Low-rank sparse subspace clustering with a clean dictionary

Cong-Zhe You, Zhen-Qiu Shu and Hong-Hui Fan

Abstract

Low-Rank Representation (LRR) and Sparse Subspace Clustering (SSC) are considered as the hot topics of subspace clustering algorithms. SSC induces the sparsity through minimizing the $l_1$-norm of the data matrix while LRR promotes a low-rank structure through minimizing the nuclear norm. In this paper, considering the problem of fitting a union of subspace to a collection of data points drawn from one more subspaces and corrupted by noise, we pose this problem as a non-convex optimization problem, where the goal is to decompose the corrupted data matrix as the sum of a clean and self-expressive dictionary plus a matrix of noise. We propose a new algorithm, named Low-Rank and Sparse Subspace Clustering with a Clean dictionary (LRS2C2), by combining SSC and LRR, as the representation is often both sparse and low-rank. The effectiveness of the proposed algorithm is demonstrated through experiments on motion segmentation and image clustering.

Keywords

Subspace clustering, low-rank representation, sparse representation

Introduction

The past few decades witnessed the data explosion, we have entered the era of big data, and an overwhelming amount of data is generated and collected every day. This poses a great challenge to process such large datasets, especially the datasets are usually very high-dimensional, even though the computer processing speed becomes faster and faster. The high-dimensionality of data not only increases the computing time but also decreases the performance due to the noise and insufficient samples in ambient space. However, the intrinsic dimension of these data is often much smaller than the dimension of the ambient space. This has motivated a variety of techniques to find the low-dimensional representations of high-dimensional data.

Subspace clustering is the problem of clustering data according to their potential low-dimensional subspaces. Is has been widely used in many fields, such as motion segmentation and face clustering in computer vision, hybrid system identification in control, community clustering in social networks. Many algorithms have been proposed, such as GPCA, Spectral Curvature Clustering (SCC), Low-Rank Representation (LRR) and its noisy variant LRSC, Sparse Subspace Clustering (SSC). Among them, SSC and LRR are considered as the state-of-the-art techniques, which are based on minimizing the nuclear norm and $l_1$-norm of the representation matrix respectively. Besides, the two algorithms are among the few subspace clustering algorithms supported by theoretical guarantees: both of them are proved to succeed when the subspaces are independent.

Sparse subspace clustering (SSC) uses $l_1$-norm minimization to find the sparsest representation of samples and form a sparse coefficient matrix. This can be considered as expressing each sample as a linear combination of the other samples. The affinity matrix is then constructed using the sparse coefficient matrix. SSC guarantees that the clustering will be successful when the samples are linearly independent, or that a disjoint clustering will be produced under certain conditions.
However, one major shortcoming of SSC is that it may not fully capture the advantages of global features, resulting in poorer performance when grossly corrupted or outlying observations are present. In addition, the SSC finds the sparsest representation of each sample individually, which leads to high computational cost.

Liu et al.\(^5\) recently presented an LRR algorithm for subspace clustering that relies on the same principle of representing a sample as a linear combination of others. LRR finds the lowest-rank representation of high-dimensional data. In addition to capturing the global structure of the data, LRR solves the convex optimization problem of nuclear norm minimization, which is considered as a surrogate for rank minimization. Several efforts have been made to improve the robustness of LRR. Chen and Yi\(^11\) extended the original LRR algorithm by integrating a symmetric constraint condition into the low-rank representation, thus avoiding the post-processing symmetrization step of constructing an affinity matrix. Yin et al.\(^12\) proposed a Laplacian regularized LRR to take into account the intrinsic geometrical structure of the data.

Although these methods are efficient and effective for solving specific clustering problems, an important shortcoming is that both LRR and SSC build an affinity matrix by using the techniques from low-rank or sparse representation. They assume that a point in a union of multiple subspaces admits a low-rank or sparse representation with respect to the dictionary formed by all the other data points, this is known as the Self-Expressiveness Property (SEP), i.e., \(D = DC\), where \(C\) is the coefficient matrix. These coefficients are then converted into symmetric and non-negative affinities, from which clustering result is obtained using spectral clustering algorithm. Although these algorithms claim that they also succeed when the data are corrupted by noise, it is not clear that the above algorithms are effective when using a corrupted dictionary.

Motivated by the above research works, in this paper, we propose a low-rank sparse subspace clustering method with a clean dictionary named as LRS2C2. The experiments conducted for different types of vision problems prove the effectiveness of the proposed LRS2C2 method. The contributions of this paper can be summarized as follows:

1. We integrate the low-rankness and sparsity constraints into a unified framework in order to get a better subspace clustering performance
2. In the general optimization framework, we try to decompose a data matrix as a self-expressive, noise-free data matrix and a noise matrix and use the noise-free data matrix as a clean dictionary. Consequently, the introduction of a clean dictionary to low-rank and sparse subspace clustering enriches the relationship between high-dimensional data samples corrupted by gross errors and provides more robust subspace clustering performance.

The rest of this paper is organized as follows. In section 2, we briefly review SSC and LRR algorithms. In section 3, we present the proposed LRS2C2 method and its implementation by using the alternating direction method of multipliers (ADMM),\(^13\) The experimental results are shown in section 4. Finally, we conclude this paper in section 5.

### Related works

In this section, we briefly introduce sparse representation, low-rank representation subspace clustering algorithm. For a given data matrix \(X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{D \times N}\), each column is a \(D\)-dimensional data vector drawn from the union of multiple linear subspaces \(\{S_i\}_{j=1}^N\) of unknown dimension. The task of subspace clustering is to cluster these data vectors into \(m\) corresponding subspaces.

#### Sparse subspace clustering

SSC was recently proposed as a solution to the subspace clustering problem.\(^9\) SSC finds the sparsest representation of the data by solving the following optimization problem

\[
\begin{align*}
\min_C & \quad \|C_i\|_0 \\
\text{s.t.} & \quad X = XC \quad \text{and} \quad \text{diag}(C) = 0
\end{align*}
\]

where \(C_i\) is the \(i\)-th column of \(C\), \(\| \cdot \|_0\) denotes the \(l_0\)-norm of a vector, i.e., the number of nonzero elements, and \(\text{diag}(C) \in \mathbb{R}^N\) is the vector of diagonal elements of \(C\). The optimal solution \(C\) of the above problem is called the sparsest representation of \(X\). Because this optimization problem is non-convex and NP-hard, the following \(l_1\)-norm convex optimization provides a surrogate for problem (1)

\[
\begin{align*}
\min_C & \quad \|C\|_1 \\
\text{s.t.} & \quad X = XC \quad \text{and} \quad \text{diag}(C) = 0
\end{align*}
\]

where \(\| \cdot \|_1\) denotes the \(l_1\)-norm of a matrix, i.e., the sum of the absolute value of all matrix elements. In the case of data corrupted by noise and sparse outliers, SSC assumes that each corrupted point can be represented by a linear combination of other corrupted points plus an outliers’ vector and a noise vector.
In particular, SSC solves the following optimization problem

\[
\min_C \|C\|_1 + \alpha \|E\|_1 + \frac{\beta}{2} \|G\|_F^2
\]

s.t. \( X = XC + E + G \) and \( \text{diag}(C) = 0 \)

where \( E \) is the matrix of sparse outlying entries and \( G \) is the noise matrix. The parameters \( \alpha > 0 \) and \( \beta > 0 \) determine the trade-off between the three terms in the optimization problem.

**Low-Rank representation**

Different from SSC, LRR finds the lowest-rank representation. LRR is formulated as the following rank minimization problem

\[
\min_C \text{rank}(C)
\]

s.t. \( X = AC \)

where the columns of \( A \) are a set of known bases or dictionary items. As it is difficult to solve the above optimization problem (2) due to the NP-hard nature of the rank function, a convex relaxation of this optimization problem is proposed as

\[
\min_C \|C\|_*
\]

s.t. \( X = AC \)

where \( \|C\|_* \) is the nuclear norm, defined as the sum of all singular values of \( C \), which is the convex envelope of the rank function. Considering the fact that samples are usually noisy or even grossly corrupted, a more reasonable objective for LRR can be expressed as

\[
\min_{C,E} \|C\|_* + \lambda \|E\|_{2,1}
\]

s.t. \( X = AC + E \)

where the \( l_{2,1} \)-norm is defined as \( \|E\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{d} e_{ij}^2} \) and the parameter \( \lambda \) is used to balance the effect of low-rank term and error term.

Therefore, in a post-processing step, one can use the coefficient \( C \) (by using SSC or LRR) to define an affinity matrix as \((C + C^T)/2\). The segmentation of the data is then obtained by applying spectral clustering to this affinity.

**Low-Rank sparse subspace clustering with a clean dictionary**

**Motivation**

It is well known that the corruptions and noise are ubiquitous in real-world data, meaning that certain observation may be arbitrarily corrupted. We cannot expect high dimensional data to always capture in well-controlled environments. Therefore, when using the original noisy data as a dictionary, the clustering performance may be seriously limited by a lack of robustness to corrupted observations. The importance of learning the dictionary from noisy data for classification and clustering has been shown in many studies.\(^{14-16}\)

As there are many complex variations to the noise encountered in observation, it is difficult to choose an appropriate strategy for removing all types of noise from real-world data. Different low-rank matrix recovery and completion techniques are suitable for different types of noise. PCA determines the best low-rank approximation when the original data are only mildly corrupted by Gaussian noise with small variance. The RPCA can be used to recover the discriminative low-rank dictionary from data corrupted by sparse noise. The objective function of RPCA is as follows

\[
\min \|A\|_* + \lambda \|E\|_1
\]

s.t. \( X = A + E \)

In this paper, we assume that given a corrupted data matrix \( D \in \mathbb{R}^{M \times N} \), we wish to decompose it as the sum of a self-expressive, noise-free data matrix \( A \), a noise matrix \( G \) and the sparse errors \( E \). We assume that the columns of the matrix \( A = \{a_1, a_2, \ldots, a_N\} \) are points in \( \mathbb{R}^M \). We also assume that \( A \) is self-expressive, which means that the clean data points can be expressed as linear combinations of themselves, i.e.

\[
a_i = \sum_{j=1}^{N} a_i e_{ij} \text{ or } A = AC
\]

where \( C = [e_{ij}] \) is the matrix of coefficients. This constraint aims to capture the fact that a point in a linear subspace can be expressed as a linear combination of other points in the same subspace. Therefore, we expect \( e_{ij} \) to be zero if point \( i \) and \( j \) are in different subspaces.

The proposed approach, which we call Low-Rank and Sparse Subspace Clustering with a Clean dictionary (LRS2C2), is based on solving the following optimization problem

\[
\min \|C\|_* + \lambda \|C\|_1 + \frac{\tau}{2} \|A - AC\|_F^2 + \frac{\xi}{2} \|G\|_F^2 + \gamma \|E\|_1
\]

s.t. \( D = A + G + E \)
where \( \|X\|_* = \sum \sigma_i(X) \), \( \|X\|_1 = \sum \|X_{ij}\| \) and \( \|X\|_F^2 = \sum_{ij} X_{ij}^2 \) are the nuclear norm, \( l_1 \)-norm and Frobenius norm of \( X \) respectively.

The proposed method combines the LRR and SSC, the first item of Eq.6 encourages \( X \) to be low-rank by minimizing \( \|C\|_\ast \), the second item encourages \( X \) to be sparse by minimizing \( \|C\|_1 \), the third item encourages \( A \) to be self-expressive by minimizing \( \|A - AC\|_F^2 \), the fourth item encourages the noise \( G \) to be as small as possible by minimizing \( \|G\|_F^2 \), and the last term ensures the error to be sparse by minimizing \( \|E\|_1 \).

### Optimization for LRS2C2

In this subsection, we derive a computationally efficient algorithm to solve the proposed objective function.

1. When \( E \) is fixed, let \((D - E) = USV^T\) be the SVD of the data matrix \((D - E)\). Then \( A \) can be expressed as

   \[
   A = UAV^T
   \]  

   where each entry of \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p) \) is obtained from each entry of \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n) \) as the solutions to

   \[
   \sigma = \psi(\lambda) = \begin{cases}
   \frac{\lambda + 1}{\sqrt{\lambda}} \lambda^{-3}, & \text{if } \lambda > \frac{1}{\sqrt{\tau}} \\
   \frac{\lambda + 1}{\sqrt{\lambda}} \lambda^{-3}, & \text{if } \lambda < \frac{1}{\sqrt{\tau}}
   \end{cases}
   \]  

   (10)

   that minimizes

   \[
   \phi(\lambda, \sigma) = \frac{\alpha}{2} (\sigma - \lambda)^2 + \begin{cases}
   \frac{\lambda + 1}{\sqrt{\lambda}} \lambda^{-3}, & \text{if } \lambda > \frac{1}{\sqrt{\tau}} \\
   \frac{\lambda + 1}{\sqrt{\lambda}} \lambda^{-3}, & \text{if } \lambda < \frac{1}{\sqrt{\tau}}
   \end{cases}
   \]  

   (11)

2. When \( A \) and \( C \) are fixed, the optimal solution for \( E \) satisfies

   \[
   -\alpha(D - A - E) + \gamma \text{sign}(E) = 0
   \]  

   (12)

   This equation has a closed-form solution

   \[
   E = S_\beta(D - A)
   \]  

   (13)

   where \( S(\cdot) \) is the shrinkage thresholding operator defined as

   \[
   S_\beta(x) = \begin{cases}
   x - \beta, & x > \beta \\
   0, & -\beta \leq x \leq \beta \\
   x + \beta, & x < -\beta
   \end{cases}
   \]  

   (14)

3. When \( A \) and \( E \) fixed, we optimize the \( C \) under the ADMM framework. To further separate the terms of variable \( C \), we add two auxiliary terms \( C_1 = C_2 = C \) to separate the two norms

   \[
   \min \|C_1\|_* + \lambda \|C_2\|_1 + \frac{\gamma}{2} \|A - AC\|_F^2
   \]

   s.t. \( C_1 = C_2 = C \)

   The augmented Lagrangian is

   \[
   L = \|C_1\|_* + \lambda \|C_2\|_1 + \frac{\gamma}{2} \|A - AC\|_F^2 + \|Y_1, C_1 - C\|
   + (Y_2, C_2 - C) + \frac{\mu}{2} (\|C_1 - C\|_F^2 + \|C_2 - C\|_F^2)
   \]  

   (16)

   where \( Y_1 \) and \( Y_2 \) are Lagrange multipliers and \( \mu > 0 \) is a penalty parameter.

   By fixing two of the variables, we can alternately update \( C_1 \), \( C_2 \) and \( C \), and then update the multipliers \( Y_1 \), \( Y_2 \). Problem (16) can be divided into the following sub-problems

   \[
   C = (\tau A^T A + 2\mu I)^{-1} (\tau A^T A + \mu C_1 + \mu C_2 - Y_1 - Y_2)
   \]

   \[
   C_1 = \arg \min \|C_1\|_* + \frac{\gamma}{2} \|C_1 - (C + \frac{Y_1}{\mu})\|_F^2
   \]

   \[
   C_2 = \arg \min \|C_2\|_1 + \frac{\gamma}{2} \|C_2 - (C + \frac{Y_2}{\mu})\|_F^2
   \]

The detailed optimization procedure is summarized in Algorithm 1. The three sub-problems in the corresponding steps are convex and have closed-form solutions. The second step can be solved by the singular value thresholding operator.\(^\text{17}\) The third step can be solved by the shrinkage thresholding operator.

After we get the coefficient matrix \( C \), we can define the affinity matrix of an undirected graph based on \( C \). In this paper, we do not use \( \|C_{ij}\| + |C_{ji}|/2 \) to compute the affinity matrix. As suggested in He et al.,\(^\text{16}\) we use the coefficient shrinkage and normalization operators as in Ma et al.,\(^\text{14}\) and construct the graph affinity matrix \( W \). This method takes into account of angular information of the principal directions of the representation. First, we consider the skinny SVD of \( C \), i.e., \( U^* \Sigma^* (V^*)^T \), and then define a matrix \( M = U^* (\Sigma^*)^{1/2} \) in which each column of \( U^* \) is assigned a weight by multiplying it by \( (\Sigma^*)^{1/2} \). The affinity matrix is defined as follows

\[
W_{ij} = \left( \frac{m_i^T m_j}{\|m_i\|_2 \|m_j\|_2} \right)^{2x}
\]
where \( m_i \) and \( m_j \) denote the \( i \)-th and \( j \)-th rows of \( M \). Finally, we use a spectral clustering to get the final result.

**Algorithm 1. LRS2C2**

**Input:** Data matrix \( D \), parameters \( \lambda, \tau, \alpha, \mu 
**Output:** ID\(_k\) = \( K \) Cluster Index

**Initialize:** \( C = C_1 = C_2 = 0, \ Y_1 = Y_2 = 0, \ \mu = 10^{-2}, \ \mu_{max} = 10^{10}, \rho = 1.1, \ \epsilon = 10^{-6} \)

1. **Low-Rank and Sparse Recovery**
   
   While not converged do
   1. Compute \( A \) by Eq. (9)
   2. Fix the other two variables and update \( C \) by
      \[
      C = (\tau A^T A + 2\mu I)^{-1}(\tau A^T A + \mu C_1 + \mu C_2 - Y_1 - Y_2)
      \]
   3. Fix the other two variables and update \( C_1 \) by
      \[
      C_1 = \arg \min \| C_1 \|_* + \frac{\tau}{2} \| C_1 - \left( C + \frac{Y_1}{\mu} \right) \|_F
      \]
   4. Fix the other two variables and update \( C_2 \) by
      \[
      C_2 = \arg \min \| C_2 \|_* + \frac{\tau}{2} \| C_2 - \left( C + \frac{Y_2}{\mu} \right) \|_F
      \]
   5. Update the multipliers:
      \[
      Y_1 = Y_1 + \mu (C_1 - C)
      \]
      \[
      Y_2 = Y_2 + \mu (C_2 - C)
      \]
   6. Update the parameter \( \mu \) by \( \mu = \min(\rho \mu, \mu_{max}) \)
   7. Check the convergence conditions
      \[
      \| C_1 - C \|_\infty < \epsilon \text{ and } \| C_2 - C \|_\infty < \epsilon
      \]
   
   End while

2. **Spectral Clustering**
   
   Compute ID\(_k\) using spectral clustering

**Experiments**

In this section, we evaluate the performance of LRS2C2 on two computer vision tasks: motion segmentation and image clustering. For the motion segmentation problem, we consider the Hopkins 155 dataset. For the image clustering problem, we consider the Extended YaleB dataset, AR dataset and MNIST datasets.

In the experiments, we use the subspace clustering error as a measure of performance, we compare LRS2C2 to state-of-the-art subspace clustering algorithms based on spectral clustering, such as LRR, \(^5\) LRSC, \(^7\) SSC\(^9\) and LSA.\(^{18}\) We choose these methods as a baseline because they have been shown to perform very well on the above tasks.

**Experiments on motion segmentation**

Motion segmentation refers to the problem of clustering a set of 2D point trajectories extracted from a video sequence into groups corresponding to different rigid-body motions. Here, the data matrix \( D \) is of dimension \( 2F \times N \), where \( N \) is the number of 2D trajectories and \( F \) is the number of frames in the video. Under the affine projection model, the 2D trajectories associated with a single rigid-body motion live in an affine subspace of \( R^{2F} \) of dimensional \( d = 1, 2 \) or \( 3 \). Therefore, the trajectories associated with \( n \) different moving objects lie in a union of \( n \) affine subspaces in \( R^{2F} \), and the motion segmentation problem reduces to clustering a collection of point trajectories according to multiple affine subspaces.

We use the Hopkins 155 motion segmentation database to evaluate the performance of LRS2C2 against that of other algorithms. The database consists of 155 sequences of two and three motions, where 120 of the videos have two motions and 35 of the videos have three motions. On average, in the dataset, each sequence of two motions have \( N = 256 \) feature trajectories and \( F = 30 \) frames while each sequence of three motions has \( N = 398 \) feature trajectories and \( F = 29 \) frames. For each sequence, the 2D trajectories are extracted automatically with a tracker and outlier are manually removed. Figure 1 shows some sample images with the feature points superimposed.

The results of applying subspace clustering algorithms to the dataset when we sue the original \( 2F \)-dimensional feature trajectories and when we project the data into a \( 4n \)-dimensional subspace (\( n \) is the number of subspaces) using PCA are shown in Tables 1 and 2, respectively.

From Tables 1 and 2, we can find that, in both cases, LRS2C2 obtains a small clustering error, outperforming the other algorithms. On the other hand, the clustering performance of different algorithms when using the \( 2F \)-dimensional feature trajectories or the \( 4n \)-dimensional PCA projections are close. This comes from the fact that the feature trajectories of \( n \) motions in a video almost perfectly lie in a \( 4n \)-dimensional linear subspace of the \( 2F \)-dimensional ambient subspace preserves the structure of the subspaces and the data; hence, for each algorithm, the clustering error in Table 1 is close to the error in Table 2. However, due to
the fact that the Hopkins 155 dataset has a relatively low-noise level, the improvement of the LRS2C2 is relatively minor.

Experiments on images clustering

In this subsection, we conduct the experiments on images clustering problems. We use three image datasets including the extended YaleB, AR and MNIST to test the performance of LRS2C2. Figure 2 shows some sample images of the three data sets. The brief information of the three datasets are summarized as follows:

The extended Yale B database contains 38 human subjects and around 64 near frontal images under different illuminations per individual, the face images of each subject correspond to a low-dimensional subspace. In this paper, we select the first 10 classes of extended Yale B database to form the test dataset which contains 640 images in total and each image is resized to $32 \times 32$ pixels.

AR database contains over 4000 face images of 126 individuals. For each individual, 26 pictures were taken in two sections (separated by two weeks) and each section contains 13 images. These images include front view of faces with different expressions, illuminations and occlusions. In this paper, we took a subset of the AR dataset containing five male subjects and five female subjects.
MNIST database has 10 subjects, corresponding to 10 handwritten digits, namely 0–9. We select a subset which consists of the first 100 samples of each subject’s training dataset.

We first applied the proposed algorithm to the full original images without any artificial corruption or missing entries. Some example images from the extended Yale B database are given in Figure 3(a). Next, we considered artificial occlusion, namely random pixel corruptions. To simulate random pixel corruptions, we replaced a certain percentage of pixels in the matrix of the image with uniformly distributed random values in the range [0, 1]. To ensure that the noise was sparse, the corrupted location and pixels were selected at random and were unknown to the algorithms. The percentage of corrupted pixels was varied from 5 to 20% in steps of

| Algorithm     | LSA | SSC | LRR | LRSC | LRS2C2 |
|---------------|-----|-----|-----|------|--------|
| Two motions   |     |     |     |      |        |
| Mean          | 4.23| 2.07| 4.10| 2.57 | 1.80   |
| Median        | 0.56| 0.00| 0.22| 0.00 | 0.00   |
| Three motions |     |     |     |      |        |
| Mean          | 7.02| 5.27| 9.89| 6.64 | 4.70   |
| Median        | 1.45| 0.40| 0.22| 1.76 | 0.56   |
| All           |     |     |     |      |        |
| Mean          | 4.86| 2.79| 5.41| 3.47 | 2.45   |
| Median        | 0.89| 0.00| 0.53| 0.09 | 0.00   |

| Algorithm     | LSA | SSC | LRR | LRSC | LRS2C2 |
|---------------|-----|-----|-----|------|--------|
| Two motions   |     |     |     |      |        |
| Mean          | 3.61| 2.14| 4.83| 2.57 | 1.84   |
| Median        | 0.51| 0.00| 0.26| 0.00 | 0.00   |
| Three motions |     |     |     |      |        |
| Mean          | 7.65| 5.29| 9.89| 6.62 | 4.59   |
| Median        | 1.27| 0.40| 0.26| 1.76 | 0.48   |
| All           |     |     |     |      |        |
| Mean          | 4.52| 2.85| 5.98| 3.47 | 2.46   |
| Median        | 0.57| 0.00| 0.59| 0.00 | 0.00   |

Figure 3. Representative examples in the extended Yale database B: (a) sample images under different illumination conditions, (b) sample images with random pixel corruptions, (c) sample images after using RPCA.
Figure 3(b) shows some examples of the images with random pixel corruptions. For a fair comparison, we preprocessed the images for the other algorithms by removing noise using the RPCA\textsuperscript{19} algorithm. Figure 3(c) shows some representative examples by applying the RPCA algorithm to face images with random pixel corruptions.

We executed each algorithm 10 times and report the mean clustering error and standard deviation in Tables 3 to 5.

From the experimental results, it is clear that the proposed LRS\textsuperscript{2C2} method achieves the lowest clustering error in the three datasets. The clustering results confirm that the LRS\textsuperscript{2C2} can significantly improve

### Table 3. Clustering error (%) of different algorithm on the extended YaleB dataset by random pixel corruptions (the parameters for LRS\textsuperscript{2C2} are $\lambda = 0.2, \tau = 0.2, \alpha = 0.03, \gamma = 0.01$).

| Corruption ratio (%) | Error | LSA | SSC | LRR | LRSC | LRS\textsuperscript{2C2} |
|----------------------|-------|-----|-----|-----|------|------------------|
| 0                    | Mean  | 36.00 | 32.80 | 47.50 | 34.65 | 24.06 |
|                      | Std.  | 2.58 | 3.17 | 2.73 | 1.47 | 0.83 |
| 5                    | Mean  | 42.45 | 34.97 | 48.84 | 35.24 | 28.97 |
|                      | Std.  | 3.00 | 3.68 | 1.60 | 1.72 | 0.94 |
| 10                   | Mean  | 44.27 | 35.08 | 50.13 | 35.58 | 30.19 |
|                      | Std.  | 1.91 | 3.30 | 2.81 | 0.97 | 0.26 |
| 15                   | Mean  | 47.23 | 36.20 | 51.13 | 36.34 | 31.67 |
|                      | Std.  | 1.29 | 2.52 | 3.10 | 2.62 | 2.28 |
| 20                   | Mean  | 51.27 | 38.32 | 51.90 | 38.97 | 32.58 |
|                      | Std.  | 3.32 | 3.42 | 3.03 | 1.43 | 1.71 |

### Table 4. Clustering error (%) of different algorithm on the AR dataset by random pixel corruptions (the parameters for LRS\textsuperscript{2C2} are $\lambda = 0.2, \tau = 0.04, \alpha = 0.03, \gamma = 0.01$).

| Corruption ratio (%) | Error | LSA | SSC | LRR | LRSC | LRS\textsuperscript{2C2} |
|----------------------|-------|-----|-----|-----|------|------------------|
| 0                    | Mean  | 37.86 | 23.57 | 2.86 | 12.83 | 2.10 |
|                      | Std.  | 1.27 | 1.09 | 0.00 | 0.87 | 0.91 |
| 5                    | Mean  | 44.05 | 26.93 | 2.86 | 13.69 | 2.25 |
|                      | Std.  | 0.65 | 1.58 | 0.30 | 0.61 | 0.69 |
| 10                   | Mean  | 44.71 | 27 | 2.86 | 15.34 | 2.33 |
|                      | Std.  | 2.02 | 1.05 | 0.00 | 0.94 | 0.71 |
| 15                   | Mean  | 44.75 | 27.45 | 2.86 | 19.81 | 2.49 |
|                      | Std.  | 2.60 | 1.27 | 0.00 | 1.08 | 0.55 |
| 20                   | Mean  | 43.64 | 29.36 | 3.07 | 2.98 | 2.56 |
|                      | Std.  | 0.75 | 3.54 | 0.35 | 0.26 | 0.33 |

### Table 5. Clustering error (%) of different algorithm on the MNIST dataset by random pixel corruptions (the parameters for LRS\textsuperscript{2C2} are $\lambda = 0.2, \tau = 10^{-3}, \alpha = 0.03, \gamma = 0.01$).

| Corruption ratio (%) | Error | LSA | SSC | LRR | LRSC | LRS\textsuperscript{2C2} |
|----------------------|-------|-----|-----|-----|------|------------------|
| 0                    | Mean  | 36.00 | 32.80 | 47.50 | 33.52 | 28.42 |
|                      | Std.  | 2.58 | 3.17 | 2.73 | 1.74 | 2.01 |
| 5                    | Mean  | 42.45 | 34.97 | 48.84 | 36.52 | 30.56 |
|                      | Std.  | 3.00 | 3.68 | 1.60 | 1.68 | 2.30 |
| 10                   | Mean  | 44.27 | 35.08 | 50.13 | 37.29 | 31.04 |
|                      | Std.  | 1.91 | 3.30 | 2.81 | 1.75 | 1.98 |
| 15                   | Mean  | 47.23 | 36.20 | 51.13 | 38.68 | 33.75 |
|                      | Std.  | 1.29 | 2.52 | 3.10 | 1.46 | 1.60 |
| 20                   | Mean  | 51.27 | 38.32 | 51.90 | 39.12 | 34.90 |
|                      | Std.  | 3.32 | 3.42 | 3.03 | 1.59 | 2.02 |
the accuracy of image clustering. In these experiments, we adopt many low-rank based methods, the results also show low-rank based method outperformed other methods. For the original dataset without artificial corruption, as the LRS2C2 can recovery a clean dictionary from the input data matrix, it has the advantage than other methods, it achieves the lowest clustering error. Among the compared methods, LRSC also has the ability to use a clean dictionary, however, it only reveals the low-rankness of the data. LRS2C2 combines the low-rankness and sparsity of the representation of the data matrix, the solution is able to benefit from both the better separation induced by the $l_1$-norm factor and the relatively denser connections promoted by the nuclear norm factor.

When the dataset was corrupted, the experimental results show that LRS2C2 consistently outperformed all the other methods, particularly for larger percentages of corrupted pixels. And as the increase of the percentage of the corruption, the proposed LRS2C2 method also has a relative advantage than other methods. These results clearly imply that LRS2C2 is more robust than the other algorithms.

Convergence analysis

In this section, we examine the convergence of our algorithm. The penalty parameter $\mu$ is upper bounded, which is guaranteed by step (2)(3)(4) in Algorithm 1. The cost function in (8) is bounded and non-increasing at each alternating minimization step [36]. Hence the sequence ($C_1$, $C_2$ and $C$) in Algorithm 1 will converge. In addition, a convergence analysis of the ADMM for non-convex problems is provided in Hong et al. (2016). Thus, the convergence of the algorithm is ensured. In order to verify the convergence of the proposed LRS2C2 algorithm, in this section, we run experimentation on the extended Yale B dataset, and draw the target function values for each iteration of the algorithm as shown in Figure 4. As you can see from the graph, the objective function decreases rapidly and converges to a certain value.

Conclusion

To effectively solve the subspace clustering problem in the case of data corrupted by noise, in this paper, we propose the LRS2C2 method, which decomposes the given data matrix to a clean data dictionary and a noise matrix. We also combine the low-rank representation and sparse representation in a unified framework to efficiently obtain the clustering results. Experiments conducted on motion segmentation and image clustering problems showed the effectiveness of our algorithm and its superiority over the state-of-the-art methods.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Natural Science Foundation of China (Grant No. 61902160, 61806088), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No.19KJB520006) and the foundation of Changzhou Science and Technology Plan (Applied Basic Research) (Grant No. CJ20190076).

ORCID iDs

Cong-Zhe You https://orcid.org/0000-0003-0906-7888
Zhen-Qiu Shu https://orcid.org/0000-0001-5737-3383

References

1. Basri R and Jacobs DW. Lambertian reflectance and linear subspaces. *IEEE Trans Pattern Anal Mach Intell* 2003; 25: 218–233.
2. Lee KC, Ho J and Kriegman DJ. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans Pattern Anal Mach Intell* 2005; 27: 684–698.
3. Vidal R, Ma Y and Sastry S. Generalized principal component analysis (GPCA). *IEEE Trans Pattern Anal Mach Intell* 2005; 27.
4. Chen G and Lerman G. Spectral curvature clustering (SCC). *Int J Comput Vis* 2009; 81: 317–330.
5. Liu G, Lin Z and Yu Y. Robust subspace segmentation by low-rank representation. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, June 21-24, 2010, pp. 663–670. Haifa, Israel: Omnipress.

Figure 4. Value of objective function versus iteration.
6. Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation. *IEEE Trans Pattern Anal Mach Intell* 2013; 35: 171–184.

7. Favaro P, Vidal R and Ravichandran A. A closed form solution to robust subspace estimation and clustering. In: *Computer vision and pattern recognition (CVPR)* 2011, June 2011, pp.1801–1807. New York, NY: IEEE.

8. Elhamifar E and Vidal R. Sparse subspace clustering. In: 2009 *IEEE conference on computer vision and pattern recognition*, 2009 June, pp.2790–2797. New York, NY: IEEE.

9. Elhamifar E and Vidal R. Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell* 2013; 35: 2765–2781.

10. Vidal R, Tron R and Hartley R. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *Int J Comput Vis* 2008; 79: 85–105.

11. Chen J and Yi Z. 2014. Subspace clustering by exploiting a low-rank representation with a symmetric constraint. arXiv:1403.2330.

12. Yin M, Gao J and Lin Z. Laplacian regularized low-rank representation and its applications. *IEEE Trans Pattern Anal Mach Intell* 2013; 38: 504–517.

13. Wen Z, Goldfarb D and Yin W. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math Prog Comp* 2010; 2: 203–230.

14. Ma L, Wang C, Xiao B, et al. Sparse representation for face recognition based on discriminative low-rank dictionary learning. In: 2012 *IEEE conference on computer vision and pattern recognition (CVPR)*, June 2012, pp.2586–2593. New York, NY: IEEE.

15. Yang M, Zhang L, Feng X, et al. Fisher discrimination dictionary learning for sparse representation. In: 2011 *IEEE international conference on computer vision (ICCV)* November 2011, pp.543–550. New York, NY: IEEE.

16. He W, Chen JX and Zhang W. Low-rank representation with graph regularization for subspace clustering. *Soft Computing* 2015; 21: 1569–1581.

17. Cai JF, Candès EJ and Shen Z. A singular value thresholding algorithm for matrix completion. *SIAM J Optim* 2010; 20: 1956–1982.

18. Yan J and Pollefeys M. A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: *European conference on computer vision–ECCV* 2006, Graz, Austria, 2006, pp.94–106. Berlin, Heidelberg: Springer.

19. Candès EJ, Li X, Ma Y, et al. Robust principal component analysis? *J Acm* 2011; 58: 1–37.

20. Hong M, Luo Z and Razaviyayn M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J Optim* 2016; 26: 337–364.