

# Structural Reweight Sparse Subspace Clustering

Ping Wang<sup>1</sup> · Bing Han<sup>1</sup> · Jie Li<sup>1</sup> · Xinbo Gao<sup>1</sup>

Published online: 1 June 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** Subspace clustering aims to segment a group of data points into a union of subspaces. Reweight sparse subspace clustering is one of the state-of-the-art algorithms which proposed an iterative weighted subspace clustering. The reweight matrix helps to improve the performance of the affinity matrix construction process but it easily falls into a local minimization. In this paper, we propose a structural reweight sparse subspace clustering algorithm which introduces the structural information into reweight subspace clustering. The structural information achieved in spectral clustering process is useful for the subsequent iterative optimization process which helps to obtain a better local minimization. The experimental results on the Extended Yale B, Hopkins 155, and COIL 20 datasets demonstrate that our algorithm achieves a better performance on subspace clustering problem.

**Keywords** Sparse subspace clustering · Motion segmentation · Structural information

## 1 Introduction

In many real-world applications, plenty of high-dimensional datasets need to be segmented into low-dimensional subspaces, e.g., computer vision, machine learning and image processing. Because the high-dimensional data processing always require higher computational capability and larger memory spaces. It has been observed that these high-dimensional datasets usually are distributed in intrinsic low-dimensional subspaces. For instance, face images taken under different illuminations lie in respective subspaces of multiple subjects. To solve the high-dimensional face recognition problem, Wan et.al. proposed the local graph embedding dimensional reduction method based on maximum margin criterion via Fuzzy Set, which was helpful to extract the representative features [1]. The motion trajectories of multiple moving objects could be approximated by several low-dimensional subspaces. The

---

✉ Bing Han  
bhan@xidian.edu.cn

<sup>1</sup> Visual Information Processing System Laboratory, School of Electronic Engineering, Xidian University, No. 2 South Taibai Road, Xi'an 710071, People's Republic of China

images of one object which is rotated several degrees lie in one subspace. Subspace clustering methods have been proposed to segment a collection of data points into a union of subspaces [2].

Subspace clustering methods is traditionally divided into four main categories: algebraic, iterative, statistical, and spectral clustering-based methods. Algebraic algorithms include matrix factorization-based approaches and algebraic-geometric approaches. Matrix factorization-based methods first obtain a rank- $r$  that  $r$  is the rank of input data matrix factorization of the data matrix, and then segment data points by the similarity matrix. So these approaches need to know the rank  $r$  and the subspaces should be independent. Generalized Principal Component Analysis (GPCA) is the typical algebraic-geometric method [3]. GPCA fits the high-dimensional data with a set of polynomials whose gradient at a data point gives the normal vector to the subspace including the point. So GPCA is sensitive to the outliers and its computation complexity grows exponential in terms of the dimensions of subspaces. Iterative methods first initialize the segmentation randomly, and fit a subspace using PCA. Then data points are assigned to its closest subspace. The iterative algorithms are sensitive to the initialization which will lead to a bad result. Statistical methods define proper generative models for subspaces. Mixtures of Probabilistic PCA (MPPCA) is based on Probabilistic PCA model and optimized by using Expectation Maximization (EM) algorithm [4]. MPPCA needs to know the number and dimensions of subspaces in advance. Agglomerative Lossy Compression (ALC) seeks for the minimization of the overall coding length of the segmented data [5]. The agglomerative procedure in ALC still need to be proved theoretically. Random Sample Consensus (RANSAC) is another statistical method [6], which randomly chooses  $d$  data points and computes a model for these points, then fit all data points to the model. RANSAC also needs to know the dimensions of subspaces. Spectral clustering-based methods are popular algorithms for subspace clustering. These methods divide subspace clustering into two steps: affinity matrix construction and spectral clustering. Spectral Local Best-fit Flats (SLBF) [7], Local Subspace Affinity (LSA) [8] and Locally Linear Manifold Clustering (LLMC) [9] cluster the data points based on the observation that a point and its neighbors always belong to the same subspace. SLBF and LSA can not handle the outliers near the subspace. When subspaces are independent, LLMC is hardly to select the number of nearest neighbors properly. SCC is based on the concept of polar curvature which is zero for data points in same subspace [10]. But its computation complexity grows fast. SSC (Sparse Subspace Clustering) introduces sparsity into the affinity matrix construction process [11]. Based on the hypothesis of data redundancy, the sparse constraint is used to separate data points of different subspaces. Because the data point could be represented by other data points of same subspace. In SSC, the  $l_1$ -norm as the convex relaxation of the  $l_0$ -norm is applied to achieve the sparsest self-expressive coefficient matrix. Then the affinity matrix could be obtained with the sparse coefficient matrix. Similar to SSC, Low Rank Representation (LRR) aims at finding the lowest-rank representation matrix [12]. Dong et al. proposed the method robust low rank subspace segmentation via joint  $\ell_{21}$ -norm minimization (LR-L21) to learn low rank representation by jointing the nuclear norm and  $\ell_{21}$ -norm minimization [13].

SSC only need the number of subspaces as a priori knowledge. Consequently, many researches have carried out a great deal of improved algorithms based on SSC. For instances, Lin et al. proposed BD-SSC and BD-LRR that pursue the block-diagonal structure by graph Laplacian constraint [14]. Inspired by enhanced  $l_1$  minimization [15], Reweighted Sparse Subspace Clustering (RSSC) was proposed for using iterative weighted  $l_1$  minimization as an alternative to the  $l_1$  minimization [16]. In order to do subspace clustering at a unified framework, Structured Sparse Subspace Clustering (SSSC) was proposed in [17]. Similarly, Wu et al. proposed a novel robust spectral subspace clustering based on least square regression

(RS<sup>2</sup>CLSR) to learn representation matrix through least square regression and the soft label was utilized to enhance the process [18].

*Contributions* RSSC proposes the iterative weighted  $l_1$  minimization to be the alternative to the  $l_1$  minimization. The main disadvantage of the method is that the optimization easily falls into a local minimization. It is observed that the structural information of the data helps to find a better local minimization. So our contributions can be summarized as follows.

1. We propose a structural reweight sparse subspace clustering model (SRSSC) by combining the structural sparse norm and weighted sparse norm into the cost function. 2. We prove that the structural information is used to help reweight subspace clustering find a better local minimum that more closely resembles the global minimum.

## 2 Related Works

*Self-expressive model* Let  $\mathbf{X} \in R^{D \times N}$  be the input data whose columns stand for the data points. Suppose that all data points are distributed in a union of subspaces  $S = \cup_{i=1}^n \{S_i\}$  whose dimension is  $d_i$ .  $n$  is the number of subspaces. The self-expressive model is proposed by SSC [11]. First, we need to assume that each data point  $x_j \in \mathbf{X} = \{x_1, x_2, \dots, x_N\}$  can be efficiently reconstructed by a combination of other data points. So  $x_j$  can be written as

$$x_j = \mathbf{X}\mathbf{z}_j, \quad \mathbf{z}_{jj} = 0, \tag{1}$$

where  $\mathbf{z}_j$  is the  $j$ -th column of the representation coefficients matrix  $\mathbf{Z} \in R^{N \times N}$ . The function could also be written in matrix form as

$$\mathbf{X} = \mathbf{X}\mathbf{Z}, \quad \text{diag}(\mathbf{Z}) = 0. \tag{2}$$

The constrain  $\text{diag}(\mathbf{Z}) = 0$  aims to eliminate the trivial solution that one data point is represented by itself. In order to obtain a sparse representation coefficient matrix  $\mathbf{Z}$ , the sparse norm is introduced into the cost function

$$\min \|\mathbf{Z}\|_0 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \text{diag}(\mathbf{Z}) = 0, \tag{3}$$

where  $l_0$ -norm counts the number of nonzero elements in a matrix. Through minimize the  $l_0$ -norm of one matrix, we could achieve the sparsest representation matrix. But the  $l_0$  minimization is the NP-hard problem. So the function should be relaxed as

$$\min \|\mathbf{Z}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \quad \text{diag}(\mathbf{Z}) = 0. \tag{4}$$

*Reweight sparse subspace clustering* As discussed in [15], the  $l_1$  minimization penalizes larger elements more heavily than smaller elements while the  $l_0$  minimization penalizes elements equally. An iterative weighted formulation of  $l_1$  minimization is designed to more democratically penalize nonzero elements. In [15], the log-sum surrogate function  $f(x) = \sum_{i=1}^n \log(|x_i| + \varepsilon)$  could closely resemble the  $l_0$  minimization. So in RSSC [16], the penalty term is added into the cost function of the sparse subspace clustering,

$$\min \|\mathbf{W} \odot \mathbf{Z}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \quad \text{diag}(\mathbf{Z}) = 0, \tag{5}$$

where  $\mathbf{W}$  is the weight matrix related to the representation coefficient matrix  $\mathbf{Z}$  and is updated by  $\mathbf{W} = \frac{\varepsilon_2}{|\mathbf{Z}| + \varepsilon_1}$ ,  $\varepsilon_1$  is used for numerical stability and to ensure that a zero-valued element will not lead to a nonzero estimate at the next iteration.

For the reason of the log-sum surrogate function is concave, RSSC easily falls into a local minimization.

### 3 Structural Reweight Sparse Subspace Clustering

Inspired by the RSSC algorithm, we propose the Sparse Reweight Sparse Subspace Clustering (SRSSC) which introduces the structural information into the objective function.

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{E}} \quad & \|\mathbf{W} \odot \mathbf{Z}\|_1 + \lambda_q \|\mathbf{Q} \odot \mathbf{Z}\|_1 + \lambda_e \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \mathbf{Z}^T \mathbf{1} = \mathbf{1}, \text{diag}(\mathbf{Z}) = 0, \end{aligned} \tag{6}$$

where  $\mathbf{E}$  is the outliers,  $\mathbf{Q}$  is the structural matrix. The parameters  $\lambda_q$  and  $\lambda_e$  balance the three terms in the objective function. As parameter  $\lambda_q$  is equal to 0, the optimization of function (6) is equivalent to standard RSSC.  $\odot$  denotes the element-wise product between two matrices.  $\mathbf{Z}^T \mathbf{1} = \mathbf{1}$  is the constraint of affine subspaces.

The update on weight matrix  $\mathbf{W}$  and structural matrix  $\mathbf{Q}$  could be written as

$$\mathbf{W}^{(k+1)} = \frac{\varepsilon_2}{|\mathbf{Z}^{(k+1)}| + \varepsilon_1} \tag{7}$$

$$\mathbf{Q}_{(t+1)}(i, j) = \begin{cases} 1 & l_{(t+1)}(i) = l_{(t+1)}(j) \\ 0 & l_{(t+1)}(i) \neq l_{(t+1)}(j) \end{cases}, \tag{8}$$

where  $l_{(t+1)}(i)$  and  $l_{(t+1)}(j)$  stand for the clustering labels of data point  $i$  and  $j$  after the  $t$ -th iteration. As shown in [16, 17], the updating of  $\mathbf{W}$  and  $\mathbf{Q}$  are not synchronous. Suppose we initialize the  $\mathbf{W}$  as all the elements to be one, and the  $\mathbf{Q}$  as zeros, the optimization is equivalent to a standard SSC.

The structural information is achieved after each iteration of weighted  $Z$  minimization. In detail, SRSSC could be divided into two steps: weighted  $Z$  minimization with fixed structural matrix  $\mathbf{Q}$  and structural matrix  $\mathbf{Q}$  updating with the result of spectral clustering. The two steps alternate until the formulation converged.

#### 3.1 The Weighted $Z$ Minimization

We fix the structural matrix  $\mathbf{Q}$  as  $\mathbf{Q}_{(t)}$  at the  $t$ -th iteration. The function (6) is changed into

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{E}} \quad & \|\mathbf{W} \odot \mathbf{Z}\|_1 + \lambda_q \|\mathbf{Q}_{(t)} \odot \mathbf{Z}\|_1 + \lambda_e \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \mathbf{Z}^T \mathbf{1} = \mathbf{1}, \text{diag}(\mathbf{Z}) = 0. \end{aligned} \tag{9}$$

The function (9) could be solved by Alternating Direction Method of Multipliers (ADMM) [19].

First, we introduce an auxiliary matrix  $\mathbf{A}$  into the function (9)

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{E}, \mathbf{A}} \quad & \|\mathbf{W} \odot \mathbf{Z}\|_1 + \lambda_q \|\mathbf{Q}_{(t)} \odot \mathbf{Z}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda_n}{2} \|\mathbf{X} - \mathbf{XA} - \mathbf{E}\|_F^2 \\ \text{s.t.} \quad & \mathbf{A}^T \mathbf{1} = \mathbf{1}, \mathbf{A} = \mathbf{Z} - \text{diag}(\mathbf{Z}). \end{aligned} \tag{10}$$

It's clearly that the solution of function (9) is consistent with function (10). Then two penalty terms should be added into the function

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{E}, \mathbf{A}} \quad & \|\mathbf{W} \odot \mathbf{Z}\|_1 + \lambda_q \|\mathbf{Q}_{(t)} \odot \mathbf{Z}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda_n}{2} \|\mathbf{X} - \mathbf{XA} - \mathbf{E}\|_F^2 \\ & + \frac{\mu}{2} \left( \|\mathbf{Z}^T \mathbf{1} - \mathbf{1}\|_2^2 + \|\mathbf{A} - (\mathbf{Z} - \text{diag}(\mathbf{Z}))\|_F^2 \right) \end{aligned}$$

$$s.t. \mathbf{A}^T \mathbf{1} = \mathbf{1}, \mathbf{A} = \mathbf{Z} - \text{diag}(\mathbf{Z}). \tag{11}$$

Note that adding the penalty terms does not change the optimal solution and makes the objective function to be strictly convex. Two Lagrangian multipliers  $\delta \in R^N$  and  $\Delta \in R^N$  are augmented for two constraints. The Lagrangian function of (11) could be written as

$$\begin{aligned} L(\mathbf{A}, \mathbf{Z}, \mathbf{W}, \mathbf{Q}, \mathbf{E}, \delta, \Delta) = & \|\mathbf{W} \odot \mathbf{Z}\|_1 + \lambda_q \|\mathbf{Q}_{(t)} \odot \mathbf{Z}\|_1 \\ & + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda_n}{2} \|\mathbf{X} - \mathbf{X}\mathbf{A} - \mathbf{E}\|_F^2 \\ & + \frac{\mu}{2} \left( \|\mathbf{Z}^T \mathbf{1} - \mathbf{1}\|_2^2 + \|\mathbf{A} - (\mathbf{Z} - \text{diag}(\mathbf{Z}))\|_F^2 \right) \\ & + \delta^T (\mathbf{A}^T \mathbf{1} - \mathbf{1}) + \text{tr} \left( \Delta^T (\mathbf{A} - (\mathbf{Z} - \text{diag}(\mathbf{Z}))) \right), \end{aligned} \tag{12}$$

where  $\text{tr}(\cdot)$  is the trace operator.

Note that  $(\mathbf{Z}^{(k)}, \mathbf{E}^{(k)}, \mathbf{A}^{(k)})$  are the optimization variables at  $k$ -th iteration.  $(\delta^{(k)}, \Delta^{(k)})$  are the Lagrange multipliers at  $k$ -th iteration.

*Update for A* Obtain  $\mathbf{A}^{(k+1)}$  by minimizing function (12) with respect to  $\mathbf{A}$  while  $(\mathbf{Z}^{(k)}, \mathbf{E}^{(k)}, \delta^{(k)}, \Delta^{(k)})$  are fixed. Setting the derivative of function (12) to be zero, we can obtain

$$\left( \lambda_z \mathbf{X}^T \mathbf{X} + \mu \mathbf{I} + \mu \mathbf{1}\mathbf{1}^T \right) \mathbf{A}^{(k+1)} = \lambda_z \mathbf{X}^T (\mathbf{X} - \mathbf{E}^{(k)}) + \mu \left( \mathbf{1}\mathbf{1}^T + \mathbf{Z}^{(k)} \right) - \delta^{(k)T} - \Delta^{(k)}. \tag{13}$$

*Update for Z* Obtain  $\mathbf{Z}^{(k+1)}$  by minimizing function (12) with respect to  $\mathbf{Z}$  while  $(\mathbf{A}^{(k+1)}, \mathbf{E}^{(k)}, \delta^{(k)}, \Delta^{(k)})$  are fixed. The update on  $\mathbf{Z}^{(k+1)}$  has a closed-form solution

$$\begin{aligned} \mathbf{Z}^{(k+1)} = & \tilde{\mathbf{Z}}^{(k+1)} - \text{diag} \left( \tilde{\mathbf{Z}}^{(k+1)} \right) \\ \tilde{\mathbf{Z}}^{(k+1)} \triangleq & \Gamma_{(1+\lambda_q)/\mu}^{\mathbf{Q}_{(t)}/\mu} \left( \mathbf{A}^{(k+1)} + \Delta^{(k)} / \mu \right), \end{aligned} \tag{14}$$

where  $\Gamma_{\eta}^{\mathbf{W}}(\cdot)$  is the shrinkage-thresholding operator which is defined as

$$\Gamma_{\eta}^{\mathbf{W}}(\mathbf{X}) = \max(|\mathbf{X}| - \eta \mathbf{W}, 0) \odot \text{sgn}(\mathbf{X}). \tag{15}$$

*Update for E* Obtain  $\mathbf{E}^{(k+1)}$  by minimizing function (12) with respect to  $\mathbf{E}$  while  $(\mathbf{A}^{(k+1)}, \mathbf{Z}^{(k+1)}, \delta^{(k)}, \Delta^{(k)})$  are fixed.

$$\mathbf{E}^{(k+1)} = \Gamma_{\lambda_e/\lambda_n} \left( \mathbf{X}\mathbf{A}^{(k+1)} - \mathbf{X} \right). \tag{16}$$

*Update for delta and Delta* Obtain  $\delta^{(k+1)}$  and  $\Delta^{(k+1)}$  with step size  $\mu$  while  $(\mathbf{Z}^{(k+1)}, \mathbf{E}^{(k+1)}, \mathbf{A}^{(k+1)})$  are fixed.

$$\begin{aligned} \delta^{(k+1)} = & \delta^{(k)} + \mu \left( \mathbf{A}^{(k+1)T} \mathbf{1} - \mathbf{1} \right) \\ \Delta^{(k+1)} = & \Delta^{(k)} + \mu \left( \mathbf{A}^{(k+1)} - \mathbf{Z}^{(k+1)} \right). \end{aligned} \tag{17}$$

*Update for W* Obtain  $\mathbf{W}^{(k+1)}$  while  $(\mathbf{Z}^{(k+1)}, \mathbf{E}^{(k+1)}, \mathbf{A}^{(k+1)})$  are fixed.

$$\mathbf{W}^{(k+1)} = \frac{\varepsilon_2}{|\mathbf{Z}^{(k+1)}| + \varepsilon_1}. \tag{18}$$

In summary, the optimization of function (9) with ADMM is summarized as Algorithm 1.

---

**Algorithm 1: Solve (9) via ADMM Algorithm**

---

**Input** : Data matrix  $\mathbf{X}$ , structural matrix  $\mathbf{Q}_{(t)}$  and maximal iteration number  $K$   
**Initialize**  $(\mathbf{A}^{(0)}, \mathbf{Z}^{(0)}, \mathbf{E}^{(0)}, \delta^{(0)}, \Delta^{(0)})$  is zeros,  $k = 0$ , and  $\mathbf{W}^{(0)} = 1$   
**While**  $k \leq K$  or not converged do  
     Update  $\mathbf{Z}^{(k)}, \mathbf{E}^{(k)}$ , and  $\mathbf{A}^{(k)}$   
     Update  $\delta^{(k)}, \Delta^{(k)}$   
      $k \leftarrow k + 1$   
**End while**  
**End**  
**Output:**  $\mathbf{Z}_{(t+1)} = \mathbf{Z}^*$

---

### 3.2 Spectral Clustering

After solving the weighted sparse optimization program in (9), we obtain the sparse representation matrix  $\mathbf{Z}_{(t+1)}$ . Next, we build a symmetric non-negative similarity matrix  $\mathbf{G} = \frac{1}{2} (|\mathbf{Z}_{(t+1)}| + |\mathbf{Z}_{(t+1)}^T|)$ . Then we apply spectral clustering to find cluster label  $\mathbf{L}_{(t+1)}$  which is essential to update  $\mathbf{Q}$ .

$$\mathbf{Q}_{(t+1)}(i, j) = \begin{cases} 1 & l_{(t+1)}(i) = l_{(t+1)}(j) \\ 0 & l_{(t+1)}(i) \neq l_{(t+1)}(j) \end{cases} . \tag{19}$$

In summary, the SRSSC program could be summarized as Algorithm 2.

---

**Algorithm 2: SRSSC**

---

**Input** : Data points  $\mathbf{X}$ , maximal iteration number  $T$  of spectral clustering, and maximal iteration number  $K$  of ADMM  
**Initialize**  $\mathbf{Q}_{(0)} = 0$ , and  $t = 0$ .  
**While**  $t < T$  or not converged do  
     Solve (10) via Algorithm 1.  
     Compute  $\mathbf{G}$  according to  $\mathbf{Z}_{(t+1)}$ .  
     Apply spectral clustering to  $\mathbf{G}$ , and obtain the label vector  $\mathbf{L}_{(t+1)}$ .  
     Update  $\mathbf{Q}_{(t+1)}$ .  
**End while**  
**End**  
**Output:**  $\mathbf{L}^*$

---

### 3.3 Convergence Analysis

As shown in [16], the update on weight matrix  $\mathbf{W}$  stems from the log-sum surrogate function. The log-sum heuristic function is concave and its minimization problem could be solved by an iterative linearization method [20]. In [15,21,22], the convergence of the reweight  $\ell_1$  minimization with log-sum surrogate function has been proved that the minimization could converge to a local minimum. While the Algorithm 1 converges to a local minimum, the proposed method may not converge. The experimental results show that our proposed method converges in practice under optimal parameters.

## 4 Experiments and Analysis

In this section, we evaluate the clustering performance of the proposed method in dealing with images clustering, and motion segmentation problems. The datasets include the Extended Yale Database B, COIL 20 and Hopkins 155.

*Experimental setup* In this paper, we compare with some state-of-the-art subspace clustering algorithms, i.e., SSC [11], LRR [12], SCC [10], LSA [8], BD-SSC [14], BD-LRR [14], LR-L21 [13], RSSC [16], SSSC [17], S-SSSC [23], and SSC-OMP [24]. S-SSSC based on SSSC defines an alternative real-valued structural matrix to binary structural matrix. SSC-OMP changes the optimization method into OMP (Orthogonal Matching Pursuit). For most of the algorithms mentioned above, the code is released by the authors based on Matlab platform. All the parameters keep the same as mentioned in their paper. And for BD-SSC, BD-LRR and LR-L21, we directly cite the results shown in their papers.

We use the clustering error to measure the performance of algorithms

$$error = \frac{1}{N} \sum_{i=1}^N \max(|p_i - g(q_i)|, 0), \quad (20)$$

where  $p_i$  is the ground truth label of point  $i$ , and  $q_i$  is the clustering label. The function  $g(\cdot)$  permutes clustering labels for matching the ground truth which could be Kuhn-Munkres method.

All the experiments are implemented in Matlab 2010b on a PC with Intel Core i3-3320 CPU at 3.30 GHz and 8.00 GB RAM.

*Experimental results on Hopkins 155 dataset* In this section, we evaluate the performance of our algorithm and other state-of-the-art algorithms on the Hopkins 155 dataset. Motion segmentation refers to the problem of motion trajectories segmentation according to their moving objects respectively. The Hopkins 155 dataset contains 155 video sequences where 120 of all videos contain two moving objects and 35 of all videos contain three moving objects, which means that the data lies in a union of 2 or 3 low-dimensional subspaces. In each video sequence, a set of  $N$  feature trajectories has been extracted and tracked by tracking algorithms. All trajectories data comprises a matrix  $\mathbf{X} = [x_1, x_2, \dots, x_N]$  and each column of  $\mathbf{X}$  is a  $2F$ -dimensional vector where  $F$  is the frame number.

For the motion segmentation experiments, we use the noisy variation, without the outlier term  $\mathbf{E}$  and with the affine constraint in the optimization process. As shown in [11], it gives the setting  $\lambda_Z = \alpha/\mu_Z$  where  $\mu_Z \triangleq \min_i \max_{j \neq i} |\mathbf{x}_i^T \mathbf{x}_j|$ . In SSC, we use  $\alpha = 800$  as shown in [11]. In LRR, we use  $\lambda = 4$  to achieve the best results without the post processing [12]. In SCC, the dimension  $d = 3$  [10]. In LSA, the nearest neighbors number  $kNeighbor = 6$  and dimension  $d = 4$  [8]. In RSSC, the numerical stability parameters  $\varepsilon_1 = 0.001$  and  $\varepsilon_2 = 0.015$  [16]. In SSSC, we use  $\alpha = 0.1$  [17]. In S-SSSC, we set  $\alpha = 0.2$  [23]. In BD-SSC, BD-LRR and LR-L21, we directly cite the best results in their papers [13, 14]. In SSC-OMP, we use  $K = 50$  [24]. In our algorithm, we set the parameters value as  $\varepsilon_1 = 0.001$ ,  $\varepsilon_2 = 0.015$  and  $\lambda_q = 0.1$ .

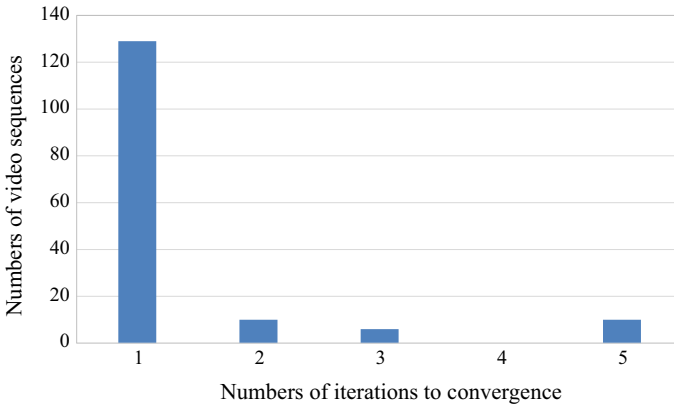
The final experimental results are represented in Table 1 where the average and median errors of subspace clustering are given.

First of all, our algorithm achieves the lowest clustering error in the Hopkins 155 dataset. The algorithms based on SSC have better performance than other algorithms which is a verification of SSC superiority over other subspace clustering algorithms based on spectral clustering. The clustering error of SSC-OMP is far worse than others. We consider the reason maybe that SSC-OMP could not deal with big data under the affine projection model

**Table 1** Clustering errors on the Hopkins 155 dataset with 2F-dimensional data points

Method	2 motions error (%)		3 motions error (%)		Total error (%)	
SSC	1.53	<b>0.00</b>	4.40	0.56	2.23	<b>0.00</b>
LRR	3.76	<b>0.00</b>	9.92	1.42	5.51	<b>0.00</b>
LR-L21	2.17	<b>0.00</b>	5.32	1.46	2.88	<b>0.00</b>
SCC	2.18	<b>0.00</b>	7.37	<b>0.19</b>	3.45	<b>0.00</b>
LSA	4.23	0.56	7.02	1.45	4.86	0.89
BD-SSC	–	–	–	–	2.90	<b>0.00</b>
BD-LRR	–	–	–	–	3.35	0.37
RSSC	0.75	<b>0.00</b>	1.68	0.48	1.08	<b>0.00</b>
SSSC	1.94	<b>0.00</b>	4.92	0.89	2.61	<b>0.00</b>
S-SSSC	1.65	<b>0.00</b>	4.27	0.61	2.24	<b>0.00</b>
SSC-OMP	31.46	33.09	44.59	45.57	34.42	35.71
Ours	<b>0.66</b>	<b>0.00</b>	<b>1.34</b>	0.21	<b>0.82</b>	<b>0.00</b>

The best results are in bold font



**Fig. 1** The histogram of numbers of iterations to convergence

successfully. Secondly, the clustering average error of our algorithm is reduced from 0.75 and 1.52 to 0.66% for the 2-motions case and from 1.68 and 4.41 to 1.34% for the 3-motion case respectively. Compared with RSSC and SSSC, we can find that the reweight sparse norm improves the performance of subspace clustering greatly. Compared with RSSC and ours, we can also find that ours reduces the cluster error greatly. So we can reach the conclusion that the reweight process is essential for the subspace clustering problem, and the structure information is also useful in achieving a better local minimization.

From the Fig. 1, we can find that our proposed algorithm mostly converges in 1 ~ 3 iterations for all 155 video sequences. It proves that our algorithm could converge to a local minimum in a few iterations.

*Experiments on Extended Yale B dataset* In this section, we evaluate the performance of our algorithm and others on the Extended Yale B dataset. The dataset contains the frontal face images of 38 subjects. Every subject is required to take 64 frontal face images under different illumination conditions. The size of each image is  $192 \times 168$ . To reduce the com-



**Table 2** Clustering errors on the Extended Yale B dataset.

Object number	2		3		5		8		10	
	Ave	Med	Ave	Med	Ave	Med	Ave	Med	Ave	Med
SSC	1.86	<b>0.00</b>	3.31	1.04	4.32	2.81	5.99	4.49	7.29	5.47
LRR	6.74	7.03	9.30	9.90	13.94	14.38	25.61	24.80	29.53	30.00
LR-L21	1.75	0.78	2.74	1.56	3.50	2.81	3.99	3.91	<b>4.32</b>	4.06
SCC	16.62	7.82	38.16	39.06	58.90	59.38	66.11	64.65	73.02	75.78
LSA	32.80	47.66	52.29	50.00	58.20	56.78	59.19	58.59	60.42	57.50
BD-SSC	3.90	–	17.70	–	27.50	–	33.20	–	39.53	–
BD-LRR	3.91	–	10.02	–	12.97	–	27.30	–	30.84	–
RSSC	0.49	<b>0.00</b>	0.92	<b>0.00</b>	1.81	0.63	3.70	1.76	5.73	3.44
SSSC	1.27	<b>0.00</b>	2.71	0.52	3.41	1.25	4.15	2.93	5.16	4.22
S-SSSC	0.94	<b>0.00</b>	2.24	1.04	4.31	2.50	6.23	5.27	11.41	7.03
SSC-OMP	0.96	0.99	2.45	1.56	3.91	2.50	6.11	3.52	7.79	4.69
Ours	<b>0.41</b>	<b>0.00</b>	<b>0.71</b>	<b>0.00</b>	<b>1.43</b>	<b>0.31</b>	<b>2.84</b>	<b>0.59</b>	4.43	<b>3.13</b>

The best results are in bold font

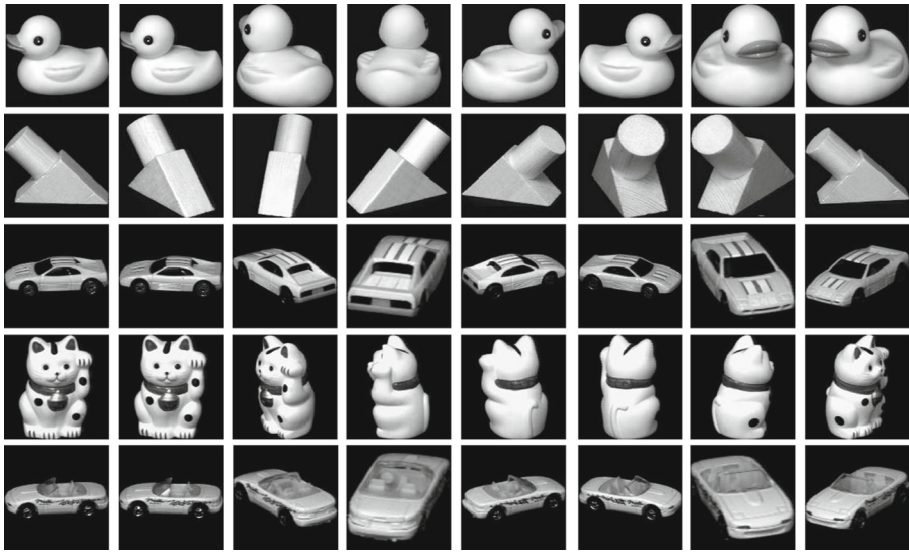
putational complexity, each face image is down-sampled to  $48 \times 42$  and represented as a 2016-dimensional vector. Following the protocol proposed in [11], we divide 38 subjects into 4 groups: 1–10, 11–20, 21–30, 31–38. We perform the combination of  $n \in \{2, 3, 5, 8, 10\}$  subjects for the first three groups and consider the all choices of  $n \in \{2, 3, 5, 8\}$  subjects for the final groups.

For the face clustering experiments, we consider the condition of noisy variation, with outlier  $\mathbf{E}$  and without the affine constrain. As shown in [11], outlier  $\mathbf{E}$  could deal with small errors due to noise in practice. And the parameter is setting  $\lambda_e = \alpha/\mu_e$  where  $\mu_e \triangleq \min_i \max_{j \neq i} \|\mathbf{x}_j\|_1$ . In SSC, we use  $\alpha = 20$  as shown in [11]. In LRR, we use  $\lambda = 0.18$  to achieve the same results in [12]. In SCC, the dimension  $d = 9$  [10]. In LSA, the nearest neighbors number  $kNeighbor = 8$  and dimension  $d = 4$  [8]. In RSSC, the numerical stability parameters  $\varepsilon_1 = 0.0002$  and  $\varepsilon_2 = 0.0014$  [16]. In SSSC [17], we use  $\alpha = 0.1$ . In S-SSSC [23], we use  $\alpha = 0.1$ . In BD-SSC, BD-LRR and LR-L21, we directly cite the best results in their papers [13, 14]. In SSC-OMP, we use  $K = 5$  [24].

In our algorithm, we use  $\lambda_q = 0.1$  and  $\varepsilon_1 = 0.006$ ,  $\varepsilon_2 = 0.0001$ . The experimental results are presented in Table 2.

From Table 2, we can find that our algorithm achieves the best performance for most conditions on the Extended Yale B dataset. Compared with RSSC and SSC, we can find that the reweight sparse norm improves the performance of subspace clustering remarkably. Compared with RSSC and our algorithm, the clustering errors are reduced from 0.49 to 0.41% for 2 objects case and from 5.73 to 4.43% for 10 objects case. When the object number increases, the numbers of outliers between clusters will increase synchronously.

*Experiments on COIL 20 dataset* In this section, we evaluate the performance of our algorithm and other state-of-the-art algorithms on COIL 20 dataset. COIL 20 contains 1440 images of 20 objects in which the background has been discard. Every object has been taken 72 images with a fixed camera while object is rotated through 360 degrees horizontally. The size of each image is  $32 \times 32$ , so the dimension of input data matrix  $\mathbf{X}$  is 1024. Some examples are shown in Fig. 2.



**Fig. 2** Examples from COIL 20 dataset

**Table 3** Clustering errors on COIL 20 dataset

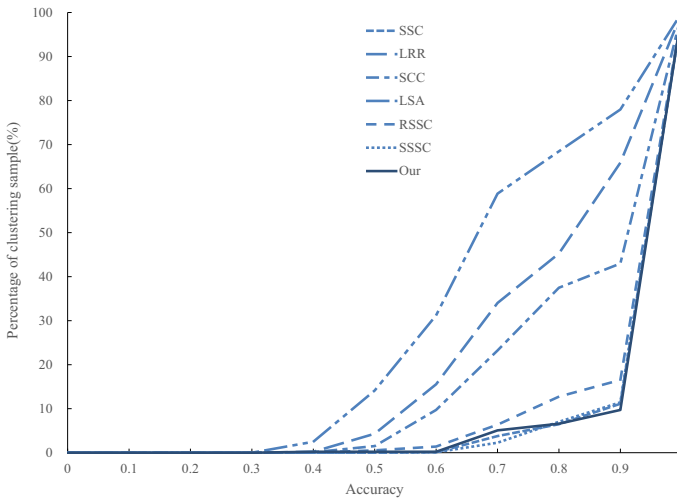
Object number	2		3		5		8	
	Ave	Med	Ave	Med	Ave	Med	Ave	Med
SSC	0.49	<b>0.00</b>	2.06	1.04	8.11	2.81	11.57	3.13
LRR	6.85	<b>0.00</b>	17.40	11.11	23.52	25.28	28.86	24.31
SCC	6.19	<b>0.00</b>	10.91	0.46	15.70	4.72	23.41	25.61
LSA	6.62	<b>0.00</b>	21.79	16.67	34.17	34.44	43.99	42.62
RSSC	0.03	<b>0.00</b>	2.72	<b>0.00</b>	6.23	<b>0.00</b>	<b>7.53</b>	<b>0.00</b>
SSSC	0.03	<b>0.00</b>	0.42	<b>0.00</b>	3.62	<b>0.00</b>	8.87	<b>0.00</b>
Ours	<b>0.01</b>	<b>0.00</b>	<b>0.079</b>	<b>0.00</b>	<b>3.19</b>	<b>0.00</b>	9.82	<b>0.00</b>

The best results are in bold font

Following the protocol proposed in [10], we first divide the 20 objects into two groups: 1–10, 11–20, and consider all the choices of  $n \in \{2, 3, 5, 8\}$  subjects in this experiment. As we consider the situation of noisy variation, with outlier  $\mathbf{E}$  and without the affine constrain. For all methods, we adjust the parameters refer to experimental parameters on the Extend Yale Dataset B dataset to achieve the best result. In SSC [11], we set  $\alpha = 10$ . In LRR [12], we set  $\lambda = 0.01$ . In SCC [10], we set the dimension  $d = 7$ . In LSA [8], the nearest neighbors number is set  $kNeight = 8$  and dimension  $d = 7$ . In RSSC [16], we set  $\varepsilon_1 = 0.1, \varepsilon_2 = 0.005$ . In SSSC [17], we use  $\alpha = 0.1$ .

In our algorithm, we use  $\lambda_q = 0.1$  and  $\varepsilon_1 = 0.2, \varepsilon_2 = 0.005$ . The experimental results are shown in Table 3.

In Table 3, we can find that all the state-of-the-art algorithms achieve competitive performance on COIL20 dataset, because the discarded images improve the dissimilarity. As we can see in Table 3, SRSSC obtains the best performance for the all cases except for 8 objects case. At the same time, we find that the clustering error increases sharply when



**Fig. 3** Percentage of clustering example whose clustering accuracy less than or equal to a given value

object number is 8. This phenomenon is due to the fast growing number of outliers. Structural information will disturb the optimization process in this situation. The clustering error increases from 7.53 to 9.82% for 8 objects cases. So our algorithm still need to be improved in the future.

Then, we calculate the percentage of clustering samples whose clustering accuracy is less than or equal to a certain value, which ranges in  $\{0, 0.1, \dots, 0.9, 1\}$ . The result are shown in Fig. 3.

From Fig. 3, we can discovery that most of clustering samples achieve accuracy higher than 0.6 for SSC, RSSC, SSSC and our algorithm. And the accuracy of more than 90% clustering samples is higher than 0.9. Generally speaking, our algorithm improve the entire accuracy on COIL20 dataset.

From the above three experiments as a whole, the clustering errors demonstrate the effectiveness of the proposed SRSSC method.

## 5 Complexity Comparisons

In this section, we compare the computational and memory complexity of our algorithm with four related algorithms. The results are shown in Table 4. For the proposed method, there are standard matrix operations in the weighted  $Z$  minimization process such as matrix multiplication and matrix inversion. Assuming that the size of input data matrix is  $m \times n$  and the maximal iteration numbers of ADMM is  $K$ , the computation complexity of the weighted  $Z$  optimization is  $O(Kn^3 + Kmn^2)$ . It takes time  $O(n^3)$  time to compute spectral clustering with  $Z$ . Then the total computational time of our proposed algorithm is  $O(Kn^3 + Kmn^2 + Tn^3)$  where  $T$  is the maximal iteration number of spectral clustering. The cost of memory is determined by the biggest matrices  $X$  and  $Z$  whose size are  $m \times n$  and  $n \times n$  respectively. The memory complexity of our algorithm is  $O(n^2 + mn)$ .

where  $n_s$  is the number of sampling iterations,  $d$  is the intrinsic dimension of input data, and  $c$  is the size of random sampling subsets of input data. The number of sampling iterations  $n_s$  will grow fast as the data points grows.

**Table 4** Computation and memory complexity comparison

Method	Computation complexity	Memory complexity
SSC	$O(Kn^3 + Kmn^2)$	$O(n^2 + mn)$
SCC	$O(n_s(d+1)^2mnc)$	$O(n(m+c))$
RSSC	$O(Kn^3 + Kmn^2)$	$O(n^2 + mn)$
SSSC	$O(Kn^3 + Kmn^2 + Tn^3)$	$O(n^2 + mn)$
Ours	$O(Kn^3 + Kmn^2 + Tn^3)$	$O(n^2 + mn)$

## 6 Conclusions

In this paper, we propose a *structural reweight subspace clustering* (SRSSC) algorithm. The reweight sparse norm is used to make a greater approximation to the  $\ell_0$ -norm, and the structural information is added to find a better local minimum. From the results on the Hopkins 155 dataset, the Extended Yale B dataset, and the COIL 20 dataset, our algorithm has much better performance than other subspace clustering algorithms. It can illustrate that our method is reliable. But the SRSSC still has some drawbacks. SRSSC is sensitive to the three parameters which bring about the problem of parameters selection. Because SRSSC has two iterative procedures, it is time consuming. In the future work, we will improve the robustness and speed up our algorithm.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China under Grant U1605252, in part by the National Key Research and Development Program of China under Grant 2016QY01W0200, the National Natural Science Foundation of China (41031064, 61572384, 61432014), China's postdoctoral fund first-class funding (2014M560752), Shanxi province postdoctoral science fund, The central university basic scientific research business fee (JBG150225), Shaanxi Key Technologies Research Program (2017KW-017).

## References

1. Wan M, Lai Z, Yang G et al (2017) Local graph embedding based on maximum margin criterion via fuzzy set. *Fuzzy Sets Syst* 318:120–131
2. Vidal R (2011) Subspace clustering. *IEEE Signal Process Mag* 28(2):52–68
3. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). *IEEE Trans Pattern Anal Mach Intell* 27(12):1945–1959
4. Archambeau C, Delannay N et al (2008) Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* 71(7):1274–1282
5. Derksen H, Ma Y, Hong W et al (2007) Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Trans Pattern Anal Mach Intell (TPAMI)* 29(9):1546–1562
6. Fischler M, Bolles R (1981) RANSAC random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24(6):381–395
7. Zhang T, Szlám A, Wang Y et al (2012) Hybrid linear modeling via local best-fit flats. *Int J Comput Vis* 100(3):217–240
8. Yan J, Pollefeys M (2006) A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: *European conference on computer vision*, pp 94–106. Springer, Berlin
9. Goh A, Vidal R (2007) Segmenting motions of different types by unsupervised manifold clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–6
10. Chen G, Lerman G (2009) Spectral curvature clustering (SCC). *Int J Comput Vis* 81(3):317–330
11. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell* 35(11):2765–2781

12. Liu G, Lin Z, Yan S et al (2013) Robust recovery of subspace structures by low-rank representation. *IEEE Trans Pattern Anal Mach Intell* 35(1):171–184
13. Dong W, Wu X J (2017) Robust low rank subspace segmentation via joint L21-norm minimization. *Neural Process Lett* 1–14
14. Feng J, Lin Z, Xu H, et al (2014) Robust subspace segmentation with block-diagonal prior. In: *IEEE conference on computer vision and pattern recognition*, pp 3818–3825
15. Candes E, Wakin M, Boyd S (2008) Enhancing sparsity by reweighted  $\ell_1$  minimization. *J Fourier Anal Appl* 14(5):877–905
16. Xu J, Xu K, Chen K et al (2015) Reweighted sparse subspace clustering. *Comput Vis Image Underst* 138:25–37
17. Li CG, Vidal R (2015) Structured sparse subspace clustering: a unified optimization framework. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 277–286
18. Wu Z, Yin M, Zhou Y, et al (2017) Robust spectral subspace clustering based on least square regression. *Neural Process Lett* 1–14
19. Boyd S, Parikh N, Chu E et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
20. Fazel M, Hindi H, Boyd S (2003) Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In: *Proceedings of American control conference*, pp 2156–2162
21. Xie Z, Hu J (2013) Reweighted L1-minimization for sparse solutions to underdetermined linear systems. In: *Proceedings of international congress on image and signal processing (CISP)*, pp 1660–1664
22. Chen X, Zhou W (2014) Convergence of the reweighted L1 minimization algorithm for L2-Lp minimization. *Comput Optim Appl* 59(1–2):47–61
23. Li CG, You C, Vidal R (2017) Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. *IEEE Trans Image Process* 26(6):2988–3001
24. You C, Robinson D, Vidal R (2016) Scalable sparse subspace clustering by orthogonal matching pursuit. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3918–3927

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.