

Subspace clustering using a symmetric low-rank representation[☆]



Jie Chen, Hua Mao, Yongsheng Sang, Zhang Yi*

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

ARTICLE INFO

Article history:

Received 18 November 2016

Revised 26 February 2017

Accepted 28 February 2017

Available online 1 March 2017

Keywords:

Low-rank representation

Subspace clustering

Affinity matrix learning

Spectral clustering

ABSTRACT

In this paper, we propose a low-rank representation with symmetric constraint (LRRSC) method for robust subspace clustering. Given a collection of data points approximately drawn from multiple subspaces, the proposed technique can simultaneously recover the dimension and members of each subspace. LRRSC extends the original low-rank representation algorithm by integrating a symmetric constraint into the low-rankness property of high-dimensional data representation. The symmetric low-rank representation, which preserves the subspace structures of high-dimensional data, guarantees weight consistency for each pair of data points so that highly correlated data points of subspaces are represented together. Moreover, it can be efficiently calculated by solving a convex optimization problem. We provide a proof for minimizing the nuclear-norm regularized least square problem with a symmetric constraint. The affinity matrix for spectral clustering can be obtained by further exploiting the angular information of the principal directions of the symmetric low-rank representation. This is a critical step towards evaluating the memberships between data points. Besides, we also develop eLRRSC algorithm to improve the scalability of the original LRRSC by considering its closed form solution. Experimental results on benchmark databases demonstrate the effectiveness and robustness of LRRSC and its variant compared with several state-of-the-art subspace clustering algorithms.

© 2017 Published by Elsevier B.V.

1. Introduction

In recent years, subspace clustering techniques have attracted much attention from researchers in many areas; for example, computer vision, machine learning, and pattern recognition. Subspace clustering is important to numerous applications such as image representation [1,2], clustering [3–7], and motion segmentation [8–12]. The generality and importance of subspaces naturally lead to the challenging problem of subspace clustering, where the goal is to simultaneously segment data into clusters that correspond to a low-dimensional subspace. To be more specific, given a set of data points drawn from a mixture of subspaces, the task is to segment all data points into their respective subspaces.

When we consider subspace clustering in real applications, there is a large amount of high-dimensional data available; for example, digital images, video surveillance, and traffic monitoring. High-dimension data increase the computational cost of algorithms and have a negative effect on performance because of noise and corrupted observations. It is typical to impose an assumption that the high-dimensional data are approximately drawn from a union

of multiple subspaces. This assumption is reasonable because data in a class can be well represented by a low-dimensional subspace of the high-dimensional ambient space. In fact, high-dimensional data often have a smaller intrinsic dimension. Examples of high-dimensional data that lie in a low-dimensional subspace of the ambient space include images of an individual's face captured under various laboratory-controlled lighting conditions, handwritten images of a digit with different rotations, translations, and thicknesses, and feature trajectories of a moving object in a video [13–15]. This has motivated the development of a number of techniques [16–20] for finding and exploiting low-dimensional structures in high-dimensional data.

A number of methods have been devised to exactly solve the subspace clustering problem [21]. Subspace clustering methods can be roughly divided into statistical learning based [22], factorization based [23,24], algebra based [25], iterative [26–28], and spectral-type based methods [4,5,12,29,30]. These methods can produce good results under the assumption that data are strictly drawn from linearly independent subspaces. However, an increase in difficulty is in part caused by the data not strictly following subspace structures (because of noise and corruptions), and can lead to indistinguishable subspace clustering. Therefore, subspace clustering algorithms that take into account the multiple subspace structure of high-dimensional data are required.

[☆] This work was supported by the National Science Foundation of China (Grant No. 61432012, U1435213, 61402306 and 61602329).

* Corresponding author:

E-mail address: zyiscu@gmail.com (Z. Yi).

Recently, some work on the Frobenius norm, sparse representation theory and rank minimization [4,5,29,31–39] have recently been proposed to alleviate some of aforementioned drawbacks. Effective approaches to subspace clustering called spectral-type based methods have been developed. These methods typically perform subspace clustering in two stages: first learn an affinity matrix (an undirected graph) from the given data, and then obtain the final clustering results using a spectral clustering algorithm [40] such as normalized cuts (NCuts) [41]. These techniques can effectively recover the multiple subspace structures when high-dimensional data is grossly corrupt. However, there are still several open questions, e.g., how to choose a good affinity matrix by building a similarity graph, and how to estimate the number and dimensions of underlying subspaces in a time efficient manner. For a given data matrix $X = [x_1, x_2, \dots, x_N] \in \mathbf{R}^{d \times N}$, each of which can be represented by the combination of the basis in the dictionary $A = [a_1, a_2, \dots, a_n] \in \mathbf{R}^{d \times n}$:

$$X = AZ, \quad (1)$$

where it refers to Z as a coefficient matrix of linear representation. Some techniques based on Frobenius norm are developed to construct L2-Graph for subspace clustering, which require low computational cost [36,37,42]. Besides, Elhamifar and Vidal [5] proposed a sparse subspace clustering (SSC) method, which uses the sparsest representation of the data points produced by l_1 -norm minimization of the coefficient matrix of linear representation to define an affinity matrix of the undirected graph. Liu et al. [35] proposed an efficient distributed framework for the computation of SSC on a shared-memory architecture. Then spectral clustering techniques are used to perform subspace clustering. The convex optimization model of SSC, under the assumption that the subspaces are either linearly independent or disjoint under certain conditions, results in a block diagonal solution. However, there is no global structural constraint on the sparse representation. Moreover, SSC needs to carefully tune the number of nearest neighbors to improve performance, which leads to another parameter.

Liu et al. [4] recently introduced low-rank representation techniques into the subspace clustering problem and established a low-rank representation (LRR) algorithm [4]. LRR assumes that the data samples are approximately drawn from a mixture of multiple low-rank subspaces. The goal of LRR is to take the correlation structure of data into account, and find the lowest-rank representation of Z using an appropriate dictionary. LRR solves the convex optimization problem of nuclear-norm minimization, which is considered as a surrogate for rank minimization. It performs subspace clustering excellently. However, the affinity for the spectral clustering input, which can be computed using a symmetrization step of the low-rank representation results [2], is not good at characterizing how other samples contribute to the reconstruction of a given sample. In addition, the motivation of a new version of LRR to use the matrix U^*U^{*T} as an affinity for the spectral clustering input remains vague without theoretical analysis [4]. Here, U^* can be found using the skinny SVD of the low-rank representation Z , i.e., $Z = U^*\Sigma^*V^{*T}$. Ni et al. [29] proposed a robust low-rank subspace segmentation method to extend LRR and improve performance. It enforces the symmetric positive semi-definite (PSD) constraint on Z to explicitly obtain a symmetric PSD matrix and avoid the symmetrization post-processing. LRR and its variations are guaranteed to produce block-diagonal affinity matrices if the points are already ordered according to their respective clusters under the independence assumption. However, nonnegativity of the values of the PSD matrix cannot be guaranteed by PSD conditions. Consequently, negative elements of the PSD matrix lack physical interpretation for visual data. Besides, if the low-rank representation matrix is considered as a pairwise affinity relationship for spectral clustering between data points, this inevitably leads to loss of in-

formation. This means that it does not fully capture the complexity of the problem, i.e., the intrinsic correlation of data points [43,44]. To tackle this difficulty, one needs to learn an affinity matrix by exploiting the structure of the low-rank representation.

In this paper, we present a low-rank representation with symmetric constraint (LRRSC) method and its variant (eLRRSC) for robust subspace clustering. In particular, our motivation is to integrate the symmetric constraint into the low-rankness property of high-dimensional data representation, to learn a symmetric low-rank matrix that preserves the subspace structures of high-dimensional data. By solving the nuclear-norm minimization problem of Z in a simple and efficient way, we can learn a symmetric low-rank matrix. For example, given a set of data points, we represent each point as a linear combination of the others, where the low-rank coefficients should be symmetric. Low-rank representation techniques often suffer from heavy computational cost when require iterative SVD operations. In contrast with these techniques, eLRRSC can obtain a symmetric low-rank representation in a closed form solution, which dramatically reduces the computational complexity. To obtain the closed form solution, we provide a proof to minimize the nuclear-norm regularized least square problem with a symmetric constraint. Consequently, eLRRSC can be adopted by large-scale subspace clustering problems because of its advantages of computational stability and efficiency. Besides, the symmetric effect of LRRSC and eLRRSC guarantees weight consistency for each pair of data points, so that highly correlated data points are represented together. As mentioned above, using a symmetric matrix as the input for subspace clustering may negatively affect the performance. To overcome this drawback, it is critical to investigate the intrinsically geometrical structure of the memberships of data points preserved in a symmetric low-rank representation, i.e., the angular information of the principal directions of the symmetric low-rank representation. This can improve the subspace clustering performance. An affinity that encodes the memberships of subspaces can be constructed using the angular information of the normalized rows of U^* , or columns of V^{*T} , obtained from the skinny SVD of the symmetric matrix Z , i.e., $Z = U^*\Sigma^*V^{*T}$. With the learned affinity matrix, spectral clustering can segment the data into clusters with the underlying subspaces they are drawn from. The proposed algorithm not only recovers the dimensions of each subspace, but also effectively learns a symmetric low-rank representation for the purpose of subspace clustering. In contrast to LRR, LRRSC and eLRRSC can obtain a coefficient matrix symmetric, and then they build a desired affinity for subspace clustering. Further details will be discussed in Section 3.

The contributions of the paper are summarized as follows:

- a) It incorporates the symmetry idea into low-rank representation learning, and can successfully learn a symmetric low-rank matrix for high-dimensional data representation. We have provided a proof for minimizing the nuclear-norm regularized least square problem with a symmetric constraint.
- b) A symmetric low-rank representation can be obtained by eLRRSC in a closed form solution, which can be solved very efficiently using SVD techniques.
- c) It exploits the intrinsically geometrical structure of the memberships of data points preserved in a symmetric low-rank matrix (i.e., the angular information of principal directions of the symmetric low-rank representation) to construct an affinity matrix with more separation ability. This significantly improves the subspace clustering performance.
- d) Compared with other state-of-the-art methods, our extensive experimental results using benchmark databases demonstrate the effectiveness and robustness of LRRSC and eLRRSC for subspace clustering.

The remainder of this paper is organized as follows. Section 2 summarizes some related work on low-rank representation techniques that inspired this work. Section 3 presents the proposed LRRSC for subspace clustering. Extensive experimental results using benchmark databases are presented in Section 4. Finally, Section 6 concludes this paper.

2. A review of previous work

Consider a set of data vectors $X = [x_1, x_2, \dots, x_n] \in \mathbf{R}^{d \times n}$, each column of which is drawn from a union of k subspaces $\{S_i\}_{i=1}^k$ with unknown dimensions. The goal of subspace clustering is to cluster data points into their respective subspaces. A main challenge in applying spectral clustering to subspace clustering is to define a good affinity matrix, each entry of which measures the similarity between data points x_i and x_j . This section provides a review of low-rank representation techniques for solving subspace clustering problems that are closely related to the proposed method.

2.1. Subspace clustering by low-rank representation

Recently, Liu et al. [4] proposed a novel objective function named the low-rank representation method for subspace clustering. Instead of seeking a sparse representation as in SSC, LRR seeks the lowest-rank representation among all the candidates that can represent the data samples as linear combinations of the bases in a given dictionary. SSC enforces that Z is sparse by imposing an l_1 -norm regularization on Z , while LRR encourages Z to be low-rank using nuclear-norm regularization. By using the nuclear norm as a good surrogate for the rank function, LRR solves the following nuclear norm minimization problem for the noise free case:

$$\min_Z \|Z\|_* \quad \text{s.t.} \quad X = AZ, \quad (2)$$

where $A = [a_1, a_2, \dots, a_n] \in \mathbf{R}^{d \times n}$ is an overcomplete dictionary, and $\|\cdot\|_*$ denotes the nuclear norm (the sum of the singular values of the matrix). When the subspaces are independent, LRR succeeds in recovering the desired low-rank representations.

In the case of data being grossly corrupted by noise or outliers, the LRR algorithm solves the following convex optimization problem:

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_l \quad \text{s.t.} \quad X = AZ + E, \quad (3)$$

where $\|\cdot\|_l$ indicates a certain regularization strategy for characterizing various corruptions. For example, the $l_{2,1}$ -norm encourages the columns of E to be zero. By choosing an appropriate dictionary A , LRR seeks the lowest-rank representation matrix of the coefficient matrix Z . This can also be used to recover the clean data from the original samples. To reduce the computational complexity, LRR uses XP^* as its dictionary, where P^* can be computed by orthogonalizing the columns of X^T .

The above optimization problem is convex, and can be efficiently solved by the inexact augmented Lagrange multipliers (ALM) technique in polynomial time with a guaranteed high performance [20]. After obtaining an optimal solution (Z^*, E^*) , Z^* is used to define an affinity matrix $|Z| + |Z|^T$ for spectral clustering.

To avoid symmetrization post-processing, Ni et al. [29] presented an improved LRR model with PSD constraints, which can be formulated as

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_l \quad \text{s.t.} \quad X = XZ + E, Z \succeq 0. \quad (4)$$

Rigorous mathematical derivations [29] show that the LRR-PSD model is the perfect case of the LRR scheme, and establishes the uniqueness of the optimal solution. By applying a scheme similar to LRR, LRR-PSD can also be efficiently solved. Then, Z^* is used to define an affinity matrix $|Z|$, after acquiring an optimal solution

(Z^*, E^*) . To reduce computational cost in the LRR scheme, Chen et al. [45] introduced the symmetric low-rank representation (SLRR) method to avoid iterative SVD operation. This significantly decrease the computational cost for the subspace clustering.

3. Exploitation of a low-rank representation with a symmetric constraint

In this section, we propose a low-rank representation with a symmetric constraint (LRRSC). Our approach takes into consideration the intrinsically geometrical structure of the symmetric low-rank representation of high-dimensional data. We first propose a low-rank representation model with a symmetric constraint, which can be efficiently calculated by solving a convex optimization problem. Then, we learn an affinity matrix for subspace clustering by exploiting the intrinsically geometrical structure of the memberships of data points preserved in the symmetric low-rank representation (i.e., the angular information of the principal directions of the symmetric low-rank representation). Finally, we discuss the convergence properties and present a computational complexity analysis of LRRSC.

3.1. Low-rank representation with symmetric constraint

When there are no errors in data X (i.e., the data are strictly drawn from k independent subspaces, and already ordered according to their respective clusters) the row space of the data, denoted by V^*V^{*T} [46] (also known as the shape interaction matrix [23]), is a block diagonal matrix that has exactly k blocks. The row space of X can be used as affinity for subspace clustering, and produces good results. It can be calculated using a closed form by computing the skinny SVD of the data matrix X , i.e., $X = U^* \Sigma^* V^{*T}$. However, real observations are often noisy. Grossly corrupted observations may reduce the performance. Low-rank representation techniques can be used to alleviate these problems [4,29].

To guarantee weight consistency for each pair of data points, we impose a symmetric constraint on the low-rank representation. Incorporating low-rank representation with a symmetric constraint in Problem (3), we consider the following convex optimization problem to seek a symmetric low-rank representation Z :

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1} \quad \text{s.t.} \quad X = AZ + E, Z = Z^T, \quad (5)$$

where the parameter $\lambda > 0$ is used as a trade-off between low-rankness and the effect of noise. The low-rankness criterion effectively captures the global structure of X . The low-rank constraint guarantees that the coefficients of samples coming from the same subspace are highly correlated. We assume that corruptions are “sample-specific”, i.e., some data vectors are corrupted and others are not. The $l_{2,1}$ -norm is used to characterize the error term E , because it encourages the columns of E to be zero. For small Gaussian noise, $\|E\|_F^2$ is an appropriate choice. Each coefficient pair (z_{ij}, z_{ji}) denotes the interaction between data points x_i and x_j . The symmetric constraint criterion is incorporated into the low-rankness property of high-dimensional data representation so that it can effectively ensure the weight consistency for each pair of data points. Consequently, the symmetric low-rank representation, which preserves the subspace structures of high-dimensional data, ensures that highly correlated data points of subspaces are represented together.

To make the objective function in Problem (5) separable, we first convert it to the following equivalent problem by introducing an auxiliary variable J :

$$\min_{Z, E, J} \|J\|_* + \lambda \|E\|_{2,1} \quad \text{s.t.} \quad X = AZ + E, Z = J, J = J^T. \quad (6)$$

The augmented Lagrangian function of Problem (6) is

$$\min_{Z, E, J=Y_1^T, Y_1, Y_2} \|J\|_* + \lambda \|E\|_{2,1} + \text{tr}[Y_1^T(X - AZ - E)] + \text{tr}[Y_2^T(Z - J)] + \frac{\mu}{2} (\|X - AZ - E\|_F^2 + \|Z - J\|_F^2), \quad (7)$$

where Y_1 and Y_2 are Lagrange multipliers, and $\mu > 0$ is a penalty parameter. The above optimization problem can be formulated as follows:

$$\min_{Z, E, J=Y_1^T, Y_1, Y_2} \|J\|_* + \lambda \|E\|_{2,1} + \frac{\mu}{2} \left(\left\| X - AZ - E + \frac{Y_1}{\mu} \right\|_F^2 + \left\| Z - J + \frac{Y_2}{\mu} \right\|_F^2 \right). \quad (8)$$

This can be effectively solved by inexact ALM [20]. The variables J , Z and E can be updated alternately at each step, while the other two variables are fixed. The updating schemes at each iteration are:

$$\begin{aligned} J_{k+1} &= \arg \min_{J=Y_1^T} \frac{1}{\mu} \|J\|_* + \frac{1}{2} \left\| J - \left(Z + \frac{Y_2}{\mu} \right) \right\|_F^2, \\ Z_{k+1} &= (I + A^T A)^{-1} \left(A^T X - A^T E + J + \frac{A^T Y_1 - Y_2}{\mu} \right), \\ E_{k+1} &= \arg \min E \lambda \|E\|_{2,1} + \frac{\mu}{2} \left\| E - \left(X - AZ + \frac{Y_1}{\mu} \right) \right\|_F^2. \end{aligned} \quad (9)$$

The complete procedure for solving Problem (9) is outlined in Algorithm 1. By choosing a proper dictionary A , LRRSC seeks the

Algorithm 1 Solving Problem (5) by Inexact ALM

Input:

data matrix X , parameters $\lambda > 0$

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 the variables as (9);

3: update the multipliers:

$$Y_1 = Y_1 + X - XZ - E;$$

$$Y_2 = Y_2 + X - \mu(Z - J);$$

4: update the parameter μ by $\mu = \min(\rho\mu, \mu_{\max})$;

5: check the convergence conditions

$$\|X - AZ - E\|_{\infty} < \varepsilon \text{ and } \|Z - J\|_{\infty} < \varepsilon;$$

6: **end while**

Output:

Z^*, E^*

lowest-rank representation matrix. Because each data point can be represented by the original data points, LRRSC uses the X as its dictionary. The last equation in Problem (9) is a convex problem and has a closed form solution. It is solved using the $l_{2,1}$ -norm minimization operator [2]. The first equation in Problem (9) is solved using the following lemma:

Lemma 1. Given any square matrix $Q \in \mathbb{R}^{n \times n}$, the unique closed form solution to the optimization problem

$$W^* = \arg \min_W \frac{1}{\mu} \|W\|_* + \frac{1}{2} \|W - Q\|_F^2, W = W^T, \quad (10)$$

takes the form

$$W^* = U_r \left(\Sigma_r - \frac{1}{\mu} \cdot I_r \right) V_r^T, \quad (11)$$

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

The proof of Lemma 1 is based on the following three lemmas.

Lemma 2. ([4] Lemma 7.1) Let U , V and W be matrices with compatible dimensions. Suppose both U and V have orthogonal columns, i.e., $U^T U = I$ and $V^T V = I$. Then we have

$$\|W\|_* = \|UWV^T\|_*. \quad (12)$$

Note that a similar equality also holds for the square of the Frobenius norm, $\|\cdot\|_F^2$.

Lemma 3. ([2] Lemma 3.1) Let A and D be square matrices, and let B and C be matrices with compatible dimensions. Then, for any block

$$\text{partitioned matrix } X = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

$$\left\| \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\|_* \geq \left\| \begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix} \right\|_* = \|A\|_* + \|D\|_*. \quad (13)$$

Note that a similar inequality also holds for the square of the Frobenius norm, $\|\cdot\|_F^2$ [29].

Lemma 4. ([16, 47, 48, 49]) Let $A \in \mathbb{R}^{m \times n}$ be a given matrix and $\|\cdot\|_1$ be the l_1 -norm. The proximal operator of

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \gamma \|x\|_1 \quad (14)$$

is $S_\gamma(x)$, i.e. the soft-thresholding operator

$$S_\gamma(x) = \begin{cases} x - \gamma, & x > \gamma \\ x + \gamma, & x < -\gamma \\ 0, & \text{else} \end{cases}.$$

Proof. (of Lemma 1) Let W^* be the unique minimizer. Then,

$$\begin{aligned} \|W^* - Q\|_F^2 &= \frac{1}{2} (\|W^* - Q\|_F^2 + \|W^* - Q^T\|_F^2) \\ &= \frac{1}{2} (\|W^* - (Q + Q^T)/2\|_F^2) + F(Q), \end{aligned} \quad (15)$$

where $F(Q)$ are entirely independent of W^* . Therefore, the original optimization can be converted to

$$W^* = \arg \min_W \frac{1}{\mu} \|W\|_* + \frac{1}{2} \|W - \tilde{Q}\|_F^2, W = W^T, \quad (16)$$

where $\tilde{Q} = (Q + Q^T)/2$.

Let $\tilde{Q} = U_r \Sigma_r V_r^T$ be the skinny SVD of a given symmetric matrix $\tilde{Q}$ of rank r , where $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ with $\{r : \sigma_r > \frac{1}{\mu}\}$ are positive singular values. More precisely, both U_r and V_r have orthogonal columns, i.e., $U_r^T U_r = I$ and $V_r^T V_r = I$. Let $W = U_r \tilde{W} V_r^T$. By Lemma 2, Problem (16) is equivalent to

$$\tilde{W}^* = \arg \min_{\tilde{W}} \frac{1}{\mu} \|\tilde{W}\|_* + \frac{1}{2} \|\tilde{W} - \Sigma_r\|_F^2, \tilde{W} = \tilde{W}^T. \quad (17)$$

Let $\tilde{W}^*$ be an optimizer of Problem (17), then $\tilde{W}^*$ must be a diagonal matrix. Assume $\tilde{W}_0$ is a non-diagonal matrix, i.e., there exists nonzero entries in the non-diagonal of $\tilde{W}^*$. Let $f(\tilde{W}) = \frac{1}{\mu} \|\tilde{W}\|_* + \frac{1}{2} \|\tilde{W} - \Sigma_r\|_F^2$. By Lemma 3 and the strict decreasing property of the square of the Frobenius norm, we can always derive $\tilde{W}_1$ by removing the non-diagonal entries of $\tilde{W}_0$ such that $f(\tilde{W}_1) < f(\tilde{W}_0)$. This contradicts the non-diagonal matrix assumption.

Let $\tilde{W} = \text{diag}(w_1, w_2, \dots, w_r)$. For diagonal matrices, the sum of the singular values is equal to the sum of the absolute values of the diagonal elements, i.e., the l_1 -norm. The optimization of Problem (17) is equivalent to

$$\tilde{W}^* = \arg \min_{\tilde{W}} \frac{1}{\mu} \|\tilde{W}\|_1 + \frac{1}{2} \|\tilde{W} - \Sigma_r\|_F^2, \tilde{W} = \tilde{W}^T. \quad (18)$$

By Lemma 4, the optimal solution to this problem is given by $\tilde{W}^* = S_{\frac{1}{\mu}}(\Sigma_r)$. Finally, we get $W^* = U_r (\Sigma_r - \frac{1}{\mu} \cdot I_r) V_r^T$. $\square$

3.2. Building an affinity graph based on the symmetric low-rank matrix

The critical task of subspace clustering in this paper mainly focus on how to learn a good affinity matrix in a time efficient manner. Using the optimal symmetric low-rank matrix Z^* from Problem (5), we need to construct an affinity graph $G = (V, E)$ associated with the corresponding adjacency matrix $W = \{w_{ij} | i, j \in V\}$, where $V = \{v_1, v_2, \dots, v_n\}$ is a set of vertices and $E = \{e_{ij} | i, j \in V\}$ is a set of edges. Note that $\{w_{ij}\}$ represents the weight of edge e_{ij} that associates vertices i and j . A fundamental problem involving affinity graph construction is how to determine the adjacency matrix W . Because each sample is represented by the others, each element z_{ij} of the matrix Z^* naturally characterizes the contribution of the sample x_j to the reconstruction of sample x_i .

A straightforward method is to use the symmetric low-rank matrix Z^* as the adjacency matrix W for spectral clustering. It can mostly achieve satisfied results. However, a few entries of the matrix are sensitive to the noise or outliers in low-rank representation. Those entries cannot reflect the real relationships of the pairwise points. The matrix Z^* cannot adequately represent the relationship between samples when there are grossly corrupted observations. Therefore, the straightforward use of the matrix Z^* as a pairwise affinity relationship between data points inevitably results in loss of information, and it does not fully capture the intrinsic correlation of data points [43,44]. Consequently, the clustering performance may seriously decline because of a lack of robustness.

Several existing methods that use the angular information of original samples to measure the relationship between samples were proposed in literatures [30,40]. However, a lack of robustness is inevitable because of corrupted samples. To enhance the ability of the low-rank representation to separate samples in different subspaces, a reasonable strategy is to derive an affinity graph with enhanced clustering information from the matrix Z^* . This preserves the subspace structures of high-dimensional data, and recovers the clustering relations among samples. If any two data points x_i and x_j 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. It has been demonstrated in recent studies of manifold learning theory [50]. To remove the effect of various noises in the high-dimensional data, we employed a mechanism of exploiting the angular information of its principal directions instead of the low-rank representation itself. As the coefficient matrix is low-rank, the angular information can hardly be effected by the few error entries of rows or column in the low-rank coefficient matrix. In other words, the elements of the affinity matrix are more robust to the noise or outliers than those of the low-rank coefficient matrix. Hence, the angular information of its principal directions is a good surrogate for symmetric low-rank representation.

Instead of directly using $|Z^*| + |(Z^*)^T|$ to define an affinity graph, we consider the mechanism driving the construction of the affinity graph from the matrix Z^* . We consider Z^* with the skinny SVD $U^* \Sigma^* (V^*)^T$. Note that U^* and V^* are the orthogonal bases of the column and row space of the matrix Z^* . Inspired by [4,30], we assign each column of U^* a weight by multiplying it by $(\Sigma^*)^{1/2}$, and each row of $(V^*)^T$ a weight by multiplying it by $(\Sigma^*)^{1/2}$. Then, we can define $M = U^* (\Sigma^*)^{1/2}$, $N = (\Sigma^*)^{1/2} (V^*)^T$. The product of two matrices can represent Z^* , i.e., $Z^* = MN$.

Because Z^* is a symmetric low-rank matrix, the absolute values of the columns of U^* and V^* always agree. The columns of U^* or V^* can span the principal directions of the symmetric low-rank matrix Z^* , and the diagonal entries reflect the relative importance of the coefficient matrix Z^* in each of these directions. Therefore, we use angular information from all of the row vectors of matrix M , or

all of the column vectors of matrix N , instead of Z^* , to define an affinity matrix W as follows:

$$[W]_{ij} = \left(\frac{m_i^T m_j}{\|m_i\|_2 \|m_j\|_2} \right)^{2\alpha} \quad \text{or} \quad [W]_{ij} = \left(\frac{n_i^T n_j}{\|n_i\|_2 \|n_j\|_2} \right)^{2\alpha}, \quad (19)$$

where m_i and m_j denote the i th and j th row of the matrix M , or n_i and n_j denote the i th and j th column of the matrix N . The values of the affinity matrix W can be distributed on the unit ball by the l_2 -norm of data vectors. Consequently, the affinity matrix W preserves angular information between data vectors but removes length information. By using $(\cdot)^{2\alpha}$ we ensure that the values of the affinity matrix W are positive inputs for the subspace clustering, and also increase the separation of points of different groups because of the geometry of the l_2 -norm ball. Finally, we apply spectral clustering algorithms such as NCuts [41] to segment the samples into a given number of clusters. Algorithm 2 summarizes the complete

Algorithm 2 The LRRSC algorithm

Input:

data matrix $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}$, number k of subspaces, regularized parameters $\lambda > 0, \alpha > 0$

1: Solving the following problem by Algorithm 1:

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1} \quad \text{s.t.} \quad X = AZ + E, Z = Z^T,$$

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

2: Compute the skinny SVD $Z^* = U^* \Sigma^* (V^*)^T$.

3: Calculate $M = U^* (\Sigma^*)^{1/2}$ or $N = (\Sigma^*)^{1/2} (V^*)^T$.

4: Construct the affinity graph matrix W , i.e.,

$$[W]_{ij} = \left(\frac{m_i^T m_j}{\|m_i\|_2 \|m_j\|_2} \right)^{2\alpha} \quad \text{or} \quad [W]_{ij} = \left(\frac{n_i^T n_j}{\|n_i\|_2 \|n_j\|_2} \right)^{2\alpha}.$$

5: Apply W to perform NCuts.

Output:

The clustering results.

subspace clustering algorithm of LRRSC.

3.3. Pursuing a symmetric low-rank representation through a closed form solution

The existing methods, e.g., a sparsity constraint and rank minimization, for obtaining reasonable data representation to characterize the correlation structure of high-dimensional data have high computational cost. Specifically, LRRSC typically requires iterative SVD operations when solving the nuclear norm optimization problem. Therefore, it suffers from a high computation cost.

In this section, we present an efficient variant of LRRSC, namely eLRRSC, to improve the scalability of LRRSC, which attempts to learn the symmetric low-rank representation by a closed form solution without iterative SVD operations. In particular, the alternative learning scheme is composed of three steps.

First, eLRRSC uses a collaborative representation with regularized least square for symmetric representation, which is also adopted in SLRR Chen et al. [45]. The optimization problem is formalized as follows:

$$\min_Z \|Z\|_F^2 + \frac{\lambda}{2} \|X - XZ\|_F^2 \quad \text{s.t.} \quad X = XZ + E, \quad (20)$$

The above problem has a closed form solution:

$$Z = (X^T X + \lambda \cdot I)^{-1} X^T X, \quad (21)$$

where $\lambda > 0$ is a parameter and I is the identity matrix of size $n \times n$.

By utilizing the self-expressiveness property of the data, we consider a general model of data representation:

$$\min f(Z) \quad s.t. \quad X = XZ + E, \quad (22)$$

where $f(Z)$ is a matrix function, i.e., $\|\cdot\|_F^2$ or $\|Z\|_*$, and E is an error term. The optimal solution Z^* is a particular representation of data X . Then we can write

$$A = XZ, \quad (23)$$

where each column of the data matrix $A = \{a_1, a_2, \dots, a_n\} \in \mathbb{R}^{m \times n}$ represents a corrected sample, i.e., $a_i = Xz_i$. In other words, each corrected sample in a union of subspaces can be efficiently reconstructed by other original samples in the dataset. The underlying assumption for the success of the eLRRSC algorithm is that the samples are drawn from the union of low-dimensional subspaces. As a result, the matrix A should be low-rank. Denote the ranks of A , X and Z by $\text{rank}(A)$, $\text{rank}(X)$ and $\text{rank}(Z)$, respectively. Therefore Z is low-rank as $\text{rank}(A) \leq \min(\text{rank}(X), \text{rank}(Z))$. Consequently, we need to construct the symmetric low-rank representation instead of the collaborative representation to characterize the correlation structure of high-dimensional data. As mentioned above, the symmetric low-rank representation of high-dimensional data play the essential role for preserving the subspace structures. Hence, we obtain an alternative symmetric low-rank Z' instead of Z from a collaborative representation of high-dimensional data by solving the following optimization problem:

$$\min_W \|Z'\|_* + \frac{\mu}{2} \|Z' - Z\|_F^2 \quad s.t. \quad Z' - Z = E, Z' = Z'^T. \quad (24)$$

The above problem can be solved by Lemma 1. Given the assumption that high-dimensional data are approximately drawn from a union of multiple subspaces, it is reasonable that eLRRSC considers a symmetric low-rank representation of high-dimensional data as a good surrogate for the collaborative representation. It is clear that eLRRSC obtained a symmetric low-rank representation through a closed form solution. However, eLRRSC does not pursue a symmetric low-rank matrix for the low-rank matrix recovery or completion. Instead, it mainly focuses on only representations of data, which is further applied to evaluate the membership between samples.

Finally, we make use of the angular information of its principal directions to get an affinity matrix after obtaining the symmetric low-rank Z^* . The angular information can be applied in the spectral clustering algorithm, such as NCuts [41], to produce the final clustering results. Algorithm 3 summarizes the complete subspace clustering algorithm of eLRRSC. The purpose of the first two steps is to pursue a symmetric low-rank representation of original data with respect to the same basis X in Algorithm 3. By making use of a closed-form solution, eLRRSC effectively provides an alternative scheme instead of LRRSC to seek a low-rank matrix, which preserves the intrinsically geometrical structure of the memberships of samples.

3.4. Relationship between eLRRSC and SLRR

We emphasize that eLRRSC is the follow-up research based on our previous work, i.e., SLRR [45]. First, eLRRSC and SLRR share the same collaborative representation with regularized least square. Then, both of them utilize the angular information of principal directions of the symmetric low-rank representation to construct the affinity matrix for the spectral clustering algorithm. However, there is a major difference in constructing a desirable low-rank symmetric matrix. In fact, each of them employ entirely different idea to obtain their own symmetric low-rank matrices.

Algorithm 3 The eLRRSC algorithm

Input:

data matrix $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}$, number k of subspaces, regularized parameters $\lambda > 0$, $\mu > 0$ and $\alpha > 0$

1: Solving the following problem:

$$\min_Z \|Z\|_F^2 + \frac{\lambda}{2} \|X - XZ\|_F^2 \quad s.t. \quad X = XZ + E,$$

and obtain the optimal solution $Z = (X^T X + \lambda \cdot I)^{-1} X^T X$.

2: Solving the following problem by Lemma 1:

$$\min_W \|Z'\|_* + \frac{\mu}{2} \|Z' - Z\|_F^2 \quad s.t. \quad Z' - Z = E, Z' = Z'^T$$

and obtain the optimal solution $Z' = U_r S V_r^T$, where $Z = U \Sigma V^T$, $S = \Sigma_r - \frac{1}{\mu} \cdot I_r$, $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ with $\{r : \sigma_r > \frac{1}{\mu}\}$ are positive singular values.

3: Calculate $M = U_r S^{1/2}$ or $N = S^{1/2} V_r^T$

4: Construct the affinity graph matrix W , i.e.,

$$[W]_{ij} = \left(\frac{m_i^T m_j}{\|m_i\|_2 \|m_j\|_2} \right)^{2\alpha} \quad \text{or} \quad [W]_{ij} = \left(\frac{n_i^T n_j}{\|n_i\|_2 \|n_j\|_2} \right)^{2\alpha}.$$

5: Apply W to perform NCuts.

Output:

The clustering results.

To obtain the symmetric low-rank matrix, SLRR attempts to pursue an alternative low-rank matrix instead of the original data matrix through existing low-rank matrix recovery techniques. To achieve competitive subspace clustering performance, SLRR need to elaborate a proper alternative low-rank matrix, which highly depends on the choice of low-rank matrix recovery techniques. This means that SLRR requires more prior knowledge of original data, i.e., types of noise. Besides, the complexity of the low-rank matrix recovery technique adopted by SLRR also may lead to high computational cost.

On the other hand, eLRRSC obtains the symmetric low-rank matrix from the collaborative representation under the assumption that high-dimensional data involves with the multiple subspace structures, without considering the noise types of original data. Hence, the overall computational cost of eLRRSC can be effectively guaranteed. Besides, the key observation is that the intrinsically geometrical structure of the samples' memberships are preserved in the symmetric low-rank representation, which is closely related to the new basis, i.e., the original data. This shows that both of them essentially focus on different construction methods on symmetric low-rank matrices respectively although a closed form solution improves the computational efficiency of large-scale subspace clustering.

3.5. Convergence properties and computational complexity analysis

The convergence properties of the exact ALM algorithm for a smooth objective function have been generally proven in [20]. The inexact variation of ALM has been extensively studied and generally converges well. Algorithm 2 performs well in practical applications. We assume that the size of X is $m \times n$, where X has n samples and each sample has m dimensions. The computational complexity of the first step in Algorithm 1 is $O(n^3)$ because it requires computing the SVD of a $n \times n$ matrix. The overall computational complexity of Algorithm 1 is $O(2n^3 + mn^2)$. When $n > m$, the computational complexity of Algorithm 1 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

Table 1

Parameter settings of different algorithms on face clustering. The parameter λ is used as a trade-off between low rankness (or sparsity) and the effect of noise (LRRSC, SLRR, LRR, LRR-PSD, SSC). The parameter α enhances the separate ability of low-rank representation between samples in different subspaces used by LRRSC. For SLRR, n is the number of subspaces, i.e., the number of subjects. For LRSC, τ and λ are two parameters weighting noise.

Method	Scenario 1	Scenario 2
LRRSC	$\lambda = 0.2, \alpha = 4$	$\lambda = 0.1, \alpha = 3$
eLRRSC	$\lambda = 35, \mu = 1e^{-3}, \alpha = 3$	$\lambda = 40, \mu = 0.1, \alpha = 2$
SLRR	$\alpha = 3, \lambda = 30$	$\alpha = 3, \lambda = 1, r = 10n$
LRR	$\lambda = 0.18, \alpha = 2$	
LRR-PSD	$\lambda = 0.2, \alpha = 4$	$\lambda = 0.1, \alpha = 3$
LRSC	$\tau = 0.4, \lambda = 0.045, \alpha = 2$	$\tau = 0.045, \lambda = 0.045, \alpha = 3$
SSC	$\lambda_e = 8/\mu_e$	$\lambda_e = 20/\mu_e$

final overall complexity of Algorithm 2 is $O(n^3)$. In our experiments, there were always less than 335 iterations.

In Algorithm 3, the computational complexity of the first two steps and the last two steps are $O(n^3)$ and $O(n^2)$, respectively. Therefore, the general complexity is $O(n^3)$. With this theoretical result, we can say that eLRRSC is significantly computational efficient than LRRSC.

4. Experiments

In this section, we evaluated the performance of the proposed LRRSC algorithm and its variant (eLRRSC) using a series of experiments on publicly available databases, the extended Yale B, Hopkins 155 and penguin databases (the Matlab source code of our method is available at <http://www.machinailab.org/users/chenjie>). We compared the performance of LRRSC with several state-of-the-art subspace clustering algorithms (LRR [4], LRR-PSD [29], SSC [5], low rank subspace clustering (LRSC) [6,7] and SLRR [45]). As there is no source code publicly available for LRR-PSD, we implemented the LRR-PSD algorithm according to its theory. For LRSC, we chose the noisy data version of (P_3) as its instance. For the other algorithms, we used the source codes provided by their authors.

We evaluate the performance of above algorithms by comparing subspace clustering errors:

$$\text{error} = \frac{N_{\text{error}}}{N_{\text{total}}}, \quad (25)$$

where N_{error} represents the number of misclassified points, and N_{total} is the total number of points. LRRSC requires two parameters, λ and α . Empirically speaking, the parameter λ should be relatively large if the data are “clean”, or smaller if they are contaminated with small noises, and the parameter α ranges from 2 to 4. For the other algorithms, we used the parameters given by the respective authors, or manually tuned the parameters to find the best results. The parameters for these methods are shown in Tables 1 and 2. All the algorithms are implemented by Matlab R2011b, and all experiments are performed on a Windows platform with Intel Core i5-2300 CPU and 16GB memory.

4.1. Experiments on face clustering

Given a collection of face images from multiple individuals, which have various illumination conditions and expression, we'd like to cluster images according to their individuals. Since this set of face images lie close to a union of 9-dimensional subspaces [13], the face clustering problem can be boiled down to image clustering problem over a union of subspaces.

In this section, we consider the Extended Yale B Database [51,52] for the face clustering problem. This database consists of 2414 frontal images from 38 individuals. There are approximately

59 – 64 images available for each person, shot under various laboratory-controlled lighting conditions. Fig. 1a shows some sample images. In order to improve the efficiency of the experiments, without losing generality, we first resize all images to 48×42 pixels, therefore each image can be regarded as a vector of 2016 dimensions. In the rest of this section, we will consider two different experimental scenarios to evaluate the performance of our proposed methods.

1. First experimental scenario: We used the first 10 classes of the Extended Yale B Database, as in [2]. This subset of the database contains 640 frontal face images from 10 subjects. To compare the clustering errors between different approaches, we first used the raw pixel values without pre-processing, considering each image as a data vector of 2016 dimensions. Then, we applied PCA to pre-process these face images using 100 and 500 feature dimensions.

We consider 640 frontal face images belonging to 10 subjects. Besides using these raw images, we also apply PCA to project these images to 100 and 500 dimension feature spaces, respectively.

Fig. 2 shows the influence of the parameters λ and α on the face clustering errors of LRRSC. In each experiment, $\alpha \in \{1, 2, 3, 4\}$. Generally, larger α leads to better clustering performance. For example, we let λ range from 0.15 to 0.4 with $\alpha = 3$. Then, the clustering error varies from 3.91% to 6.41% (Fig. 2a). When we let λ range from 0.15 to 0.4 with $\alpha = 1$, the clustering error varies from 15.31% to 42.19%. However, note that if α is too large (i.e., $\alpha = 4$), LRRSC must narrow the range of parameter λ to obtain the desired result. This can also be observed in Fig. 2b. Fig. 2c shows that LRRSC performs well for a large range of λ . This is benefited from the 100-dimensional data obtained by applying PCA with noise removal. Consequently, LRRSC is capable of a stable face clustering performance when λ is chosen according to the noise level.

The three different feature dimensions of the face images required 32, 118, 114 and 113 iterations. Table 3 shows the proposed eLRRSC has most promising performance. For example, eLRRSC achieved a low clustering error of 2.97% for the original data, and improved the clustering accuracy by at least 18% when compared with LRR, LRSC and SSC. We observed the same advantages when using our proposed method for the 500- and 100-dimensional data obtained using PCA. The clustering results for each algorithm using raw or reduced dimension data are very similar, which suggests that the face images of an individual lie close to a union of subspaces. These clustering results confirmed that the affinity calculated from the symmetric low-rank representation significantly improves the clustering accuracy when the data are grossly contaminated by noise, and that it outperforms the other algorithms. LRR compares favorably against the other algorithms. LRR-PSD, SSC, and LRSC have very similar clustering results.

Fig. 3 showed the computational times of the competing algorithms corresponding to results outside parentheses in Table 3. We can see that the computation costs of eLRRSC, SLRR and LRSC significantly outperformed the other approaches. This is because both of them can obtain a closed form solution of the low-rank representation, which they use to build the affinity. Thus, they can run much faster than the other approaches. On other hand, LRRSC, LRR, LRR-PSD have comparable computational times because of their efficient convex optimization techniques.

Since clustering performance of LRRSC and eLRRSC is closely related with the choice of the parameters, we conducted another experiment to illustrate the effect of estimation of the parameters of LRRSC and eLRRSC using various number of training samples. We designed four groups of training samples, where each group randomly selected 5, 10, 15 and 20 images of each person respectively. The rest are used for testing. Because the training and testing samples are selected randomly, 10 different training and test sample sets are chosen for parameter evaluation. The final clustering

Table 2
Parameter settings of different algorithms on motion segmentation.

Method	The Hopkins 155 motion database		The penguin motion database
	Scenario 1	Scenario 2	Scenario 1
LRRSC	$\lambda = 3.3, \alpha = 2$	$\lambda = 3, \alpha = 3$	$\lambda = 0.05, \alpha = 4$
eLRRSC	$\lambda = 5e^{-3}, \mu = 0.2, \alpha = 2$	$\lambda = 5e^{-3}, \mu = 0.1, \alpha = 2$	$\lambda = 5.5, \alpha = 4, \mu = 30$
SLRR	$\alpha = 2, \lambda = 5e^{-3}$	$\alpha = 2, \lambda = 5e^{-3}, r = 4n$	$\lambda = 5e^{-3}, \alpha = 4$
LRR	$\lambda = 4, \alpha = 2$		$\lambda = 0.1, \alpha = 2$
LRR-PSD	$\lambda = 3.3, \alpha = 2$	$\lambda = 3, \alpha = 3$	$\lambda = 0.05, \alpha = 4$
LRSC	$\tau = 420, \lambda = 5000, \alpha = 2$		$\tau = 5, \lambda = 10, \alpha = 2$
SSC	$\lambda_z = 800/\mu_z$		$\lambda_z = 240/\mu_z$



(a) The original sample images



(b) The corrupted sample images with the 10% random pixel corruptions

Fig. 1. Example images of multiple individuals from the Extended Yale B database.

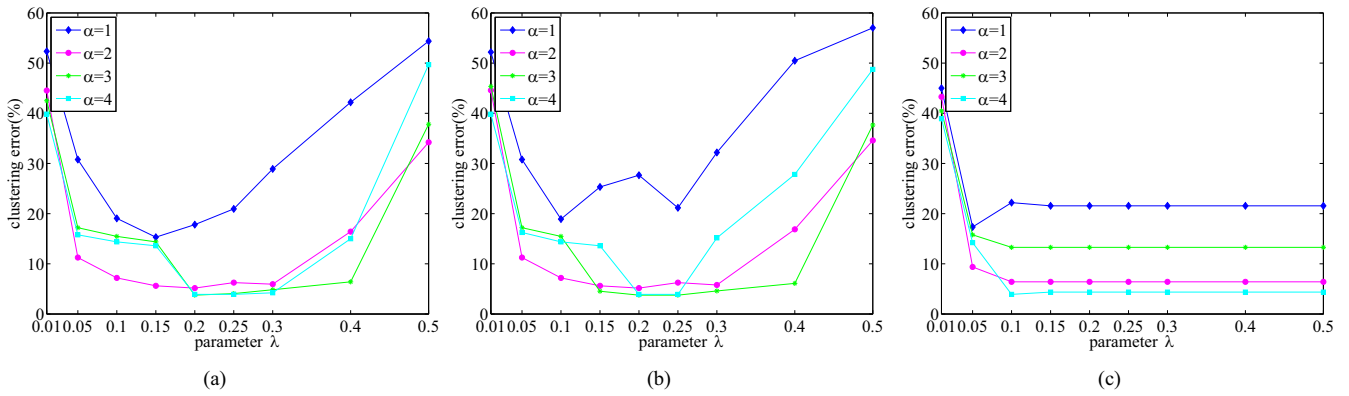


Fig. 2. Clustering error given different λ and α combinations, using the first 10 classes in the Extended Yale B Database. (a) Raw data. (b) The 500-dimensional data obtained by applying PCA. (c) The 100-dimensional data obtained by applying PCA.

Table 3
Clustering error (%) of different algorithms on the first 10 classes of the Extended Yale B Database.

Algorithm	LRRSC	eLRRSC	SLRR	LRR	LRR-PSD	LRSC	SSC
Dim. = 100	4.37	4.22	4.22	21.09	38.44	36.47	36.56
Dim. = 300	3.91	3.91	3.91	20.63	38.12	35.78	35.47
Dim. = 500	3.91	2.97	3.13	21.56	35.47	36.67	35
Raw data	3.91	2.97	–	20.94	35.47	36.97	35

Table 4

Clustering error (%) and computational time (seconds) of LRRSC and eLRRSC on the first 10 classes of the Extended Yale B Database using different number of training samples for parameter evaluation.

Algorithm	LRRSC			eLRRSC		
	Mean	Std.	Time	Mean	Std.	Time
5	7.68	3.8	119.14	6.22	3.29	34.24
10	8.15	4.73	96.65	8.2	4.58	31.27
15	9.2	4.44	77.82	9.06	4.92	29.58
20	9.34	4.42	68.53	11.34	5.11	27.3

result is computed by averaging the recognition rates from these ten experiments. We set parameter α range from 2 to 4. For LRRSC, we let λ range from 0.1 to 0.3 in steps of 0.05. For eLRRSC, we let λ range from 5 to 40 in steps of 5, and let μ range from 0.01 to 0.3 in steps of 0.02 respectively.

Table 4 shows the mean clustering error and standard deviation of LRRSC and eLRRSC when the number of an individual's images varies from 5 to 20. LRRSC and eLRRSC obtained similar clustering results under different numbers of training samples. How-

ever, eLRRSC achieved a lower computation cost because its solution can be computed in closed form. Besides, it can be seen from Table 4 that the mean clustering error gradually raises as the number of samples increases. The larger the number of samples there is in the experiment, the greater the computational complexity there is for clustering regarding as an unsupervised manner.

Finally, we evaluated the performance and robustness of LRRSC and eLRRSC as well as the other methods on a more challenging set of face images using artificial occlusion, namely random pixel corruptions. To simulate random pixel corruptions, the locations of corrupted pixels of face images were chosen randomly with uniformly distributed random values in the range [0, 1]. The percentage of pixel corruption levels was varied from 10 to 40% in steps of 10%. Fig. 1b shows some examples of the face images with random 10% pixel corruptions. All experiments were repeated 10 times. Table 5 shows the average clustering error. Some experiment results are given by our previous work [45]. The results demonstrate that LRRSC, eLRRSC and SLRR obtain similar clustering accuracies. At corruption percentages of 10% and 20%, SLRR obtains the lowest clustering error. Besides, LRRSC consistently outperformed all

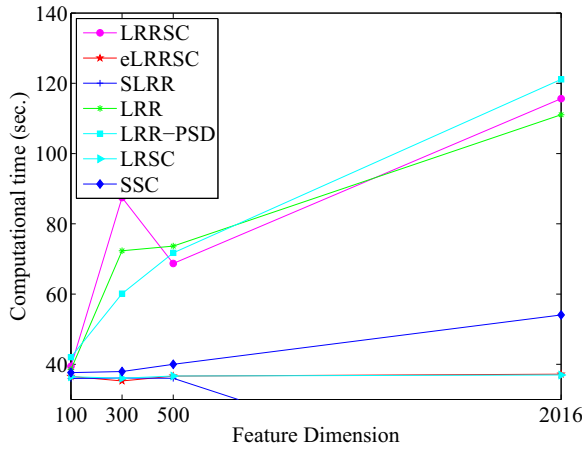


Fig. 3. Computational time (seconds) of each algorithm for different feature dimensions, using the first 10 classes of the Extended Yale B Database.

Table 5

Clustering error (%) by applying different algorithms on the first 10 classes of the Extended Yale Database B contaminated by random pixel corruptions.

Ratio (%)	LRRSC	eLRRSC	SLRR	LRR	LRR-PSD	LRSC	SSC
10	12.16	11.28	9.23	21.38	25.31	16.22	32.84
20	12.25	11.51	10.34	24.77	26.01	17.47	39.44
30	12.79	13.18	13.69	30.44	30.55	17.2	43.84
40	12.23	12.58	14.59	31.72	31.02	20.72	48.95

Table 6

Average clustering error (%) for different number of subjects, applying each algorithm to the Extended Yale B database.

Algorithm	LRRSC	eLRRSC	SLRR	LRR	LRR-PSD	LRSC	SSC
2 Subjects							
Mean	1.78	1.32	1.29	2.54	3.04	4.25	1.86
Median	0.78	0.78	0.78	0.78	2.34	3.13	0
3 Subjects							
Mean	2.61	2.08	1.94	4.23	4.33	6.07	3.24
Median	1.56	1.56	1.56	2.6	3.91	5.73	1.04
5 Subjects							
Mean	3.19	2.5	2.72	6.92	10.45	10.19	4.33
Median	2.81	2.19	2.5	5.63	7.19	7.5	2.82
8 Subjects							
Mean	4.01	3.02	3.21	13.62	23.86	23.65	5.87
Median	3.13	2.34	2.93	9.67	28.61	27.83	4.49
10 Subjects							
Mean	3.7	3.28	3.49	14.58	32.55	31.46	7.29
Median	3.28	2.81	2.81	16.56	34.06	28.13	5.47

the other methods for larger percentages of corrupted pixels, i.e., corruption percentages of 30% and 40%. Compared with the other competing methods, LRRSC, eLRRSC and SLRR are slightly more stable as the percentage of corruption increases.

2. Second experimental scenario: We used the experimental settings from [5]. We divided the 38 subjects into four groups, where subjects 1 to 10, 11 to 20, 21 to 30, and 31 to 38 correspond to four different groups. For each of the first three groups, we considered all choices of $n \in \{2, 3, 5, 8, 10\}$. For the last group, we considered all choices of $n \in \{2, 3, 5, 8\}$. We tested each choice (i.e., each set of n subjects) using each algorithm. Finally, the mean and median subspace clustering errors for different number of subjects were computed using all algorithms. Note that we applied these clustering algorithms to the normalized face images.

Table 6 shows the clustering results of various approaches using different number of subjects. When considering two subjects, eLRRSC achieved a clustering error of 1.32%. The eLRRSC algorithm consistently obtained lower average clustering errors than

the other algorithms when the number of subjects increased. For example, there was nearly 4% improvement in clustering accuracy compared with SSC for 10 subjects. Note that SSC performed better than the other original LRR-based approaches (LRR, LRR-PSD, and LRSC) in terms of the average clustering error. This confirms that our proposed method is very effective and robust to different number of subjects for face clustering.

Fig. 4 shows examples of the affinity graph matrix produced by the different algorithms for the Extended Yale B Database with five subjects. It is clear from Fig. 4 that the affinity graph matrix produced by LRRSC and eLRRSC has a distinct block-diagonal structure, whereas the other approaches do not. The clear block-diagonal structure implies that each subject becomes highly compact and the different subjects are better separated. Note that the number of iterations taken by our algorithm is always less than 335. The average numbers of iterations for the five different sets ($n \in \{2, 3, 5, 8, 10\}$ subjects) in the proposed method are 308, 256, 202, 182 and 166.

Fig. 5 shows the mean computational times for the Extended Yale B database with different number of subjects. Note that the computational times of eLRRSC, SLRR and LRSC are still much lower than that of the other algorithms. However, LRSC does not perform well, especially as the number of subjects increases. LRRSC, LRR, and SSC have comparable computational times in these experiments.

4.2. Experiments on motion segmentation

Motion segmentation refers to the problem of segmenting feature trajectories of multiple rigidly moving objects into their corresponding spatiotemporal regions in a video sequence. The feature trajectories from a single rigid motion lie in a linear subspace of at most four dimensions [14]. Motion segmentation can be regarded as a subspace clustering problem.

4.2.1. The Hopkins 155 database

We first consider the Hopkins 155 database [53] for the motion segmentation problem. It consists of 155 sequences of two motions or three motions. Fig. 6 shows some example frames from two video sequences with feature trajectories. There are 39 – 550 data vectors drawn from two, three or five motions for each sequence, which correspond to a subspace. Each sequence is a separate clustering task.

We considered two scenarios to evaluate the performance of the applicability of LRRSC and eLRRSC to motion segmentation. We first used the original feature trajectories associated with each motion in an affine subspace, i.e., enforced that the coefficients sum to 1. Then, we projected the original data into a $4n$ -dimensional subspace using PCA, where n is the number of subspaces.

The clustering performance for all 155 sequences was largely affected by λ . Fig. 7a and b show the clustering errors of LRRSC using two experimental settings for different λ over all 155 sequences. In Fig. 7a, when λ was between 1 and 6 the clustering error varied between 1.5% and 2.67%; when λ was between 2.4 and 4, the clustering error remained very stable, varying between 1.5% and 1.7%. In Fig. 7b, when λ was between 1 and 6, the clustering error varied from 1.56% to 3.18%; when λ was between 2.4 and 4, the clustering error varied from 1.56% to 2.31% and remained very stable. This implies that LRRSC performs well under a wide range of values of λ . In addition, choosing λ for each sequence may improve the clustering performance, especially when the data are grossly corrupted by noise.

Tables 7 and 8 show the average clustering errors of the different algorithms on all 155 sequences of the Hopkins 155 database, using two experimental settings. In both experimental settings, eLRRSC obtained competitive clustering results and significantly

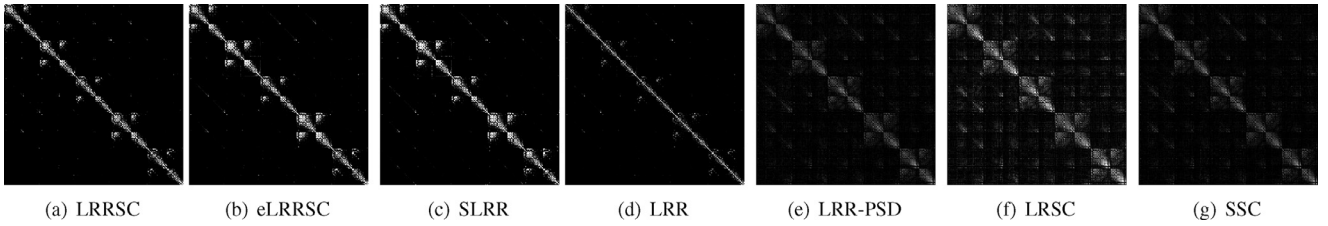


Fig. 4. Examples of the affinity graph matrix produced when using different algorithms for the Extended Yale B Database with five subjects.

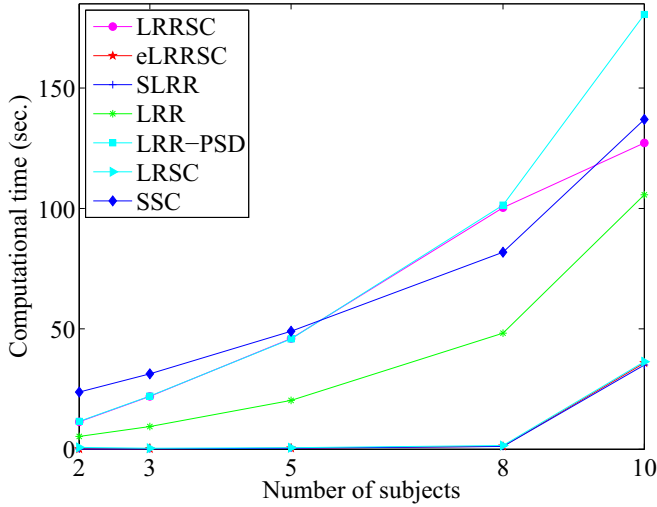


Fig. 5. Comparison of mean computational time (seconds) of each algorithm applied to the Extended Yale B database, using different number of subjects.

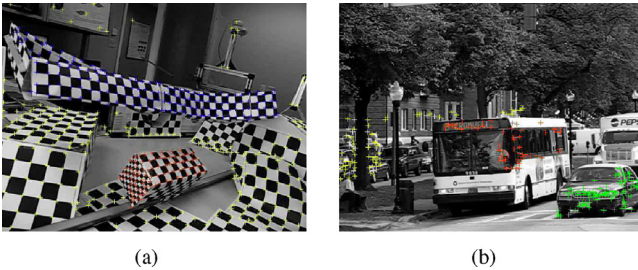


Fig. 6. Example frames from two video sequences with feature trajectories involving three motions.

Table 7
Average clustering error (%) and mean computation time (seconds) when applying the different algorithms to the Hopkins 155 database, with the 2F-dimensional data points.

Algorithm	Error				Time
	mean	median	std.	max.	
LRRSC	1.5	0	4.36	33.33	4.71
eLRRSC	0.86	0	3.39	34.33	0.09
SLRR	0.88	0	3.63	38.06	0.09
LRR	1.71	0	4.86	33.33	1.29
LRR-PSD	5.38	0.55	10.18	45.79	4.35
LRSC	4.73	0.59	8.8	40.55	0.14
SSC	2.23	0	7.26	47.19	1.02

outperformed the other algorithms. Specifically, LRRSC obtained 1.5% and 1.56%, and eLRRSC obtained 0.86% and 1.3% clustering errors for the two experimental settings. This confirms the effectiveness and robustness of LRRSC and eLRRSC for the segmentation of different motion subspaces by exploiting the angular information of the principal directions of the symmetric low-rank representa-

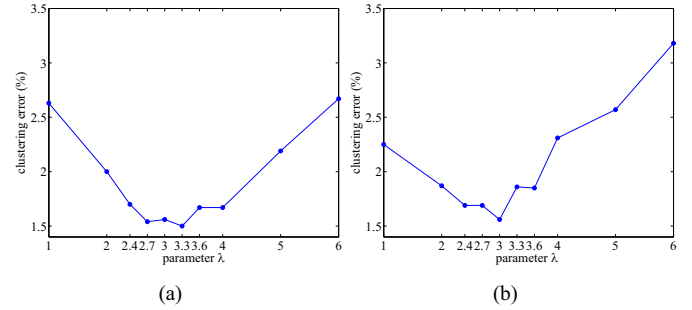


Fig. 7. The influences of the parameter λ of LRRSC. (a) The clustering error of LRRSC on the Hopkins 155 database with the 2F-dimensional data points. (b) The clustering error of LRRSC on the Hopkins 155 database with the 4n-dimensional data points obtained by applying PCA.

Table 8
Average clustering error (%) and mean computation time (seconds) when applying the different algorithms to the Hopkins 155 database, with the 4n-dimensional data points obtained using PCA.

Algorithm	Error				Time
	mean	median	std.	max.	
LRRSC	1.56	0	5.48	43.38	4.62
eLRRSC	1.3	0	4.97	39.73	0.08
SLRR	1.3	0	5.1	42.16	0.07
LRR	2.17	0	6.58	43.38	0.69
LRR-PSD	5.78	0.57	10.6	45.79	5.23
LRSC	4.89	0.63	8.91	40.55	0.13
SSC	2.47	0	7.5	47.19	0.93

tion of each motion. We note that the accuracy of LRRSC was very similar for the two experimental settings. This confirms that the feature trajectories of each sequence in a video approximately lie close to a 4n-dimensional linear subspace of the 2F-dimensional ambient space [14]. The computational costs of eLRRSC, SLRR and LRSC are much lower than the other algorithms. Hence, LRRSC and eLRRSC are effective and robust methods for motion segmentation. The computational cost of LRR is lower than the LRR-based methods (LRRSC and LRR-PSD). This is because LRR applies the dictionary learning method to improve performance, while LRRSC uses one SVD computation at each iteration and the original data's dictionary.

4.2.2. The penguin motion database

In this section, we further compared the clustering performance obtained by LRRSC and eLRRSC against the other competing algorithms on a more challenging data set. The penguin motion database contains six non-rigidly moving objects over 42 annotated frames of a video sequence, which is drawn from the SegTrack dataset [54]. We presented four scenarios consisting of four data sets of different feature trajectories to illustrate the robustness and effectiveness of LRRSC and eLRRSC. In particular, we sampled 50, 100, 150 and 200 points of feature trajectories at the

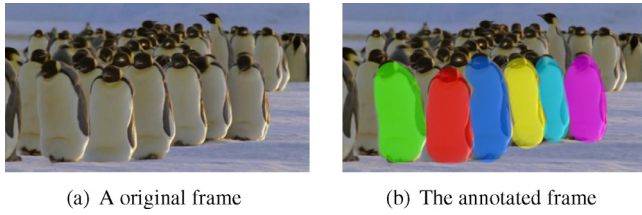


Fig. 8. A example frame of a video sequence involving six motions.

Table 9

Clustering error (%) of different algorithms on the penguin motion database with different number of feature trajectories.

Algorithm	LRRSC	eLRRSC	SLRR	LRR	LRR-PSD	LRSC	SSC
50	4	3.33	20	27.67	22	21	20.33
100	7.33	3.17	18.67	26.5	22.33	20.5	23.5
150	8.22	3.78	17.67	21	22.33	22.11	24.56
200	8	3.42	18.09	21.83	28.5	21.67	20.5

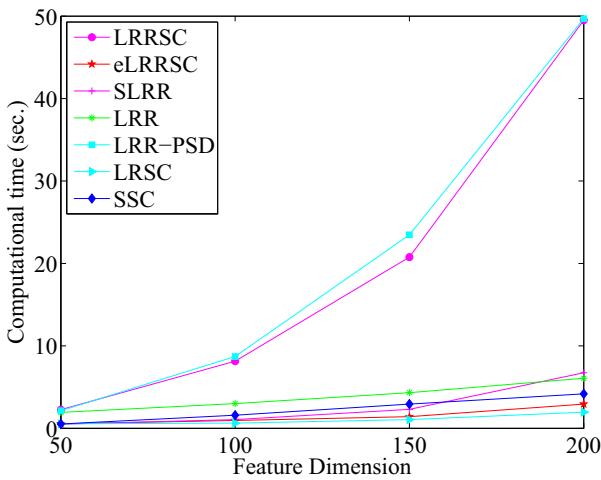


Fig. 9. Comparison of computational time (seconds) of each algorithm applied to the penguin motion database, using different number of feature trajectories.

average intervals from each annotated frame of the penguin motion database respectively. This experiment is more challenging for all the algorithms. A typical example frame of a video sequence and its annotated frame are illustrated in Fig. 8.

We reported clustering errors for four scenarios in Table 9, which demonstrates that our proposed eLRRSC achieves a consistently high clustering accuracy. Besides, LRRSC also performs well in four scenarios. This suggests that the angular information of the principal directions of the symmetric low-rank representation have the potential to make non-rigidly moving objects of a video sequence separable when the number of motions increases. All the results show that SSC always performs poorly under different feature trajectories. Fig. 9 shows the computational costs of all competing methods. Clearly, eLRRSC, SLRR and LRSC achieve similar computational costs overall, which are lower than that of the other methods.

5. Discussions

LRR-based techniques such as LRR and LRRSC compared with sparsity based methods able to capture globally linear structures of data. However, there are still several significant problems. For LRRSC and eLRRSC, how to choose proper parameters is an open problem in practice. Although the parameter α of LRRSC can be easily chosen empirically by its limited range, it is difficult to es-

timate the parameter λ without prior knowledge. In addition, it is important to develop dictionary learning algorithms, which may significantly improve the subspace clustering performance.

Then, we discussed three differences among the above LRR-based methods. First, they contain different objective functions. For example, LRRSC and LRP-PSD imposed different constraints on low-rank representation, i.e., a symmetric constraint and semi-definite guarantees, respectively. In particular, LRRSC only aimed to obtain a symmetric matrix while LRR-PSD pursued a semi-definite matrix at the first step of the optimizations. In fact, we should emphasize that it is easy to validate that their corresponding efficiency, i.e., Lemma 1 and Theorem 14 [29], are distinct using synthetic data or some instances. For instance, different optimal solutions at the second step of the eLRRSC algorithm with relative large λ can be achieved if we used semi-definite guarantees instead of symmetric constraint with respective optimization theory. Moreover, the procedures of the proof of two optimization theories are different. However, both of Lemma 1 and Theorem 14 can achieve the identical result if and only if Q is a symmetric positive semi-definite matrix in Lemma 1. This is because the results of SVD and eigenvalue decomposition of the matrix are identical if a symmetric positive semi-definite matrix is given.

Compared with LRR, LRR-PSD and LRSC, the second difference is LRRSC and eLRRSC utilize the angular information of its principal directions of symmetric low-rank representation to build similarity graph under the unified subspace clustering framework. The utilization of the angular information in the proposed algorithm guarantees its robustness towards evaluating the memberships between each pair of data points. Experimental results further demonstrated that it will lead to a significant improvement on the clustering performance. Overall, we can argue that combination of the two improvements plays an important role in the low-rank representation and achieves satisfied results in the subspace clustering.

The last difference is how to decrease the computation cost among the competing methods. The computational complexity of some existing LRR-based methods which require iterative SVD operations becomes computationally impracticable in large-scale subspace clustering problems. Hence, pursuing a closed form solution is a positive way to avoid iterative SVD operations. Frobenius-norm plays a critical role in eLRRSC. By making use of the Frobenius norm, eLRRSC obtained a collaborative representation of high-dimensional data. Then the nuclear norm is employed in eLRRSC as a common surrogate for low-rank criterion. Finally, eLRRSC pursues a symmetric low-rank matrix preserving the low-dimensional subspace structures from the collaborative representation in a closed form solution. The experimental results illustrated that the computation costs of eLRRSC, SLRR and LRSC are much lower than that of other algorithms.

6. Conclusions

In this paper, we proposed a method that used a low-rank representation with a symmetric constraint to solve the problem of subspace clustering, using the assumption that high-dimensional data are approximately drawn from a union of multiple subspaces. In contrast with existing low-rank based algorithms, LRRSC integrates the symmetric constraint into the low-rankness property of high-dimensional data representation, which can be efficiently calculated by solving a convex optimization problem. The affinity matrix for spectral clustering can be obtained by further exploiting the angular information of the principal directions of the symmetric low-rank representation. This is a critical step towards understanding the memberships between high-dimensional data points. To speed up the optimization procedures of LRRSC, we also developed the efficient variant of LRRSC (eLRRSC), which

considers a closed form solution. Extensive experiments on benchmark databases showed that LRRSC and its variant produce very competitive results for subspace clustering compared with several state-of-the-art subspace clustering algorithms, and demonstrated its robustness when handling noisy real-world data.

Acknowledgments

The authors thank the anonymous reviewers for their thorough and valuable comments and suggestions.

References

- [1] W. Hong, J. Wright, K. Huang, Y. Ma, Multiscale hybrid linear models for lossy image representation, *IEEE Trans. Image Process.* 15 (12) (2006) 3655–3671.
- [2] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, *ICML*, 2010.
- [3] J. Ho, M. Yang, J. Lim, K. Lee, Clustering appearances of objects under varying illumination conditions, *CVPR*, 2003.
- [4] 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.
- [5] E. Elhamifar, R. Vidal, Sparse subspace clustering algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.
- [6] P. Favaro, R. Vidal, A closed form solution to robust subspace estimation and clustering, *CVPR*, 2011.
- [7] R. Vidal, P. Favaro, Low rank subspace clustering (LRSC), *Pattern Recognit. Lett.* (2013).
- [8] J. Yan, M. Pollefeys, A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate, in: *ECCV*, 2006, pp. 94–106.
- [9] S. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (10) (2010) 1832–1845.
- [10] L. Zappella, X. Lladó, E. Provenzi, J. Salvi, Enhanced local subspace affinity for feature-based motion segmentation, *Pattern Recognit.* 44 (2) (2011) 454–470.
- [11] D. Pham, S. Budhaditya, D. Phung, S. Venkatesh, Improved subspace clustering via exploitation of spatial constraints, in: *CVPR*, 2012, pp. 550–557.
- [12] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, N. Yu, Non-negative low rank and sparse graph for semi-supervised learning, *CVPR*, 2012.
- [13] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2) (2003) 218–233.
- [14] T. Boult, L. Brown, Factorization-based segmentation of motions, in: *IEEE Workshop on Proceedings of the Visual Motion*, 1991, pp. 179–186.
- [15] H. Qu, Z. Yi, A new algorithm for finding the shortest paths using pcnns, *Chaos, Solitons Fractals* 33 (4) (2007) 1220–1229.
- [16] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Ser. B* 58 (1) (1996) 267–288.
- [17] D. Donoho, For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solution, *Comm. Pure Appl. Math.* 59 (6) (2006) 797–829.
- [18] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization, in: *NIPS*, 2009, pp. 2080–2088.
- [19] E.J. Candès, Y. Plan, Matrix completion with noise, *Proc. IEEE* 6 (98) (2010) 925–936.
- [20] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, *arXiv preprint arXiv:1009.5055* (2011).
- [21] R. Vidal, A tutorial on subspace clustering, *IEEE Signal Process. Mag.* 28 (2) (2011) 52–68.
- [22] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (6) (1981) 381–395.
- [23] J.P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, *Int. J. Comput. Vision* 29 (3) (1998) 159–179.
- [24] V. Govindu, A tensor decomposition for geometric grouping and segmentation, *CVPR*, 2005.
- [25] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA), *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (12) (2005) 1945–1959.
- [26] J. Ho, M.Y.J. Lim, K. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, *CVPR*, 2003.
- [27] 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 Trans. Autom. Control* 54 (6) (2009) 1341–1347.
- [28] Z. Yi, Foundations of implementing the competitive layer model by lotka-volterra recurrent neural networks, *IEEE Trans. Neural Netw.* 21 (3) (2010) 494–507.
- [29] Y. Ni, J. Sun, S.Y. X. Yuan, L. Cheong, Robust low-rank subspace segmentation with semidefinite guarantees, in: *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, 2010, pp. 1179–1188.
- [30] F. Lauer, C. Schnörr, Spectral clustering of linear subspaces for motion segmentation, in: *ICCV*, 2009, pp. 678–685.
- [31] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, *J. Fourier Anal. Appl.* 14 (5–6) (2008) 877–905.
- [32] B. Recht, M. Fazel, P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Rev.* 52 (3) (2010) 471–501.
- [33] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [34] K. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, *Pacific J. Optim.* 15 (6) (2010) 615–640.
- [35] B. Liu, X. Yuan, Y. Yu, Q. Liu, D.N. Metaxas, Decentralized robust subspace clustering, in: *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 3539–3545.
- [36] C. Lu, J. Tang, M. Lin, L. Liang, S. Yan, Z. Lin, Correntropy induced l_2 graph for robust subspace clustering, in: *ICCV*, 2013, pp. 1801–1808.
- [37] X. Peng, Z. Yu, Z. Yi, H. Tang, Constructing the l_2 -graph for robust subspace learning and subspace clustering, *IEEE Trans. Cybern. PP* (99) (2016) 1–14.
- [38] 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 Trans. Neural Netw. Learn. Syst.* 27 (12) (2016) 2499–2512.
- [39] S. Du, Y. Ma, Y. Ma, Graph regularized compact low rank representation for subspace clustering, *Knowl.-Based Syst.* 118 (2017) 56C69.
- [40] U.V. Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (4) (2007) 395–416.
- [41] J. Shi, J. Malik, S. Sastry, Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 888–905.
- [42] X. Peng, Y. Rui, B. Zhang, H. Tang, Z. Yi, Fast low-rank representation based spatial pyramid matching for image classification, *Knowl.-Based Syst.* 90 (2015) 14–22.
- [43] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, S. Belongie, Beyond pairwise clustering, *CVPR*, 2005.
- [44] D. Zhou, J. Huang, B.B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding, in: *NIPS*, 2006, pp. 1601–1608.
- [45] J. Chen, H. Zhang, H. Mao, Y. Sang, Z. Yi, Symmetric low-rank representation for subspace clustering, *Neurocomputing* 173 (3) (2016) 1192–1202.
- [46] S. Wei, Z. Lin, Analysis and improvement of low rank representation for subspace segmentation, *arXiv:1107.1561* (2011).
- [47] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.* 43 (1) (2001) 129–159.
- [48] E.T. Hale, W. Yin, Y. Zhang, Fixed-point continuation for l_1 -minimization: methodology and convergence, *SIAM J. Optim.* 19 (3) (2008) 1107–1130.
- [49] N. Parikh, S. Boyd, Proximal algorithms, *Found. Trends Optim.* (2013) 1–96.
- [50] 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.
- [51] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (5) (2005) 684–698.
- [52] 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) (2001) 643–660.
- [53] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, *CVPR*, 2007.
- [54] D. Pham, S. Budhaditya, D. Phung, S. Venkatesh, Video segmentation by tracking many figure-ground segments, in: *ICCV*, 2013, pp. 2192–2199.