Neural Processing Letters
https://doi.org/10.1007/s11063-021-10597-5

Subspace Clustering with Block Diagonal Sparse Representation

Xian Fang[1,2] · Ruixun Zhang[3] · Zhengxin Li[1,2] · Xiuli Shao[1,2]

Accepted: 13 July 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
Structured representation is of remarkable significance in subspace clustering. However, most of the existing subspace clustering algorithms resort to single-structured representation, which may fail to fully capture the essential characteristics of data. To address this issue, a novel multi-structured representation subspace clustering algorithm called block diagonal sparse representation (BDSR) is proposed in this paper. It takes both sparse and block diagonal structured representations into account to obtain the desired affinity matrix. The unified framework is established by integrating the block diagonal prior into the original sparse subspace clustering framework and the resulting optimization problem is iteratively solved by the inexact augmented Lagrange multipliers (IALM). Extensive experiments on both synthetic and real-world datasets well demonstrate the effectiveness and efficiency of the proposed algorithm against the state-of-the-art algorithms.

Keywords Subspace clustering · Multi-structured representation · Sparse structure · Block diagonal structure · Spectral clustering

1 Introduction

Subspace clustering, also known as subspace segmentation, aims to partition a set of data samples approximately drawn from a union of linear subspaces [1]. It has been widely applied in many fields, such as image clustering [2–5], image classification [6,7], image compression [8,9], video separation [10] and video summarization [11]. Among the popular subspace clustering methods, the spectral-type based methods are extremely promising [12], which attracts the increasing attention of researchers. These kinds of methods typically perform subspace clustering in two stages, that is, first learning affinity matrix that encodes the subspace membership information from the given data, and then applying spectral clustering on the learned affinity matrix to obtain the final clustering result.

✉ Xiuli Shao
shaoxl@nankai.edu.cn

1 College of Computer Science, Nankai University, Tianjin 300350, China

2 Tianjin Key Laboratory of Network and Data Security Technology, Tianjin 300350, China

3 MIT Laboratory for Financial Engineering, Cambridge, MA 02139, USA

Published online: 23 July 2021

Up to now, three dominant branches of structured representation have been designed and embedded in subspace clustering algorithms, i.e., sparse representation, low-rank representation and block diagonal representation. Accordingly, the fundamental algorithms are referred as sparse subspace clustering (SSC) [13,14], low-rank representation (LRR) [15,16] and block diagonal representation (BDR) [17], respectively. In general, the former two build the affinity matrix by implicitly seeking the sparsest or the lowest rank linear representation of each sample relative to the rest samples or the entire data samples, whereas the latter one explicitly pursues block diagonal structure to calculate the affinity matrix. On the basis of these algorithms, lots of improved algorithms are constantly emerging [18–29]. When the observed data samples are insufficient or contaminated by an overwhelming amount of noises, latent low-rank representation (LLRR) [30] is brought forward. Least squares regression (LSR) [31] is implemented to encourage the grouping effect, where ridge regression may have similar behavior as LRR in the context of statistics. Non-linear latent space sparse subspace clustering (NLS3C) [32] is devised as the kernel extension of SSC to complex non-linear manifold learning. In view of the influence of symmetric components, low-rank representation with symmetric constraint (LRRSC) [33] is put forward. Discriminative and coherent subspace clustering (DCSC) [34] is modeled, which enforces the coherence and discrimination of the affinity matrix as well as the label. Symmetry constrained latent low rank representation with converted nuclear norm SLLRRC [35] is presented by introducing a kind of converted nuclear norm and integrating strategy of the symmetric constraint. Although encouraging performance has been met, the algorithms mentioned above are still challenging due to the lack of reliable guidance for global characteristics identification. In other words, they just focus on single-structured representation, and none of them are able to comprehensively understanding the diversity of characteristics. To this end, several multi-structured representation subspace clustering algorithms have been proposed in recent years. Correlation adaptive subspace segmentation (CASS) [36] is brought forward, which uses the trace Lasso to adaptively interpolate SSC and LSR. Low-rank sparse subspace clustering (LRSSC) [37] is implemented for the sparsity and low-rankness of the representation. Implicit block diagonal low-rank representation (IBDLR) [38] is devised by combining the block diagonal representation and low-rank representation. As a matter of fact, the multi-structured representation exploits intrinsic properties, i.e., complementarity and consistency, among different structures to jointly enhance the generalization ability of learning models.

Motivated by the success of multi-structured representation for subspace clustering, we investigate the joint structure of the sparse and block diagonal representations. Devoted to taking advantage of the flexible calculation of sparse structure and the real distribution of block diagonal structure, we therefore propose the block diagonal sparse representation (BDSR) in this paper. The proposed algorithm is capable of learning the desired affinity matrix, which enjoys both sparse and block diagonal structures, yielding reliable clustering result. The strength of the algorithm is that it explores the internal association of data from two aspects of complementarity and consistency without bringing extra computing costs. In addition, we utilize the inexact augmented Lagrange multipliers (IALM) [39] to derive closed form solution to the resulting optimization problem, hence the algorithm is executed simply and quickly. The main contributions of this paper can be summarized below.

(1) We propose a novel multi-structured representation subspace clustering algorithm by simultaneously incorporating the sparse constraint and block diagonal prior.
(2) We develop an optimization strategy for the minimization problem of the proposed algorithm. In this approach, the closed form solutions of all subproblems are derived.

(3) We conduct experiments on a large number of benchmark datasets, and the experimental results demonstrate the superiority of the proposed algorithm over other state-of-the-art algorithms.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 elaborates on the details of the proposed algorithm and the corresponding solving method. Section 4 gives the experimental results followed by concluding remarks in Sect. 5.

2 Related Works

2.1 Notations

All matrices, vectors and scalars used in this paper are shown in boldface uppercase, boldface lowercase and lowercase letters, respectively. For instance, $\boldsymbol{M}$ is a matrix, $\boldsymbol{v}$ is a vector and s is a scalar. In particular, $\boldsymbol{I}$ is used to denote the identity matrix, and the all zero and all one vector or matrix are denoted as $\mathbf{0}$ and $\mathbf{1}$, respectively. $\boldsymbol{M}^{\mathrm{T}}$ and $\boldsymbol{M}^{-1}$ denote the transpose and the inverse of $\boldsymbol{M}$, respectively, while $\boldsymbol{v}^{\mathrm{T}}$ is the transpose of $\boldsymbol{v}$. Diag($\boldsymbol{v}$) converts $\boldsymbol{v}$ into a diagonal matrix whose i-th diagonal entry being the i-th entry of $\boldsymbol{v}$, while diag($\boldsymbol{M}$) is a vector with its i-th entry being the i-th diagonal entry of $\boldsymbol{M}$. If all entries of $\boldsymbol{M}$ are non-negative, we denote $\boldsymbol{M} \geq 0$. If $\boldsymbol{M}$ is positive semi-definite, we denote $\boldsymbol{M} \succeq 0$. In addition, tr($\boldsymbol{M}$) and sign($\boldsymbol{M}$) denote the trace function and the sign function of $\boldsymbol{M}$, respectively. The only used vector norm is the ℓ_2 norm, denoted by $\|\boldsymbol{v}\|_2$. Some matrix norms are also be used, including the ℓ_0 norm, the ℓ_1 norm, the spectral norm, the Frobenius norm and the ℓ_∞ norm, denoted by $\|\boldsymbol{M}\|_0$, $\|\boldsymbol{M}\|_1$, $\|\boldsymbol{M}\|_2$, $\|\boldsymbol{M}\|_F$ and $\|\boldsymbol{M}\|_\infty$, respectively. For $\boldsymbol{M}$ and $\boldsymbol{N}$, we denote $\boldsymbol{N} \preceq \boldsymbol{M}$ if $\boldsymbol{M} - \boldsymbol{N} \succeq 0$. In addition, $\langle \boldsymbol{M}, \boldsymbol{N} \rangle$ and $\boldsymbol{M} \odot \boldsymbol{N}$ denote the inner product and the Hadamard product between $\boldsymbol{M}$ and $\boldsymbol{N}$, respectively.

2.2 Sparse Subspace Clustering

Suppose that there is a set of high dimensional data samples $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, each column of which is drawn from a union of k independent subspaces $\{\mathcal{S}_i\}_{i=1}^k$ with unknown dimensions, where d is dimension of samples and n is number of samples. The task of subspace clustering is to divide the samples according to the subspaces they are drawn from.

Sparse subspace clustering (SSC) relies on the fact that each sample can be expressed as a sparse linear combination of other data samples. The minimization problem of SSC is given as

$$\min_{\boldsymbol{Z},\boldsymbol{E}} \|\boldsymbol{Z}\|_0 + \lambda \|\boldsymbol{E}\|_F^2 \quad \text{s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}, \operatorname{diag}(\boldsymbol{Z}) = \mathbf{0} \tag{1}$$

where $\boldsymbol{Z} \in \mathbb{R}^{n \times n}$ is the coefficient matrix, $\boldsymbol{E} \in \mathbb{R}^{d \times n}$ is the noise matrix and $\lambda \in (0, \infty)$ is the trade-off parameter. SSC wants $\boldsymbol{Z}$ to be sparse and $\boldsymbol{Z}$ is expected not to be the trivial solution. The problem in Eq. (1) is essentially an NP-hard problem, the tractable relaxation is given as

$$\min_{\boldsymbol{Z},\boldsymbol{E}} \|\boldsymbol{Z}\|_1 + \lambda \|\boldsymbol{E}\|_F^2 \quad \text{s.t. } \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}, \operatorname{diag}(\boldsymbol{Z}) = \mathbf{0} \tag{2}$$

Definition 1 [15] For a collection of subspaces $\{\mathcal{S}_i\}_{i=1}^k$, it is said to be independent if and only if the sum of the dimensions of all elements is equal to the dimension of their direct sum.

In theory, it is proved that the affinity matrix $\boldsymbol{A} = (|\boldsymbol{Z}| + |\boldsymbol{Z}|^{\mathrm{T}})/2$ obtained by SSC satisfies the block diagonal property when the underlying subspaces are independent. Such block diagonal property, however, is often fragile and weak. Therefore, block diagonal constraint is actually helpful as prior knowledge to alleviate this situation.

2.3 Block Diagonal Regularizer

For accurate clustering, the graph Laplacian matrix should represent several connected components. It is absolutely compatible to require the affinity matrix to be block diagonal.

Without loss of generality, it can be assumed that $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k]$ is ordered, where $\boldsymbol{X}_i \in \mathbb{R}^{d \times n_i}$ denotes the data samples belong to $\mathcal{S}_i$ $(i = 1, 2, \ldots, k)$ and $\sum_{i=1}^k n_i = n$.

Definition 2 [17] For any affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, the k-block diagonal regularizer is defined as

$$\|\boldsymbol{A}\|_{\boxed{k}} = \sum_{i=n-k+1}^{n} \varphi_i(\boldsymbol{L}_{\boldsymbol{A}}) \tag{3}$$

where $\boldsymbol{L}_{\boldsymbol{A}}$ is the Laplacian matrix of $\boldsymbol{A}$ defined as $\boldsymbol{L}_{\boldsymbol{A}} = \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1}) - \boldsymbol{A}$ and $\varphi_i(\boldsymbol{L}_{\boldsymbol{A}})$ denote the i-th eigenvalue of $\boldsymbol{L}_{\boldsymbol{A}}$ in decreasing order.

Theorem 1 *[38] Let $\varphi_i(\cdot)$ be the i-th eigenvalue of a certain square matrix in decreasing order, $\boldsymbol{L}_{\boldsymbol{A}} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{L}_{\boldsymbol{A}} \succeq 0$, then we have*

$$\sum_{i=n-k+1}^{n} \varphi_i(\boldsymbol{L}_{\boldsymbol{A}}) = \min_{\boldsymbol{B}} \langle \boldsymbol{L}_{\boldsymbol{A}}, \boldsymbol{B} \rangle \quad \text{s.t. } \boldsymbol{0} \preceq \boldsymbol{B} \preceq \boldsymbol{I}, \mathrm{tr}(\boldsymbol{B}) = k \tag{4}$$

where $\boldsymbol{B} \in \mathbb{R}^{n \times n}$.

Intuitively, the block diagonal regularizer is a soft regularizer for imposing block diagonal constraint. Based on the theorem above, the block diagonal regularizer can be transformed into a minimization surrogate.

3 Block Diagonal Sparse Representation

3.1 Problem Formulation

With the goal of providing a new unified optimization framework, the sparse constraint and block diagonal prior are combined for collaborative subspace clustering, which is called block diagonal sparse representation (BDSR). Different from most subspace clustering algorithms which only focus on single-structured representation, the algorithm takes into account the multi-structured representation of both sparse and block diagonal. Besides, the symmetry constraint is introduced to ensure the weight consistency for each pair of data samples, while retaining the geometrical subspace structure of the data, so that highly correlated data points of subspaces can be represented together. Compared with the current multi-structured subspace

clustering algorithms, the algorithm can better recover the subspace structure hidden in the high dimensional dataset and obtain the more exact segmentation effect. The minimization problem of BDSR is given as

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2+\lambda_1\|\boldsymbol{Z}\|_1+\lambda_2\|\boldsymbol{A}\|_{\boxed{k}} \quad \text{s.t. } \boldsymbol{A}=\frac{|\boldsymbol{Z}|+|\boldsymbol{Z}|^{\mathrm{T}}}{2} \tag{5}$$

where $\lambda_1 \in (0, \infty)$ and $\lambda_2 \in (0, \infty)$ are two trade-off parameters, which balances the importance of each term. The first term can be treated as the regularization term, the second term promotes the sparsity of the coefficient matrix, while the third term is adopted to guarantee the block diagonal structure of the affinity matrix.

Merging Eq. (3) and Eq. (4) into Eq. (5), then the minimization problem can be reformulated as

$$\min_{\boldsymbol{Z},\boldsymbol{B}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}\|_F^2+\lambda_1\|\boldsymbol{Z}\|_1+\lambda_2\langle \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1})-\boldsymbol{A},\boldsymbol{B}\rangle \quad \text{s.t. } \boldsymbol{A}=\frac{|\boldsymbol{Z}|+|\boldsymbol{Z}|^{\mathrm{T}}}{2}, \boldsymbol{0}\preceq\boldsymbol{B}\preceq\boldsymbol{I}, \mathrm{tr}(\boldsymbol{B})=k \tag{6}$$

3.2 Optimization Procedure

Since each term in Eq. (6) is closely related to one of the variables $\boldsymbol{Z}$ to be optimized, the problem in which is non-convex. Fortunately, it can be approximatively solved by the inexact augmented Lagrange multipliers (IALM). By adding two auxiliary variables $\boldsymbol{P} \in \mathbb{R}^{d\times n}$ and $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, Eq. (6) can be further reformulated as

$$\begin{aligned}&\min_{\boldsymbol{Z},\boldsymbol{B},\boldsymbol{P},\boldsymbol{Q}} \frac{1}{2}\|\boldsymbol{P}\|_F^2+\lambda_1\|\boldsymbol{Q}\|_1+\lambda_2\langle \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1})-\boldsymbol{A},\boldsymbol{B}\rangle\\ &\text{s.t. } \boldsymbol{A}=\frac{|\boldsymbol{Z}|+|\boldsymbol{Z}|^{\mathrm{T}}}{2}, \boldsymbol{0}\preceq\boldsymbol{B}\preceq\boldsymbol{I}, \mathrm{tr}(\boldsymbol{B})=k,\\ &\quad \boldsymbol{P}=\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}, \boldsymbol{Q}=\boldsymbol{Z}\end{aligned} \tag{7}$$

The augmented Lagrange function of Eq. (7) can be written as

$$\begin{aligned}&\mathcal{L}(\boldsymbol{Z},\boldsymbol{B},\boldsymbol{P},\boldsymbol{Q},\boldsymbol{Y}_1,\boldsymbol{Y}_2,\mu)\\ &=\frac{1}{2}\|\boldsymbol{P}\|_F^2+\lambda_1\|\boldsymbol{Q}\|_1+\lambda_2\langle \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1})-\boldsymbol{A},\boldsymbol{B}\rangle\\ &\quad+\mathrm{tr}(\boldsymbol{Y}_1^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}-\boldsymbol{P}))+\mathrm{tr}(\boldsymbol{Y}_2^{\mathrm{T}}(\boldsymbol{Z}-\boldsymbol{Q}))\\ &\quad+\frac{\mu}{2}(\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}-\boldsymbol{P}\|_F^2+\|\boldsymbol{Z}-\boldsymbol{Q}\|_F^2)\\ &=\frac{1}{2}\|\boldsymbol{P}\|_F^2+\lambda_1\|\boldsymbol{Q}\|_1+\lambda_2\langle \mathrm{Diag}(\boldsymbol{A}\boldsymbol{1})-\boldsymbol{A},\boldsymbol{B}\rangle\\ &\quad+\mathcal{H}(\boldsymbol{Z},\boldsymbol{P},\boldsymbol{Q},\boldsymbol{Y}_1,\boldsymbol{Y}_2,\mu)-\frac{1}{2\mu}(\|\boldsymbol{Y}_1\|_F^2+\|\boldsymbol{Y}_2\|_F^2)\end{aligned} \tag{8}$$

where $\boldsymbol{Y}_1 \in \mathbb{R}^{d\times n}$ and $\boldsymbol{Y}_2 \in \mathbb{R}^{n\times n}$ are two Lagrange multipliers, $\mu \in (0, \infty)$ is the penalty parameter, $\boldsymbol{A}=(|\boldsymbol{Z}|+|\boldsymbol{Z}|^{\mathrm{T}})/2$, $\boldsymbol{0}\preceq\boldsymbol{B}\preceq\boldsymbol{I}$, $\mathrm{tr}(\boldsymbol{B})=k$ and $\mathcal{H}(\boldsymbol{Z},\boldsymbol{P},\boldsymbol{Q},\boldsymbol{Y}_1,\boldsymbol{Y}_2,\mu)=\mu/2(\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{Z}-\boldsymbol{P}+\boldsymbol{Y}_1/\mu\|_F^2+\|\boldsymbol{Z}-\boldsymbol{Q}+\boldsymbol{Y}_2/\mu\|_F^2)$.

Due to the non-smoothness of the problem in Eq. (8) with respect to variables $\boldsymbol{Z}$, $\boldsymbol{B}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$, we optimize it by alternatively minimizing each variable while fixing the others for each iteration. The key challenge of the whole optimization procedure lies in the following four subproblems.

$\boldsymbol{Z}$-subproblem: Updating $\boldsymbol{Z}$ with others fixed. The subproblem of Eq. (8) with respect to $\boldsymbol{Z}$ is

$$\begin{aligned} \boldsymbol{Z} &= \operatorname*{argmin}_{\boldsymbol{Z}} \lambda_2 \langle \mathrm{Diag}(\boldsymbol{A}\mathbf{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle + \mathcal{H}(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{Y}_1, \boldsymbol{Y}_2, \mu) \\ &\text{s.t. } \boldsymbol{A} = \frac{|\boldsymbol{Z}| + |\boldsymbol{Z}|^{\mathrm{T}}}{2} \end{aligned} \tag{9}$$

Like [40], Eq. (9) can be reformulated as

$$\begin{aligned} \boldsymbol{Z}^{(t+1)} &= \operatorname*{argmin}_{\boldsymbol{Z}} \lambda_2 \langle \mathrm{Diag}(\boldsymbol{A}\mathbf{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle + \frac{\mu\eta}{2} \|\boldsymbol{Z} - \boldsymbol{Z}^{(t)}\|_F^2 \\ &\quad + \langle \nabla_{\boldsymbol{Z}} \mathcal{H}(\boldsymbol{Z}^{(t)}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{Y}_1, \boldsymbol{Y}_2, \mu), \boldsymbol{Z} - \boldsymbol{Z}^{(t)} \rangle \\ &= \operatorname*{argmin}_{\boldsymbol{Z}} \frac{\lambda_2}{\mu\eta} \langle \mathrm{Diag}(\boldsymbol{A}\mathbf{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle + \frac{1}{2} \|\boldsymbol{Z} - (\boldsymbol{Z}^{(t)} + \frac{1}{\eta}(\boldsymbol{X}^{\mathrm{T}} \\ &\quad (\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu})))\|_F^2 \\ &\quad \text{s.t. } \boldsymbol{A} = \frac{|\boldsymbol{Z}| + |\boldsymbol{Z}|^{\mathrm{T}}}{2} \end{aligned} \tag{10}$$

where $\nabla_{\boldsymbol{Z}} \mathcal{H}(\boldsymbol{Z}^{(t)}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{Y}_1, \boldsymbol{Y}_2, \mu)$ denotes the partial differential of $\mathcal{H}$ to $\boldsymbol{Z}$, $\eta = \|\boldsymbol{X}\|_2^2$ and t is number of iterations. Note that the term $\langle \mathrm{Diag}(\boldsymbol{A}\mathbf{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle$ has $|\boldsymbol{Z}|$ but not $\boldsymbol{Z}$, while the term $\|\boldsymbol{Z} - (\boldsymbol{Z}^{(t)} + 1/\eta(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \boldsymbol{Y}_1/\mu) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \boldsymbol{Y}_2/\mu)))\|_F^2$ is the opposite. As a result, the entries of $\boldsymbol{Z}$ must share the same sign as the ones of $\boldsymbol{Z}^{(t)} + 1/\eta(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \boldsymbol{Y}_1/\mu) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \boldsymbol{Y}_2/\mu))$. Let $\boldsymbol{Z} = \hat{\boldsymbol{Z}} \odot \mathrm{sign}(\boldsymbol{Z}^{(t)} + 1/\eta(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \boldsymbol{Y}_1/\mu) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \boldsymbol{Y}_2/\mu)))$, then $\hat{\boldsymbol{Z}}$ is the solution to

$$\begin{aligned} \hat{\boldsymbol{Z}}^{(t+1)} &= \operatorname*{argmin}_{\boldsymbol{Z}} \frac{\lambda_2}{\mu\eta} \langle \mathrm{Diag}(\boldsymbol{A}\mathbf{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle + \frac{1}{2} \|\boldsymbol{Z} - |(\boldsymbol{Z}^{(t)} + \frac{1}{\eta}(\boldsymbol{X}^{\mathrm{T}} \\ &\quad (\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu})))|\|_F^2 \\ &\quad \text{s.t. } \boldsymbol{A} = \frac{\boldsymbol{Z} + \boldsymbol{Z}^{\mathrm{T}}}{2}, \boldsymbol{Z} \geq 0 \end{aligned} \tag{11}$$

It is obvious that Eq. (11) is equivalent to

$$\begin{aligned} \hat{\boldsymbol{Z}}^{(t+1)} &= \operatorname*{argmin}_{\boldsymbol{Z}} \frac{1}{2} \|\boldsymbol{Z} - (|(\boldsymbol{Z}^{(t)} + \frac{1}{\eta}(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}) \\ &\quad - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu})))| - \frac{\lambda_2}{2\mu\eta}(\boldsymbol{D}_{\boldsymbol{B}} + \boldsymbol{D}_{\boldsymbol{B}}^{\mathrm{T}}))\|_F^2 \\ &\quad \text{s.t. } \boldsymbol{Z} \geq 0 \end{aligned} \tag{12}$$

where $\boldsymbol{D}_{\boldsymbol{B}} = \mathrm{diag}(\boldsymbol{B})\mathbf{1}^{\mathrm{T}} - \boldsymbol{B}$. Thus, Eq. (12) has the closed form solution as

$$\begin{aligned} \hat{\boldsymbol{Z}}^{(t+1)} &= \max(\mathbf{0}, |(\boldsymbol{Z}^{(t)} + \frac{1}{\eta}(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}) - (\boldsymbol{Z}^{(t)} - \\ &\quad \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu})))| - \frac{\lambda_2}{2\mu\eta}(\boldsymbol{D}_{\boldsymbol{B}} + \boldsymbol{D}_{\boldsymbol{B}}^{\mathrm{T}})) \end{aligned} \tag{13}$$

Solving out $\hat{\boldsymbol{Z}}^{(t+1)}$, then $\boldsymbol{Z}^{(t+1)}$ can be recovered by

$$\boldsymbol{Z}^{(t+1)} = \hat{\boldsymbol{Z}}^{(t+1)} \odot \text{sign}(\boldsymbol{Z}^{(t)} + \frac{1}{\eta}(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z}^{(t)} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}) - (\boldsymbol{Z}^{(t)} - \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu}))) \tag{14}$$

***B*-subproblem**: Updating $\boldsymbol{B}$ with others fixed. The subproblem of Eq. (8) with respect to $\boldsymbol{B}$ is

$$\begin{aligned} \boldsymbol{B} = &\underset{\boldsymbol{B}}{\text{argmin}} \langle \text{Diag}(\boldsymbol{A}\boldsymbol{1}) - \boldsymbol{A}, \boldsymbol{B} \rangle \\ &\text{s.t. } \boldsymbol{0} \preceq \boldsymbol{B} \preceq \boldsymbol{I}, \text{tr}(\boldsymbol{B}) = k \end{aligned} \tag{15}$$

where $\boldsymbol{A} = (|\boldsymbol{Z}| + |\boldsymbol{Z}|^{\mathrm{T}})/2$. Thus, Eq. (15) can be solved in a closed form with

$$\boldsymbol{B} = \boldsymbol{G}\boldsymbol{G}^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{G} \in \mathbb{R}^{n \times k}$, the column vectors of it are composed of the k eigenvectors associated with the smallest k eigenvalues of the term $\text{Diag}(\boldsymbol{A}\boldsymbol{1}) - \boldsymbol{A}$.

***P*-subproblem**: Updating $\boldsymbol{P}$ with others fixed. By removing the irrelevant terms with respect to $\boldsymbol{P}$ in Eq. (8), we deal with the subproblem as

$$\boldsymbol{P} = \underset{\boldsymbol{P}}{\text{argmin}} \frac{1}{2}\|\boldsymbol{P}\|_F^2 + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z} - \boldsymbol{P} + \frac{\boldsymbol{Y}_1}{\mu}\|_F^2 \tag{17}$$

Eq. (17) is strongly convex, which can be directly solved. Taking the derivation with respect to $\boldsymbol{P}$, we have

$$\boldsymbol{P} = \frac{1}{1+\mu}(\mu\boldsymbol{X} - \mu\boldsymbol{X}\boldsymbol{Z} + \boldsymbol{Y}_1) \tag{18}$$

***Q*-subproblem**: Updating $\boldsymbol{Q}$ with others fixed. We only preserve the relevant terms with respect to $\boldsymbol{Q}$ in Eq. (8) and deal with the subproblem as

$$\boldsymbol{Q} = \underset{\boldsymbol{Q}}{\text{argmin}}\, \lambda_1\|\boldsymbol{Q}\|_1 + \frac{\mu}{2}\|\boldsymbol{Z} - \boldsymbol{Q} + \frac{\boldsymbol{Y}_2}{\mu}\|_F^2 \tag{19}$$

It is obvious that Eq. (19) is equivalent to

$$\boldsymbol{Q} = \underset{\boldsymbol{Q}}{\text{argmin}} \frac{\lambda_1}{\mu}\|\boldsymbol{Q}\|_1 + \frac{1}{2}\|\boldsymbol{Q} - (\boldsymbol{Z} + \frac{\boldsymbol{Y}_2}{\mu})\|_F^2 \tag{20}$$

Eq. (20) can be solved by the shrinkage operator, which has the closed form solution as

$$\boldsymbol{Q} = \max(\boldsymbol{0}, |\boldsymbol{Z} + \frac{\boldsymbol{Y}_2}{\mu}| - \frac{\lambda_1}{\mu}\boldsymbol{1}) \odot \text{sign}(\boldsymbol{Z} + \frac{\boldsymbol{Y}_2}{\mu}) \tag{21}$$

After $\boldsymbol{Z}$, $\boldsymbol{B}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are updated, the Lagrange multipliers $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, and penalty parameter μ can be easily updated by the following rule

$$\begin{cases} \boldsymbol{Y}_1^{(t+1)} = \boldsymbol{Y}_1^{(t)} + \mu(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{Z} - \boldsymbol{P}) \\ \boldsymbol{Y}_2^{(t+1)} = \boldsymbol{Y}_2^{(t)} + \mu(\boldsymbol{Z} - \boldsymbol{Q}) \\ \mu^{(t+1)} = \min(\mu_{\max}, \rho\mu^{(t)}) \end{cases} \tag{22}$$

where $\mu_{\max}$ and ρ denote the maximum value of μ and the learning rate, respectively. The updating process is terminated when the convergence conditions are satisfied, i.e., $\|\boldsymbol{X} - \boldsymbol{XZ} - \boldsymbol{P}\|_\infty < \varepsilon$ and $\|\boldsymbol{Z} - \boldsymbol{Q}\|_\infty < \varepsilon$, where ε denotes the preset threshold.

The detailed procedure to solve the problem stated in Eq. (6) by IALM is outlined in Algorithm (1). Once the coefficient matrix $\boldsymbol{Z}$ is obtained, the affinity matrix $\boldsymbol{A}$ can be further defined to perform certain spectral clustering methods, such as normalized cuts (NCut) [41]. The complete procedure of the BDSR algorithm is outlined in Algorithm (2).

Algorithm 1: Solving the problem in Eq. (6) by IALM.

Input: Data matrix X, parameters λ_1, λ_2.
Output: Coefficient matrix Z.
Initialize: $P^{(0)} = Y_1^{(0)} = 0$, $Z^{(0)} = B^{(0)} = Q^{(0)} = Y_2^{(0)} = 0$, $\mu^{(0)} = 10^{-2}$, $\mu_{\max} = 10^6$, $\rho = 1.1$, $\varepsilon = 10^{-6}$ and $t = 0$.
while *not converged* **do**
  Fix the others and update $Z^{(t+1)}$ by Eq. (14);
  Fix the others and update $B^{(t+1)}$ by Eq. (16);
  Fix the others and update $P^{(t+1)}$ by Eq. (18);
  Fix the others and update $Q^{(t+1)}$ by Eq. (21);
  Update $Y_1^{(t+1)}$, $Y_2^{(t+1)}$ and $\mu^{(t+1)}$ by Eq. (22);
  $t \leftarrow t + 1$;
end

Algorithm 2: The BDSR algorithm.

Input: Data matrix $\boldsymbol{X}$, number of subspaces k, parameters λ_1, λ_2.
Output: Clustering result.
Obtaining the coefficient matrix $\boldsymbol{Z}$ using Algorithm (1);
Constructing the affinity matrix $\boldsymbol{A} = (|\boldsymbol{Z}| + |\boldsymbol{Z}|^{\mathrm{T}})/2$;
Applying spectral clustering to $\boldsymbol{A}$ and segmenting the data samples into k clusters;

3.3 Computational Complexity Analysis

The running time of BDSR is mainly consumed by updating the variables $\boldsymbol{Z}$, $\boldsymbol{B}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$. Updating of $\boldsymbol{Z}$ involves both matrix multiplication of $d \times n$ matrix and $n \times n$ matrix and matrix addition of two $n \times n$ matrices, which has the complexity of $\mathcal{O}(dn^2 + n^2)$ in each iteration. Updating of $\boldsymbol{B}$ involves the eigenvalue decomposition operation of $n \times n$ matrix, i.e., matrix multiplication of $n \times k$ matrix and $k \times n$ matrix, which has the complexity of $\mathcal{O}(kn^2)$ in each iteration. Updating of $\boldsymbol{P}$ involves matrix multiplication of $d \times n$ matrix and $n \times n$ matrix, which has the complexity of $\mathcal{O}(dn^2)$ in each iteration. Updating of $\boldsymbol{Q}$ involves matrix addition of two $n \times n$ matrices, which has the complexity of $\mathcal{O}(n^2)$ in each iteration. In summary, the computational complexity of each iteration of BDSR is about $\mathcal{O}(2dn^2 + kn^2 + 2n^2)$. In practice, $d \gg k$ holds for most cases. Therefore, the overall time complexity of BDSR can be roughly considered as $\mathcal{O}(tdn^2)$.

3.4 Convergence Analysis

Theorem 2 *Let $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{B}, \boldsymbol{P}, \boldsymbol{Q})$ be the objective function related to variables $\boldsymbol{Z}$, $\boldsymbol{B}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$, and $\mathcal{G}(\boldsymbol{B})$ be the indicator function of constraint condition related to variable $\boldsymbol{B}$ in Eq. (7), we can gain*

$$\begin{aligned}
&\mathcal{F}(\boldsymbol{Z}^{(t+1)}, \boldsymbol{B}^{(t+1)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) + \mathcal{G}(\boldsymbol{B}^{(t+1)}) \\
&\quad \leq \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t)}, \boldsymbol{Q}^{(t)}) + \mathcal{G}(\boldsymbol{B}^{(t)}) - \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \hat{\boldsymbol{Z}}^{(t+1)}\|_F^2 \\
&\quad -(1+\mu)\|\boldsymbol{P}^{(t)} - \boldsymbol{P}^{(t+1)}\|_F^2 - (1 + \frac{\mu}{2\lambda_1})\|\boldsymbol{Q}^{(t)} - \boldsymbol{Q}^{(t+1)}\|_F^2
\end{aligned} \tag{23}$$

where both μ and λ_1 are non-negative.

It is easy to discover that $\mathcal{F}(\boldsymbol{Z}, \boldsymbol{B}, \boldsymbol{P}, \boldsymbol{Q}) + \mathcal{G}(\boldsymbol{B})$ is strictly monotonic decreasing. Thus, BDSR can converge to an optimal solution in limited iteration. The proof of Theorem 2 is given in the Appendix.

4 Experiments

4.1 Experimental Setup

We compare our BDSR algorithm against other subspace clustering algorithms including SSC, LRR, LRRSC, SLLRRC, BDR and IBDLR using synthetic and real-world datasets. Note that BDR owns two versions, i.e., BDR-B and BDR-Z. All of the methods are coded by the MATLAB R2013b platform and all of the experiments are tested on a PC of Windows 10 with Intel(R) Core(TM) i7-9700 CPU and 32GB memory. The clustering error is adopted as the metric to evaluate and discuss the performance of algorithms, which is defined as

$$\text{err} = 1 - \frac{1}{n}\sum_{i=1}^{n}\delta(\hat{l}_i, \text{map}(l_i)) \tag{24}$$

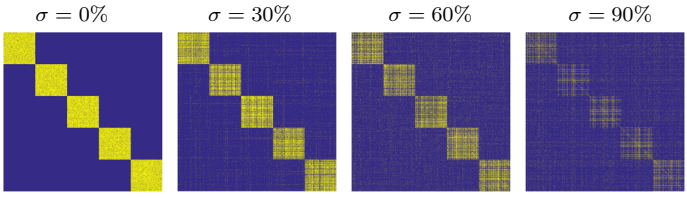
where $\{\hat{l}_i\}_{i=1}^n$ and $\{l_i\}_{i=1}^n$ denote the estimated labels and the ground truth ones of data samples, respectively, map$(\cdot)$ depicts the optimal mapping from each clustering label to the ground truth label and $\delta(\hat{l}_i, \text{map}(l_i))$ equals to 1 if $\hat{l}_i = \text{map}(l_i)$ or equals to 0 otherwise. The smaller value of the metric indicates the better result, and the best results in experiments are portrayed in boldface. Note that we report experimental results of contrast algorithms on the premise of setting the parameters suggested by the authors or manually tuning them to satisfaction.

4.2 Experiments on Synthetic Dataset

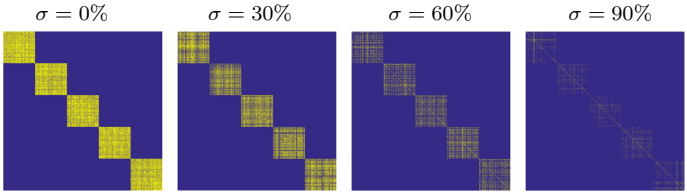
To verify the effectiveness of the proposed BDSR, we first conduct experiments in comparison to IBDLR on the synthetic dataset in this section. Similar to [42], we generate data matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k] \in \mathbb{R}^{1000\times 1000}$ drawn from $k = 5$ linear subspaces $\{\mathcal{S}_i\}_{i=1}^k$, where $\boldsymbol{X}_i \in \mathbb{R}^{1000\times 200}$ $(i = 1, 2, \ldots, k)$. Specifically, the bases of subspace $\{\boldsymbol{U}_i\}_{i=1}^k \in \mathbb{R}^{1000\times r}$ are computed by $\boldsymbol{U}_{i+1} = \boldsymbol{T}\boldsymbol{U}_i$ $(i = 1, 2, \ldots, k-1)$, where $r = 5$ denotes the rank of each subspace, $\boldsymbol{T} \in \mathbb{R}^{1000\times 1000}$ is a random orthogonal matrix and the column vectors of $\boldsymbol{U}_1$ are composed of the first r columns of the left singular matrixes computed from a random matrix.

Table 1 Clustering error (%) and computational time (s) of IBDLR and BDSR on the synthetic dataset with different noise ratios

Ratio	Algorithm	Error	Time
0%	IBDLR	**0.00**	68.49
	BDSR	**0.00**	**24.20**
30%	IBDLR	3.45	194.71
	BDSR	**0.00**	**24.58**
60%	IBDLR	4.25	436.04
	BDSR	**0.00**	**24.86**
90%	IBDLR	6.73	640.15
	BDSR	**2.38**	**25.24**



(a) Binarized $\boldsymbol{A}$ by IBDLR



(b) Binarized $\boldsymbol{A}$ by BDSR

Fig. 1 The affinity matrix $\boldsymbol{A}$ produced by IBDLR and BDSR on the synthetic dataset with different noise ratios

After that, data from the i-th subspace is sampled by $\boldsymbol{X}_i = \boldsymbol{U}_i \boldsymbol{S}$, where $\boldsymbol{S} \in \mathbb{R}^{r \times 200}$ is a random matrix. We prepare four different scenarios in our experiment. That is, the dataset is considered under four kinds of noise ratios of $\sigma \in \{0\%, 30\%, 60\%, 90\%\}$. Here $\sigma = 0\%$ means that there is no noise to corrupt the dataset. When there is noise to corrupt the dataset, $\sigma = 30\%$, $\sigma = 60\%$ or $\sigma = 90\%$ of the samples selected to be contaminated are regenerated by adding Gaussian noise with mean 0 and standard deviation $0.1\|\boldsymbol{x}_i\|_2$. Considering the randomness, each experiment is independently repeated 20 times and the average results are recorded.

Table 1 shows the clustering results of two algorithms on the synthetic dataset with different noise ratios. As can be seen from Table 1, in the case that the data is not corrupted by noise, both of the two algorithms achieve perfect clustering segmentation with an error of 0, although the computational time of IBDLR is slightly longer than that of BDSR. However, in the case that the data is corrupted by noise, BDSR significantly outperforms IBDLR whether in clustering error or computational time. We also observed that the clustering error and computational time of BDSR are always stable with the increase of noise ratio, which proves the strong robustness.

Figure 1 visualizes the affinity matrix $\boldsymbol{A}$ produced by IBDLR and BDSR on the synthetic dataset with different noise ratios. For ease of observation, the binarized $\boldsymbol{A}$ is defined here.

That is, if $\boldsymbol{A}_{ij} \geq \tau$, then let $\boldsymbol{A}_{ij}$ be 1, otherwise 0, where $\tau = 10^{-1}$. Ideally, the affinities of data samples from the same cluster should be one, leading to the coherent property, while those from different clusters should be zero, resulting in the discriminative property. In other words, the ideal affinity matrix $\boldsymbol{A}$ should exhibit a salient block diagonal structure, where the entries on the diagonal block correspond to the data within the cluster, while the entries off the diagonal block correspond to the data outside the cluster. From Fig. 1, it can be seen that the affinity matrix produced by BDSR has a clearer and cleaner block diagonal structure than that produced by IBDLR, which tends to divide clusters correctly according to the corresponding subspace.

4.3 Experiments on Real-World Datasets

In order to demonstrate the efficiency of the proposed BDSR, we further conduct experiments on three kinds of real-world datasets from object clustering, face clustering and motion segmentation in this section, where BDSR is compared with SSC, LRR, LRRSC, SLLRRC, BDR-B, BDR-Z and IBDLR.

4.3.1 Object Clustering

The COIL20 database.[1] a famous object clustering dataset, is comprised of 1440 images from 20 objects taken from different angles [43]. There are 72 images for each object and the size of each image is 128×128 pixels. Fig. 2 shows some original image samples from the COIL20 database. To reduce memory requirements, each image sample is downsampled to 32×32 pixels, thus forming a 1024-dimensional vector. After that, we evenly divide the 38 subjects into two groups, where the first groups correspond to subjects 1 to 10 and the last groups correspond to subjects 11 to 20. In the experiment, we arrange two scenarios to carry out. That is, the database is considered under two kinds of subject classes $c_1 \in \{2, 8\}$. As a result, there are s_1 combinations in both two scenarios, where $s_1 = 2 \times \binom{10}{2} = 2 \times \binom{10}{8} = 90$. For these tasks, the average results are recorded.

Table 2 shows the clustering results of various algorithms on the COIL20 database. As can be seen from Table 2, BDSR obtains the best results in most cases in terms of all metrics. We can also observe that the performance of the multi-structured representation algorithms, such as BDSR and IBDLR, are generally better than that of the single-structured representation algorithms, such as SSC, LRR, LRRSC, SLLRRC, BDR-B and BDR-Z. The reason is that, compared with the single-structured representation, the multi-structured representation provides more complementary and consistent information. Besides, IBDLR achieves nearly the identical level of efficiency as BDSR, but lacks a large degree of effectiveness. This is due to it aims at solving the clustering problem for non-linear data which consumes time to convert the data from the original space to a specific kernel space and is also difficult to find a suitable kernel space. Figure 3 shows the convergence curves of BDSR on the COIL20 database with different subject classes. From Fig. 3, it can be seen that the value of residual error decreases monotonically in each iteration and rapidly approaches an accumulation point. As shown in the experimental study, it usually converges in less than 100 iterations.

[1] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

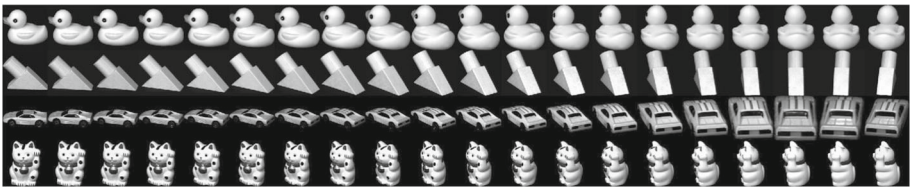


Fig. 2 Some original image samples from the COIL20 database

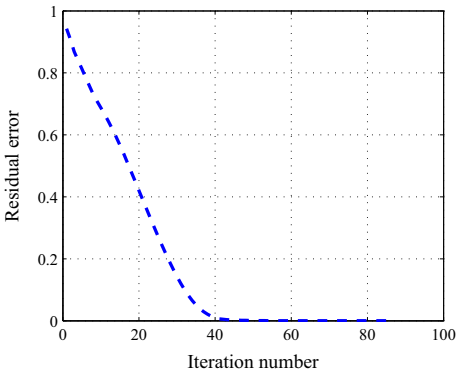
Table 2 Clustering error (%) and computational time (s) of various algorithms on the COIL20 database

Class	Algorithm	Error				Time
		Mean	Med.	Max.	Std.	
2	SSC	1.47	0.83	2.57	1.53	9.75
	LRR	2.23	1.77	3.11	3.21	4.41
	LRRSC	2.17	1.54	3.29	3.02	8.64
	SLRRC	1.34	0.63	2.37	1.04	5.72
	BDR-B	0.96	0.56	1.54	0.86	6.74
	BDR-Z	0.93	0.53	1.33	0.77	6.74
	IBDLR	**0.88**	0.48	**1.15**	0.84	9.85
	BDSR	0.89	**0.43**	1.19	**0.74**	**2.58**
8	SSC	3.53	3.36	5.34	7.35	21.34
	LRR	3.83	2.96	5.71	6.22	25.73
	LRRSC	3.41	3.02	4.02	3.57	29.94
	SLRRC	3.13	2.63	4.64	3.11	23.61
	BDR-B	2.12	1.46	2.85	2.91	26.46
	BDR-Z	1.95	1.27	2.63	2.64	26.46
	IBDLR	1.40	0.95	**1.75**	**1.07**	33.56
	BDSR	**1.19**	**0.94**	2.43	1.67	**11.74**

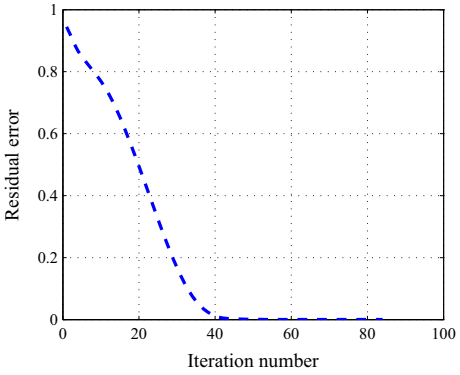
4.3.2 Face clustering

The Extended Yale B database[2] is a commonly used face clustering dataset, which contains 2414 images from 38 people acquired under different pose and illumination conditions [44, 45]. Each person has about 59 to 64 images with 168×192 pixels. Figure 4 shows some original image samples from the Extended Yale B database. To reduce memory requirements, each image sample is downsampled to 42×48 pixels, thus forming a 2016-dimensional vector. After that, we divide the 38 subjects into four groups, where the first three groups correspond to subjects 1 to 10, subjects 11 to 20 and subjects 21 to 30, and the last groups correspond to subjects 31 to 38. In the experiment, we also arrange two scenarios to carry out. That is, the database is considered under two kinds of subject classes $c_2 \in \{5, 8\}$. As a result, there are s_3 and s_4 combinations in two scenarios, respectively, where $s_2 = 3 \times \binom{10}{5} + 1 \times \binom{8}{5} = 812$ and $s_3 = 3 \times \binom{10}{8} + 1 \times \binom{8}{8} = 163$. For these tasks, the average results are recorded.

[2] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.



(a) 2 subject classes



(b) 8 subject classes

Fig. 3 Convergence curves of BDSR on the COIL20 database with different subject classes

Table 3 shows the clustering results of various algorithms on the Extended Yale B database. As can be seen from Table 3, the performance of BDSR is the best and far better than other competitors, especially LRR. In particular, IBDLR has the smallest standard deviation of clustering error on the dataset with 8 subject classes.

Figure 5 shows the affinity matrix $\boldsymbol{A}$ produced by BDSR using the t-distributed stochastic neighbor embedding (t-SNE) [46] on the Extended Yale B database with different subject classes. From Fig. 5, it can be seen that the affinity matrix $\boldsymbol{A}$ has great distinguishing power because each category in the affinity matrix is quite scattered from the visualizations.

4.3.3 Motion Segmentation

As a well-known motion segmentation dataset, the Hopkins 155 database[3] consists of 120 sequences of two motions, 35 sequences of three motions and 1 sequence of five motions [47]. Figure 6 shows some original image samples from the Hopkins 155 database. On the one hand, we keep the original feature trajectories of each motion in $2F$-dimensional subspace without any preprocessing, where F denotes the number of frames in the video. On the other hand, we project the original data into $4k$-dimensional subspace by using principal component

[3] http://www.vision.jhu.edu/data/hopkins155/.

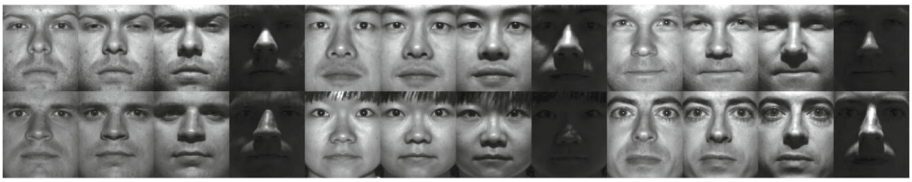


Fig. 4 Some original image samples from the Extended Yale B database

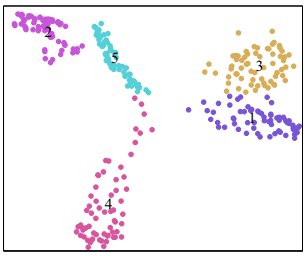
Table 3 Clustering error (%) and computational time (s) of various algorithms on the Extended Yale B database

Class	Algorithm	Error				Time
		Mean	Med.	Max.	Std.	
5	SSC	4.33	2.82	10.43	4.60	48.86
	LRR	6.92	5.63	9.78	8.64	25.10
	LRRSC	3.19	2.81	4.17	4.57	47.81
	SLRRC	3.04	2.72	3.25	4.01	31.62
	BDR-B	4.62	2.47	5.52	6.39	46.74
	BDR-Z	3.46	2.85	5.03	2.48	46.74
	IBDLR	2.75	2.54	3.54	2.75	40.23
	BDSR	**1.63**	**0.98**	**1.95**	**0.95**	**21.10**
8	SSC	5.87	4.49	12.53	4.13	84.08
	LRR	13.62	9.67	25.44	15.35	36.96
	LRRSC	4.01	3.13	6.47	4.01	81.77
	SLRRC	3.14	3.07	5.84	3.26	47.16
	BDR-B	3.45	2.68	7.14	1.29	47.68
	BDR-Z	4.73	4.73	7.78	2.83	47.68
	IBDLR	3.64	3.21	3.96	**0.76**	63.44
	BDSR	**1.82**	**1.24**	**2.74**	1.35	**27.76**

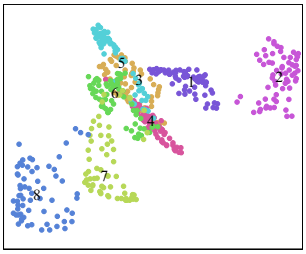
analysis (PCA) [48,49], where k denotes the number of subspaces. In the experiment, we arrange two scenarios to carry out again. That is, the database is considered under two kinds of feature dimensions of $\omega \in \{2F, 4k\}$. Since each sequence can be regarded as a separate segmentation task, there are a total of 156 segmentation tasks. For these tasks, the average results are recorded.

Table 4 shows the clustering results of various algorithms on the Hopkins 155 database. As can be seen from Table 4, BDSR obtains the mean value of clustering error of 0.74% on the dataset with $2F$ feature dimensions, which is 1.91%, 0.97%, 0.76%, 0.20%, 0.69%, 0.43% and 0.21% improvements over SSC, LRR, LRRSC, SLLRRC, BDR-B, BDR-Z and IBDLR, respectively. Besides, BDSR obtains the mean value of clustering error of 0.82% on the dataset with $4k$ feature dimensions, which is 1.83%, 1.35%, 0.74%, 0.69%, 1.25%, 1.14% and 0.66% improvements over these algorithms, respectively. In addition, all algorithms get 0.00% median value of clustering error and LRR has the smallest computing time on the dataset with $4k$ feature dimensions.

Figure 7 shows the mean clustering accuracy of BDSR using different parameters on the Hopkins 155 database with different feature dimensions. We tune two parameters λ_1 and λ_2 from the wide range of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. From Fig. 7, it can be seen that BDSR is not very sensitive to these parameters, and it achieves relatively higher mean



(a) 5 subject classes



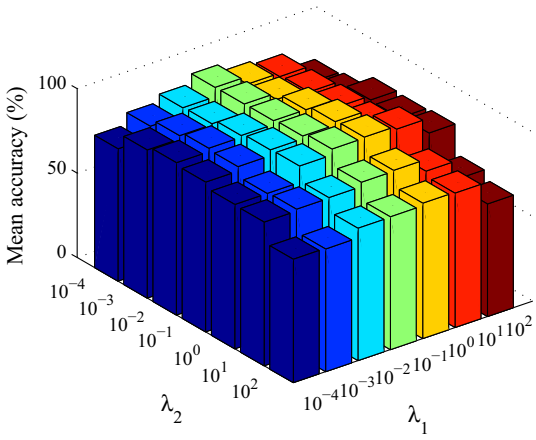
(b) 8 subject classes

Fig. 5 The affinity matrix $\boldsymbol{A}$ produced by BDSR using t-SNE on the Extended Yale B database with different subject classes

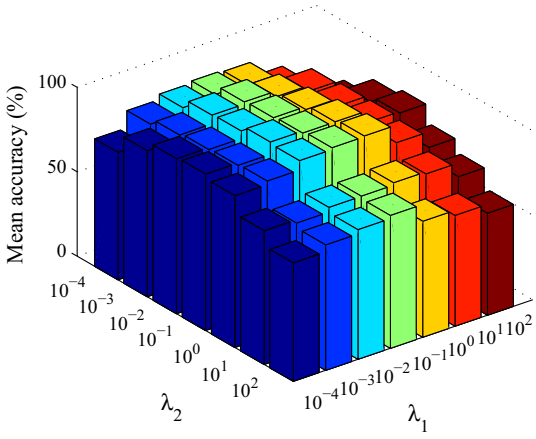
Fig. 6 Some original image samples from the Hopkins 155 database

Table 4 Clustering error (%) and computational time (s) of various algorithms on the Hopkins 155 database

Dim.	Algorithm	Error				Time
		Mean	Med.	Max.	Std.	
$2F$	SSC	2.65	**0.00**	46.97	7.61	0.97
	LRR	1.71	**0.00**	33.33	4.86	1.14
	LRRSC	1.50	**0.00**	33.33	4.36	4.34
	SLRRC	0.94	**0.00**	23.52	3.87	2.51
	BDR-B	1.43	**0.00**	41.41	5.45	1.26
	BDR-Z	1.17	**0.00**	37.69	5.73	1.26
	IBDLR	0.95	**0.00**	26.83	4.37	6.52
	BDSR	**0.74**	**0.00**	**21.84**	**3.08**	**0.68**
$4k$	SSC	2.65	**0.00**	45.50	7.53	0.89
	LRR	2.17	**0.00**	43.38	6.58	**0.53**
	LRRSC	1.56	**0.00**	43.38	5.48	4.25
	SLRRC	1.51	**0.00**	40.84	5.28	0.89
	BDR-B	2.07	**0.00**	49.49	6.11	1.24
	BDR-Z	1.96	**0.00**	48.48	6.56	1.24
	IBDLR	1.48	**0.00**	45.16	6.09	5.46
	BDSR	**0.82**	**0.00**	**39.69**	**3.95**	0.65



(a) $2F$ feature dimensions



(b) $4k$ feature dimensions

Fig. 7 Mean clustering error of BDSR using different parameters on the Hopkins 155 database with different feature dimensions

clustering accuracy when the values of λ_1 and λ_2 are in the interval 10^{-2} to 10^{0}. It is worth mentioning that we also report the result in the case that λ_1 or λ_2 is equal to 0. In $2F$ feature dimensions, when $\lambda_1 = 0$ along with $\lambda_2 = 10^{0}$ and $\lambda_1 = 10^{-1}$ along with $\lambda_2 = 0$, BDSR can obtain the mean clustering accuracy of 97.53% and 98.16%, respectively. In $4k$ feature dimensions, when $\lambda_1 = 0$ along with $\lambda_2 = 10^{0}$ and $\lambda_1 = 10^{-2}$ along with $\lambda_2 = 0$, BDSR can obtain the mean clustering accuracy of 97.67% and 98.29%, respectively.

5 Conclusions and Further Works

In this paper, we propose a novel algorithm called BDSR for subspace clustering, which combines sparse constraint and block diagonal prior so that they interactively enforce each other to have expected properties. As a result, the obtained affinity matrix not only has a

sparse structure, but also has a block diagonal structure, which strengthens its coherence and discrimination. Moreover, an IALM-based strategy is developed to solve the resulting optimization problem. To verify the performance of BDSR, a considerable amount of experiments are conducted. Experimental results demonstrate that the proposed algorithm is very competitive compared with other algorithms and highly robust in handling noisy data.

There are a few possible future research directions we need to point out. The parameters in our algorithm are still heuristic, and it is thus meaningful to study the way to determine these optimal parameters. We are also interested in extending the algorithm to the deep framework to further improve its performance utilizing non-linear information. Besides, how to import the low-rank constraint into the algorithm and solve the corresponding minimization problem reasonably seems to be another topic worth considering.

Acknowledgements The authors would like to gratefully acknowledge the editors and the anonymous reviewers for their valuable comments. This research is supported in part by the National Natural Science Foundation of China under Grant 61973173.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Appendix

The proof of Theorem 2 is given as follows.

Proof For simplicity, Eq. (12) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{Z}}^{(t+1)} &= \underset{\boldsymbol{Z}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{Z} - \varpi_1\|_F^2 \\ &\text{s.t. } \boldsymbol{Z} \geq 0 \end{aligned} \tag{25}$$

Then, we have

$$\begin{aligned} \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \varpi_1\|_F^2 &= \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \hat{\boldsymbol{Z}}^{(t+1)} + \hat{\boldsymbol{Z}}^{(t+1)} - \varpi_1\|_F^2 \\ &\geq \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \hat{\boldsymbol{Z}}^{(t+1)}\|_F^2 + \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t+1)} - \varpi_1\|_F^2 \end{aligned} \tag{26}$$

Hence, we can obtain

$$\begin{aligned} \mathcal{F}(\boldsymbol{Z}^{(t+1)}, \boldsymbol{B}^{(t+1)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) &\leq \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t+1)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) \\ -\frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \hat{\boldsymbol{Z}}^{(t+1)}\|_F^2 \end{aligned} \tag{27}$$

From Eq. (15), we can directly obtain

$$\begin{aligned} &\mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t+1)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) + \mathcal{G}(\boldsymbol{B}^{(t+1)}) \\ &\leq \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) + \mathcal{G}(\boldsymbol{B}^{(t)}) \end{aligned} \tag{28}$$

For simplicity, Eq. (17) can be rewritten as

$$\boldsymbol{P} = \underset{\boldsymbol{P}}{\operatorname{argmin}} \|\boldsymbol{P}\|_F^2 + \mu\|\boldsymbol{P} - \varpi_2\|_F^2 \tag{29}$$

Then, we have

$$\begin{aligned}\|\boldsymbol{P}^{(t+1)}\|_F^2 + \mu\|\boldsymbol{P}^{(t+1)} - \varpi_2\|_F^2 \leq \|\boldsymbol{P}^{(t)}\|_F^2 + \mu\|\boldsymbol{P}^{(t)} - \varpi_2\|_F^2 \\ -((1+\mu))\|\boldsymbol{P}^{(t)} - \boldsymbol{P}^{(t+1)}\|_F^2\end{aligned} \tag{30}$$

Hence, we can obtain

$$\begin{aligned}\mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) \leq \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t)}, \boldsymbol{Q}^{(t+1)}) \\ -(1+\mu)\|\boldsymbol{P}^{(t)} - \boldsymbol{P}^{(t+1)}\|_F^2\end{aligned} \tag{31}$$

For simplicity, Eq. (19) can be rewritten as

$$\boldsymbol{Q} = \underset{\boldsymbol{Q}}{\operatorname{argmin}} \|\boldsymbol{Q}\|_1 + \frac{\mu}{2\lambda_1}\|\boldsymbol{Q} - \varpi_3\|_F^2 \tag{32}$$

Then, we have

$$\begin{aligned}\|\boldsymbol{Q}^{(t+1)}\|_F^2 + \frac{\mu}{2\lambda_1}\|\boldsymbol{Q}^{(t+1)} - \varpi_3\|_F^2 \leq \|\boldsymbol{Q}^{(t)}\|_F^2 + \frac{\mu}{2\lambda_1}\|\boldsymbol{Q}^{(t)} - \varpi_3\|_F^2 \\ -((1+\frac{\mu}{2\lambda_1}))\|\boldsymbol{Q}^{(t)} - \boldsymbol{Q}^{(t+1)}\|_F^2\end{aligned} \tag{33}$$

Hence, we can obtain

$$\begin{aligned}\mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t)}, \boldsymbol{Q}^{(t+1)}) \leq \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t)}, \boldsymbol{Q}^{(t)}) \\ -(1+\frac{\mu}{2\lambda_1})\|\boldsymbol{Q}^{(t)} - \boldsymbol{Q}^{(t+1)}\|_F^2\end{aligned} \tag{34}$$

Combining Eq. (27), Eq. (28), Eq. (31) and Eq. (34), we can gain

$$\begin{aligned}\mathcal{F}(\boldsymbol{Z}^{(t+1)}, \boldsymbol{B}^{(t+1)}, \boldsymbol{P}^{(t+1)}, \boldsymbol{Q}^{(t+1)}) + \mathcal{G}(\boldsymbol{B}^{(t+1)}) \leq \\ \mathcal{F}(\boldsymbol{Z}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{P}^{(t)}, \boldsymbol{Q}^{(t)}) + \mathcal{G}(\boldsymbol{B}^{(t)}) - \frac{1}{2}\|\hat{\boldsymbol{Z}}^{(t)} - \hat{\boldsymbol{Z}}^{(t+1)}\|_F^2 \\ -(1+\mu)\|\boldsymbol{P}^{(t)} - \boldsymbol{P}^{(t+1)}\|_F^2 - (1+\frac{\mu}{2\lambda_1})\|\boldsymbol{Q}^{(t)} - \boldsymbol{Q}^{(t+1)}\|_F^2\end{aligned} \tag{35}$$

□

References

1. Lance P, Ehtesham H, Huan L (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explor Newslett 6(1):90–105
2. Zhenyue Z, Keke Z (2012) Low-rank matrix approximation with manifold regularization. IEEE Trans Patt Anal Mach Intell 35(7):1717–1729
3. Liansheng Z, Haoyuan G, Zhouchen L, Yi M, Xin Z, Nenghai Y (2012) Non-negative low rank and sparse graph for semi-supervised learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2328–2335
4. Mahdi A, Patel Vishal M (2018) Multimodal sparse and low-rank subspace clustering. Inform Fusion 39:168–177
5. Wencheng Z, Jiwen L, Jie Z (2019) Structured general and specific multi-view subspace clustering. Patt Recogn 93:392–403
6. John W, Yang Allen Y, Arvind G, Shankar SS, Yi M (2008) Robust face recognition via sparse representation. IEEE Trans Patt Anal Mach Intell 31(2):210–227
7. Chunjie Z, Jing L, Qi T, Changsheng X, Hanqing L, Songde M (2011) Image classification by non-negative sparse coding, low-rank and sparse decomposition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1673–1680

8. Ori B, Michael E (2008) Compression of facial images using the K-SVD algorithm. J Vis Commun Image Represent 19(4):270–282
9. Przemysław S, Śmieja M, Krzysztof M (2015) Subspaces clustering approach to lossy image compression. In: International conference on computer information systems and industrial management applications (CISIM), pp 571–579
10. Derin BS, Martin L, Rafael M, Katsaggelos Aggelos K (2012) Sparse Bayesian methods for low-rank matrix estimation. IEEE Trans Sig Process 60(8):3964–3977
11. Jing Z, Yue S, Peiguang J, Jing L, Yuting S (2019) A structure-transfer-driven temporal subspace clustering for video summarization. Multimedia Tools Appl 78(17):24123–24145
12. René V (2011) Subspace clustering. IEEE Sig Process Mag 28(2):52–68
13. Elhamifar E, René V (2009) Sparse subspace clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2790–2797
14. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans Patt Anal Mach Intell 35(11):2765–2781
15. Liu G, Zhouchen L, Yong Y (2010) Robust subspace segmentation by low-rank representation. In: Proceedings of the international conference on machine learning (ICML), pp 663–670
16. Guangcan L, Zhouchen L, Shuicheng Yan J, Sun Yong Yu, Yi M (2013) Robust recovery of subspace structures by low-rank representation. IEEE Trans Patt Anal Mach Intell 35(1):171–184
17. Canyi L, Jiashi F, Zhouchen L, Mei T, Shuicheng Y (2019) Subspace clustering by block diagonal representation. IEEE Trans Patt Anal Mach Intell 41(2):487–501
18. Patel Vishal M, René V (2014) Kernel sparse subspace clustering. In: Proceedings of the IEEE international conference on image processing (ICIP), pp 2849–2853,
19. Jun Yu, Yong R, Dacheng T (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
20. Jun Yu, Dacheng T, Meng W, Yong R (2014) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
21. Wanjun C, Erhu Z, Zhuomin Z (2016) A Laplacian structured representation model in subspace clustering for enhanced motion segmentation. Neurocomputing 208:174–182
22. Jun W, Daming S, Dansong C, Yongqiang Z, Junbin G (2016) LRSR: low-rank-sparse representation for subspace clustering. Neurocomputing 214:1026–1037
23. He W, Chen Jim X, Weihua Z (2017) Low-rank representation with graph regularization for subspace clustering. Soft Comput 21(6):1569–1581
24. Guang LC, Chong Y, René V (2017) Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. IEEE Trans Image Process 26(6):2988–3001
25. Shiqiang D, Yide M, Yurun M (2017) Graph regularized compact low rank representation for subspace clustering. Knowled Based Syst 118:56–69
26. Lai W, Xiaofeng W, Aihua W, Rigui Z, Changming Z (2018) Robust subspace segmentation by self-representation constrained low-rank representation. Neural Process Lett 48(3):1671–1691
27. Yanxi C, Gen L, Yuantao G (2018) Active orthogonal matching pursuit for sparse subspace clustering. IEEE Sig Process Lett 25(2):164–168
28. Chaoqun H, Jun Yu, Jian Z, Xiongnan J, Kyong-Ho L (2018) Multimodal face-pose estimation with multitask manifold deep learning. IEEE Trans Indus Inform 15(7):3952–3961
29. Jun Yu, Min T, Hongyuan Z, Dacheng T, Yong R (2019) Hierarchical deep click feature prediction for fine-grained image recognition. IEEE Trans Patt Anal Mach Intell
30. Liu G, Shuicheng Y (2011) Latent low-rank representation for subspace segmentation and feature extraction. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1615–1622
31. Canyi L, Hai M, Zhongqiu Z, Lin Z, Deshuang H, Shuicheng Y (2012) Robust and efficient subspace segmentation via least squares regression. In: Proceedings of the European conference on computer vision (ECCV), pp 347–360
32. Patel Vishal M, Nguyen Hien V, René V (2013) Latent space sparse subspace clustering. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 225–232
33. Jie C, Hua M, Yongsheng S, Zhang Y (2017) Subspace clustering using a symmetric low-rank representation. Knowled Based Syst 127:46–57
34. Huazhu C, Weiwei W, Xiangchu F, Ruiqiang H (2018) Discriminative and coherent subspace clustering. Neurocomputing 284:177–186
35. Xian F, Zhixin T, Feiyang S, Jialiang Y (2019) Robust subspace clustering via symmetry constrained latent low rank representation with converted nuclear norm. Neurocomputing 340:211–221
36. Lu C, Jiashi F, Zhouchen L, Shuicheng Y (2013) Correlation adaptive subspace segmentation by trace lasso. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1345–1352

37. Wang YX, Xu H, Leng C (2013) Provable subspace clustering: when LRR meets SSC. In: Proceedings of the conference on neural information processing systems (NeurIPS), pp 64–72
38. Xingyu X, Xianglin G, Guangcan L, Jun W (2017) Implicit block diagonal low-rank representation. IEEE Trans Image Process 27(1):477–489
39. Zhouchen L, Minming C, Yi M (2009) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical Report UILU-ENG-09-2215
40. Zhi H, Lin L, Suhong Z, Yi S (2016) Learning group-based sparse and low-rank representation for hyperspectral image classification. Patt Recogn 60:1041–1056
41. Jianbo S, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Patt Anal Mach Intell 22(8):888–905
42. Feng J, Zhouchen L, Huan X, Shuicheng Y (2014) Robust subspace segmentation with block-diagonal prior. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3818–3825
43. Nene Samer A, Nayar Shree K, Hiroshi M (1996) Columbia object image library (COIL-20). Technical Report CUCS-005-96
44. Georghiades Athinodoros S, Belhumeur Peter N, Kriegman David J (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Patt Anal Mach Intell 23(6):643–660
45. Chih LK, Jeffrey H, Kriegman David J (2005) Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans Patt Anal Mach Intell 27(5):684–698
46. van der Maaten L, Geoffrey H (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
47. Tron R, René V (2007) A benchmark for the comparison of 3-D motion segmentation algorithms. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
48. Karl P (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(11):559–572
49. Harold H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24(6):417–441

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