Rigidly Self-Expressive Sparse Subspace Clustering

Linbo Qiao[1,2(✉)], Bofeng Zhang[1], Yipin Sun[1], and Jinshu Su[1,2]

[1] College of Computer, National University of Defense Technology, Changsha 410073, China
[2] National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China
{qiao.linbo,bfzhang,yipinsun,sjs}@nudt.edu.cn

Abstract. Sparse subspace clustering is a well-known algorithm, and it is widely used in many research field nowadays, and a lot effort has been contributed to improve it. In this paper, we propose a novel approach to obtain the coefficient matrix. Compared with traditional sparse subspace clustering (SSC) approaches, the key advantage of our approach is that it provides a new perspective of the self-expressive property. We call it rigidly self-expressive (RSE) property. This new formulation captures the rigidly self-expressive property of the data points in the same subspace, and provides a new formulation for sparse subspace clustering. Extensions to traditional SSC could also be cooperating with this new formulation. We present a first-order algorithm to solve the nonconvex optimization, and further prove that it converges to a KKT point of the nonconvex problem under certain standard assumptions. Extensive experiments on the Extended Yale B dataset, the USPS digital images dataset, and the Columbia Object Image Library shows that for images with up to 30 % missing pixels the clustering quality achieved by our approach outperforms the original SSC.

Keywords: Sparse subspace clustering · Rigidly self-expressive · Optimization method

1 Introduction

Subspace clustering naturally arises with the emergence of high-dimensional data. It refers to the problem of finding multiple low-dimensional subspaces underlying a collection of data points sampled from a high-dimensional space and simultaneously partitioning these data into subspaces. Making use of the low intrinsic dimension of input data and the assumption of data lying in a union of linear or affine subspaces, subspace clustering assigns each data point to a subspace, in which data points residing in the same subspace belong to the same cluster. Subspace clustering has been applied to various areas such as computer vision [11],

L. Qiao—The work was partially supported by the National Natural Science Foundation of China under Grant No. 61303264 and Grant No. 61202482.

H. Cao et al. (Eds.): PAKDD 2016 Workshops, LNAI 9794, pp. 101–114, 2016.
DOI: 10.1007/978-3-319-42996-0_9

signal processing [16], and bioinformatics [13]. In the past two decades, numerous algorithms to subspace clustering have been proposed, including K-plane [3], GPCA [24], Spectral Curvature Clustering [4], Low Rank Representation (LRR) [15], Sparse Subspace Clustering (SSC) [7], *etc.* Among these algorithms, the recent work of Sparse Subspace Clustering (SSC) has been recognized to enjoy promising empirical performance.

In this paper, we propose a novel approach beyond SSC to obtain the coefficient matrix. Compared with the approaches mentioned above, the key advantage of our approach is that it provides a new perspective of the self-expressive property. We call it rigidly self-expressive (RSE) property. The model that we build for subspace clustering incorporates rigidly self-expressive property to obtain the coefficient matrix. This formulation generalizes traditional SSC, and captures the expressive property of the data points in the same subspace. We present a first-order algorithm to solve the nonconvex optimization, and further prove that it converges to a KKT point of the nonconvex problem under certain standard assumptions. Extensive experiments on the Extended Yale B dataset [14], the USPS digital images dataset [12], and the Columbia Object Image Library (COIL20) [17] show that for images with up to 30 % missing pixels the clustering quality achieved by our approach outperforms the original SSC.

2 Sparse Subspace Clustering

Assume there are N data points $y_i \in \mathbb{R}^{d_i}, i = 1, \cdots, N$ lying in the union of the linear or affine subspaces $S_i, i = 1, ..., n$, each of dimension $d_i \leq M$ for $i = 1, \cdots, n$. The observed data is reshaped as an matrix with each data point as one column in it

$$Y = [y_1 \ \cdots \ y_N] = [Y_1 \ \cdots \ Y_N]\Gamma \tag{1}$$

where $Y \in \mathbb{R}^{M \times N}$ with each data point as one column of the matrix and $\Gamma \in \mathbb{R}^{N \times N}$ is an unknown permutation matrix. The subspace clustering focus on the number of subspaces and the membership of each data point to its corresponding subspace.

The SSC algorithm [8] assumes the data can be sparsely represented by the data in a union of subspaces, which is called *self-expressive*. Based on the observation, we can write each data point as

$$y_i = Yc_i, \quad c_{ii} = 0, \tag{2}$$

where $c_i = [c_{i1} \ c_{i2} \ \cdots \ c_{iN}]$ and the constraint $c_{ii} = 0$ eliminates the trivial solution of writing a point as a linear combination of itself. When the data points are well distributed inside each subspace, the relationship among the data points is theoretically analyzed in [9], which formulate the problem as

$$\begin{aligned} &\min_{C} \|C\|_1 \\ &s.t. \ Y = YC, \quad \mathrm{diag}(C) = 0. \end{aligned} \tag{3}$$

where $C = [c_1\ c_2\ \cdots\ c_N] \in \mathbb{R}^{N\times N}$ is the matrix whose i-th column corresponds to the sparse representation of y_i, c_i, and $\mathrm{diag}(C) \in \mathbb{R}^N$ is the vector of the diagonal elements of C.

Considering there are outliers and noise in real world data, the model is extended as a more practical one

$$\begin{aligned} &Y = YC + E + Z, \\ &\mathrm{diag}(C) = 0. \end{aligned} \tag{4}$$

where E denotes the matrix form of outlying entries which has only a few large non-zero elements, and Z denotes the matrix form of noise,the formulation (3) is extended to handle the real-world problems with considering

$$\begin{aligned} \min_{(C,E,Z)}\ & \|C\|_1 + \lambda_e\|E\|_1 + \tfrac{\lambda_z}{2}\|Z\|_F^2 \\ s.t.\quad & Y = YC + E + Z, \\ & \mathrm{diag}(C) = 0. \end{aligned} \tag{5}$$

In the formulation, $Y \in R^{M\times N}$ is the original data set without any noise and outliers, Z and E correspond to noise and sparse outliers respectively, $\lambda_e > 0$, $\lambda_z > 0$ are two trade-off parameters balancing the three terms in the objective function, and the $\mathrm{diag}(C)$ is a vector whose elements are the diagonal entries of the matrix C.

For the case that the data corrupted by outliers, Candes *et al.* [22,23] give geometric insights which show that the method would succeed when the dataset is corrupted with possibly overwhelmingly many outliers. For the case that the data is corrupted by noise Z Wang and Xu [25] show that when the amount of noise is small enough, the subspaces are sufficiently separated, and the data are well distributed, the matrix of coefficients gives the correct clustering with high probability.

3 Rigidly Self-Expressive SSC

The sparse subspace clustering algorithm mainly takes advantage of the *self-expressiveness* property of the data. Specifically speaking, each data point in a union of subspaces can be efficiently expressed as a combination of other points in the same dataset.

However, in many real-world problems, data are always corrupted by noise and sparse outlying entries at the same time due to measurement noise or data collection techniques. For instance, in clustering of human faces, images can be corrupted by errors due to speculation, cast shadow, and occlusion [26]. Similarly, in the motion segmentation problem, because of the malfunctioning of the tracker, feature trajectories can be corrupted by noise or can have entries with large errors [28]. In such cases, the data do not lie perfectly in a union of subspaces, which means that *the noise and outlier free data* $\hat{Y}$ *does not exactly lie in the union of the subspaces of observed data* Y, and the observed data points Y lie in a space contaminated with outlier E and noise Z. Considering

the self-expressiveness in the formulation (5), it is not appropriate to use the linear combination YC to represent the data points $\hat{Y}$.

In this paper we improve the SSC formulation in [9], and present a rigidly self-expressive formulation, named Rigidly Self-Expressive Sparse Subspace Clustering (RSE-SSC), using a sparse linear combination $\hat{Y}C$ of the original subspaces to represent the original data points $\hat{Y}$ without noise and outliers, and the observed data set is expressed as the sum of the original data set and noise and outliers. Here we consider the optimization problem with the same objective value function with the original SSC, but totally different constraints on data points. Letting $\hat{Y}$ the data points lie in the original subspaces without outliers nor noise, and the constraints

$$\begin{aligned} &Y = \hat{Y} + E + Z, \\ &\hat{Y} = \hat{Y}C, \\ &\mathrm{diag}(C) = 0. \end{aligned} \tag{6}$$

are used to express the sparse representation. The observed data Y is naturally expressed as the sum of the exactly original data set and noise and outliers. And the appropriate sparse subspace clustering (RSE-SSC) is formulated as (7)

$$\begin{aligned} \min_{(\hat{Y},C,E,Z)} \quad & \|C\|_1 + \lambda_e \|E\|_1 + \tfrac{\lambda_z}{2}\|Z\|_F^2 \\ \text{s.t.} \quad & Y = \hat{Y} + E + Z, \\ & \hat{Y} = \hat{Y}C, \\ & \mathrm{diag}(C) = 0. \end{aligned} \tag{7}$$

In the formulation, $\hat{Y} \in R^{M\times N}$ is the original data set without any noise and outliers, Z and E correspond to noise and sparse outliers respectively, $\lambda_e > 0$, $\lambda_z > 0$ are two trade-off parameters balancing the three terms in the objective function, and the diag(C) is a vector whose elements are the diagonal entries of the matrix C.

The sparse coefficients again encode information about memberships of data to subspaces, which are used in a spectral clustering framework, as before. The algorithm for RSE-SSC is shown in the following section.

4 Solving the Sparse Optimization Problem

The proposed convex programs can be solved via generic convex solvers. However, the computational costs of generic solvers typically is high and these solvers do not scale well with the dimension and the number of data points. In this section, we study efficient implementations of the proposed sparse optimizations using an Alternating Direction Method of Multipliers (ADMM) method [1,2]. We first consider the most general optimization problem

$$\begin{aligned} \min_{(\hat{Y},C,E,Z)} \quad & \|C\|_1 + \lambda_e \|E\|_1 + \tfrac{\lambda_z}{2}\|Z\|_F^2 \\ \text{s.t.} \quad & \hat{Y} = \hat{Y}C, \\ & Y = \hat{Y} + E + Z, \\ & \mathrm{diag}(C) = 0. \end{aligned} \tag{8}$$

Utilizing the equality constraint $Y = \hat{Y} + E + Z$, we can eliminate Z from the optimization problem (7) to generate a concise formulation without Z which is equivalent to solve the original optimization problem. The optimization problem is formulated as

$$\begin{aligned} \min_{(\hat{Y},C,E)} \quad & \|C\|_1 + \lambda_e \|E\|_1 + \tfrac{\lambda_z}{2}\|Y - \hat{Y} - E\|_F^2 \\ \text{s.t.} \quad & \hat{Y} = \hat{Y}C, \\ & \operatorname{diag}(C) = 0. \end{aligned} \tag{9}$$

where Z is eliminated from the optimization program, and there are fewer variables and this will cut down the computational cost in each iteration as each variable shall be updates in each iteration of ADMM.

However there is no simple way to update C in each iteration, in order to obtain efficient updates on the optimization variables, we introduce an auxiliary matrix $A \in \mathbb{R}^{N\times N}$, and equivalently transform the optimization problem (9) to the optimization problem

$$\begin{aligned} \min_{(\hat{Y},C,E,A)} \quad & \|C\|_1 + \lambda_e \|E\|_1 + \tfrac{\lambda_z}{2}\|Y - \hat{Y} - E\|_F^2 \\ \text{s.t.} \quad & \hat{Y} = \hat{Y}A, \\ & A = C - \operatorname{diag}(C). \end{aligned} \tag{10}$$

It should be noted that the solution $(\hat{Y}; C; E)$ of optimization problem (10) concides with the solution of optimization problem (9), also concides with the solution of optimization problem (7). As shown in the Algorithm 1, the updating of variable C is much simpler in each iteration. Following the ADMM framework, we add those two constraints $\hat{Y} = \hat{Y}A$ and $A = C - \operatorname{diag}(C)$ of (10) as two penalty terms with parameter $\rho > 0$ to the objective function, and also introduce Lagrange multipliers for the two equality constraints, then we can write the augmented Lagrangian function of (10) as

$$\begin{aligned} & \mathcal{L}(\hat{Y}, C, E, A; \Delta_1, \Delta_2) \\ := & \|C\|_1 + \lambda_e \|E\|_1 + \tfrac{\lambda_z}{2}\|Y - \hat{Y} - E\|_F^2 \\ & + \frac{\rho}{2}\|\hat{Y}A - \hat{Y}\|_F^2 + \frac{\rho}{2}\|A - C + \operatorname{diag}(C)\|_F^2 \\ & + \operatorname{tr}(\Delta_1^\top(\hat{Y}A - Y)) + \operatorname{tr}(\Delta_2^\top(A - C + \operatorname{diag}(C))). \end{aligned} \tag{11}$$

where the matrix $\Delta_1, \Delta_2 \in \mathbb{R}^{N\times N}$ is the Lagrange multipliers for the two equality constraints in (10), $\operatorname{tr}(\cdot)$ denotes the trace operator of a given matrix. The algorithm then iteratively updates the variables $A, C, E, \hat{Y}$ and the Lagrange multipliers Δ_1, Δ_2.

In the k-th iteration, we denote the variables by A^k, C^k, E^k, $\hat{Y}^k$, and the Lagrange multipliers by Δ_1^k, Δ_2^k. Given A^k, C^k, E^k, $\hat{Y}^k$,Δ_1^k, Δ_2^k, ADMM iterates as follows

Algorithm 1. ADMM algorithm for solving program (10)

Input: $\hat{Y}^0$,A^0,C^0, E^0,Δ_1^0,Δ_2^0
Output: $\hat{Y}$,A,C,E,Δ_1,Δ_2

1: **Initialization:**Set $maxIter = 10^4$, $k = 0$, $Terminate = False$
2: **while**(Terminate == False) **do**
3: Update A^{k+1} by solving the following system of linear equations

$$\rho((\hat{Y}^k)^\top \hat{Y}^k + I)A = \rho((\hat{Y}^k)^\top \hat{Y}^k + C^k) + \rho \text{diag}(C^k) + (\hat{Y}^k)^\top \Delta_1^k - \Delta_2^k,$$

4: Update C^{k+1} as $C^{k+1} = J^{k+1} - \text{diag}(J^{k+1})$, where $J^{k+1} = \mathcal{T}_{\frac{1}{\rho}}(A^{k+1} + \Delta_2^k/\rho)$,
5: Update E^{k+1} as $E^{k+1} = \mathcal{T}_{\frac{\lambda_e}{\lambda_z}}(\hat{Y}^k - Y)$,
6: Update $\hat{Y}^{k+1}$ by the solving system of linear equations

$$\hat{Y}(\lambda_z I + \rho(A^{k+1} - I)(A^{k+1} - I)^\top) = \lambda_z(Y - E^{k+1}) - \Delta_1(A^{k+1} - I)^\top,$$

7: Update Δ_1^{k+1} as $\Delta_1^{k+1} := \Delta_1^k + \rho(\hat{Y}^{k+1}A^{k+1} - \hat{Y}k + 1)$,
8: Update Δ_2^{k+1} as $\Delta_2^{k+1} := \Delta_2^k + \rho(A^{k+1} - C^{k+1})$,
9: $k = k + 1$,
10: **if** $k \geq maxIter$ or $\|\hat{Y}A^k - \hat{Y}\|_F^2 \leq \epsilon$ or $\|A^k - C^k\|_F^2 \leq \epsilon$ or $\|A^k - A^{k-1}\| \leq \epsilon$ or $\|C^k - C^{k-1}\| \leq \epsilon$ or $\|E^k - E^{k-1}\| \leq \epsilon$ or $\|\hat{Y}^k - \hat{Y}^{k-1}\| \leq \epsilon$ **then**
$Terminate = True$
11: **end if**
12: **end while**

$$\hat{Y}^{k+1} := \underset{\hat{Y}}{\text{argmin}}\, \mathcal{L}(\hat{Y}, A^{k+1}, C^{k+1}, E^{k+1}; \Delta_1^k, \Delta_2^k), \tag{12a}$$

$$A^{k+1} := \underset{A}{\text{argmin}}\, \mathcal{L}(\hat{Y}^k, A, C^k, E^k; \Delta_1^k, \Delta_2^k), \tag{12b}$$

$$C^{k+1} := \underset{C}{\text{argmin}}\, \mathcal{L}(\hat{Y}^k, A^{k+1}, C, E^k; \Delta_1^k, \Delta_2^k), \tag{12c}$$

$$E^{k+1} := \underset{E}{\text{argmin}}\, \mathcal{L}(\hat{Y}^k, A^{k+1}, C^{k+1}, E; \Delta_1^k, \Delta_2^k), \tag{12d}$$

$$\Delta_1^{k+1} := \Delta_1^k + \rho(\hat{Y}A - \hat{Y}), \tag{12e}$$

$$\Delta_2^{k+1} := \Delta_2^k + \rho(A - C). \tag{12f}$$

Then we show how to solve the six subproblems in (12a), (12b), (12c), (12d), (12e) and (12f) in the ADMM algorithm. After all these subproblems solved, we will give the framework of the algorithm that summarize our ADMM algorithm for solving (10) in Algorithm 1.

These three steps are repeated until convergence is achieved or the number of iterations exceeds a maximum iteration number. The algorithm will stop when these conditions satisfied $\|\hat{Y}A^k - \hat{Y}\|_F^2 \leq \epsilon, \|A^k - C^k\|_F^2 \leq \epsilon, \|A^k - A^{k-1}\| \leq \epsilon, \|C^k - C^{k-1}\| \leq \epsilon, \|E^k - E^{k-1}\| \leq \epsilon, \|\hat{Y}^k - \hat{Y}^{k-1}\| \leq \epsilon$, where ϵ denotes the error tolerance for the primal and dual residuals. In practice, the choice of $= 10^4$ works well in real experiments. In summary, Algorithm 1 shows the updates for the ADMM implementation of the optimization problem (10).

5 Convergence Analysis

Similar to [21,27], in this section, we provide a convergence analysis for the proposed ADMM algorithm showing that under certain standard conditions, any limit point of the iteration sequence generated by Algorithm 1 is a KKT point of (10).

Theorem 1. *Let* $X := (\hat{Y}, C, W, E)$ *and* $\{X^k\}_{k=1}^{\infty}$ *be generated by Algorithm 1. Assume that* $\{X^k\}_{k=1}^{\infty}$ *is bounded and* $\lim_{k\to\infty}(X^{k+1} - X^k) = 0$. *Then any accumulation point of* $\{X^k\}_{k=1}^{\infty}$ *is a KKT point of problem* (10).

For ease of presentation, we define $S_1 := \{C|\text{diag}(C) = 0\}$, and use $\mathcal{P}_S(\cdot)$ to denote the projection operator onto set S. It is easy to verify that the KKT conditions for (10) are:

$$\partial\mathcal{L}_Y(\hat{Y}, A, C, E; \Delta_1, \Delta_2) = 0, \tag{13a}$$
$$\partial\mathcal{L}_A(\hat{Y}, A, C, E; \Delta_1, \Delta_2) = 0, \tag{13b}$$
$$\partial\mathcal{L}_C(\hat{Y}, A, C, E; \Delta_1, \Delta_2) = 0, \tag{13c}$$
$$\partial\mathcal{L}_E(\hat{Y}, A, C, E; \Delta_1, \Delta_2) = 0, \tag{13d}$$
$$\hat{Y} - \hat{Y}A = 0, \tag{13e}$$
$$\mathcal{P}_S(A) = A, \tag{13f}$$

where $\partial f(x)$ is the subdifferential of function f at x.

Proof. Assume $\hat{X} := (\hat{Y}, \hat{C}, \hat{W}, \hat{E})$ is a limit point of $\{X^k\}_{k=1}^{\infty}$. We will show that $\hat{X}$ satisfies the KKT conditions in (13). As (12a) and (12b) are guaranteed by the algorithm construction, it directly implies that $\hat{X}$ satisfies (13a) and (13b).

We rewrite the updating rules 12e and 12f in Algorithm 1 as

$$E^{k+1} - E^k = \mathcal{T}_{\frac{\lambda_e}{\lambda_z}}(\hat{Y}^k - Y) - \mathcal{T}_{\frac{\lambda_e}{\lambda_z}}(\hat{Y}^{k-1} - Y) \tag{14a}$$
$$C^{k+1} - C^k = J^{k+1} - \text{diag}(J^{k+1}) - (J^k - \text{diag}(J^k)) \tag{14b}$$
$$\Delta_1^{k+1} - \Delta_1^k = \rho(\hat{Y}A - \hat{Y}), \tag{14c}$$
$$\Delta_2^{k+1} - \Delta_2^k = \rho(A - C). \tag{14d}$$

The assumption $\lim_{k\to\infty}(X^{k+1} - X^k) = 0$ implies that the left hand sides in (15) all go to zero. Therefore,

$$\rho(\hat{Y}A - \hat{Y}) \to 0 \tag{15a}$$
$$\rho(A - C) \to 0 \tag{15b}$$

Hence, we only need to verify that $\hat{X}$ satisfies the other two conditions in (13).

For convenience, let us first ignore the projection in (13e). Then for $\tau_2 > 0$, (13e) is equivalent to

$$\tau_2\rho Y^T(Y - YC) + C \in \tau_2\partial\|C\|_1 + C \triangleq \mathcal{Q}_{\tau_2}(C) \tag{16}$$

with the scalar function $\mathcal{Q}_{\tau_2}(t) := \tau_2 \partial(|t|_1) + t$ applied element-wise to C. It is easy to verify that $\mathcal{Q}_{\tau_2}(t)$ is monotone and $\mathcal{Q}_{\tau_2}^{-1}(t) = \text{shrink}_1(t, \tau_2)$. By applying $\mathcal{Q}_{\tau_2}^{-1}(\cdot)$ to both sides of (16), we get

$$C = \text{shrink}_1(\tau_2 \rho Y^T (Y - YC) + C, \tau_2). \tag{17}$$

By invoking the definition of g_C^k leads to

$$\hat{C} = \mathcal{P}_{S_2}(\text{shrink}_1(\tau_2 \rho \hat{Y}^T (\hat{Y} - \hat{Y}\hat{C}) + \hat{C}, \tau_2)),$$

which implies that $\hat{X}$ satisfies (13c) and (13d) when the projection function is considered.

In summary, $\hat{X}$ satisfies the KKT conditions (13). Thus, any accumulation point of $\{X^k\}_{k=1}^{\infty}$ is a KKT point of problem (10).

6 Experiments

In this section, we apply our RSE-SSC approach to three different data sets. We will mainly focus on the comparison of our approach with SSC proposed in [9]. The MATLAB codes of SSC were downloaded from http://www.cis.jhu.edu/~ehsan/code.htme.

6.1 The Dataset

We applied our Algorithm 2 to three public datasets: the Extended Yale B dataset[1], the USPS digital images dataset[2], and the COIL20 dataset[3].

The Extended Yale B dataset is a well-known dataset for face clustering, which consists of images taken from 38 human subjects, and 64 frontal images for each subject were acquired under different illumination conditions and a fixed pose. To reduce the computational cost and memory requirements of the algorithms, we downsample the raw images into the size of 48×42. Thus, each image is in dimension of $2,016$.

The USPS dataset is relatively difficult to handle, in which there are $7,291$ labeled observations and each observation is a digit of 16×16 grayscale image and of different orientations. The number of each digit varies from 542 to $1,194$. To reduce the time and memory cost of the expriement, we randomly chose 100 images of each digit in our experiment.

The COIL20 dataset is a database consisting of $1,440$ grayscale images of 20 objects. Images of the 20 objects were taken to pose intervals of 5 degrees, which results in 72 images per object. All $1,440$ normalized images of 20 objects are used in our experiment.

[1] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.
[2] http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html.
[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Algorithm 2. Algorithm for RSE-SSC

Input: A set of points $\{M_i\}_{i=1}^N$ (with missing entries and outlying entries) lying in a union of L linear subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^L$

Output: Clustering results of the data points

1: Normalization: normalize the data points.
2: RSE-SSC solve the RSE-SSC optimization problem (7) by algorithm introduced in Sect. 4.
3: Post-processing: for each C_i, keep its largest T coefficients in absolute magnitude, and set the remaining coefficients to zeros.
4: Similarity graph: form a similarity graph with N nodes representing the data points, and set the weights of the edges between the nodes by $W = |C| + |C|^T$.
5: Clustering: apply the normalized spectral clustering approach in [18] to the similarity graph.

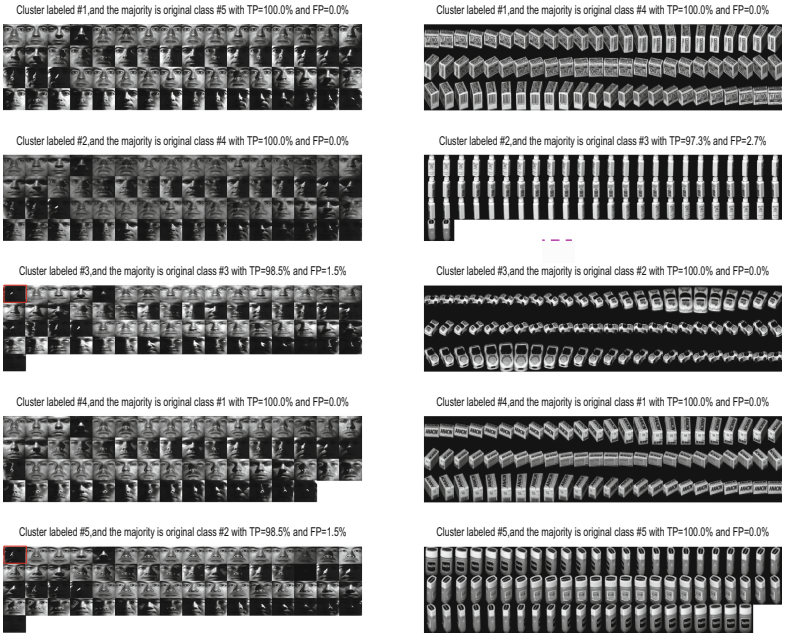
6.2 Post-Processing and Spectral Clustering

After solving optimization problem (7), we obtain the clustered images and the sparse coefficient matrix C. Similar to [6], we perform some post-processing procedure on C. For each coefficient vector C_i, we keep its largest T coefficients in absolute magnitude and set the remaining coefficients to zeros. The affinity matrix W associated with the weighted graph $\mathcal{G}$ is then constructed as $W = |C|+|C|^T$. To obtain the final clustering result, we apply the normalized spectral clustering approach proposed by Ng *et al.* [18]. Thus, the whole procedure of our RSE-SSC based clustering approach can be described as in Algorithm 2.

For SSC [7], we find that it performs better without the normalization of data and post-processing of the coefficient matrix C. As a result, we ran the codes provided by the authors directly. Note that the same spectral clustering algorithm was applied to the coefficient matrix obtained by SSC.

6.3 Implementation Details

Considering that the number of subspaces affects the clustering and recovery performance of algorithms, we applied algorithms under cases of $L = 3, 5, 8$, where L denotes the number of subspaces, i.e., the number of different subjects. To shorten the testing time, L subjects were chosen in the following way. In the Extended Yale B dataset, all the 38 subjects were divided into four groups, where the four groups correspond to subjects 1 to 10, 11 to 20, 21 to 30, and 31 to 38, respectively. L subjects were chosen from the same group. For example, when $L = 5$, the number of possible 5 subjects is $3\binom{10}{5} + \binom{8}{5} = 812$. Among these 812 possible choices, 20 trials were randomly chosen to test the proposed algorithms under the condition of L subspaces. To study the effect of the fraction of missing entries in clustering and recovery performance of algorithms, we artificially corrupted images into 3 missing rate levels 10 %, 20 % to 30 %. To make the comparison of different algorithms as fair as possible, we randomly generated the missing images first, and then all algorithms were applied to the same randomly corrupted images to cluster the images and recover the missing pixels.



(a) Clustered images from Extended Yale (b) Clustered images from Columbia objects

Fig. 1. Clustered images from Extended Yale B face images and Columbia objects images within 5 subspaces with 10 % entries missing

To generate corrupted images with a specified missing fraction range from, we randomly removed squares whose size is no larger than 10×10, repeatedly until the total fraction of missing pixels is no less than the specified fraction.

In Algorithm 2 (RSE-SSC), we initialise Y by filling in each missing pixel with the average value of corresponding pixels in other images with known pixel value. We implement SSC proposed in [7] in two different ways. In Algorithm "SSC-AVE", we fill in the missing entries in the same way as "RSE-SSC", and in Algorithm "SSC-0", we fill in the missing entries by 0. We use the subspace clustering error (SCE), which is defined as

$$SCE := (\#\text{ of misclassified images})/(\text{total }\#\text{ of images}),$$

to demonstrate the clustering quality. For each set of L with different percentage of missing pixels, the averaged SCE over 20 random trials are calculated.

In Algorithm 1, we choose $\lambda_e = \lambda_z = 5 \times 10^{-2}$, total loss threshold $tol = 5 \times 10^{-6}$, maximum iterations $maxIter = 5 \times 10^3$ in all experiments. We set $\rho = 1000/\mu$, where $\mu := \min_i \max_{j \neq i} |Y_i^T Y_j|$ to avoid trivial solutions. It should be noted that ρ actually changes in each iteration, because matrix Y is updated in each iteration. To accelerate the convergence of Algorithm 1, we adopt the following stopping criterion to terminate the algorithm $\|\hat{Y}A^k - \hat{Y}\|_F^2 + \|A^k - C^k\|_F^2 + \|A^k - A^{k-1}\| + \|C^k - C^{k-1}\| + \|E^k - E^{k-1}\| + \|\hat{Y}^k - \hat{Y}^{k-1}\| < tol$ or the iteration number exceeds the $maxIter$.

Table 1. SCE-Mean of RSE-SSC and SSC with different fraction of missing entries for different datasets

Missing rate=10 %			Missing rate=20 %			Missing rate=30 %		
SSC-0	SSC-AVE	RSE-SSC	SSC-0	SSC-AVE	RSE-SSC	SSC-0	SSC-AVE	RSE-SSC
Yale B dataset: 3 subjects								
5.83	5.20	3.78	27.00	7.57	6.12	51.84	14.68	8.24
Yale B dataset: 5 subjects								
5.62	4.76	5.25	40.32	17.23	8.29	54.28	32.23	13.75
Yale B dataset: 8 subjects								
7.81	6.45	5.72	42.18	40.37	15.42	60.62	46.57	18.19
USPS dataset: 3 subjects								
0.07	0.07	0.07	9.07	9.33	7.43	10.52	9.97	8.34
USPS dataset: 5 subjects								
8.76	7.68	3.74	23.51	21.69	18.54	24.55	21.06	20.74
USPS dataset: 8 subjects								
12.50	10.11	7.86	36.81	34.59	21.35	46.93	29.08	26.53
COIL20 dataset: 3 subjects								
6.99	6.74	5.87	38.47	19.75	7.63	45.74	19.72	8.94
COIL20 dataset: 5 subjects								
7.79	9.17	7.13	41.76	18.60	9.36	46.74	26.36	11.82
COIL20 dataset: 8 subjects								
11.55	9.44	9.56	41.94	28.32	15.27	48.51	35.54	18.62

6.4 Results

We report the experimental results in Table 1. It shows the SCE of three algorithms, where "SCE-mean" represent the mean of the SCE in percentage over 20 random trials. From Table 1, we can see that when the images are incomplete, for example when $L = 3$, the mean of SCE is usually smaller than 7 %. This means that the percentage of misclassified images is smaller than 7 %. It can been seen that our Algorithm 2 gives better results on clustering errors in most situations. Especially, when $spa = 20\,\%$ and 30 %, the mean clustering errors are much smaller than the ones given by SSC-AVE and SSC-0, this phenomena is more obvious with the increase of number of subjects. These comparison results show that our RSE-SSC model can cluster images very robustly and greatly outperforms SSC.

The subfigures (a) and (b) of Fig. 1 show the clustering results of one instance of $L = 5$ using Algorithm 2 for the Extended Yale B dataset and the COIL dataset. After applying our algorithm, each image is labled with a class ID, and comparing to the original class ID we could calculate the True Positive Rate and False Positive Rate of each original individual. The misclassified images in each cluster are labeled with colored rectangles and the true positive rate (TP) and false positive rate are also given as the title of the subfigures. Most of the misclassified images, as we can see, are not in good illumination conditions and

they are thus difficult to be classified. Removing these illumination conditions will improve the experimental results a lot. We only show the results of one instance of $L = 5$ for the Extended Yale B dataset and the COIL20 dataset. The results on all datasets will be fully displayed in a longer version.

It should be noted that, due to the limited space, we just present results on the Extended Yale B dataset and the COIL dataset with 10 % entries missing, the other two missing rate level will be provided in a longer version or in the form of supplementary material. In the supplementary material, Fig. 1, Fig. 2 and Fig. 3 show the results for the Extended Yale B dataset, the COIL dataset and the USPS dataset with $L = 5$ using Algorithm 2. Figure 4, Fig. 5 and Fig. 6 show the accordingly clustering results of $L = 8$ using Algorithm 2 for these datasets.

7 Conclusions

Sparse subspace clustering is a well-known algorithm, and it is widely used in many research field nowadays, and a lot effort has been contributed to improve it. In this paper, we propose a novel approach to obtain the coefficient matrix. Compared with traditional sparse subspace clustering (SSC) approaches, the key advantage of our approach is that it provides a new perspective of the self-expressive property. We call it rigidly self-expressive (RSE) property. This new formulation captures the rigidly self-expressive property of the data points in the same subspace, and provides a new formulation for sparse subspace clustering. Extensions to traditional SSC could also be cooperate with this new formulation, and this could lead to a serial of approaches based on rigidly self-expressive property. We present a first-order algorithm to solve the nonconvex optimization, and further prove that it converges to a KKT point of the nonconvex problem under certain standard assumptions. Extensive experiments on the Extended Yale B dataset, the USPS digital images dataset, and the Columbia Object Image Library show that for images with up to 30 % missing pixels the clustering quality achieved by our approach outperforms the original SSC.

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends in Mach. Learn. **3**(1), 1–122 (2011)
2. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
3. Bradley, P.S., Mangasarian, O.L.: K-plane clustering. J. Global Optim. **16**(1), 23–32 (2000)
4. Chen, G.L., Lerman, G.: Spectral curvature clustering (SCC). Int. J. Comput. Vis. **81**(3), 317–330 (2009)
5. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**(11), 1413–1457 (2004)

6. Dyer, E.L., Sankaranarayanan, A.C., Baraniuk, R.G.: Greedy feature selection for subspace clustering. J. Mach. Learn. Res. **14**(1), 2487–2517 (2013)
7. Elhamifar, E.: Sparse modeling for high-dimensional multi-manifold data analysis. Ph.D. thesis, The Johns Hopkins University (2012)
8. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proceedings of CVPR, vol. 1–4, pp. 2782–2789 (2009)
9. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2765–2781 (2013)
10. Grant, M., Boyd, S., Ye, Y.: CVX: matlab software for disciplined convex programming (2008)
11. Ho, J., Yang, M.-H., Lim, J., Lee, K.-C., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: Proceedings 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I-11–I-18. IEEE (2003)
12. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Mach. Intell. **16**(5), 550–554 (1994)
13. Kriegel, H.-P., Krger, P., Zimek, A.: Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. Knowl. Discov. Data (TKDD) **3**(1), 1 (2009)
14. Lee, K.-C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
15. Liu, G., Xu, H., Yan, S.: Exact subspace segmentation and outlier detection by low-rank representation. In: Proceedings of AISTATS, pp. 703–711 (2012)
16. Lu, Y.M., Do, M.N.: Sampling signals from a union of subspaces. IEEE Sig. Proc. Mag. **25**(2), 41–47 (2008)
17. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-20). Technical report CUCS-005-96 (1996)
18. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. Proc. NIPS **14**, 849–856 (2002)
19. Ramamoorthi, R.: Analytic PCA construction for theoretical analysis of lighting variability in images of a lambertian object. IEEE Trans. Pattern Anal. Mach. Intell. **24**(10), 1322–1333 (2002)
20. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
21. Shen, Y., Wen, Z., Zhang, Y.: Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. Optim. Methods Softw. **29**(2), 239–263 (2014)
22. Soltanolkotabi, M., Candes, E.J.: A geometric analysis of subspace clustering with outliers. Ann. Stat. **40**(4), 2195–2238 (2012)
23. Soltanolkotabi, M., Elhamifar, E., Candes, E.J.: Robust subspace clustering. Ann. Stat. **42**(2), 669–699 (2014)
24. Vidal, R.E.: Generalized principal component analysis (GPCA): an algebraic geometric approach to subspace clustering and motion segmentation. Ph.D. thesis, University of California at Berkeley (2003)
25. Wang, Y.-X., Xu, H.: Noisy sparse subspace clustering. In: Proceedings of ICML, pp. 89–97 (2013)

26. Wright, J., Ganesh, A., Rao, S., Peng, Y., Ma, Y.: Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization. In: Proceedings of NIPS, pp. 2080–2088 (2009)
27. Xu, Y., Yin, W., Wen, Z., Zhang, Y.: An alternating direction algorithm for matrix completion with nonnegative factors. Front. Math. China **7**(2), 365–384 (2012)
28. Zappella, L., Llado, X., Salvi, J.: Motion segmentation: a review. Artif. Intell. Res. Dev. **184**, 398–407 (2008)