optimization procedure based on the augmented Lagrange multipliers method

We first apply the ALM method to Problem (14) [46]. To facilitate the optimization, we first convert Problem (14) into the following equivalent problem by introducing an auxiliary variable J:

$$\begin{aligned}\min_{Z,E,J}\quad & \|J\|_* + \lambda\|X - AZ\|_l + \frac{\beta}{2}tr\left(Z^T LZ\right)\\ s.t.\quad & Z = J, J = J^T.\end{aligned} \tag{16}$$

The above optimization problem can be rewritten using (12) as follows:

$$\begin{aligned}\min_{Z,E,J}\quad & \|J\|_* + \lambda\|E\|_l + \frac{\beta}{2}tr\left(Z^T LZ\right)\\ s.t.\quad & X = AZ + E, Z = J, J = J^T.\end{aligned} \tag{17}$$

Then, we get the corresponding augmented Lagrangian function:

$$\begin{aligned}\min_{Z,E,J=J^T,Y_1,Y_2}\quad & \|J\|_* + \lambda\|E\|_l + \frac{\beta}{2}tr\left(Z^T LZ\right)\\ & + tr[Y_1^T(X - AZ - E)] + tr[Y_2^T(Z - J)]\\ & + \frac{\mu}{2}\left(\|X - AZ - E\|_F^2 + \|Z - J\|_F^2\right),\end{aligned} \tag{18}$$

where Y_1 and Y_2 are Lagrange multipliers, μ is a positive penalty parameter, and $\|\cdot\|_F$ is the Frobenius norm. Using linear algebra techniques, the above optimization problem can be rewritten as:

$$\begin{aligned}\min_{Z,E,J=J^T,Y_1,Y_2}\quad & \|J\|_* + \lambda\|E\|_l + \frac{\beta}{2}tr\left(Z^T LZ\right)\\ & + \frac{\mu}{2}\left(\left\|X - AZ - E + \frac{Y_1}{\mu}\right\|_F^2 + \left\|Z - J + \frac{Y_2}{\mu}\right\|_F^2\right).\end{aligned} \tag{19}$$

Problem (19) can be minimized separately with respect to variables J, Z and E. In particular, variables J, Z and E can be updated alternately by fixing the other two variables, and then, the Lagrange multipliers Y_1 and Y_2 are updated separately. Because each variable in Problem (19) can be solved iteratively, the updating schemes at the $(k+1)$th iteration are:

$$J_{k+1} = \arg\min_{J=J^T}\frac{1}{\mu}\|J\|_* + \frac{1}{2}\left\|J - \left(Z + \frac{Y_2}{\mu}\right)\right\|_F^2 \tag{20}$$

$$Z_{k+1} = \left(I + A^TA + \frac{\beta}{\mu}L\right)^{-1}\left(A^TX - A^TE + J + \frac{A^TY_1 - Y_2}{\mu}\right) \tag{21}$$

$$E_{k+1} = \arg\min \lambda\|E\|_l + \frac{\mu}{2}\left\|E - \left(X - AZ + \frac{Y_1}{\mu}\right)\right\|_F^2 \tag{22}$$

Note that Problems (20) and (22) have closed-form solutions. Problem (20) can be solved using the following lemma. The choice of $\|\cdot\|_l$ is related to types of corruptions. For example, $\|\cdot\|_{2,1}$ is adopted to characterize the sample-specific corruptions. Besides, $\|\cdot\|_1$ is an appropriate choice for random corruptions, and $\|\cdot\|_F^2$ is chosen for small Gaussian noise. The complete optimization for solving Problem (14) is summarized in Algorithm 1.

Algorithm 1 Solving Problem (14) by inexact ALM.

Input:

data matrix X, parameters $\lambda > 0$, $\beta > 0$.

Initialize: $Z = J = 0, E = 0, Y_1 = Y_2 = 0, \mu = 10^{-2}, \mu_{\max} = 10^{10}, \rho = 1.1, \varepsilon = 10^{-6}$

1: **while** not converged **do**

2: Update J with the other two variables Z and E fixed:
$J = \arg\min_{J=J^T} \frac{1}{\mu}\|J\|_* + \frac{1}{2}\left\|J - \left(Z + \frac{Y_2}{\mu}\right)\right\|_F^2$.

3: Update Z with the other two variables J and E fixed:
$Z = \left(I + X^TX + \frac{\beta}{\mu}L\right)^{-1}\left(X^TX - X^TE + J + \frac{X^TY_1 - Y_2}{\mu}\right)$.

4: Update E with the other two variables J and Z fixed:
$E = \arg\min \lambda\|E\|_{2,1} + \frac{\mu}{2}\left\|E - \left(X - XZ + \frac{Y_1}{\mu}\right)\right\|_F^2$.

5: Update the multipliers:
$$Y_1 = Y_1 + X - XZ - E;$$
$$Y_2 = Y_2 + X - \mu(Z - J).$$

6: Update parameter μ by $\mu = \min(\rho\mu, \mu_{\max})$;

7: Check the convergence conditions
$$\|X - AZ - E\|_\infty < \varepsilon \text{ and } \|Z - J\|_\infty < \varepsilon.$$

8: **end while**

Output:

Z^*, E^*

Lemma 1 (Lemma 1 [41]). *For any given square matrix $Q \in \mathbf{R}^{n\times n}$, the minimizer of the following optimization problem is unique.*

$$W^* = \arg\min_W \frac{1}{\mu}\|W\|_* + \frac{1}{2}\|W - Q\|_F^2, W = W^T. \tag{23}$$

The minimizer has the following closed form

$$W^* = U_r\left(\Sigma_r - \frac{1}{\mu}\cdot \mathrm{I}_r\right)V_r^T, \tag{24}$$

where $\widetilde{Q} = U\Sigma V^T$ is the singular value decomposition (SVD) of the symmetric matrix $\widetilde{Q} = (Q + Q^T)/2$, $\Sigma_r = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$ where $\{r : \sigma_r > \frac{1}{\mu}\}$ are positive singular values, U_r and V_r are the corresponding singular vectors of matrix $\widetilde{Q}$, and I_r is an $r \times r$ identity matrix.

Then we adopt ALM method to Problem (15) [46]. We first introduce an auxiliary variable J to separate the variable in Problem (15).

$$\begin{aligned} &\min_{Z,E,J} \quad \|Z\|_* + \gamma\|J\|_1 + \lambda\|E\|_l \\ &s.t. \quad Z = J, J = J^T. \end{aligned} \tag{25}$$

The augmented Lagrange function of Problem (25) is

$$\begin{aligned} &L(Z, J, E, Y_1, Y_2, \mu) \\ &= \|Z\|_* + \gamma\|J\|_1 + \lambda\|E\|_l + tr[Y_1^T(X - AZ - E)] \\ &\quad + tr[Y_2^T(Z - J)] + \frac{\mu}{2}\left(\|X - AZ - E\|_F^2 + \|Z - J\|_F^2\right) \\ &= \|Z\|_* + \lambda\|E\|_l + \gamma\|J\|_1 \\ &\quad + \frac{\mu}{2}\left(\left\|X - AZ - E + \frac{Y_1}{\mu}\right\|_F^2 + \left\|Z - J + \frac{Y_2}{\mu}\right\|_F^2\right). \end{aligned} \tag{26}$$

The updating schemes at the $(k+1)$th iteration in Problem (19) are:

$$\begin{aligned} Z_{k+1} &= \arg\min_{Z=Z^T} \|Z\|_* + \langle \nabla f(Z_k),\ Z - Z_k\rangle + \frac{\eta}{2}\|Z - Z_k\|_F^2 \\ &= \arg\min_{Z=Z^T} \|Z\|_* + \frac{\mu\eta}{2}\left\|Z - Z_k + \frac{1}{\eta}\right. \\ &\quad \left.\times\left[-A^T\left(X - AZ_k - E + \frac{Y_1}{\mu}\right) + \left(Z_k - J + \frac{Y_2}{\mu}\right)\right]\right\|_F^2, \end{aligned} \tag{27}$$

$$J_{k+1} = \arg\min \gamma\|J\|_1 + \frac{\mu}{2}\left\|J - \left(Z + \frac{Y_2}{\mu}\right)\right\|_F^2, \tag{28}$$

$$E_{k+1} = \arg\min \lambda\|E\|_l + \tfrac{\mu}{2}\left\|E - \left(X - AZ + \tfrac{Y_1}{\mu}\right)\right\|_F^2, \tag{29}$$

where $f(Z) = \frac{\mu}{2}(\|X - AZ - E + \frac{Y_1}{\mu}\|_F^2 + \|Z - J + \frac{Y_2}{\mu}\|_F^2)$ and ∇f is the partial differential of f with respect to Z. The complete optimization for solving (15) is similar to Problem (14) in Algorithm 1.

3.4. Construction of a graph for subspace learning

The general purpose of subspace learning is to learn a transformation matrix from the original high-dimensional data. By transforming the original high-dimensional data into the low-dimension subspace, the transformation matrix can be used to exploit the intrinsic low-dimensional structure of high-dimensional data. The similarity between the vertex pairs is measured by a graph that characterizes some of the geometric structures of the high-dimensional data. Therefore, the procedure for graph construction has a great impact on the potential of the graph-oriented subspace learning algorithms.

For SLPP-M and SLPP-S, the procedure for graph construction is very important. Each column of the symmetric low-rank representation matrix Z^* characterizes how the other samples contribute to the reconstruction of a corresponding sample. In other words, each element z_{ij} of matrix Z^* measures the relationship between samples i and j. We consider an affinity graph, $G = (V, E)$, where $V = \{v_1, v_2 \ldots, v_n\}$ is the set of vertices and $E = \{e_{ij} | i, j \in V\}$ is the set of edges. Edge e_{ij} connects vertices i and j. We adopt z_{ij} to represent the weight of edge e_{ij}. By solving Problem (8), we can use the affinity graph G to obtain a transformation matrix P. The complete procedure for SLPP-M is summarized in Algorithm 2. The complete procedure for SLPP-M is similar to Algorithm 2 except the first step of solving the optimization problem. A corresponding low-dimensional representation of samples X can be transformed by

$$Y = P^TX, \tag{30}$$

using transformation matrix P. Finally, the subsequent classification task can be performed using the low-dimensional representation Y with reduced computational cost.

With class label information of samples available, SLPP-M and SLPP-S can easily be extended to two supervised versions, i.e., S-SLPP-M and S-SLPP-S, respectively. We take into account the class label information to adjust the weight between pairs of samples. In particular, S-SLPP-M and S-SLPP-S encourage the coefficients of intra-class samples to be highly correlated, while making the coefficients of inter-class samples as independent as possible. To implement the intra-class attraction and inter-class repulsion of samples

Algorithm 2 SLPP-M algorithm.

Input:

data matrix X, parameters $\lambda > 0$, $\beta > 0$.

1: Solve the following problem by Algorithm 1:

$$\min_{Z,E} \|Z\|_* + \lambda\|E\|_{2,1} + \frac{\beta}{2}tr\left(Z^T L Z\right)$$
$$s.t. \quad X = XZ + E, Z = Z^T,$$

and obtain the optimal solution (Z^*, E^*).

2: Construct the weight matrix W of an affinity graph G using Z^*.

3: Solve the following generalized eigenvalue decomposition problem:

$$XMX^T p = \lambda XX^T p,$$

where $M = (I - W)^T((I - W)$ and p is an eigenvector. The transformation matrix $P = [p_0, p_1, \ldots, p_{d-1}]$ is composed of the d eigenvectors corresponding to the d smallest negative eigenvalues, i.e., $\lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{d-1}$.

Output:

Matrix P

well, the weight matrix W of the affinity graph G is constructed as follows:

$$W = (1-\mu)\begin{bmatrix} 0 & z_{12} & \ldots & z_{1k} \\ z_{21} & 0 & \ldots & z_{2k} \\ \ldots & \ldots & \ldots & \ldots \\ z_{k1} & z_{k2} & \ldots & 0 \end{bmatrix} + (1+\mu)\begin{bmatrix} z_{11} & 0 & \ldots & 0 \\ 0 & z_{22} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & z_{kk} \end{bmatrix},$$

where parameter μ is used to balance the effects of the intra-class compactness and inter-class scantiness in the weight matrix. Using weight matrix W, S-SLPP-M and S-SLPP-S can obtain a transformation matrix by solving Problem (8). If we set parameter $\mu = 0$, SLPP-M and SLPP-S can be regarded as special cases of S-SLPP-M and S-SLPP-S, respectively.

3.5. Computational complexity analysis

We assume that matrix X of size $m \times n$ consists of n samples, where each column of the matrix is an m-dimensional sample. The computational complexity of the first step in Algorithm 1 is $O(n^3)$ because it involves calculating the SVD of an $n \times n$ matrix. The computational complexities of the second and third steps are $O(n^3)$ and $O(mn^2)$, respectively. If we consider $n > m$, the computational complexity of Algorithm 1 for each iteration can be considered to be $O(n^3)$. The computational complexity of Algorithm 2 is $O(tn^3) + O(n^3)$, where t is the number of iterations. Therefore, the final computational complexity of Algorithm 2 is $O(tn^3)$. In practical applications, Algorithm 2 generally converges well.

3.6. Comparison with related data representation techniques

The similarity matrix is used to measure the relationship between data points. The construction of a similarity matrix is a critical step in the graph embedding framework. NPE uses k-nearest-neighbor method to construct a similarity matrix, where each element is a pairwise Euclidean distance. However, a fixed number of neighborhood lacks of adaptivity in real applications. Besides, the data noises are inevitable in practice. Hence, the robustness for the construction of a similarity matrix is a desirable property.

Sparse representation-based techniques can adaptively determine number of neighbors and robust to noises by solving 1_1-norm optimization problems. However, these techniques ignore the underlying global structural information of data. By inheriting the advantages of LRR, the SLPP framework considers the global structure of high-dimensional data. Besides, this framework further exploits local structure by using manifold or sparsity regularization, which encourages the coefficients of samples from the same subspace to be highly correlated. In addition, it imposes a symmetric constraint on the low-rankness property of high-dimensional data representation. The symmetric similarity matrix obtained by the SLPP framework characterizes the global and local structure of high-dimensional data, which acquires a better estimate of the underlying subspace.

4. Experiments

In this section, we discuss the series of experiments conducted to evaluate the performance of the proposed SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S algorithms on publicly available databases, namely, the extended Yale B, AR, Hopkins 155 and COIL-20 databases. We compared these proposed algorithms with PCA, LPP, NPE, and neighborhood components analysis (NCA) [47], which are the most popular linear dimensionality reduction techniques for publicly available databases. The source code for four competitive algorithms was obtained from the Matlab toolbox of dimensionality reduction [48].

To further demonstrate the performance of the proposed algorithms, we also considered the three special cases:

(1) a symmetric matrix of LRR, i.e., $Z^* = Z + Z^T$, where Z is calculated by LRR, as a surrogate for the weight matrix of NPE, which is denoted as LRR-NPE.
(2) only considering manifold constraint of Problem (22) without symmetry, which is denoted as SLPP1.
(3) only considering symmetry of Problem (22) without manifold constraint, which is denoted as SLPP2.

This provided an intuitive evaluation baseline for evaluation. Because of the sample-specific corruptions in our experiments, Problem (22) and (29) are solved using the $l_{2,1}$-norm minimization operator [49].

For comparative purposes, the nearest neighbor (NN) classifier with Euclidean distance was employed in the subsequent classification tasks after extracting the features of the high-dimensional data. To overcome the small sample size problem, we first applied PCA as preprocessing retaining almost 98% energy. All experiments were performed on a personal computer running Windows 7 with an Intel Core i5-2300 CPU and 16 GB memory. The code was implemented on Matlab R2013b.

4.1. Experimental settings

4.1.1. Databases

Four benchmark databases were used in our experiments, that is, the extended Yale B, AR, USPS and COIL-20 databases. Details of the four databases are summarized below.

- Extended Yale B database [50,51]. This database contains 38 individuals and about 2414 frontal images, captured under various laboratory-controlled lighting conditions. There are around 59–64 images available for each individual. Each face image was manually cropped and normalized to size 48×42 pixels. Fig. 2 shows some image samples of five individuals from the extended Yale B database. We randomly selected 20, 30, 40, and 50 face images for each individual as training samples. All the remaining face images were used for testing.
- AR database [52]. This database contains over 4000 face images of 126 individuals. The images of each individual were taken

Fig. 2. Sample images of five individuals from the extended Yale B database.

Fig. 3. Sample images of three individuals with illumination and expression variations, sunglasses and scarves from the AR database.

Fig. 4. Sample images of ten digit characters from the USPS database.

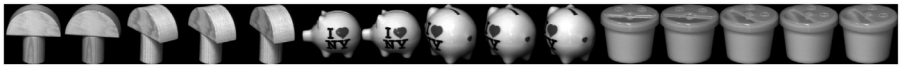

Fig. 5. Sample images of three objects from the COIL-20 database.

in two separate sessions. In each session, there are 13 images of each individual, consisting of three images with sunglasses, three with scarves, and the remaining seven with illumination and expression variations. We used a normalized face image of size 165×120 pixels. Fig. 3 shows some image samples of three individuals with illumination and expression variations, sunglasses, and scarves from the AR database. We chose a subset of the AR database containing 50 male individuals and 50 female individuals. We used the face images with illumination and expression variations for training and testing.

- USPS database [53]. The USPS handwritten digit database includes ten classes and 11,000 images in total. There are about 1100 images available for each digit character. In our experiments, each digit image was manually cropped and normalized to size 16×16 pixels. Fig. 4 shows some typical samples of 10 digit characters from the USPS database.
- COIL-20 database [54]. This database includes 1440 images with black background for 20 different subjects. Each subject has 72 images captured in equally space views. In our experiments, each object image was cropped to size 128×128 pixels. Fig. 5 shows some typical samples of three objects from the COIL-20 database.

4.1.2. Parameter selection

In our experiments, SLPP-M requires two key parameters: the noise regularization parameter λ and the manifold regularization parameter β. SLPP-S also includes two parameters: the noise regularization parameter λ and the sparsity regularization parameter γ.

We need another parameter μ for S-SLPP-M and S-SLPP-S. The selection of λ is more complicated than that of β because λ is closely related to the prior knowledge of different types of noise. Correspondingly, β is used to measure the effects of the local manifold structure of the high-dimensional data.

In the experiments, we obtained nearly satisfactory results when β was selected from the candidate value set $\{1e^{-8}, 1e^{-7}, 1e^{-6}\}$. Besides, the value of the parameter γ was selected from a set $\{1e^{-4}, 1e^{-3}, 1e^{-2}\}$. Similarly, parameter μ for S-SLPP-M varied between 0.1 and 0.4. If the value of parameter μ was set too large or too small, the class label information could not be utilized effectively in the construction of the graph.

Besides, we need to determine the appropriate value for the feature dimension r using an NN classifier. We can use a cross-validation mechanism to select r. The parameter settings for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S are given for each experiment. For the other algorithms, the parameters of each algorithm were carefully chosen and their best recognition results are reported. The bold numbers denote the highest recognition rates and the lowest standard deviations for each experiment.

4.2. Experiments on face recognition

To extensively investigate the effectiveness and robustness of the proposed methods as well as the other competing methods on face recognition, we conducted two kinds of experiments to examine the performance of these algorithms on the extended Yale B and AR databases, respectively. In the experiments, random projection (RP) was also employed to reduce the dimensionality of the facial images. We first applied the proposed algorithm to the original facial images in Section 4.2.1. Next, we considered actual occlusion of facial images in Section 4.2.2.

4.2.1. Face recognition on original facial images

We first examined the performance of these algorithms on the original facial images from the extended Yale database B. The SLPP-M and S-SLPP-M parameters were set to $\lambda = 5e^{-3}$ and $\beta = 1e^{-7}$. The SLPP-S and S-SLPP-S parameters were set to $\lambda = 0.1$ and $\gamma = 1e^{-4}$. Parameter μ of S-SLPP-M and S-SLPP-S ranged from 0.1 to 0.2 depending on the number of training samples. Fig. 6 shows the influence of the feature dimension for identical λ, β and γ values using different numbers of training samples on the face recognition accuracies of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S. We observe that S-SLPP-M significantly outperformed SLPP-M in most cases. For example, the dimensionality is ranged from 20 to 100 using 20 randomly selected images of each individual as training samples. Besides, S-SLPP-S achieved better performance than SLPP-S. Then, the face recognition accuracies of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S vary from 67.17% to 92.81%, 71.59% to 93.71%, 87.97% to 93.23% and 77.33% to 93.89%, respectively, as shown in Fig. 6(a). The optimal results of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S are obtained using 70, 90, 90 and 80 feature dimensions, respectively. We also observed similar results in Fig. 6(b)–(d). This demonstrates that S-SLPP can effectively improve the accuracy of face recognition when class label information of training samples is available.

We conducted 10 random experiments for each training size. The final recognition result was computed by averaging the recognition rates from these 10 experiments. Table 1 shows the face recognition accuracies and standard deviations of the different algorithms. The feature dimension for NN ranges from 70 to 90 according to the number of training samples. It is clear that our

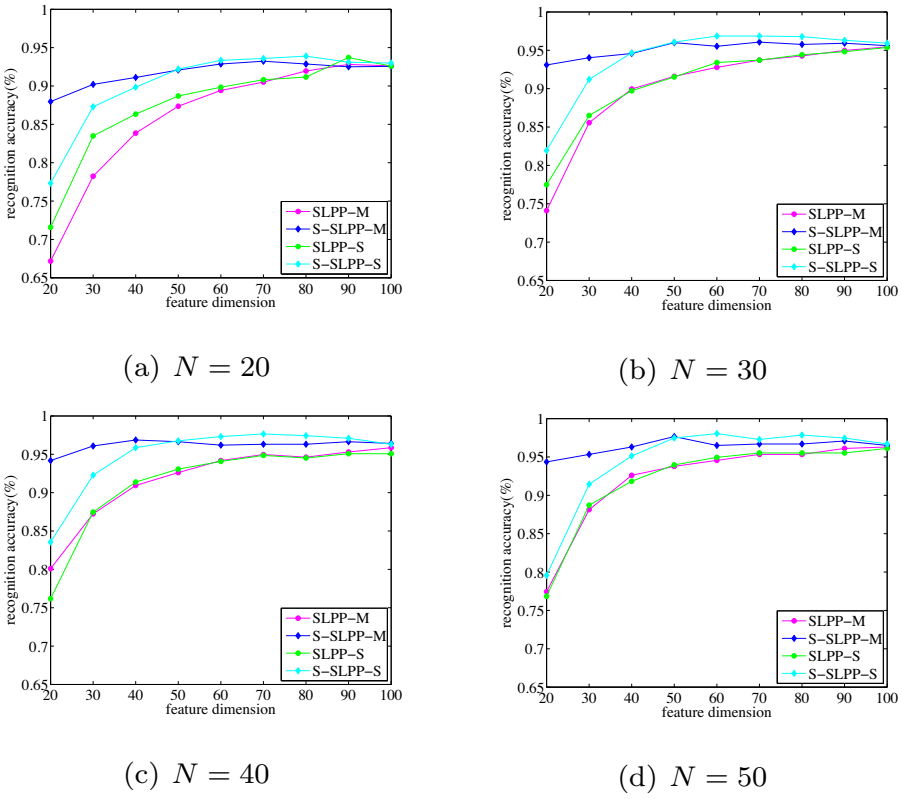

Fig. 6. Changes for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S in recognition accuracy when varying feature dimension using different numbers of training samples in the extended Yale B database.

Table 1
Face recognition rates and standard deviations (%) for different algorithms on the extended Yale B database.

Number	ACC.	SLPP-M	S-SLPP-M	SLPP-S	S-SLPP-S	SLPP1	SLPP2	LRR-NPE	NPE	LPP	PCA	RP	NCA
20	Mean	91.95	92.48	93.07	**93.48**	91.77	91.85	91.57	91.34	91.4	67.5	59.98	90.16
	Std.	0.82	0.74	**0.4**	0.67	0.48	0.51	0.68	0.66	0.57	0.98	2.24	0.94
30	Mean	95.21	95.28	95.4	**95.96**	94.92	94.61	94.18	94.14	94.14	75.11	67.67	94.07
	Std.	0.55	0.5	**0.43**	0.45	0.46	0.47	0.6	0.66	**0.5**	0.89	1.37	0.5
40	Mean	96.25	96.42	96.53	**97.11**	95.76	95.6	95.21	95.02	95.08	79.22	71.09	95.83
	Std.	0.38	**0.33**	0.35	0.42	0.44	0.38	0.52	0.48	0.64	1.29	1.86	0.7
50	Mean	96.73	97	97.08	**98.05**	96.32	96.2	96.09	95.45	95.51	82	72.9	96.69
	Std.	0.76	0.52	0	0	0.49	0.34	0.55	0.62	**0.49**	1.53	2.42	0.75

proposed S-SLPP-S method has greater recognition accuracy than the other algorithms. S-SLPP-S achieves face recognition accuracies of 93.48%, 95.96%, 97.11% and 98.05%, respectively, when randomly selecting 20, 30, 40, and 50 face images as training samples for each individual. In practice, class label information of training samples is often unavailable. We also observed that SLPP-M and SLPP-S still obtained competitive recognition results and outperformed the other unsupervised subspace learning algorithms. This shows that SLPP-M and SLPP-S can characterize the intrinsic structure of high-dimensional data well. Besides, using LRR, the graph-oriented methods (SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S, SLPP1, SLPP2 and LRR-NPE) improved their recognition performance compared with NPE using various similarity strategies. This shows that SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S are effective robust methods for graph construction.

Then, we evaluated the face recognition performance of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S as well as the other methods on the AR database using the clean samples. For each individual, the seven images from Session 1 were used for training, and the seven images from Session 2 for testing. The SLPP-M parameters were set to $\lambda = 0.8$ and $\beta = 1e^{-8}$, while the S-SLPP-M parameters were set to $\lambda = 0.2$, $\beta = 1e^{-6}$, and $\mu = 0.1$. The SLPP-S parameters were set to $\lambda = 4.5$ and $\beta = 1e^{-4}$, while the S-SLPP parameters were set to $\lambda = 5e^{-3}$, $\gamma = 1e^{-2}$, and $\mu = 0.1$. The feature dimension for NN was set to 100. Fig. 7 shows the changes in recognition accuracy

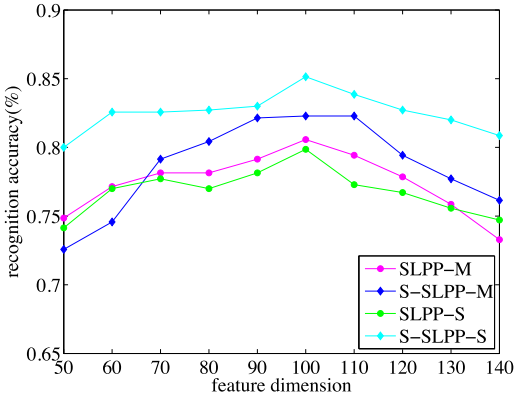

Fig. 7. Changes for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S in recognition accuracy when varying feature dimension on the AR database with the clean samples.

Table 2
Face recognition rates (%) for different algorithms on the AR database using the clean samples.

Algorithm	SLPP-M	S-SLPP-M	SLPP-S	S-SLPP-S	SLPP1	SLPP2	LRR-NPE	NPE	LPP	PCA	RP	NCA
ACC.	80.57	83.43	79.86	**85.14**	77.71	75.57	74.86	69.57	63.43	71.71	69.57	67

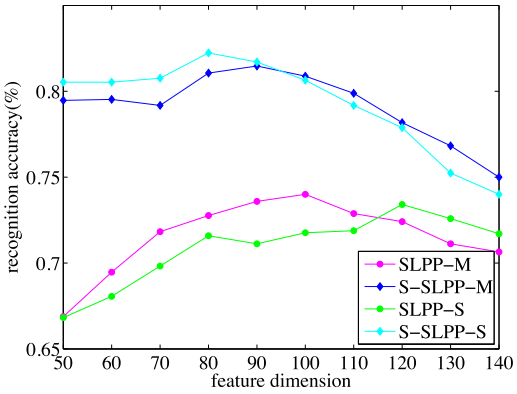

Fig. 8. Changes for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S in recognition accuracy when varying feature dimension on the AR database with the occlusion samples.

for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S with different feature dimensions. S-SLPPs still shows its distinct advantage as a supervised method. The recognition accuracies for SLPP-M and S-SLPP-M vary from 77.86% to 80.57%, and 79.14% to 83.43% with their feature dimensions ranging from 80 to 120, respectively. The recognition accuracies of SLPP-S and S-SLPP-S vary from 74.71% to 79.86%, and 80.86% to 85.14%, respectively. This indicates that the recognition performance of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S remained consistently stable for a large range of feature dimensions.

Table 2 shows the face recognition accuracies and standard deviations of different algorithms. S-SLPP-S has better recognition performance than the other algorithms. For example, S-SLPP-S significantly improved the recognition accuracy by at least 10% when compared with the other algorithms, i.e., NPE, PCA, RP, NCA, achieving a high recognition accuracy of 85.14%. SLPP-M achieved a recognition accuracy of 80.57%, and performed better than the other algorithms when considering only unsupervised execution. The improvement by S-SLPP-M and S-SLPP-S indicates the importance of symmetric low-rank representation of high-dimensional data with local constraint, i.e., manifold or sparsity regularization, in constructing the graph.

4.2.2. Face recognition on corrupt facial images

We explored the performance and robustness of these methods on a more challenging set of actual corrupt facial images. We selected seven neutral images and two corrupted images (one with sunglasses and the other with a scarf) from Session 1 for training. The rest of the images from Sessions 1 and 2 were used for testing. The SLPP-M parameters were set to $\lambda = 0.8$ and $\beta = 1e^{-8}$. The S-SLPP-M parameters were set to $\lambda = 0.2$, $\beta = 1e^{-6}$ and $\mu = 0.1$. The SLPP-S parameters were set to $\lambda = 3$ and $\beta = 1e^{-2}$. The S-SLPP-S parameters were set to $\lambda = 0.02$, $\beta = 1e^{-2}$ and $\mu = 0.1$. The feature dimensions of NN were set to 100, 90, 120 and 80 for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S, respectively.

Fig. 8 shows the comparative recognition accuracy for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S with varying feature dimensions on corrupt facial images. As expected, S-SLPP-S still significantly outperforms the other methods for the different feature dimensions. As shown in Fig. 8, S-SLPP-M and S-SLPP-S perform well for a large range of feature dimensions. For example, letting the feature dimension range from 80 to 120 with λ and β set as mentioned above for S-SLPP-M and S-SLPP-S, the recognition accuracies vary from 72.77% to 74%, and 78.18% to 81.06%, respectively. The recognition accuracies of SLPP-S and S-SLPP-S vary from 71.12% to 73.41%, and 77.88% to 82.24%, respectively. These recognition results show that SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S are capable of preserving the embedded geometric structure of high-dimensional data.

Table 3 shows the face recognition accuracies of different algorithms. S-SLPP-M and S-SLPP-S achieve a recognition accuracy of 81.47% and 82.35%, respectively. It is clear that the recognition accuracy by S-SLPP-S significantly outperforms that of the other methods. For example, S-SLPP-S improved the recognition accuracy by 15% compared with LRR-NPE. We also observed that the recognition performance by LRR-NPE outperforms that of NPE by a very small margin. This further illustrates the robustness and effectiveness of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S. This confirms that our proposed method is very effective and robust when the data are grossly corrupted by noise. In contrast, PCA and RP do not perform well, because they are very sensitive to large errors, such as occlusion and disguise.

4.3. Experiments on handwritten digit recognition

We evaluated the recognition rates of the proposed SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S with different training sizes on the USPS database. We designed six groups of training samples, with each group consisting of 30, 40, 50 60, 70, and 100 randomly selected images of each digit character, respectively. The remaining digit character images were used for testing. We repeated each experiment 10 times, and the final classification performance was computed by averaging the recognition rates over the standard deviation.

The SLPP-M parameters were set to $\lambda = 0.2$. The parameter β for SLPP-M was set to $1e^{-7}$ or $1e^{-6}$ according to the different training sizes. The S-SLPP-M parameters were set to $\lambda = 0.2$ and $\mu = 0.2$. The parameter β for S-SLPP-M was set to $1e^{-8}$ or $1e^{-7}$ according to the different training sizes. The SLPP-S parameters were set to $\lambda = 0.3$ and $\gamma = 1e^{-4}$. The S-SLPP-S parameters were set to $\lambda = 0.1$ and $\gamma = 0.1$, and $\mu = 0.2$. The feature dimensions of NN for both SLPP-M and S-SLPP-M were set to 15 and 25, respectively.

Fig. 9 shows comparative recognition accuracies of an experiment for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S with different numbers of feature dimensions using varied numbers of training samples. As shown in Fig. 9(a), S-SLPP-S achieves the best recognition results when feature dimension is more than 15. Moreover, SLPP-S performs better than SLPP-M when feature dimension is more than 20. However, SLPP-M can achieve similar performance as S-SLPP-M for a large range of feature dimension sizes when increasing the number of training samples, as shown in Fig. 9(b)–(f). SLPP-M and SLPP-S realize improved recognition performance when more unlabeled training samples are available.

The average recognition rates and standard deviations of the different algorithms are reported in Table 4 for various training sizes. We observed that the recognition rates of these algorithms can be greatly improved with an increase in the number of training samples. It can be observed from the experimental results that NCA achieves the best recognition results. However, it is clear that S-SLPP-M and S-SLPP-S consistently obtained higher recognition rates than the other algorithms for different numbers of training

Table 3
Face recognition rates (%) for different algorithms on the AR database with the occluded samples.

Algorithm	SLPP-M	S-SLPP-M	SLPP-S	S-SLPP-S	SLPP1	SLPP2	LRR-NPE	NPE	LPP	PCA	RP	NCA
ACC.	74	81.47	74.41	**82.35**	72.29	72.53	67.35	65.65	61.94	60.65	58.88	64.49

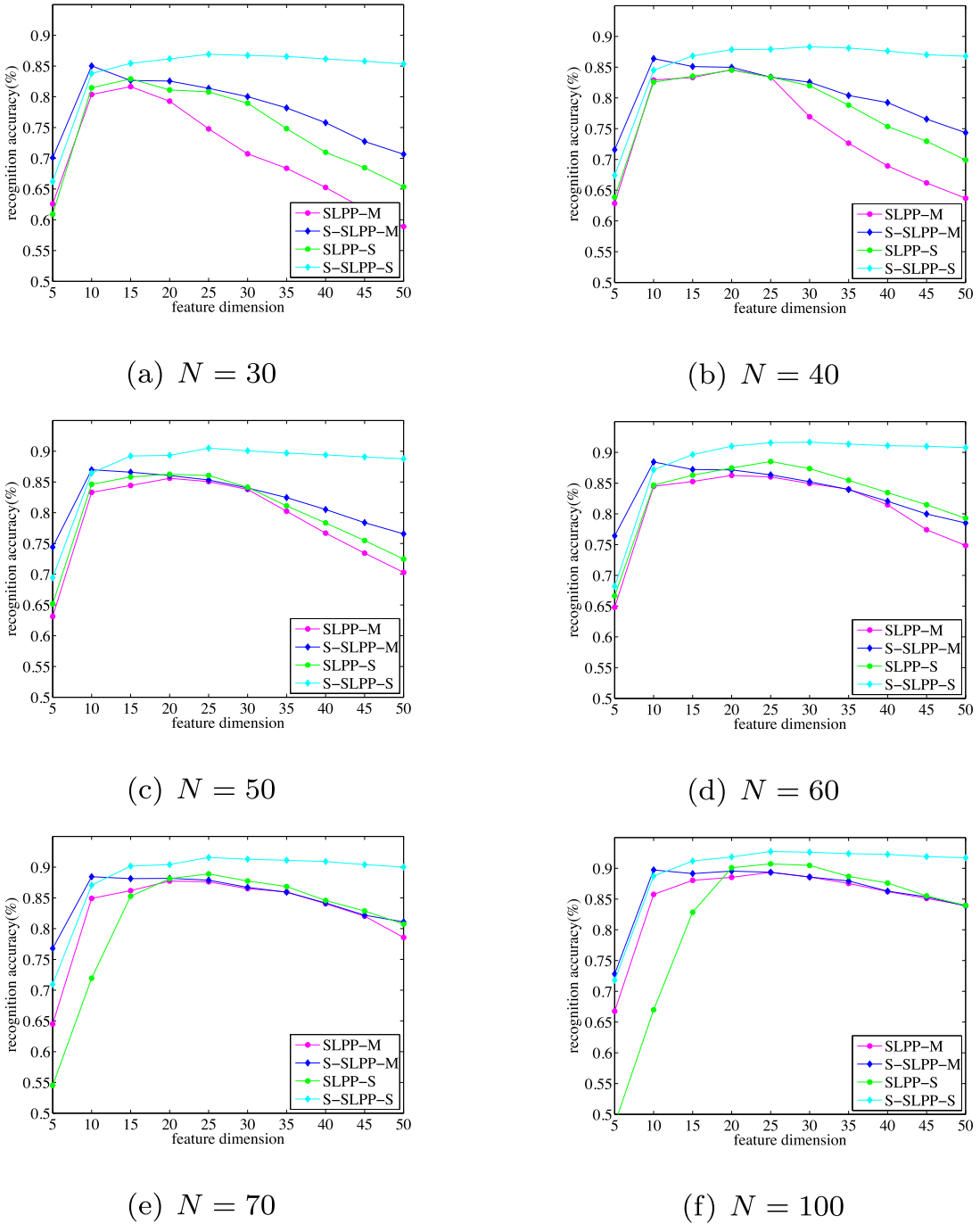

Fig. 9. Changes in recognition accuracy for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S when varying the feature dimension for different numbers of training samples in the USPS database.

samples, except the NCA algorithm. By constructing the new similarity matrix, the recognition accuracy of the proposed approaches significantly outperforms that of NPE. For example, the recognition accuracy of S-SLPP-S is nearly 12% higher than NPE with 30 training samples. LRR-NPE has a lower standard deviation than the other methods. However, the difference in standard deviation between LRR-NPE and S-SLPP-S decreased gradually when the number of training samples increased from 30 to 100.

4.4. Experiments on object recognition

In this experiment, we evaluated the proposed SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S algorithm for object recognition on the

Table 4
Recognition accuracies and standard deviations (%) for different algorithms on the USPS database.

Number	Error	SLPP-M	S-SLPP-M	SLPP-S	S-SLPP-S	SLPP1	SLPP2	LRR-NPE	NPE	LPP	PCA	NCA
30	Mean	81.21	83.4	81.93	87.51	81.05	81.1	80.98	75.69	73.21	78.78	**89.52**
	Std.	1.13	1.68	0.9	**0.67**	0.92	0.99	1.27	1.11	1.73	1.91	0.81
40	Mean	83.84	85.57	84.55	89.84	83.39	83.66	83.13	78.63	76.05	81.33	**90.84**
	Std.	0.96	1.2	0.77	0.7	0.72	0.79	**0.64**	1.2	1.68	1.78	0.76
50	Mean	85.6	86.52	86.18	90.86	85.28	85.45	85.09	81.05	78.63	83.13	**91.76**
	Std.	0.62	1.08	0.79	**0.58**	0.61	0.76	0.68	0.87	1.62	1.62	0.65
60	Mean	86.86	87.44	87.29	91.66	86.3	86.72	86.64	82.54	80.48	84.27	**92.7**
	Std.	0.7	0.64	0.73	0.49	0.6	0.7	0.63	0.73	1.01	1.28	**0.39**
70	Mean	87.7	88.71	88.1	92.13	87.34	87.57	87.46	84.12	82.03	85.3	**93.35**
	Std.	0.52	0.67	0.52	**0.45**	0.56	0.53	0.5	0.89	1.15	1.18	0.69
100	Mean	89.65	90.21	90.01	93.76	89.43	89.51	89.27	87.33	85.78	87.13	**94.5**
	Std.	0.54	0.74	0.44	**0.37**	0.46	0.5	0.53	0.61	0.85	1.25	0.48

Table 5
Recognition accuracies and standard deviations (%) for different algorithms on the COIL-20 database.

Number	Error	SLPP-M	S-SLPP-M	SLPP-S	S-SLPP-S	SLPP1	SLPP2	LRR-NPE	NPE	LPP	PCA	NCA
20	Mean	96.7	97.1	97.89	**98.65**	96.54	96.44	96.38	85.15	86.69	94.75	98
	Std.	0.69	0.81	**0.44**	0.6	0.56	0.8	0.71	0.81	1	0.71	0.74
30	Mean	98.65	98.58	99.4	**99.64**	98.4	98.45	98.33	87.96	91.57	97.05	99.14
	Std.	0.67	0.77	0.58	0.55	0.65	0.6	0.75	0.91	0.77	0.8	**0.51**
40	Mean	99.51	99.42	99.69	**99.84**	99.6	99.62	99.55	91.2	94.02	98.73	99.73
	Std.	0.26	0.76	0.3	**0.16**	0.23	0.27	0.24	0.98	0.82	0.37	0.28
50	Mean	99.68	99.59	99.77	**99.96**	99.53	99.63	99.5	92.64	95.8	99.2	99.82
	Std.	0.29	0.26	0.28	**0.1**	0.24	0.16	0.33	1.14	0.76	0.34	0.21

COIL-20 database. Four groups of different numbers of object samples, i.e., 20, 30, 40 and 50 randomly selected object images per each category, were used to construct the training sets. The rest of the object images were contained in the test sets. Each experiment is repeated 10 times, and we reported the average recognition rates over the standard deviation for each compared algorithm. The parameters λ and β for SLPP-M and S-SLPP-M were set to $1e^{-3}$ and $1e^{-6}$, respectively. The parameter μ for S-SLPP-M was set to 0.2. The parameters λ and γ for SLPP-S was set to 1 and $1e^{-2}$, respectively. The parameters λ and γ for S-SLPP-S was set to 1 and $1e^{-2}$, respectively. The parameter μ for S-SLPP-S varies from 0.2 to 0.4 depending on the number of training samples. The feature dimensions for NN for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S were set to 15.

Fig. 10 shows comparative recognition accuracies of an experiment for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S with different numbers of feature dimensions using varying numbers of training samples. The parameters λ and β for SLPP-M and S-SLPP-M were set to the same values, i.e., $\lambda = 1e^{-3}$ and $\beta = 1e^{-6}$. As shown in Fig. 10, the increasing numbers of training samples can effectively improve the recognition performance of SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S. In addition, S-SLPP-S achieved best recognition performance under the different numbers of training samples. However, SLPP-M, SLPP-S, and S-SLPP-M achieve similar performance as the number of training samples gradually increases.

Table 5 shows the average recognition rates and standard deviations of all compared algorithms. It can be observed from Table 5 that S-SLPP-S achieves the best recognition results. Besides, NCA achieves higher recognition rates than the other algorithms for different numbers of training samples, except the S-SLPP-S algorithm. However, the difference in the average recognition rates among SLPP-M, S-SLPP-M, SLPP-S and NCA was gradually reduced when the number of training samples increases from 20 to 50. Moreover, the recognition accuracy by SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S significantly outperforms that of NPE for different numbers of training samples. For example, the recognition accuracies of SLPP-S are nearly 11% and 7% higher than NPE with 20 and 50 training samples, respectively. Similarly, LRR-NPE also outperforms NPE in all cases. In contrast, NPE and LPP do not perform well. We experimentally observed that similarity matrix can significantly influence the recognition accuracy. Our SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S approaches consistently outperform NPE, which further demonstrates the importance of characterizing the relationship among data samples.

4.5. Discussion

First, local regularization is important for exploiting the underlying subspace structure of high-dimensional data in the SLPP framework. In particular, the manifold regularization takes into account the local manifold structure of high-dimensional data while the sparsity regularization considers the local geometric structure of high-dimensional data. Therefore, it enables us to capture the intrinsic structure information to learn multiple discriminative subspaces of high-dimensional data. Moreover, the recognition performance is robust to the setting of the manifold or sparsity regularization parameter in the experiments. In other words, the incorporation of manifold or or sparsity regularization in the symmetric low-rank representation only increases the new parameter setting in the SLPP framework. On the contrary, in the experiments, SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S significantly improve their recognition performance compared with LRR-NPE.

Second, we observed that NCA achieved the best recognition results on handwritten digit recognition. We elaborately choose several state-of-the-art algorithms to evaluate the performance of the proposed algorithms on publicly available databases. This provided an intuitive evaluation baseline for comparison. The difference of digit recognition rates between NCA and S-SLPP-S decreased gradually when the number of training samples increased from 30 to 100. Hence, more training samples are needed in the SLPP framework since the proposed algorithms characterize the low-rank structure of the given samples. What's more, we also observed that digit recognition rates by S-SLPP-S outperforms significantly that of LRR-NPE and NPE. It further demonstrates that the importance of graph construction in subspace learning and the advantages of exploiting the global and local structures of high-dimensional data in LRR.

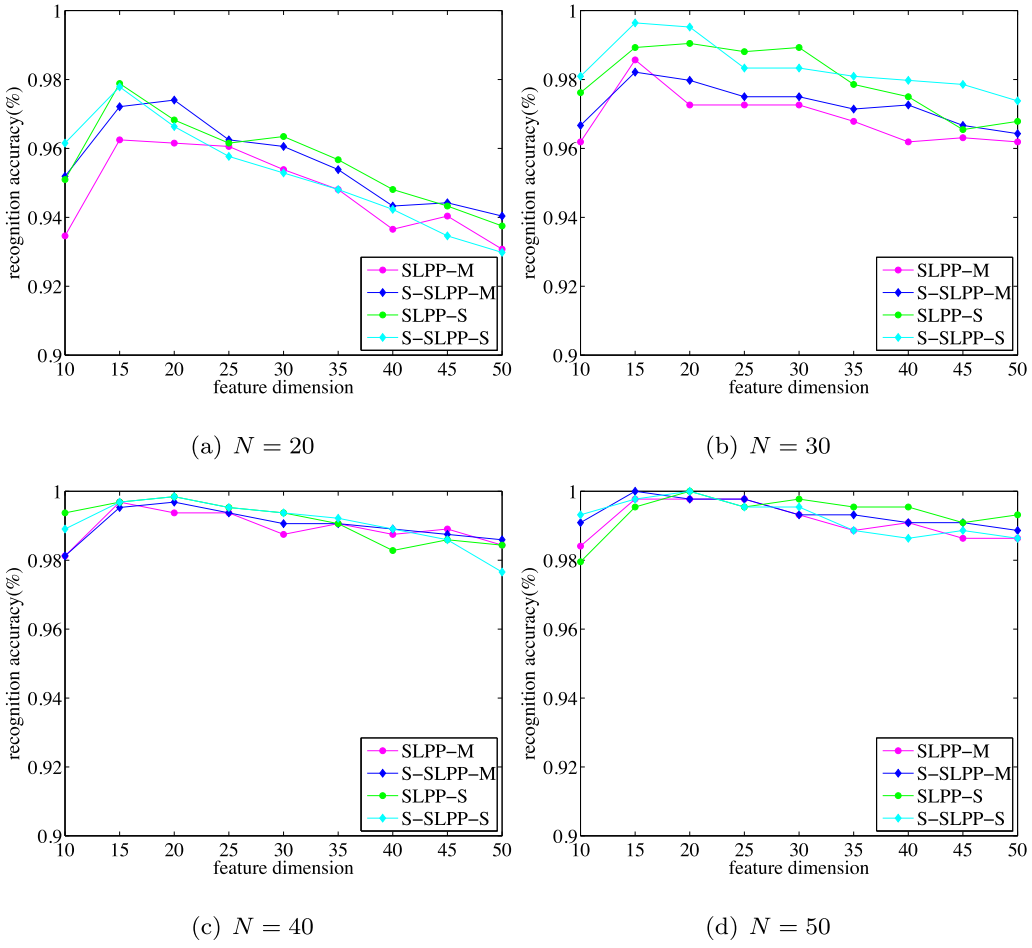

Fig. 10. Changes for SLPP-M, SLPP-S, S-SLPP-M and S-SLPP-S in recognition accuracy when varying feature dimension using different numbers of training samples in the COIL-20 database.

Finally, the SLPP framework provides a more flexible model of graph-oriented subspace learning, thereby unifying supervised and unsupervised methods. Most existing graph-oriented subspace learning algorithms either consider or ignore the advantages of class label information during subspace learning. However, one cannot confirm that training samples contain class label information without prior knowledge. This means that their benefit may be limited in practical applications. Since we adopted an intuitive criterion for graph construction by making use of the class label information of training samples, the recognition performance of S-SLPP-M and S-SLPP-S almost outperform that of SLPP-M and SLPP-S in the experiments, respectively. In addition, we found that more training samples can improve the recognition performance of SLPP-M and SLPP-S.

5. Conclusions

In this paper, we presented a symmetric low-rank representation framework incorporating the local structure of high-dimensional data for robust subspace learning. The proposed framework considers a local regularization with a symmetric low-rank representation to learn a similarity matrix, which greatly enriches the relationship among high-dimensional data by solving the low-rank optimization problem. The SLPP framework uses the similarity matrix to construct an affinity graph and then effectively obtains a transformation matrix for dimensionality reduction. Compared with other LRR-oriented subspace learning algorithms, the proposed algorithms consider both the global structure and local manifold structure of high-dimensional data. This enables the proposed algorithms to provide a more accurate data representation than others. Hence, the transformation matrix effectively preserves the low-dimensional structure features of high-dimensional data. Experimental results on benchmark databases demonstrated the effectiveness of the proposed methods for subspace learning by comparison of several popular subspace learning algorithms.

Although SLPP-M, SLPP-S and their variants are effective methods, they requires more computation time than NPE and LPP to achieve a better similarity matrix, especially for subspace learning with large-scale high-dimensional samples. Nonconvex low-rank optimization techniques may be good surrogates for solving low-rank optimization problem. In addition, estimating two proper parameters of the proposed methods without prior knowledge of noise is intractable. In future work, we will focus on these problems for practical applications.

References

[1] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Trans, Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[2] H.P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Transactions on Knowledge Discovery from Data 3 (1) (2009) 1–57.
[3] H. Qu, Z. Yi, A new algorithm for finding the shortest paths using pcnns, Chaos, Solitons & Fractals 33 (4) (2007) 1220–1229.
[4] L. Zhang, Z. Yi, S.L. Zhang, P.A. Heng, Activity invariant sets and exponentially stable attractors of linear threshold discrete-time recurrent neural networks, IEEE Transactions on Automatic Control 54 (6) (2009) 1341–1347.
[5] Z. Yi, Foundations of implementing the competitive layer model by lotkac–volterra recurrent neural networks, IEEE Transactions on Neural Networks 21 (3) (2010) 494–507.

[6] X. Peng, H. Tang, L. Zhang, Z. Yi, S. Xiao, A unified framework for representation-based subspace clustering of out-of-sample and large-scale data, IEEE Transactions on Neural Networks and Learning Systems 27 (12) (2016) 2499–2512.

[7] H. Zhu, R. Vial, S. Lu, X. Peng, H. Fu, Y. Tian, X. Cao, Yotube: Searching action proposal via recurrent and static regression networks, IEEE Transactions on Image Processing 27 (6) (2018) 2609–2622.

[8] J.T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, R.S.M. Goh, Transfer hashing: From shallow to deep, IEEE Transactions on Neural Networks and Learning Systems (2018) 1–11, doi:10.1109/TNNLS.2018.2827036.

[9] T. Boult, L. Brown, Factorization-based segmentation of motions, in: IEEE Workshop on Proceedings of the Visual Motion, 1991, pp. 179–186.

[10] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Trans. Pattern Anal. and Mach. Intell. 25 (2) (2003) 218–233.

[11] M. Turk, A. Pentland, Face recognition using eigenfaces, in: Proceedings of the CVPR, 1991, pp. 586–591.

[12] P. Belhumeur, J. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.

[13] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using lapla-cianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.

[14] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, Proceedings of the ICCV (2005) 1208–1213.

[15] J. Gou, Z. Yi, Locality-based discriminant neighborhood embedding, The Computer Journal 56 (9) (2013) 1063–1082.

[16] D. Cai, X. He, J. Han, Spectral regression for efficient regularized subspace learning, Proceedings of the ICCV (2007) 1–8.

[17] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multi-view features for image re-ranking, IEEE Transactions on Multimedia 16 (1) (2014) 159–168.

[18] X. Peng, J. Lu, Z. Yi, R. Yan, Automatic subspace learning via principal coefficients embedding, IEEE Trans. Cybern. 47 (11) (2017) 3583–3596.

[19] C. Deng, Z. Chen, X. Liu, X. Gao, D. Tao, Triplet-based deep hashing network for cross-modal retrieval, IEEE Transactions on Image Processing 27 (8) (2018) 3893–3903.

[20] K. Papachristou, A. Tefas, I. Pitas, Symmetric subspace learning for image analysis, IEEE Trans. Image Processing 23 (12) (2014) 5683–5697.

[21] T. Wimalajeewa, Y. Eldar, P.K. Varshney, Subspace recovery from structured union of subspaces, IEEE Trans. Information Theory 61 (4) (2015) 2101–2114.

[22] S.Yan X.H, Y. Hu, H. Zhang, Learning a locality preserving subspace for visual recognition, in: Proceedings of the ICCV, 2003, pp. 385–392.

[23] W. Yu, X. Teng, C. Liu, Face recognition using discriminant locality preserving projections, Image and Vision computing 24 (3) (2006) 239–248.

[24] L. Zhu, S. Zhu, Face recognition based on orthogonal discriminant locality preserving projections, Neurocomputing 70 (7) (2007) 1543–1546.

[25] L. Zhang, L. Qiao, S. Chen, Graph-optimized locality preserving projections, Pattern Recognition 43 (6) (2010) 1993–2002.

[26] J. Lu, Y. Tan, Regularized locality preserving projections and its extensions for face recognition, IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics 40 (3) (2010) 958–963.

[27] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10 (5) (1998) 1299–1319.

[28] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural computation 12 (10) (2000) 2385–2404.

[29] J. Tenenbaum, V. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

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

[31] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Proceedings of the NIPS (2002) 585–591.

[32] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.

[33] B. Cheng, J. Yang, S. Yan, Y. Fu, T.S. Huang, Learning with l1-graph for image analysis, IEEE Trans. Image Processing 19 (4) (2010) 858–866.

[34] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, S. Ji, Discriminant sparse neighborhood preserving embedding for face recognition, Pattern Recognition 45 (8) (2012) 2884–2893.

[35] Z. Zhang, S. Yan, M. Zhao, Pairwise sparsity preserving embedding for unsupervised subspace learning and classification, IEEE Trans. Image Processing 22 (12) (2013) 4640–4651.

[36] Y. Fang, R. Wang, B. Dai, X. Wu, Graph-based learning via auto-grouped sparse regularization and kernelized extension, IEEE Trans. Knowledge and Data Engineering 27 (1) (2015) 142–154.

[37] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.

[38] Q. Yao, J.T. Kwok, W. Zhong, Fast low-rank matrix learning with nonconvex regularization, Proceedings of the ICDM (2015) 539–548.

[39] S. Li, Y. Fu, Learning robust and discriminative subspace with low-rank constraints, IEEE Trans. Neural Networks and Learning Systems doi:10.1109/TNNLS.2015.2464090.

[40] Q. Yao, J.T. Kwok, Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity, Journal of Machine Learning Research 18 (1) (2018) 1–52.

[41] Y. S. J. Chen, H. Mao, Z. Yi., Subspace clustering using a symmetric low–rank representation 127 (1) (2017) 46–57.

[42] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Trans. Image Processing 20 (5) (2011) 1327–1336.

[43] D. Cai, X. He, J. Hang, T. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.

[44] X. Lu, Y. Wang, Y. Yuan, Graph-regularized low-rank representation for destriping of hyperspectral images, IEEE Trans. Geoscience and Remote Sensing 51 (7) (2013) 4009–4018.

[45] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Review 51 (1) (2009) 34–81.

[46] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv preprint arXiv:1009.5055.

[47] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, in: Proceedings of the NIPS, 2004, pp. 513–520.

[48] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.

[49] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, Proceedings of the ICML (2010) 663–670.

[50] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. and Mach. Intell. 27 (5) (2005) 684–698.

[51] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2011) 643–660.

[52] A. Martinez, R. Benavente, The ar face database, CVC Tech. Report No. 24.

[53] J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.

[54] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (coil-20), Technical Report CUCS-005-96.

Jie Chen received the B.S. degree in Software Engineering and the M.S. degree in Computer Science from Sichuan University, Chengdu, China, in 2005 and 2008, respectively. Currently he is pursuing his Ph.D. degree at Machine Intelligence Laboratory, College of Computer Science, Sichuan University. His research interests focus on pattern recognition, computer vision and machine learning.

Hua Mao received the B.S. degree and M.S. degree in Computer Science from Electronic Science and Technology of China (UESTC) in 2006 and 2009, respectively. She received her Ph.D. degree in Computer Science and Engineering in Aalborg University, Denmark in 2013. Her current research interests include Neural Networks and Big Data.

Haixian Zhang received the B.S. degree in Mathematics and Computer Science from Henan Normal University in 2003, and received the M.S. degree and Ph.D. degree in Applied Mathematics from University of Electronic Science and Technology of China (UESTC) in 2006 and 2010, respectively. Her current research interests include theories and applications of neural networks.

Zhang Yi received the Ph.D. degree in mathematics from the Institute of Mathematics, The Chinese Academy of Science, Beijing, China, in 1994. Currently, he is a Professor at the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. He is the co-author of three books: Convergence Analysis of Recurrent Neural Networks (Kluwer Academic Publishers, 2004), Neural Networks: Computational Models and Applications (Springer, 2007), and Subspace Learning of Neural Networks (CRC Press, 2010). He was an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems (2009 2012), and He is an Associate Editor of IEEE Transactions on Cybernetics (2014). His current research interests include Neural Networks and Big Data.