Neural Networks 88 (2017) 1–8

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Orthogonal self-guided similarity preserving projection for classification and clustering

Xiaozhao Fang [a], Yong Xu [b,*], Xuelong Li [c], Zhihui Lai [d], Shaohua Teng [a], Lunke Fei [b]

[a] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, 510006, China

[b] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, 518055, Guangdong, China

[c] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Chinese Academy of Sciences, Xian, 710119, Shaanxi, China

[d] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518055, Guangdong, China

ARTICLE INFO

Article history:
Received 23 August 2016
Received in revised form 20 November 2016
Accepted 5 January 2017
Available online 16 January 2017

Keywords:
Dimensionality reduction
Intrinsic structure
Subspace clustering
Feature representation

ABSTRACT

A suitable feature representation can faithfully preserve the intrinsic structure of data. However, traditional dimensionality reduction (DR) methods commonly use the original input features to define the intrinsic structure, which makes the estimated intrinsic structure unreliable since redundant or noisy features may exist in the original input features. Thus a dilemma is that (1) one needs the most suitable feature representation to define the intrinsic structure of data and (2) one should use the proper intrinsic structure of data to perform feature extraction. To address the problem, in this paper we propose a unified learning framework to simultaneously obtain the optimal feature representation and intrinsic structure of data. The structure is learned from the results of feature learning, and the features are learned to preserve the refined structure of data. By leveraging the interactions between the process of determining the most suitable feature representation and intrinsic structure of data, we can capture accurate structure and obtain the optimal feature representation of data. Experimental results demonstrate that our method outperforms state-of-the-art methods in DR and subspace clustering. The code of the proposed method is available at "http://www.yongxu.org/lunwen.html ".

1. Introduction

In many computer vision and pattern recognition applications, high-dimensional data often contain some redundant or noisy features. The performance of algorithm may drop exponentially as the dimensionality of data increases (Elhamifar & Vidal, 2013; Lai, Xu, Yang, Tang, & Zhang, 2013; Mardani, Mateos, & Giannakis, 2015; Wang, Nie, Yang, Gao, & Yao, 2015). Therefore, it is essential to seek a low-dimensional representation for the original high-dimensional data (Fang et al., 2014; Jing-Yan Wang & Gao, 2015). Principal component analysis (PCA) is an unsupervised dimensionality reduction (DR) method which maps high dimensional data into a low-dimensional subspace by seeking the direction of maximum variance for optimal data reconstruction (Fan et al., 2014; Li, Pang, & Yuan, 2010). Locally linear embedding (LLE) (Roweis & Saul, 2010) and Laplacian eigenmap (LE) (Mikhail & Niyogi, 2001) were recently proposed to discover the intrinsic manifold structure of data. However, LE cannot deal with new data points, which is commonly referred to as the "out-of-sample" problem. Locality preserving projection (LPP) method can address this problem because the obtained projection matrix can directly deal with new data (Niyogi, 2014). Local learning projection (LLP) was also proposed to address the same problem (Wu, Yu, Yu, & Scholkopf, 2007). Neighborhood preserving embedding (NPE) was proposed to preserve the local neighborhood structure on the manifold (He, Cai, Yan, & Zhang, 2005). Recently, many neural networks methods were proposed to perform data representation (Huang, 1999; Huang & Du, 2008; Huang & Jiang, 2012; Lemme, Reinhart, & Steil, 2012).

Semi-supervised learning (SSL) can utilize both limited labeled data and abundant yet unlabeled data to seek a suitable data representation for boosting algorithmic performance. Graph based SSL (G-SSL) methods have been successfully applied in capturing desired structures of data. In addition, the Hessian regularization can also be viewed as a graph embedding to explore the local geometry structure of data for boosting the performance of SSL (Liu, Liu, Tao, Wang, & Lu, 2015; Tao, Jin, Liu, & Li, 2013; Wang & Huang, 2009; Wang, Huang, & Xu, 2010). The performance of

* Corresponding author.
E-mail address: yongxu@ymail.com (Y. Xu).

http://dx.doi.org/10.1016/j.neunet.2017.01.001
0893-6080/

G-SSL method heavily relies on the graph construction process. Thus, lots of researches focus on the problem of constructing a suitable graph. For example, ℓ_1 graph (Yan & Wang, 2009), local linear reconstruction graph proposed in LLE (LLE graph) (Roweis & Saul, 2010), sparse probability graph (SPG) (He, Zheng, Hu, & Kong, 2011) (SPG in essence is a sparse coding problem with the non-negative constraint) and low-rank representation graph (LRR graph) (Liu, Lin, & Yu, 2010) have been proposed. Although these methods obtain empirical success, there are still some disadvantages. For example, reconstruction and minimization in traditional LLE are only processed within the sample neighbors defined by some conventional graphs construction methods such as k nearest neighbor (kNN-graph) or ϵ graph. Such procedure cannot provide an adaptive neighbor for the algorithm. kNN-graph and ϵ graph have the similar problem. Differing from conventional graph construction methods, ℓ_1 graph is sparse. It is well known that ℓ_1 graph is only to find the sparse representation for data reconstruction. However, the best data reconstruction does not mean the best discriminating power (Patel, Nguyen, & Vidal, 2013; Zhang, Zhou, & Chang, 2012). The LRR graph is usually dense, which is undesirable for G-SSL. Moreover, LRR allows the data to "cancel each other out" by subtraction. In other words, the weight of the graph in LRR may be negative, which lacks physical interpretation for the visual data. What is more, almost all these methods construct the graph structure on the original high-dimensional feature space, which is unnecessary to be best for characterizing the pairwise data relationship due to the fact that some redundant or noisy features may exist in the original feature representation.

In the unsupervised scenario, one needs the most suitable feature representation to define the structure of data and simultaneously one needs the data structure to perform feature extraction. However, both of them are not known in advance. Facing with such dilemma, in this paper we propose a novel orthogonal self-guided similarity preserving projection (OSSPP) method which can simultaneously learn them. Specifically, the similarity structure information of data is encoded by reconstruction coefficients of the projected data, and at the same time the projected data are required to respect the similarity structure via the similarity preserving regularization term. By leveraging the interactions between these two essential tasks, we are able to learn the best of them. In other words, the projection matrix and reconstruction coefficient matrix can be mutually improved during the process of learning. Fig. 1 presents an overview of our proposed OSSPP. We explicitly enforce the reconstruction coefficient matrix to be non-negative so that the reconstruction coefficient matrix can be directly used as the graph weights. Although we do not explicitly impose the sparsity constraint on the reconstruction coefficients, the problem to determine the reconstruction coefficient matrix is naturally converted to a weighted non-negative sparse coding problem. In this way, OSSPP allows the reconstruction coefficients matrix to be sparsity, datum-adaptive neighborhood, which is useful to construct a sparse graph for subspace clustering. It is obvious that OSSPP is somewhat similar with sparsity preserving projections (SPP) (Qiao, Chen, & Tan, 2010) without any explicit sparsity constraint. Thus, the projection learned by OSSPP contains natural discriminative information. We conduct extensive experiments on public data sets for DR and SSL tasks. Please note that this work is an extension to previous conference publication (Fang, Xu, Zhang, Lai, & Shen, 2015). This work adds the application of semi-supervised subspace clustering and analysis of the difference between our method and related methods.

The remainder of this paper is organized as follows: in Section 2, the related work is presented. Section 3 describes OSSPP method. Section 4 presents the experimental results. Finally, Section 5 offers our conclusion.

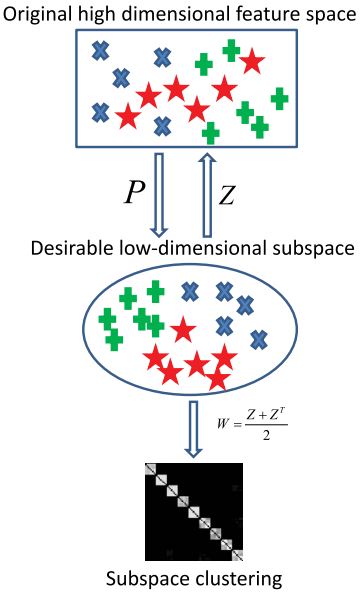

Fig. 1. Overview of OSSPP. In our framework, P projects the original data into a desirable low-dimensional subspace for learning Z and at the same time Z is also used to refine P. Doing so, they can be mutually improved. Finally, we use P and W to perform DR and subspace clustering, respectively.

2. Related work

Since our work is based on latent space sparse subspace clustering (LS3C) (Patel et al., 2013), we briefly review its formulations for the sake of completeness. Let $X = [x_1, x_2, \ldots, x_n] \in \Re^{m\times n}$ be a collection of n training samples $\{x_i \in \Re^m\}_{i=1}^n$ drawn from a union c of linear subspaces $\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_c$ of dimensions $\{d_\ell\}_{\ell=1}^c$ in $\Re^m$. Let $X_i \in \Re^{m\times n_\ell}$ be a sub-matrix of X of rank d_ℓ with $n_\ell > d_\ell$ training samples that lie in Ω_ℓ with $n_1 + n_2 + \cdots + n_c = n$. Each sample in X can be well represented by a linear combination of at most d_ℓ samples in X.

$$x_i = Xz_i, \qquad z_{ii} = 0, \qquad \|z_i\|_0 \le d_\ell \tag{1}$$

where z_i is the reconstruction coefficient vector. Considering all training samples (matrix form), we can rewrite the above formulation as

$$\min \|Z\|_1, \quad \text{s.t. } X = XZ,\ \text{diag}(Z) = 0 \tag{2}$$

where $\|Z\|_1 = \sum_{i=1}^n \sum_{j=1}^n |Z_{ij}|$ is the ℓ_1-norm of representation coefficient matrix Z.

In real-world applications, the data are often contaminated by some arbitrary noise E, *i.e.*, $X = XZ + E$, we may reformulate problem (2) as

$$\min_Z \|Z\|_1 + \gamma\|X - XZ\|_F^2, \quad \text{s.t. } \text{diag}(Z) = 0 \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm.

LS3C embeds samples into a low-dimensional space and simultaneously finds the sparse code in this space. Let $P \in \Re^{t\times m}$ be the projection matrix that projects the training samples from the original high-dimensional feature space $\Re^m$ into a latent output space of dimensionality t. By minimizing the following cost function

$$\begin{aligned} &[P^*, Z^*] = \min_{P,Z} J(P, Z, X) \\ &\text{s.t. } PP^T = I, \quad \text{diag}(Z) = 0, \end{aligned} \tag{4}$$

where

$$J(P, Z, X) = \|Z\|_1 + \gamma_1 \|PX - PXZ\|_F^2 + \gamma_2 \|X - P^T PX\|_F^2.$$

The goal of the first two terms of J is to insure sparsity and reconstruction of data in the reduced space. The last term, which is a PCA-like regularization term, ensures that the projection can hold the main energy of data. γ_1 and γ_2 are non-negative constants that control reconstruction and PCA-like regularization, respectively. In LS3C, the rows of P are required to be orthogonal. Then the above formulation can be written as

$$\begin{aligned} &[P^*, Z^*] = \min_{P,C} J(P, Z, X) \\ &\text{s.t. } PP^T = I, \quad Z^T 1 = 1, \quad \text{diag}(Z) = 0 \end{aligned} \tag{5}$$

where $Z^T 1 = 1$ is the affine constraint which is used to deal with the data that lie on a union of affine rather than linear subspace. Although the sparsity structure is preserved in LS3C, the similarity structure may be lost. The similarity structure of the data is as important as the sparsity property for discriminant analysis and subspace clustering, which is verified by the subsequent experimental results.

3. Orthogonal Self-guided Similarity Preserving Projections (OSSPP)

In this section, we introduce our orthogonal self-guided similarity preserving projection (OSSPP) method which can be used to perform DR and subspace clustering. Unlike previous DR methods that firstly encode the similarity structure information of data as graph relationship and then enforce the projected data to respect the graph structure, OSSPP uses the reconstruction coefficients of the projected data to encode the similarity structure information and simultaneously requires the projected data to respect the similarity structure during the procedure of DR. In this way, these two tasks can be mutually improved so that we can obtain the most suitable feature representation and accurate similarity structure of data. A natural assumption is that if Z can capture the similarity, then any two projected data points $P^T x_i$ and $P^T x_j$ that are close in the intrinsic geometry of the data distribution have a big weight Z_{ij}. A reasonable criterion for choosing a "good" map is to minimize the objective function $\sum_i^n \sum_j^n \|P^T x_i - P^T x_j\|^2 Z_{ij}$. It will ensure that $P^T x_i$ and $P^T x_j$ are close in the projected low-dimensional subspace. Based on the above insights, we propose the following objective function for OSSPP.

$$\begin{aligned} &[P^*, Z^*] = \min_{P,Z} F(P, Z) \\ &\text{s.t. } P^T P = I, \ \text{diag}(Z) = 0, \quad Z \geq 0, \ \forall i \end{aligned} \tag{6}$$

where

$$\begin{aligned} F(P, Z) = &\|P^T X - P^T XZ\|_F^2 + \alpha \|X - PP^T X\|_F^2 \\ &+ \beta \sum_{i=1}^n \sum_{j=1}^n \|P^T x_i - P^T x_j\|^2 Z_{ij} \end{aligned}$$

where reconstruction coefficient matrix Z is required to be non-negative so that it can be directly used as graph weights. We do not impose the affine constraint on the reconstruction coefficient matrix so that it can provide more freedom to well capture the similarity structure of data. The last term in function F is the similarity preserving regularization term which aims to require the projected data to respect the similarity structure during the procedure of DR. The whole learning processing is driven by the philosophy that the optimal feature representation and intrinsic structure of data jointly constitute a harmonic system, where the optimal feature representation is finally invariant with respect to the intrinsic structure on the similarity graph.

3.1. Optimization

In this section, we propose an iterative update rule to solve problem (6) of OSSPP. Specifically, the first step of the optimization algorithm solves P by fixing Z and the second step solves Z by fixing P.

Solve P by fixing Z:

If Z is fixed, the optimization problem defined in (6) can be written as

$$\begin{aligned} P^* = &\arg\min_P \|P^T X - P^T XZ\|_F^2 \\ &+ \alpha \|X - PP^T X\|_F^2 + \beta Tr(P^T XLX^T P) \\ &\text{s.t. } P^T P = I \end{aligned} \tag{7}$$

where $L = D - Z$ is graph Laplacian and D is a diagonal matrix with $D_{jj} = \sum_k Z_{jk}$. $Tr(\cdot)$ is the trace operator of matrix.

Considering the constraint $P^T P = I$, we can further transform (7) into

$$\begin{aligned} P^* = &\arg\min_P Tr(P^T (X - XZ)(X - XZ)^T P) \\ &+ \alpha Tr(X^T X - P^T XX^T P) + \beta Tr(P^T XLX^T P). \end{aligned} \tag{8}$$

Let $(X - XZ)(X - XZ)^T = M$, then (8) can be written as

$$\begin{aligned} &P^* = \arg\min_P Tr(P^T (M - \alpha XX^T + \beta XLX^T) P) \\ &\text{s.t. } P^T P = I. \end{aligned} \tag{9}$$

The solution of (9) can be obtained by solving the minimum eigenvalues problem:

$$(M - \alpha XX^T + \beta XLX^T) p_i = \lambda p_i. \tag{10}$$

Let $P = [p_1, \ldots, p_d]$ be the solution of (10). Column vectors $p_i s (i = 1, \ldots, d)$ correspond to the eigenvectors corresponding to the first d smallest eigenvalues.

Solve Z by fixing P:

If P is fixed, the optimization problem defined in (6) can be written as

$$\begin{aligned} &\min_Z \|P^T X - P^T XZ\|_F^2 + \beta \sum_{i=1}^n \sum_{j=1}^n \|P^T x_i - P^T x_j\|^2 Z_{ij} \\ &\text{s.t. } \text{diag}(Z) = 0, \quad Z \geq 0. \end{aligned} \tag{11}$$

(11) can be written as

$$\begin{aligned} &\min_Z \|H - HZ\|_F^2 + \beta Tr(\Theta (R \odot Z)) \\ &\text{s.t. } \text{diag}(Z) = 0, \quad Z \geq 0, \ \forall i \end{aligned} \tag{12}$$

where $H = P^T X = [h_1, \ldots, h_n] \in \Re^{d \times n}$, $R_{ij} = \|P^T x_i - P^T x_j\|^2 (R = [r_1, \ldots, r_n] \in \Re^{n \times n})$ and $\Theta \in \Re^{n \times n}$ is a matrix with all elements as 1. $\odot$ is the Hadamard operation. The optimization problem in (12) can be decomposed into n independent sub-problems for each coding coefficient $z_i (i = 1, \ldots, n)$ corresponding to $h_i (i = 1, \ldots, n)$ and each sub-problem is a weighted non-negative sparse coding problem.

$$\begin{aligned} &\min_{z_i} \sum_{k=1}^n r_i^k z_i^k + \beta \|h_i - Hz_i\|^2 \\ &\text{s.t. } z_i \geq 0, \quad z_i^i = 0, \ \forall i \end{aligned} \tag{13}$$

where z_i^k and r_i^k are the kth elements of the vectors z_i and r_i, respectively. Many algorithms, such as basis pursuit (SP) (Qiao et al., 2010) and fast iterative shrinkage and thresholding (FISTA) (Zhang et al., 2012) can be used to solve (13). Here, the alternating direction method (ADM) (Yang, Chou, Zhang, Xu, & Yan, 2013;

Yang & Zhang, 2011) is used to solve the optimization problem (13). Thus, we convert (13) into the following problem

$$\min_{z\geq 0} \|z\|_{r,1} + \beta\|h_i - H_{-i}z\|_2^2 \tag{14}$$

where H_{-i} represents the vectors $\{h_1, \ldots, h_{i-1}, h_{i+1}, \ldots, h_n\}$ and $\|\cdot\|_{r,1}$ is the weighted ℓ_1 (semi-) norm defined as $\|z\|_{r,1} \triangleq \sum_{k=1}^{n} r^k|z^k|$. Let $z = f$, we have

$$\min_{f\geq 0, z} \|f\|_{r,1} + \beta\|h_i - H_{-i}z\|_2^2, \quad \text{s.t. } z = f. \tag{15}$$

The augmented Lagrangian function of problem (15) is

$$\mathcal{J} = \arg\min_{f\geq 0, z} \|f\|_{r,1} + \beta\|h_i - H_{-i}z\|_2^2 + \langle y, z - f\rangle + \frac{\mu}{2}\|z - f\|_2^2 \tag{16}$$

where y is the Lagrange multiplier and $\mu > 0$ is the penalty parameter. The variables are updated alternately by minimizing the Lagrangian function, with other variables fixed. The iteration stops when the convergence conditions are met. We provide details of solving (16) in the following.

Step 1. Update z: Updating z by solving the following problem.

$$\mathcal{J} = \arg\min_{z} \beta\|h_i - H_{-i}z\|_2^2 + \frac{\mu}{2}\left\|z - f + \frac{y}{\mu}\right\|_2^2 \tag{17}$$

which can be rewritten as

$$\mathcal{J} = \arg\min_{z} \beta\|h_i - H_{-i}z\|_2^2 + \frac{\mu}{2}\|z - b\|_2^2 \tag{18}$$

where $b = f - \frac{y}{\mu}$. By setting the derivative $\frac{\partial \mathcal{J}}{\partial z} = 0$, we obtain

$$z = \left(\beta(H_{-i})^T(H_{-i}) + \frac{\mu}{2}I\right)^{-1}\left(\beta(H_{-i})^T h_i + \frac{\mu}{2}b\right). \tag{19}$$

Step 2. Update f: Updating f by solving the following problem.

$$\mathcal{J} = \arg\min_{f\geq 0} \|f\|_{r,1} + \frac{\mu}{2}\left\|z - f + \frac{y}{\mu}\right\|_2^2 \tag{20}$$

which has the following closed form solution by the one-dimensional shrinkage (or soft thresholding) formula:

$$f_i^k = \max\left(0, shrink\left(z_i^k + \frac{y_i^k}{\mu}, \frac{r_i^k}{\mu}\right)\right). \tag{21}$$

We obtain the solutions of (6) by updating P and Z iteratively. The overall algorithm of OSSPP is described in detail in Algorithm 1.

Algorithm 1 : OSSPP

Input: Training samples matrix X; Parameters α, β; Dimensionality of low-dimensional feature space d;
Initialization: Initializing Z as a similarity matrix by k nearest neighbor graph;
while not converged **do**
1. Update P by solving (9)
2. Update Z by solving (12)

end while
Output: Projections P and reconstruction coefficient matrix Z

3.2. Sparse graph construction

Once the sparse reconstruction coefficient matrix Z is obtained, we can construct an undirected graph $G = (V, E)$ associated with a weight matrix $W = \{w_{ij}\}$, where $V = \{v_i\}_{i=1}^n$ represents the vertex set and each vertex corresponding to a sample x_i. $E = \{e_{ij}\}$ is the set of edge and each edge e_{ij} associating nodes v_i and v_j with a weight w_{ij}. The next problem is how to define the graph weight matrix W when the vertex set V is given.

The non-negative weighted sparse coding in OSSPP can guarantee that each sample is associated with only a few samples and thus the graph derived from Z is naturally sparse. Since each sample can be represented by the other samples, a column vector $z_i (i = 1, \ldots, n)$ of Z naturally characterizes the other samples contribution in reconstructing of x_i. Such information is helpful to recover the clustering structure among samples. Thus we can directly define the graph weight matrix W as

$$W = (Z + Z^T)/2. \tag{22}$$

3.3. Different from NNLRS (Zhuang et al., 2012)

In our OSSPP method, we use sparse reconstruction coefficient matrix Z as graph to perform semi-supervised subspace clustering. To our best knowledge, non-negative low-rank and sparse graph (NNLRS) is the originally designed for the semi-supervised clustering problem by using the low-rank and sparse reconstruction coefficient matrix. The objective function of NNLRS is

$$\min_{Z,E} \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_{2,1} \quad \text{s.t. } X = AZ + E, \quad Z \geq 0 \tag{23}$$

where $\|Z\|_*$ is the nuclear norm (i.e., the sum of the singular values). Once obtaining the optimal Z^*, column vectors of Z^* are normalized by $z_i^* = z_i^*/\|z_i\|_2^*$ and the elements in each column vector are pruned by a predefined threshold θ, namely,

$$z_{ij}^* = \begin{cases} z_{ij}^*, & \text{if } z_{ij}^* \geq \theta \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Although we also use the reconstruction coefficient matrix to perform the semi-supervised clustering, our OSSPP is quite different from NNLRS in the following two aspects:

(1) In NNLRS, the reconstruction coefficient matrix is defined in the original feature space, which is unnecessary to be best for characterizing the similarity of sample pairs because some noisy features may exist in such original feature representation. In contrast, our OSSPP constructs the graph in the derived optimal low-dimensional feature space, which better characterizes the similarity than the graph built directly in the original feature space.

(2) For NNLRS, in order to obtain the optimal graph, we need to prune the learned affinity matrix, i.e., some elements of the affinity matrix should be set to 0 by a given threshold value. However, how to estimate the optimal threshold value is data set dependent. In our OSSPP, we directly use the reconstruction coefficient matrix as the graph to conduct semi-supervised subspace clustering without any pruning operation since the learned reconstruction coefficient is optimal.

4. Experiments and analysis

In this section, we apply OSSPP for dimensionality reduction and subspace clustering, along with showing our experimental results. Let (P^*, Z^*) be the solution of (6), then we use the obtained P^* and Z^* to perform dimensionality reduction and subspace clustering, respectively. All algorithms are implemented via Matlab 2010b.

4.1. Experiment settings

Our experiments are conducted on four public data sets: YaleB face image data set (Zhang & Yang, 2014), CMU PIE (PIE) face image

Table 1
Classification error rates of different algorithms with NN classifier under different number of training samples. The bold numbers are the best results (both mean error rate and optimal dimensionality).

Data set	PCA	LPP	NPE	SPP	LRPP	LS3C	OSSPP
USPS (10)	27.09 ± 1.73 (49)	27.00 ± 1.86 (20)	26.07 ± 2.30 (20)	19.30 ± 1.28 (47)	22.20 ± 1.43 (65)	17.50 ± 1.46 (48)	**15.59 ± 1.36 (48)**
USPS (20)	21.25 ± 1.78 (46)	19.60 ± 0.71 (26)	17.86 ± 2.05 (20)	13.22 ± 0.78 (39)	14.40 ± 1.53 (70)	12.67 ± 1.83 (44)	**10.60 ± 0.67 (47)**
USPS (30)	17.95 ± 0.78 (50)	16.13 ± 0.88 (32)	14.27 ± 0.68 (40)	11.27 ± 0.54 (38)	10.62 ± 1.68 (65)	10.10 ± 1.54 (46)	**8.76 ± 0.66 (48)**
COIL20 (3)	39.70 ± 2.19 (48)	–	–	22.67 ± 2.07 (20)	28.50 ± 2.00 (85)	**20.28 ± 1.84 (76)**	**20.68 ± 2.77 (83)**
COIL20 (5)	33.61 ± 1.69 (48)	38.62 ± 3.96 (24)	40.52 ± 1.93 (35)	**15.16 ± 1.83 (30)**	22.56 ± 1.75 (50)	17.34 ± 1.69 (30)	**15.30 ± 1.96 (30)**
COIL20 (7)	28.01 ± 1.78 (44)	25.52 ± 1.82 (37)	24.07 ± 2.19 (54)	12.23 ± 1.45 (58)	13.32 ± 1.78 (45)	12.50 ± 1.45 (43)	**11.89 ± 2.06 (28)**
YaleB (20)	35.42 ± 1.59 (318)	17.19 ± 0.34 (61)	16.65 ± 0.96 (61)	16.08 ± 0.68 (61)	18.20 ± 1.69 (55)	15.30 ± 1.11 (39)	**13.89 ± 1.12 (37)**
YaleB (30)	26.75 ± 1.56 (325)	14.44 ± 0.94 (61)	14.21 ± 0.93 (61)	12.51 ± 0.92 (61)	13.30 ± 1.80 (50)	11.39 ± 1.76 (36)	**9.59 ± 2.01 (30)**
YaleB (40)	21.42 ± 1.14 (450)	13.33 ± 0.81 (61)	12.56 ± 0.89 (61)	11.59 ± 1.08 (61)	10.45 ± 2.01 (30)	10.32 ± 1.25 (33)	**8.86 ± 0.95 (30)**
PIE (15)	30.52 ± 0.91 (410)	8.32 ± 0.74 (68)	5.83 ± 0.75 (63)	4.91 ± 0.43 (68)	3.86 ± 1.11 (65)	3.58 ± 0.82 (58)	**3.06 ± 0.31 (58)**
PIE (20)	25.45 ± 1.17 (290)	6.31 ± 0.65 (68)	4.03 ± 0.73 (55)	3.52 ± 0.31 (53)	3.30 ± 0.82 (60)	3.32 ± 0.67 (57)	**3.02 ± 0.34 (60)**
PIE (25)	22.73 ± 1.07 (280)	4.93 ± 0.59 (68)	3.34 ± 0.52 (60)	3.00 ± 0.46 (68)	3.15 ± 0.75 (65)	2.93 ± 0.64 (58)	**2.54 ± 0.42 (60)**

data set (Zhang & Yang, 2014), COIL-20 object image data set (Lai, Wong, Jin, Yang, & Xu, 2012) and USPS digit image data set (Nie, Wang, & Huang, 2014; Zheng, Bu, Chen, & Wang, 2011).

The YaleB data set: This data set has 38 individuals, each subject having around 64 near frontal images under different illuminations. The images are cropped and then resized to 32 × 32 pixels.

The PIE data set: In this experiment, we choose the face images from the frontal pose (C27) and each subject has around 49 images from varying illuminations and facial expressions. The images are cropped and then resized to 64 × 64 pixels.

The COIL20 data set: This data set consists of images of 20 objects, and each object has 72 images captured from varying angles at intervals of five degrees. All images in this data set are resized to 32 × 32 pixels.

The USPS data set: This handwritten digit data set contains 9298 handwritten digit images and each image is cropped and then resized to 16 × 16 pixels.

All images in these data sets use gray-level features to perform classification. For the sake of computational efficiency, PCA is used as a preprocessing step to preserve 98% energy of data for the USPS data set, and 95% energy of data for the YaleB, COIL20 and PIE face data sets.

4.2. Dimensionality reduction

We directly apply P^* to map both of the training samples and test samples into the desired low-dimensional subspace and use the NN classifier (based on Euclidean distance) to perform classification since there is no need of parameter tuning. For each data set, we randomly select different training samples from per subject for training and rest for testing. All experiments are run 10 times (unless otherwise stated) and then the mean classification accuracy and standard deviation are reported.

We compare OSSPP with some popular unsupervised dimensionality reduction methods such as PCA, LPP (Niyogi, 2014), NPE (He et al., 2005), SPP (Qiao et al., 2010), low-rank preserving projection (LRPP) (Lu et al., 2016) and LS3C (Patel et al., 2013). (We use the learned projection matrix P to perform DR.) For LPP, we select its model parameters, i.e., neighborhood size k and kernel width t by searching them in a large range of candidates and report the best classification results. This strategy is also used to determine the neighborhood size k in NPE. For SPP and LS3C, we adopt the Thresholding Algorithm to solve the ℓ_1-minimization optimization problem. For OSSPP, we select parameters α and β from {0.001, 0.004, 0.007, 0.01, 0.04, 0.07, 0.1, 0.4} and {0.001, 0.002, 0.003, . . . , 0.009, 0.01}, respectively. Table 1 shows classification results on these data sets. Note that the numbers in parentheses (after each data set) are the number of training samples selected from each class of data set and the numbers in parentheses (below the experiments results) are the optimal dimensionalities after dimensionality reduction. From Table 1, we have the following observations:

(1) PCA generally gets much worse performance than LPP, NPE, SPP and OSSPP. Only on the COIL20 data set, its classification performance is better than LPP and NPE when only three and five images per class are selected for training. Moreover, we find that when three images per class are selected for training on the COIL20 data set, LPP and NPE do not work in such small number of training samples.
(2) LPP and NPE generally outperform PCA with lower dimensions. This indicates that by preserving the local structure of the data, the classification accuracy can be improved. That is, when NN classifier (nearest neighbor search) is used, the local structure seems to be important than global structure.
(3) OSSPP consistently outperforms all the compared methods when we use the NN classifier. This suggests that the projection matrix learned by OSSPP contains more discriminative information than that learned by compared methods, which is benefit from the weighted non-negative sparse coding for the solution of reconstruction coefficients matrix Z. In SPP, the sparse reconstruction coefficient matrix is firstly calculated and then the dimensionality reduction is performed. Such two independent steps cannot guarantee an overall optimum. In contrast, our OSSPP performs the reconstruction coefficients learning and dimensionality reduction in a single optimization step which can guarantee an overall optimum. Thus, OSSPP obtains the best classification results.
(4) OSSPP outperforms LS3C in most of cases. The main reason is that the learned sparse reconstruction coefficients are influenced by the dimension reduction, and vice versa. The learning process can hopefully boost the performance of each task. Therefore, OSSPP obtains the better classification performance. Although LS3C also uses the simultaneous learning strategy, the low-dimensional embedding of data is not explicitly required to preserve the similarity. So the improvement of classification performance of LS3C is not very obvious.
(5) We also note that when we select few samples as training samples, the optimal dimensionality in general is higher. For example, on the COIL20 data set, the value of dimension is 83 in the first case, 50 higher than the dimension in the second case. The reason may be that in the case of few training samples the similarity is not preserved well.
(6) The low-rank preserving projection can well capture the global subspaces structure of data and the sparsity preserving projection can well capture the local similarity structure of data (Zhuang et al., 2012). From the experiments in Table 1, we can see that SPP and LRPP alternately beat each other on the

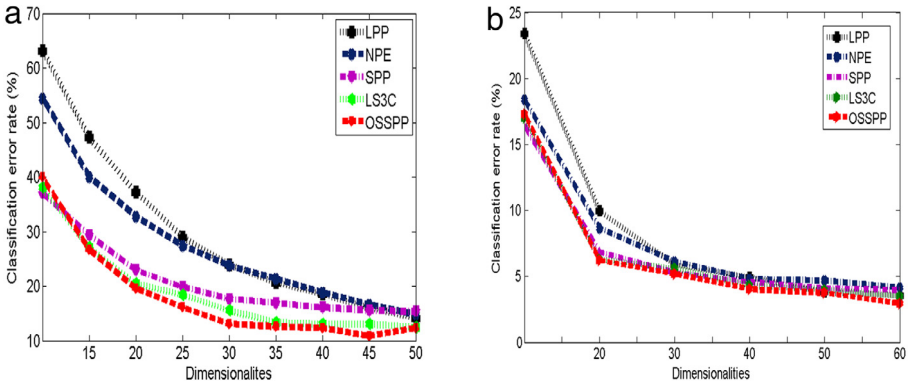

Fig. 2. Classification error rates (%) of different algorithms with NN classifier versus dimensionalities on the (a) YaleB and (b) PIE face data sets. For the YaleB and PIE data sets, we randomly select 30 and 20 samples per subject for training and use the remaining for testing, respectively.

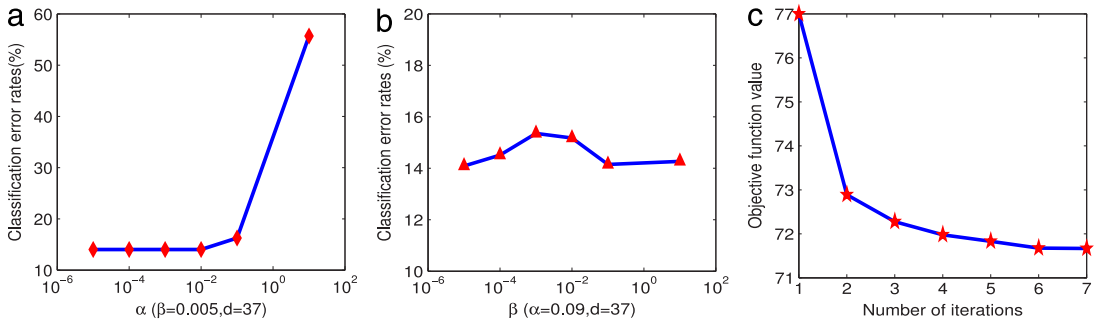

Fig. 3. Parameters sensitivity and convergence: (a) and (b) show the performance of OSSPP vs. the parameters α and β, respectively. (c) shows the convergence curve of OSSPP. We randomly select 20 images per subject for training and use the remaining for testing on the YaleB data set.

four data sets. Thus, we argue that the important of the global subspace structure and local similarity structure of data is data set dependent.

Classification error rates versus dimensionalities on the YaleB and PIE data sets are shown in Fig. 2. We compare the dimensionalities up to 50 and 60 for the YaleB and PIE data sets, respectively. Again, OSSPP performs better than the other methods in almost of dimensionalities.

We examine the parameter sensitivity of OSSPP to classification error rate. OSSPP requires two parameters α and β to be set in advance. α is used to hold the main energy of data, while β is to ensure the similarity preserving on the projection. Fig. 3(a) and (b) show the parameters sensitivity of OSSPP. From Fig. 3(a) and (b), we can see that the performance of OSSPP is robust to the parameter α when $\alpha \leq 10^{-2}$. Moreover, OSSPP is not sensitive to the parameter β in the given wide range (see Fig. 3(b)). In practice, we first fix α due to its more stronger robustness than β and then select the optimal value of β from the given set.

Fig. 3(c) shows that the objective function value decreases very fast. After only about 6–7 iterations, the objective value converges, which suggests that our iterative update rule is very effective.

4.3. Semi-supervised subspace clustering

All semi-supervised subspace clustering experiments are conducted on the YaleB (Lai, Xu, Chen, Yang, & Zhang, 2014; Zhang & Yang, 2014), COIL20 (Lai et al., 2012, 2015) and PIE (Zhang & Yang, 2014) data sets. For the sake of computational efficiency, for the YaleB and PIE data sets, we only use the first 20 persons in the YaleB data set and first 40 persons in the PIE data set. For each data set, we randomly select different samples per subject as labeled samples and use the rest as unlabeled samples and all experiments are run 10 times and then the mean classification error rate (%) is reported.

We carry out the semi-supervised clustering experiments on the derived graph weight matrix W using the existing G-SSL method, Gaussian field and harmonic function (GFHF) (Deng, Choi, Jiang, Wang, & Wang, 2016; Fang, Xu, Li, Lai, & Wong, 2015; Zhu, Ghahramani, & Lafferty, 2013).

We denote the sample set as $X = [x_1, \ldots, x_u, \ldots, x_n] \in \Re^{m\times n}$, where $x_i|_{i=1}^{u}$ and $x_i|_{u+1}^{n}$ are labeled and unlabeled data, respectively. We define a binary label matrix $Y \in \Re^{n\times c}$ (c is the total number of classes) with $Y_{ij} = 1$ if x_i has label $y_i = j (j = 1, 2, \ldots, c)$; $Y_{ij} = 0$, otherwise. GFHF estimates prediction labels matrix $F \in \Re^{n\times c}$ by minimizing the following objective function

$$\min_F \sum_{i,j=1}^{n} \|F_i - F_j\|^2 W_{ij} + \lambda_\infty \sum_{i=1}^{u} \|F_i - Y_i\|^2 \tag{25}$$

where F_i and Y_i are the ith rows of F and Y, respectively, and λ_∞ is very large number such that $\sum_{i=1}^{u} \|F_i - Y_i\|^2 = 0$, or $F_i = Y_i, \forall i = 1, 2, \ldots, u$.

The graphs used in our experiments for comparison include:

k **NN-graph**: We set Gaussian kernel parameter σ as 1. There are two configurations for constructing graphs, denoted as k **NN5** and k **NN8**, where the numbers of nearest neighbors are set to 5 and 8, respectively.

LLE-graph (Roweis & Saul, 2010): We construct LLE-graphs, denoted as **LLE8** and **LLE10**, where the numbers of nearest neighbors are 8 and 10, respectively.

ℓ_1**-graph**: Following the lines of (Yan & Wang, 2009), we construct the ℓ_1-graph.

SPG-graph (He et al., 2011): In He et al. (2011), the **SPG** is essentially a lasso problem. We construct the **SPG**-graph following the lines of He et al. (2011).

LS3C-graph (Patel et al., 2013): The matrix Z^* produced by (5) is firstly converted into $|Z^*|$, where $|\cdot|$ is absolute value of a matrix,

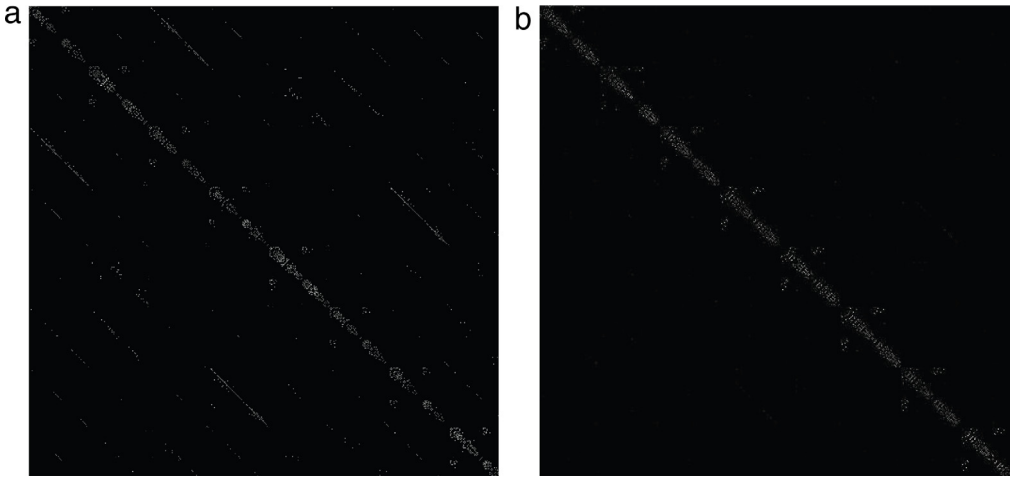

Fig. 4. Visualization of graph weight matrices of (a) SPG-graph, (b) OSSPP-graph on the YaleB data set, in which we randomly select 25 images per subject as labeled samples and use the remaining as unlabeled samples.

Table 2
Classification error rates (%) of GFHF with different graphs under different number of the labeled samples. The bold numbers are the lowest error rates.

Data set	*k*NN5	*k*NN8	LLE8	LLE10	ℓ_1-graph	SPG	LS3C	NNLRS	OSSPP
YaleB (15)	45.84	50.83	36.80	37.84	41.68	**12.17**	**12.20**	**12.28**	**12.38** (**32**)
YaleB (25)	41.73	44.62	30.05	30.05	32.29	8.26	9.31	8.89	**7.35** (**38**)
YaleB (35)	35.76	37.37	25.27	23.84	25.09	8.18	8.00	7.74	**5.87** (**36**)
YaleB (45)	29.83	33.15	20.99	17.13	16.02	6.35	3.86	2.77	**2.48** (**36**)
COIL20 (5)	9.03	13.21	9.02	11.87	5.67	20.97	18.13	6.28	**4.56** (**29**)
COIL20 (10)	3.87	8.47	4.11	4.83	3.39	10.48	8.14	4.92	**1.93** (**34**)
COIL20 (15)	3.42	6.05	2.89	3.59	2.46	6.58	6.40	2.56	**0.88** (**28**)
COIL20 (20)	2.11	5.09	1.45	1.63	2.38	3.85	3.17	1.45	**0.37** (**28**)
PIE (5)	36.54	43.99	39.04	41.21	29.08	**12.98**	27.71	13.58	13.49 (61)
PIE (10)	27.49	35.07	26.84	28.96	19.91	9.31	13.87	**8.89**	9.77 (62)
PIE (15)	19.16	26.31	19.15	20.56	14.30	5.16	5.31	5.72	**5.05** (**63**)
PIE (20)	18.67	25.41	16.16	16.50	12.96	5.09	4.49	4.54	**4.33** (**64**)

and then $|Z^*|$ is used to construct the graph weight matrix W according to (14).

NNLRS-graph (Zhuang et al., 2012): In Zhuang et al. (2012), the NNLRS is essentially the problem of solving a constraint low-rankness and sparsity minimization objective. We construct the NNLRS-graph following the lines of Zhuang et al. (2012).

OSSPP-graph: Reconstruction coefficients matrix Z in (7) is used to construct graph weight matrix W according to (22).

Since the problem of SPG is similar to that of OSSPP (two non-negative sparse graph learning methods). Thus, we give the visualization of graph weight matrices of SPG and OSSPP in Fig. 4. From this figure, two observations can be made: (1) The edges in OSSPP-graph are sparse than SPG-graph; (2) There is much less inter-subject adjacency structure in OSSPP-graph than in SPG-graph, which means the OSSPP-graph delivers strong discriminant information and thus is more effective for label propagation than SPG-graph.

The mean classification error rates of different graphs are shown in Table 2. Note that numbers in parentheses (after each data set) are the number of labeled samples selected from each class of data set and numbers in parentheses (after the experiments results of OSSPP) are the optimal dimensionalities after dimensionality reduction. From this table, the following conclusions can be drawn:

(1) In most cases, OSSPP consistently achieves the lowest mean classification error rates in comparison with other graphs. In many cases, the improvements are rather obvious. This indicates that the graph produced by OSSPP is more informative and suitable for label propagation. When the number of labeled samples is small, the performance of OSSPP slightly decreases on the YaleB and PIE data sets.

(2) The goals of LS3C and NNLRS are somewhat similar to OSSPP in subspace clustering. However, in LS3C, the projected data are not required to respect the similarity structure during the procedure of dimensionality reduction so that the reconstruction coefficient matrix does not effectively capture the similarity of data. Thus the label information cannot be accurately propagated. In NNLRS, the reconstruction coefficient matrix is defined in the original high-dimensional feature space which is also unnecessary to be best for characterizing the similarity of data. OSSPP addresses these problems by solving the objective (6) and thus the OSSPP outperforms them in most cases.

5. Conclusion

This paper proposes a novel dimensionality reduction method, called orthogonal self-guided similarity preserving projection (OSSPP) for dimensionality reduction and semi-supervised subspace clustering. The core idea of OSSPP is that OSSPP simultaneously obtains the feature representation and intrinsic similarity structure of data. In this way, these two tasks can be mutually improved and we eventually can determine the most suitable feature representation and capture the accurate similarity structure. Extensive experiments on both DR and semi-supervised subspace clustering show the effectiveness of OSSPP.

Acknowledgments

This paper is partially supported by the National Basic Research Program of China (973 Program) (Grant No. 2012CB316400), the National Natural Science Foundation of China (Grant Nos. 61370163, and 61332011).

References

Deng, Z. H., Choi, K. S., Jiang, Y. Z., Wang, J., & Wang, S. T. (2016). A survey on soft subspace clustering. *Information Sciences*, *348*(20), 84–106.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2765–2781.

Fang, X. Z., Xu, Y., Li, X. L., Fan, Z. Z., Liu, H., & Chen, Y. (2014). Locality and similarity preserving embedding for feature selection. *Neurocomputing*, *128*, 304–315.

Fang, X. Z., Xu, Y., Li, X. L., Lai, Z. H., & Wong, W. K. (2015). Learning a nonnegative sparse graph for linear regression. *IEEE Transactions on Image Processing*, *24*(9), 2760–2771.

Fang, X. Z., Xu, Y., Zhang, Z., Lai, Z. H., & Shen, L. L. (2015). Orthogonal self-guided similarity preserving projections. 2015, ICIP.

Fan, Z., Xu, Y., Zuo, W., Tan, J., Lai, Z., & Zhang, D. (2014). Modified principal component analysis: An integration of multiple similarity subspace models. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(8), 1538–1552.

He, X., Cai, D., Yan, S., & Zhang, H. (2005). Neighborhood preserving embedding. 2015, ICCV, 2 (pp. 1208–1213).

He, R., Zheng, W., Hu, B., & Kong, W. (2011). Non-negative sparse coding for discriminative semi-supervised learning. 2011, CVPR (pp. 2849–2856).

Huang, D. S. (1999). Radial basis probabilistic neural networks: Model and application. *International Journal of Pattern Recognition and Artificial Intelligence*, *13*(7), 1083–1101.

Huang, D. S., & Du, Ji-Xiang (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. *IEEE Transactions on Neural Networks*, *19*(12), 2099–2115.

Huang, D. S., & Jiang, Wen (2012). A general CPL-AdS methodology for fixing dynamic parameters in dual environments. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(5), 1489–1500.

Jing-Yan Wang, J., & Gao, X. (2015). Max-min distance nonnegative matrix factorization. *Neural Networks*, *61*, 75–84.

Lai, Z. H., Wong, W. K., Jin, Z., Yang, J., & Xu, Y. (2012). Sparse approximation to the eigensubspace for discrimination. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(12), 1948–1960.

Lai, Z. H., Wong, W. K., Xu, Y., Yang, J., Tang, J. H., & Zhang, D. (2015). Approximate orthogonal sparse embedding for dimensionality reduction. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(4), 723–735.

Lai, Z. H., Xu, Y., Chen, Q. C., Yang, J., & Zhang, D. (2014). Multilinear sparse principal component analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(10), 1942–1950.

Lai, Z. H., Xu, Y., Yang, J., Tang, J., & Zhang, D. (2013). Sparse tensor discriminant analysis. *IEEE Transactions on Image Processing*, *22*(10), 3904–3915.

Lemme, A., Reinhart, R. F., & Steil, J. J. (2012). Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. *Neural Networks*, *33*, 194–203.

Li, X., Pang, Y., & Yuan, Y. (2010). L1-norm-based 2dpca. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *40*(4), 1170–1175.

Liu, G., Lin, Z., & Yu, Y. (2010). Robust subspace segmentation by low-rank representation. 2010, ICML (pp. 663–670).

Liu, W. F., Liu, H. L., Tao, D. P., Wang, Y. J., & Lu, K. (2015). Multiview Hessian regularized logistic regression for action recognition. *Signal Processing*, *110*, 101–107.

Lu, Y., Lai, Z., Xu, Y., Li, X., Zhang, D., & Yuan, C. (2016). Low-rank preserving projections. *IEEE Transactions on Cybernetics*, *46*(8), 1900–1913.

Mardani, M., Mateos, G., & Giannakis, G. B. (2015). Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing*, *63*(10), 2663–2677.

Mikhail, B., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. 2001, NIPS, 14 (pp. 585–591).

Nie, F. P., Wang, X. Q., & Huang, H. (2014). Clustering and projected clustering with adaptive neighbors. 2014, KDD (pp. 977–986).

Niyogi, X. (2014). Locality preserving projections. *IEEE Transactions on Cybernetics*, *44*(10), 1738–1746.

Patel, V., Nguyen, H., & Vidal, R. (2013). Latent space sparse subspace clustering. 2013, ICCV (pp. 225–232).

Qiao, L., Chen, S., & Tan, X. (2010). Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, *43*(1), 331–341.

Roweis, S., & Saul, L. (2010). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Tao, D. P., Jin, L. W., Liu, W. F., & Li, X. L. (2013). Hessian regularized support vector machines for mobile image annotation on the cloud. *IEEE Transactions on Multimedia*, *15*(4), 833–844.

Wang, X. F., & Huang, D. S. (2009). A novel density-based clustering framework by using level set method. *IEEE Transactions on Knowledge and Data Engineering*, *21*(11), 1515–1531.

Wang, X. F., Huang, D. S., & Xu, H. (2010). An efficient local Chan-Vese model for image segmentation. *Pattern Recognition*, *43*(3), 603–618.

Wang, R., Nie, F. P., Yang, X. J., Gao, F. F., & Yao, M. L. (2015). Robust 2DPCA with non-greedy L1-norm maximization for image analysis. *IEEE Transactions on Cybernetics*, *45*(5), 1108–1112.

Wu, M., Yu, K., Yu, S., & Scholkopf, B. (2007). Local learning projections. 2007, ICML, 16 (pp. 1039–1046).

Yan, S., & Wang, H. (2009). Semi-supervised learning by sparse representation. 2009, SDM (pp. 792–801).

Yang, J., Chou, D., Zhang, L., Xu, Y., & Yan, J. Y. (2013). Sparse representation classifier steered discriminative projection with applications to face recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(7), 1023–1035.

Yang, J., & Zhang, Y. (2011). Alternating direction algorithms for l1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, *33*(1), 250–278.

Zhang, J., & Yang, J. (2014). Linear reconstruction measure steered nearest neighbor classification framework. *Pattern Recognition*, *47*, 1709–1720.

Zhang, L., Zhou, W., & Chang, P. (2012). Kernel sparse representation-based classifier. *IEEE Transactions on Signal Processing*, *60*(4), 1684–1695.

Zheng, M., Bu, J., Chen, C., & Wang, C. (2011). Graph regularized sparse coding for image representation. *IEEE Transactions on Image Processing*, *20*(5), 1327–1336.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2013). Semi-supervised learning using gaussian fields and harmonic functions. 2013, ICML (pp. 912–919).

Zhuang, L., Gao, H., Lin, Z., Ma, Y., Zhang, X., & Yu, N. (2012). Non-negative low rank and sparse graph for semisupervised learning. 2012, CVPR (pp. 2328–2335).