Learning Idempotent Representation for Subspace Clustering

Lai Wei, Shiteng Liu, Rigui Zhou, Changming Zhu, and Jin Liu, Member, IEEE

Abstract—The critical point for the success of spectral-type subspace clustering algorithms is to seek reconstruction coefficient matrices that can faithfully reveal the subspace structures of data sets. An ideal reconstruction coefficient matrix should have two properties: 1) it is block-diagonal with each block indicating a subspace; 2) each block is fully connected. We find that a normalized membership matrix naturally satisfies the above two conditions. Therefore, in this paper, we devise an idempotent representation (IDR) algorithm to pursue reconstruction coefficient matrices approximating normalized membership matrices. IDR designs a new idempotent constraint. And by combining the doubly stochastic constraints, the coefficient matrices which are close to normalized membership matrices could be directly achieved. We present an optimization algorithm for solving IDR problem and analyze its computation burden as well as convergence. The comparisons between IDR and related algorithms show the superiority of IDR. Plentiful experiments conducted on both synthetic and real-world datasets prove that IDR is an effective subspace clustering algorithm.

Index Terms—Subspace clustering, idempotent matrix, doubly stochastic constraint, normalized membership matrix.

I. INTRODUCTION

HIGH-DIMENSIONAL data samples that emerged in computer vision fields could be viewed as generated from a union of linear subspaces [1], [2], [3], [4]. subspace clustering, whose goal is to partition the data samples into several clusters with each cluster corresponding to a subspace, has attracted lots of researchers’ attention. In the past decades, many kinds of subspace clustering algorithms have been proposed [5], [6], [7], [8], [9], [10]. Among them, spectral-type methods showed more excellent performance in many applications such as motion segmentation, face clustering, and so on [9], [10], [11], [12].

Without loss of generality, suppose that a clean data matrix

\[ X = [X_1, X_2, \ldots, X_k] \in \mathbb{R}^{d \times n} \] contains \( n \) data samples drawn from \( k \) subspaces. \( X_i \subset X \) denotes the sub-matrix including \( n_i \) data samples lying in the \( i \)-th subspace, where \( \sum_{i=1}^{k} n_i = n \). And if \( i \neq j \) (\( i, j = 1, 2, \ldots, k \)), \( X_i \cap X_j = \emptyset \). The framework of spectral-type subspace clustering algorithms is divided into three parts. First, they learn a reconstruction coefficient matrix \( Z \in \mathbb{R}^{n \times n} \) satisfying \( X = XZ \). Second, an affinity matrix \( A \) is built by using the obtained reconstruction coefficient matrix, i.e.,

\[ [A]_{ij} = ([Z]_{ij} + [\mathbf{Z}^\top]_{ij})/2, \]

where \([A]_{ij} \) and \([Z]_{ij} \) denote the \((i, j)\)-th element of \( A \) and \( Z \) respectively, \( Z^\top \) is the transpose of \( Z \). Finally, a certain spectral clustering algorithm, e.g., normalized cuts (Ncuts) [13], is used to get the final clustering results by using \( A \). It could be seen that the performance of a spectral-type algorithm mainly relies on the learned reconstruction matrix. A good coefficient matrix should have inter-subspace sparsity and intra-subspace connectivity. Namely, if \( x_i \) and \( x_j \) belong to the same subspace, \([Z]_{ij} > 0 \). Otherwise, \([Z]_{ij} = 0 \).

Different spectral-type methods use different regularizers to produce coefficient matrices with different characteristics. For instance, sparse subspace clustering (SSC) [9], [14] pursues sparse reconstruction coefficient matrices by introducing a sparse constraint [15]. Low-rank representation (LRR) [10], [16] seeks a low-rank reconstruct coefficient matrix by minimizing the nuclear norm of the coefficient matrix. Least square regression (LSR) [17] defines a Frobenius norm regularizer and searches a dense reconstruction coefficient matrix. Block diagonal representation (BDR) [18] provides a \( k \) block-diagonal reconstruction coefficient matrix by minimizing the sum of the \( k \) smallest eigenvalues of the coefficient matrix’s Laplacian regularizer. Though these representative methods achieve promising results in different kinds of subspace clustering tasks, the obtained coefficient matrices still have some drawbacks. The coefficient matrices gotten by SSC are usually too sparse to lack connectedness within each subspace. The block-diagonal constraint used in BDR may not lead to the correct clustering, since each block still may not be fully connected. On the other hand, although the connectedness within subspaces is guaranteed in the dense coefficient matrices constructed by LRR and LSR, the coefficients of the inter-subspace samples are usually non-zero. To get away with the dilemmas, three different types of methods emerge.

First, some extensions of classical regularizers are developed. For example, Xu et al. developed a reweighted SSC method that used the reciprocal of each element of a coefficient matrix to define a weighted sparse constraint for the coefficient matrix [19]. Zhang et al. extended the nuclear norm regularizer used in LRR to a kind of Schatten-\( p \) norm regularizer [20]. Xu et al. proposed a scaled simplex representation by adding
the non-negative constraint and scaled affine constraint of the coefficient matrix obtained in LSR [21].

Second, mixed regularizers of coefficient matrices are applied. Li et al. proposed a structured sparse subspace clustering (SSSC) [22] by adding a re-weighted $l_1$-norm regularizer into SSC. Elastic net (EN) method defines a combination of $l_1$-norm and Frobenius regularizer of the coefficient matrices [23], [24]. Zhuang et al. integrated sparse constraint and nuclear norm regularizer to propose a non-negative low-rank and sparse representation method (NNLRSR) [25], Tang et al. generalized NNLRSR and devised a structure-constrained LRR (SCLRR) [26]. Lu et al. presented a graph-regularized LRR (GLRR) algorithm by minimizing the nuclear norm and the Laplacian regularizer of the coefficient matrix simultaneously [27]. Tang et al. designed a dense block and sparse representation (DBSR) method which used the $2$-norm (the maximal singular value) and $l_1$ norm regularizers to compute a dense block and sparse coefficient matrix [28].

Third, classical spectral-type subspace clustering algorithms are stacked to build cascade models. Wei et al. devised a sparse relation representation by stacking SSC and LRR [29]. Sui et al. also provided a similar method to show the effectiveness of cascade models [30]. These extended methods outperform the classical algorithms to a certain extent, but they still may not guarantee to produce ideal coefficient matrices.

From the viewpoint of a spectral clustering algorithm, the best affinity matrix $M^*$ of data set $X$ should have the following properties: If $x_i$ and $x_j$ belong to the same cluster, then $[M^*]_{ij} = 1$. Otherwise, $[M^*]_{ij} = 0$. In the correlation clustering domain, $M^*$ is called a membership matrix [31], [32], [33]. However, researchers proved that a variation of the membership matrix called the normalized membership matrix, is also adequate for spectral clustering. The normalized membership matrix $A^*$ corresponding to $M^*$ is expressed as follows:

$$A^* = \begin{pmatrix} \frac{1}{n_1} & 1_{n_1} & 1_{n_1}^T \ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{n_k} & 1_{n_k} & 1_{n_k}^T \end{pmatrix},$$  

(1)

where $1_{ni}$ is a column vector with $n_i$ elements and each element equals 1.

Back to the domain of subspace clustering, suppose an affinity matrix $A$ is a normalized membership matrix. As we mentioned above, in a spectral-type subspace clustering algorithm, an affinity matrix is defined as $[A]_{ij} = (|Z|_{ij} + |Z|_{ji})/2$. If we force $Z = Z^T$ and $|Z|_{ij} \geq 0$ (for all $i, j$), then it could be deduced that the reconstruction coefficient matrix $Z$ is also a normalized membership matrix. We can see $Z$ is definitely inter-subspace sparse and intra-subspace connected. Moreover, the property that each element in a block (i.e., $Z_k$) equals $1/n_k(>0)$ means $Z$ is fully connected in each block. Hence, this kind of coefficient matrix is better than that obtained by BDR. Fig. 1 presents the two coefficient matrices obtained by BDR and the proposed algorithm on a synthetic data set. We can see that all coefficient matrices are block diagonal, but each block in the coefficient matrices obtained by the proposed algorithms is denser than those of BDR. Then the subspace structure of the data set could be revealed more faithfully by using the proposed algorithm.

In [33], Lee et al. also suggested constructing a normalized membership matrix for subspace clustering. However, the so-called membership representation (MR) algorithm [33] takes three steps to finally get the coefficient matrix. First, a certain subspace clustering algorithm, such as SSC or LRR, is used to get an initial coefficient matrix. Second, MR seeks a membership matrix by using the obtained initial coefficient matrix. Finally, a normalized membership matrix is computed with the obtained membership matrix. In the last two steps, Augmented Lagrangian method (ALM) [34] is applied to solve the corresponding optimization problems. Hence, besides the computation time used for finding an initial coefficient matrix, the time cost in the last two steps of MR is also high.

In this paper, we invent a new method to find a coefficient matrix that is close to a normalized membership matrix as much as possible. The motivation of the proposed algorithm is the self-expressiveness property of the reconstruction coefficient vectors obtained by subspace clustering algorithms. As we know, spectral-type subspace clustering algorithms assume that the original data samples obey the self-expressiveness property [9], i.e., each data point can be well reconstructed by a linear combination of other points in the given dataset. The self-expressiveness property of the obtained coefficient vectors means each coefficient vector could be linearly reconstructed by other coefficient vectors. Based on this proposition and the doubly stochastic constraints [35], [36], an idempotent representation (IDR) method for subspace clustering is proposed. Different from MR, IDR could directly obtain a normalized membership matrix. And by comparative analysis with some related algorithms, IDR shows its superiority. Extensive experiments conducted on both synthetic and real-world databases also prove the effectiveness and efficiency of IDR.

The rest of the paper is organized as follows: we introduce the general formulation of spectral-type subspace clustering algorithms in Section II. In Section III, we propose the idea of idempotent representation (IDR) and the optimization algorithm for solving IDR problem. Further discussions of IDR, such as the analysis of the convergence and complexity of the optimization algorithm, and the connections between IDR and the related algorithms, are given in Section IV. Comparative subspace clustering experiments on both the synthetic data set and real-world...
data sets are performed in Section V. Section VI presents the conclusions.

II. PRELIMINARY

Though there is a wide variety of existing spectral-type subspace clustering algorithms, the general objective function of these algorithms could be expressed as follows:

\[
\min_{Z,E} \Omega(X, Z) \quad \text{s.t.} \quad X = XZ,
\]

where \( \Omega(X, Z) \) usually denotes a certain regularizer of \( Z, X \in \mathbb{R}^{d \times n} \) is a data matrix. In real applications, data is often noisy or corrupted. Hence, the more robust version of the above problem could be defined as follows:

\[
\min_{Z,E} \Omega(X, Z) + \lambda \Phi(E), \quad \text{s.t.} \quad X = XZ + E,
\]

where \( \Phi(E) \) is a certain measurement of \( E \). \( \lambda \) is a positive parameter that is used to balance the effects of \( \Omega(X, Z) \) and \( \Phi(E) \). Moreover, some algorithms add some additional constraints of \( Z \) which could be expressed as \( \Theta(Z) \). The main differences between the existing subspace clustering algorithms are the definitions of \( \Omega(\cdot, \cdot), \Phi(\cdot) \) and \( \Theta(\cdot) \). We use Table I to summarize the formulations of \( \Omega(X, Z), \Phi(E) \) and \( \Theta(Z) \) of some representative subspace clustering algorithms.

III. IDEMPOTENT REPRESENTATION

A. Motivation

The key point of the spectral-type subspace clustering algorithms is that they all assume the data samples obey the self-expressiveness property [9]. Namely, each data sample could be approximately reconstructed by a linear combination of other data points in the given dataset with tolerable errors. Thus, \( X \approx XZ \), and \( Z \) records the reconstruction relationship of the original data samples.

In addition, as described in [10], [16], the obtained coefficient matrix \( Z \) is a representation of the original data matrix \( X \) with \( z_i \) being the representation of \( x_i \). Here, \( z_i \) and \( x_i \) are the \( i \)-th columns of \( Z \) and \( X \) respectively. Then it is reasonable to assume that the coefficient vectors also obey the self-expressiveness property (Self-expressiveness property of coefficient vectors), namely, each coefficient vector could be linearly reconstructed by other coefficient vectors. Thus,

\[ Z \approx ZW, \]

where \( W \) is a reconstruction coefficient matrix corresponding to \( Z \). Moreover, we could hope \( W \) to be close to \( Z \). The reason is that if \( Z \) is a good representation of \( X \), \( Z \) should follow the reconstruction relationship of the original data set, and \( Z \) just records the reconstruction relationship of the original data samples. Therefore, the following equation holds

\[ Z \approx XZ = Z^2. \]

The above equation means that \( Z \) is approximate to an idempotent matrix.

It is easy to verify that an \( n \times n \) identity matrix \( I_n \) is idempotent and the solution to the problem \( X = XZ \). Then for a spectral-type subspace clustering algorithm, the above idempotent constraint (5) is not sufficient for finding a good coefficient matrix. Fortunately, it could be checked that a normalized membership matrix is also an idempotent matrix. Hence, we will show how to add some necessary constraints to compel an idempotent reconstruction coefficient matrix to be a normalized membership matrix.

Lee et al. pointed out that an idempotent matrix is a normalized membership matrix if and only if it is doubly stochastic [33]. And a doubly stochastic matrix \( Z \in \mathbb{R}^{n \times n} \) can be completely described by the following doubly stochastic constraints [35], [36]:

\[ 1^\top_n Z = 1^\top_n, \quad Z = Z^\top, \quad Z \geq 0. \]

However, these constraints still can not prevent \( Z \) to be \( I_n \). As mentioned above, for revealing the subspace structure of data set \( X \) with \( k \) subspaces faithfully, a coefficient matrix should be \( k \) block-diagonal. For an idempotent and doubly stochastic coefficient matrix \( Z \), we could simply let \( Tr(Z) = k \), then \( Z \) would be \( k \) block-diagonal. By integrating these constraints and the general formulation of subspace clustering algorithms, we could define the idempotent representation (IDR) problem as follows:

\[
\min_{Z} \|Z\|_{id} + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad X = XZ + E, \quad 1^\top_n Z = 1^\top_n, \quad Z = Z^\top, \quad Z \geq 0, \quad Tr(Z) = k, \quad \|Z\|_{id} = \|Z - Z^2\|_F^2.
\]

In most real applications, partial data samples are corrupted, we use \( l_{2,1} \) norm to measure the error term \( E \). It could also alleviate the influence of outliers.

TABLE I

| Algorithms | \( \Omega(X, Z) \) | \( \Phi(E) \) | \( \Theta(Z) \) |
|------------|------------------|-----------------|----------------|
| SSC       | \( \|Z\|_1 \)   | \( \|E\|_1 \)   | \( \text{diag}(Z) = 0_n \) |
| LRR       | \( \|Z\|_2 \)   | \( \|E\|_2 \)   | \( \text{diag}(Z) = 0_n \) |
| LSR       | \( \|Z\|_F \)   | \( \|E\|_F \)   | \( \text{diag}(Z) = 0_n \), \( \|Z - Z^\top\|_F = 0_n \) |
| BDR       | \( \|Z\|_k \)   | \( \|E\|_2 \)   | \( \text{diag}(Z) = 0_n \), \( Z = Z^\top \), \( Z \geq 0 \) |
| RSSL       | \( \|W \odot Z\|_1 \) | \( \|E\|_1 \)   | \( \text{diag}(Z) = 0_n \) |
| SSCS      | \( \{(I_n + \gamma Q) \odot Z\}|_1 \) | \( \|E\|_1 \)   | \( \text{diag}(Z) = 0_n \) |
| EN        | \( \|Z\|_F + \gamma \|Z^\top\|_F \) | \( \|E\|_1 \)   | \( \text{diag}(Z) = 0_n \) |
| SCLR       | \( \|Z\|_2 + \gamma \|Z\|_2 \) | \( \|E\|_2 \)   | - |
| GLRR       | \( \|Z\|_2 + \gamma \|Z\|_2 \) | \( \|E\|_2 \)   | - |
| DBSR       | \( \|Z\|_2 + \gamma \|Z\|_2 \) | \( \|E\|_2 \)   | - |
| CASE [37]  | \( \sum_{i=1}^n \|X \text{Diag}(z_i)\|_2 \) | \( \|E\|_F \)   | - |

Note: \( \gamma > 0 \) is a parameter. \( \text{diag}(Z) \) denotes a column vector composed by the elements in the diagonal of \( Z, 0_n \) is a column vector with each element equals 0. In BDR, \( \|Z\|_k \) is a constraint which forces \( Z \) to be \( k \) block-diagonal. In RSSL, \( W \in \mathbb{R}^{n \times n} \) is a weighted matrix with \( \|W\|_{ij} = 1/(\|Z\|_{ij} + \epsilon) \), where \( \epsilon > 0 \) is a small parameter. In SSCS, \( Q \in \mathbb{R}^{n \times n} \) is a weighted matrix updated by the segmentation results in each iteration and \( I_n \) is an \( n \times n \) identity matrix. In GLRR, \( Tr(\cdot) \) denotes the trace of a matrix and \( L \) is the Laplacian matrix built by using \( k \)-nearest neighbors (KNN) [38] and \( X \). In CASE, \( \text{Diag}(z_i) \) is an operation that construct a diagonal matrix whose diagonal is \( z_i \), \( z_i \) is the \( i \)-th column of \( Z \).
All these restrictions imposed on $Z$ will limit its representation capability. For alleviating this problem, we introduce an intermediate-term and propose the following relaxed problem:

$$\min_{Z,S} \|Z - S\|_F^2 + \gamma \|S - SC\|_F^2 + \lambda \|E\|_{2,1},$$
$$\text{s.t.} \quad X = XZ + E, \quad 1_n^T S = 1_n^n, S = S^T, S \geq 0, \quad Tr(S) = k,$$  
(8)

where $\gamma$ is also a positive parameter.

**B. Optimization**

Similar to solving the existing subspace clustering problems, we use ALM [34] to find the solutions to IDR problem (e.g., (8)). First, we need to transfer (8) to the following equivalent problem:

$$\min_{Z,S,C,D} \|Z - S\|_F^2 + \gamma \|S - SC\|_F^2 + \lambda \|E\|_{2,1},$$
$$\text{s.t.} \quad X = XZ + E, \quad S = C, \quad S = S^T, S \geq 0, \quad 1_n^T C = 1_n^n, S = D, \quad Tr(D) = k,$$  
(9)

where $C, D$ are two auxiliary variables. The corresponding augmented Lagrangian function of (9) could be expressed as follows:

$$\mathcal{L} = \|Z - S\|_F^2 + \gamma \|S - SC\|_F^2 + \lambda \|E\|_{2,1} + \langle Y_1, X - XZ - E \rangle + \langle Y_2, S - C \rangle$$
$$+ \langle Y_3, 1_n^T C - 1_n^n \rangle + \langle Y_4, S - D \rangle$$
$$+ \mu/2 \|X - XZ - E\|_F^2 + \|S - C\|_F^2$$
$$+ \|1_n^T C - 1_n^n\|_F^2 + \|S - D\|_F^2,$$  
(10)

where $Y_1, Y_2, Y_3$ and $Y_4$ are four Lagrangian multipliers and $\mu > 0$ is an additional parameter. By minimizing $\mathcal{L}$, the variables $Z, S, C, D, E$ could be optimized alternately while fixing others.

1. **Fix Other Variables and Update $Z$:** In the $h$-th iteration (Suppose $h$ is the current number of iterations),

$$Z^{h+1} = \arg \min_Z \|Z - S^h\|_F^2 + \langle Y_1^h, X - XZ - E^h \rangle$$
$$+ \mu^h/2 \|X - XZ - E^h\|_F^2$$
$$= \arg \min_Z \|Z - S^h\|_F^2 + \mu^h/2 \|X - XZ - E^h\|_F^2$$
$$+ Y_1^h/\mu^h\|_F^2,$$  
(11)

where $S^h, Y_1^h$ and $\mu^h$ are the updated variables. It could be easily verified that

$$Z^{h+1} = (2I_n + \mu^h X^T X)^{-1} (2S^h + \mu^h (X^T X - X^T E^h)$$
$$+ X^T Y_1^h),$$  
(12)

where $(\cdot)^{-1}$ denotes the pseudo-inverse of a matrix.

2. **Fix Other Variables and Update $S$:** Similar to updating $Z$,

$$S^{h+1} = \arg \min_S \|Z^{h+1} - S\|_F^2 + \gamma \|S - SC^h\|_F^2$$
$$+ \langle Y_2^h, S - C^h \rangle + \langle Y_4^h, S - D^h \rangle$$
$$+ \mu^h/2 \|S - C^h\|_F^2 + \|S - D^h\|_F^2$$
$$= \arg \min_S \|Z^{h+1} - S\|_F^2 + \gamma \|S - SC^h\|_F^2$$
$$+ \mu^h/2 \|S - C^h\|_F^2 + \|S - D^h\|_F^2$$
$$+ Y_2^h/\mu^h\|_F^2 + Y_4^h/\mu^h\|_F^2 + \|S - D^h\|_F^2,$$  
(13)

Hence,

$$S^{h+1} = \left(2Z^{h+1} + \mu^h C^h - Y_2^h + \mu^h D^h - Y_4^h\right)$$
$$\left((2 + \mu^h)I_n + 2\gamma(I_n - C^h)(I_n - C^h)^T\right)^{-1},$$  
(14)

Because of the non-negative and symmetric constraints on $S$, we further let $S^{h+1} = \max(S^{h+1}, 0)$ and $S^{h+1} = (S^{h+1} + (S^{h+1})^T)/2$.

3. **Fix Other Variables and Update $C$:** We also could find

$$C^{h+1} = \arg \min_C \|S^{h+1} - S^{h+1}C\|_F^2 + \langle Y_3^h, S^{h+1} - C \rangle$$
$$+ \langle Y_4^h, S^{h+1} - D \rangle$$
$$+ \mu^h/2 \|S^{h+1} - C\|_F^2$$
$$+ \|1_n^T C - 1_n^n\|_F^2,$$  
(15)

Then

$$C^{h+1} = \left(2\gamma(S^{h+1})^T S^{h+1} + \mu^h (I_n + 1_n 1_n^T)\right)^{-1}$$
$$\left(2\gamma(S^{h+1})^T S^{h+1} + Y_3^h - 1_n 1_n^T + \mu^h (S^{h+1} + 1_n 1_n^T)\right).$$  
(16)

4. **Fix Other Variables and Update $D$:** For updating $D$, we could get the following problem:

$$\min_D \|Y_4^h - S^{h+1} - D\|_F^2 + \mu^h/2 \|S^{h+1} - D\|_F^2,$$
$$\text{s.t.} \quad Tr(D) = k.$$  
(17)

where $T = S^{h+1} + Y_4^h/\mu^h$. Note that the constraint is just imposed on the diagonal elements of $D$, hence $[D^{h+1}]_{ij} = [T]_{ij}$, if $i \neq j(i, j = 1, 2, \ldots, n)$. Let $d = diag(D^{h+1})$ and $t = diag(T)$, then we have

$$\min_d \|d - t\|_2, \quad \text{s.t.} \quad 1_n^T d = k.$$  
(18)

This problem could be solved by any off-the-shelf quadratic programming solver. We here provide a more efficient method to achieve the solution to Problem (18). The Lagrangian function of Problem (18) is

$$\mathcal{L} = \|d - t\|_2^2 - \eta(1_n^T d - k),$$  
(19)

where $\eta > 0$ is a Lagrange multiplier. By taking the derivative of (19) w.r.t. $d$ and set it to zero, we have

$$2(d - t) - \eta 1_n = 0.$$  
(20)

Then for the $j$-th element of $d$, we have

$$d_j - t_j - \eta/2 = 0,$$  
(21)

where $d_j$ and $t_j$ are the $j$-th element of $d$ and $t$ respectively. According to the constraint $1_n^T d = k$ in Problem (18), then

$$\eta = 2(k - 1_n^T t)/n.$$  
(22)

Hence,

$$d = t + 1_n(k - 1_n^T t)/n.$$  
(23)
By summarizing the above computations,
\[
D^{h+1} = T + \text{Diag} \left(1_n(k - 1^T_d \text{diag}(T))/n \right).
\] (24)

5. Fix other variables and update E: From (10), it could be easily obtained as follows:
\[
E_{h+1} = \arg \min_{E} \lambda ||E||_{2,1} + <Y^h, X - XZ^{h+1} - E > \\
+ \mu^h ||X - XZ^{h+1} - E||_F^2 \\
= \arg \min_{E} \lambda ||E||_{2,1} + \mu^h/2 ||X - XZ^{h+1} - E > \\
+ Y^h/\mu^h||E||_F^2.
\] (25)
The above problem could be solved by following the Lemma 4.1 presented in [10].

6. Fix Other Variables and Update Parameters: The precise updating schemes for the Langrangian multipliers
\[
Y_1, Y_2, Y_3, Y_4 \text{ and parameter } \mu \text{ existed in (10) are summarized as follows:}
\]
\[
Y_1^{h+1} = Y^h + \mu^h(X - XZ^{h+1} - E^{h+1}),
\]
\[
Y_2^{h+1} = Y^h + \mu^h(S^{h+1} - C^{h+1}),
\]
\[
Y_3^{h+1} = Y^h + \mu^h(1_n^T C^{h+1} - 1^n),
\]
\[
Y_4^{h+1} = Y^h + \mu^h(S^{h+1} - D^{h+1}),
\]
\[
\mu^{h+1} = \min(\mu_{\text{max}}, \rho \mu^h),
\] (26)
where \( \mu_{\text{max}} \) and \( \rho \) are two given positive parameters.

C. Algorithm

We summarize the algorithmic procedure of IDR in Algorithm 1. For a data set, once the solutions to IDR are obtained, we use Z and S to define two affinity graphs \( G_1 \) and \( G_2 \) as \( |G_1|_{ij} = (||Z||_{ij} + ||Z||_{ij})/2 \) and \( |G_2|_{ij} = (||S||_{ij} + ||S||_{ij})/2 \). Then Neuts is consequently performed on the two graphs to get two clustering results. Finally, the best one would be chosen as the final result.

IV. FURTHER ANALYSES

A. Complexity Analysis

We can see that the complexity of Algorithm 1 is mainly determined by the updating of five variables \( Z, S, C, D, E \). In each iteration, these variables all have closed-form solutions. For updating \( Z, S \) and \( C \), it needs to compute the pseudo-inverse of an \( n \times n \) matrix, hence the computation burden is \( O(n^3) \). For updating \( D \), it takes \( O(n^2) \) to compute the multiplier of an \( n \times n \) matrix. And for updating \( E \) by using Lemma 1 presented in [10], its time cost is \( O(n) \). Hence, the time complexity of Algorithm 1 in each iteration taken together is \( O(n^3) \). In all our experiments, the number of iterations of Algorithm 1 is less than 300, hence its total complexity is \( O(n^3) \).

B. Convergence Analysis

We present a theoretical convergence proof of the proposed Algorithm 1 in this subsection.

Algorithm 1: Idempotent Representation (IDR).

Input:
Data set \( X = [x_1, x_2, \ldots, x_n] \in \mathcal{R}^{d\times n} \), parameters \( \gamma, \lambda \), the number of subspaces \( k \), the maximal number of iteration \( \text{Maxiter} \).

Output:
The coefficient matrix \( Z^* \) and \( S^* \) and the noise term \( E^* \).

1: Initialize the parameters, i.e.,
\( h = 0, \mu^h = 10^{-6}, \mu_{\text{max}} = 10^4, \rho = 1.1, \varepsilon = 10^{-7} \) and
\( Y^h = Y^h = Y^h = 0 \), randomly Initialize \( Z^h, S^h, C^h, D^h \in \mathcal{R}^{n\times n} \).

2: while \( ||X - XZ^h - E^h||_{\infty} > \varepsilon, ||S^h - C^h||_{\infty} > \varepsilon, \)
\( ||S^h - D^h||_{\infty} > \varepsilon, ||1^n C^h - 1^n||_{\infty} > \varepsilon \) and
\( h < \text{Maxiter} \)
3: Update \( Z^{h+1}, S^{h+1}, C^{h+1}, D^{h+1} \) by using Eq.(12),
(14), (16) and (24) respectively, and further let \( S^{h+1} = \max(S^{h+1}, 0) \), \( S^{h+1} = (S^{h+1} + (S^{h+1})^T)/2 \);

4: Update \( E^{h+1} \) by solving Problem (25);

5: Update \( Y_1^{h+1}, Y_2^{h+1}, Y_3^{h+1}, Y_4^{h+1} \) and \( \mu^{h+1} \) by using
(26);

6: \( h = h + 1 \);
7: end while
8: return \( Z^* = Z^h, S^* = S^h \).

Proposition 1: Algorithm 1 is convergent and the sequence \( \{Z^h, S^h, C^h, D^h, E^h\} \) generated by Algorithm 1 would converge to a stationary point.

Proof: Algorithm 1 aims to minimize the Lagrangian function of (10) by alternately updating the variables \( Z, S, C, D, E \). First, from the updating rule of \( Z^{h+1} \) in (12), we have
\[
Z^{h+1} = \arg \min_{Z} \mathcal{L}(Z^h, S^h, C^h, D^h, E^h). \] (27)
It could be easily deduced that \( \mathcal{L}(Z^h, S^h, C^h, D^h, E^h) \) is \( \beta \)-strongly convex w.r.t. \( Z \). The following inequality holds:
\[
\mathcal{L}(Z^{h+1}, S^h, C^h, D^h, E^h) \leq \mathcal{L}(Z^h, S^h, C^h, D^h, E^h) - \beta/2 ||Z^{h+1} - Z^h||_F^2. \] (28)
Here we use Lemma B.5 in [39].

Second, variables \( C, D, E \) have similar properties of \( Z \). Hence, the corresponding inequalities of these variables similar to (28) would hold. By adding these inequalities, we have
\[
\mathcal{L}(Z^{h+1}, S^{h+1}, C^{h+1}, D^{h+1}, E^{h+1}) \leq \mathcal{L}(Z^h, S^{h+1}, C^h, D^h, E^h) - \beta/2 ||Z^{h+1} - Z^h||_F^2 \\
+ ||C^{h+1} - C^h||_F^2 + ||D^{h+1} - D^h||_F^2 + ||E^{h+1} - E^h||_F^2. \] (29)
Third, for the variable \( S \), when \( S^{h+1} \) is computed by using (14). We further let \( S^{h+1} = \max(S^{h+1}, 0) \), \( S^{h+1} = (S^{h+1} + (S^{h+1})^T)/2 \). These additional constraints will not affect the convergence of the algorithm.

Suppose a variable \( H \) is most close to \( S \) and satisfies \( H \geq 0 \) and \( H = H^T \), then \( H \) should be the solution to the following
problem:
\[
\min_{\mathbf{H}} \| (\mathbf{S}^h + 1) - \mathbf{H} \|_F = \| (\mathbf{S}^h + 1) \|_F - \mathbf{H} \|_F
\]
\[
= \frac{1}{2} \| (\mathbf{S}^h + 1) \|_F^2 + \frac{1}{2} \| (\mathbf{S}^h + 1) \|_F^2 - \mathbf{H} \|_F^2,
\]
(30)
where \((\mathbf{S}^h + 1)^+ = \max(0, \mathbf{S}^h + 1)\). By taking the derivative of above problem w.r.t. \(\mathbf{H}\), we have \(\mathbf{H} = \frac{1}{2}(\mathbf{S}^h + 1) + (\mathbf{S}^h + 1)^+\). This means \(\mathbf{S}^h + 1\) with additional constraints is the solution to (10) in-th iteration. Hence, we have
\[
\mathcal{L}(\mathbf{Z}^h, \mathbf{S}^h + 1, \mathbf{C}^h, \mathbf{D}^h, \mathbf{E}^h) \leq \mathcal{L}(\mathbf{Z}^h, \mathbf{S}^h, \mathbf{C}^h, \mathbf{D}^h, \mathbf{E}^h)
\]
(31)
By combing inequalities (29) and (31) together, we can conclude \(\mathcal{L}(\mathbf{Z}^h, \mathbf{S}^h, \mathbf{C}^h, \mathbf{D}^h, \mathbf{E}^h)\) monotonically decreasing and thus it is upper bounded. Moreover, by summing inequality (29) over \(h = 1, 2, \ldots\), we have
\[
\mathcal{L}(\mathbf{Z}^h + 1, \mathbf{S}^h + 1, \mathbf{C}^h + 1, \mathbf{D}^h + 1, \mathbf{E}^h + 1)
\]
\[
+ \sum_{h=1}^{+\infty} \frac{\beta}{2} \| \mathbf{Z}^h + 1 - \mathbf{Z}^h \|_F^2 + \| \mathbf{C}^h + 1 - \mathbf{C}^h \|_F^2
\]
\[
+ \| \mathbf{D}^h + 1 - \mathbf{D}^h \|_F^2 + \| \mathbf{E}^h + 1 - \mathbf{E}^h \|_F^2
\]
\[
\leq \mathcal{L}(\mathbf{Z}^0, \mathbf{S}^0, \mathbf{C}^0, \mathbf{D}^0, \mathbf{E}^0).
\]
(32)
Moreover, in (10), \(\| \mathbf{S} - \mathbf{C} \|_F^2, \| \mathbf{I}_n^{\top} \mathbf{C} - \mathbf{I}_n \|_F^2, \| \mathbf{S} - \mathbf{D} \|_F^2\) are all strongly convex, by invoking the first-order necessary and the updating schemes of Lagrange multipliers ((26)), we have \(\mathcal{L}(\mathbf{Z}^h + 1, \mathbf{S}^h + 1, \mathbf{C}^h + 1, \mathbf{D}^h + 1, \mathbf{E}^h + 1) \geq \sigma [40]\), where \(\sigma\) is a fixed value. This implies when \(h \to +\infty\),
\[
\mathbf{Z}^h + 1 - \mathbf{Z}^h \to 0, \mathbf{C}^h + 1 - \mathbf{C}^h \to 0,
\]
\[
\mathbf{D}^h + 1 - \mathbf{D}^h \to 0, \mathbf{E}^h + 1 - \mathbf{E}^h \to 0.
\]
(33)
Namely, \(\{\mathbf{Z}^h, \mathbf{C}^h, \mathbf{D}^h, \mathbf{E}^h\}\) would convergent to a stationary point of (10). \(\mathbf{S}\) is computed based on other variables, hence \(\mathbf{S}^h\) will converge to a stable value. Therefore, the convergence of Algorithm 1 is guaranteed.

### C. Block-Diagonal Property of Idempotent Constraint

In this subsection, we give the following proposition.

**Proposition 2:** The coefficient matrix \(\mathbf{Z}\) obtained by IDR is block-diagonal.

**Proof:** First, IDR problem, i.e., (8), falls into the general subspace clustering formulation, namely
\[
\min_{\mathbf{Z}} \Omega(\mathbf{X}, \mathbf{Z}) \text{ s.t. } \mathbf{X} = \mathbf{XZ}.
\]
(34)
Second, for any permutation matrix \(\mathbf{P}\), it is easy to verify that (8) satisfies \(\Omega(\mathbf{X}, \mathbf{Z}) = \Omega(\mathbf{PX}, \mathbf{PZP}^\top)\) which is the first EBD (enforced block diagonal) condition [18].

Third, the previous subsection has proved that the equivalent problem of (8) has a unique solution. Hence, it could be deduced that (8) also has a unique solution. Based on the above explanations and **Theorem 3** in [18], we can conclude that the coefficient matrix obtained by IDR is block-diagonal.

### D. Comparative Analysis With Related Algorithms

We now discuss the relationships between IDR and some related algorithms.

1) **Comparative Analysis With Membership Representation (MR):** As we mentioned in Section I, MR also proposes to learn a normalized membership matrix as a reconstruction coefficient matrix [33]. However, MR consists of three steps:

   First, an initial coefficient matrix is leaned by using SSC or LRR. Second, a membership matrix \(\mathbf{M}\) is constructed by solving a weighted sparse optimization problem with certain constraints. Third, after \(\mathbf{M}\) is obtained, a normalized membership matrix \(\mathbf{Z}\) is achieved by optimizing a trace minimization problem.

   Besides the computation for finding the initial coefficient matrix, MR also needs to use ALM to solve the problems that existed in the second and third steps. Hence, MR is very time-consuming. Additionally, the performance of MR depends on the learned initial coefficient matrices. The parameter in SSC or LRR will influence its performance. How to choose an initial coefficient matrix is not reported in [33]. Moreover, the three hyper-parameters that existed in MR will make the tuning of the parameters difficult.

2) **Comparative Analysis With Doubly Stochastic Subspace Clustering (DSSC) [41]:** Based on the descriptions in Section III-A, it could be seen that the normalized membership matrix obtained by IDR is a special case of doubly stochastic matrices. Recently, Lim et al. devised a doubly stochastic subspace clustering (DSSC) algorithm [41] that pursued a doubly stochastic coefficient matrix. By using two different strategies to solve DSSC problem, two different models, namely joint DSSC (J-DSSC) and approximation DSSC (A-DSSC), are presented. Among them, A-DSSC is a two-step algorithm that first uses LSR or EN to get an initial coefficient matrix and computes a doubly stochastic matrix consequently. On the other hand, the computation burden of J-DSSC is high, because, in each iteration of J-DSSC, two intermediate \(n \times n\) matrices should be iteratively updated by using linear alternating direction method of multipliers (ADMM) [42]. Moreover, we also could see that DSSC has three hyper-parameters which will also lead the difficulties in parameters adjustment.

3) **Comparative Analysis With Self-Representation Constrained LRR (SRLRR) [43]:** The idempotent constraint of a coefficient matrix is first proposed in our previous work [43]. The main problem existing in SRLRR is that we have not built solid theoretical connections between \(\mathbf{Z}\) and a normalized membership matrix. The nuclear norm minimization and the affine constraint (i.e., \(\mathbf{1}_n^\top \mathbf{Z} = \mathbf{1}_n^\top\) [44]) in SRLRR are used to avoid \(\mathbf{Z}\) to degenerate to \(\mathbf{I}_n\) or \(\mathbf{0}_{n \times n}\). This is totally different from IDR.

Researchers may doubt why we impose the idempotent constraint on the coefficient matrix \(\mathbf{Z}\) instead of the affinity matrix \(\mathbf{A}\). The reasons come from two sides. First, according to the descriptions in Section III-A, we could see the idempotent constraint of the coefficient matrix is directly deduced based on the self-expressiveness property of coefficient vectors. It makes intuitive physical sense as opposed to placing the constraint on the affinity matrix. On the other hand, because the obtained
coefficient matrix of IDR also satisfies the doubly stochastic constraints, it could be easy to deduce that the constructed affinity matrix is also an idempotent matrix.

E. Comparative Analysis With Other Graph Construction Methods

Spectral-type subspace clustering algorithms present different affinity graph construction methods. Besides the existing spectral-type subspace clustering algorithms, there are some representative graph construction methods. For example, Li et al. proposed a rank-constrained graph construction (RCGC) method for spectral clustering and subspace learning [45]. RCGC also hopes to find a low-rank affinity graph, but its optimization process is different from BDR. Moreover, RCGC needs a pre-defined affinity graph. Zhu et al. devised a dense affinity graph construction method [46]. In our opinion, the main difference between Zhu’s method and LSR is that they use different error terms. LSR aims to minimize the reconstruction error of a data matrix while Zhu’s method tries to minimize the Laplacian regularizer of the data matrix. Hence, we can see that the graph construction method presented in IDR is totally different from the existing graph construction methods.

V. EXPERIMENTS

A. Experiment Setup

1) Datasets: Both synthetic and real-world data sets are applied in our experiments to verify the effectiveness of IDR. Seven benchmark databases including Hopkins 155 motion segmentation data set [47], ORL face image database [48], AR face image database [49] and MNIST handwritten digital database,2 Umist 3, COIL20 [50] and COIL40 databases are used for evaluation.

2) Comparison Methods: The closely related algorithms such as SSC [14], LRR [10], [16], LSR [17], BDR [18], MR [33] and DSSC [41] will be used for deep comparative analysis.4 And some modern subspace clustering algorithms including low-rank subspace clustering (LRSC) [5], sparse subspace clustering by orthogonal matching pursuit (SSCOMP) [51], scaled simplex representation based subspace clustering (SSRSC) [52], graph filter LSR (FLSR) [53], graph convolutional subspace clustering (GCSC) [54] will be also tested on some databases for comparison. All the experiments are conducted on a Windows-based machine with an Intel i7-4790 CPU with 20-GB memory and MATLAB R2017b.

3) Parameters Setting: Because the value of parameters will influence the performance of the evaluated algorithms, for each compared method, we will tune all the parameters by following the suggestions in corresponding references and retain those with the best performance on each data set. The chosen parameter settings for those closely related algorithms are given in Table II. Especially, for MR, when SSC or LRR is used to achieve the initial coefficient matrix, the parameter corresponding to the two algorithms would be chosen in [0.001,0.01,0.1,1,5,10] according to the description in [33].

4) Evaluation Metrics: For all the evaluated algorithms, we use the obtained coefficient matrices to construct the affinity graphs without any post-processing. For the performance evaluation, we use segmentation accuracy (i.e., clustering accuracy) (SA) or segmentation error (SE), which is defined as follows:

\[ SA = \frac{1}{n} \sum_{i=1}^{n} \delta(g_i, f(o_i)), SE = 1 - SA, \]  

(35)

where \( g_i \) and \( o_i \) represent the ground truth and the output label of the \( i \)-th point respectively, \( \delta(x,y) = 1 \) if \( x = y \), and \( \delta(x,y) = 0 \) otherwise, and \( f(o_i) \) is the best mapping function that permutes clustering labels to match the ground truth labels.

B. Experiments on a Synthetic Data Set

We generate 5 subspaces for each dimension \( d = 5 \) in an ambient space of dimension \( D = 20 \). We sample 50 data points from each subspace and construct a \( D \times 250 \) data matrix \( X \). Moreover, a certain percentage \( p = 0 \sim 100\% \) of the data vectors are chosen randomly to add Gaussian noise with zero mean and variance \( 0.3\|x\|^2 \). Finally, the evaluated algorithms are used to segment the data into 5 subspaces. For a certain \( p \), the experiments would repeat 20 trials. Therefore, there would be a total of 220 subspace clustering tasks.

Similar experiments could be found in some existing references [16], [22]. But in these experiments, the parameters of the corresponding evaluated algorithm would be fixed when the algorithm is performed on each sub-database. Then by changing the parameter(s), the best results with certain parameter(s) would be finally selected. However, performing subspace clustering on a sub-database should be viewed as a sole segmentation task. In our experiments, we hence will let the parameter(s) of each algorithm vary in the corresponding interval sets in Table II and record the highest segmentation accuracies of the evaluated algorithms on each sub-database. Then the mean of these highest segmentation accuracies (averaged from 20 random trials) of

| Methods | Parameters |
|---------|------------|
| SSC    | \( \lambda \in \{0.0001,0.01,0.1,1,10,25,50,100,200,500,600,800,1000\} \) |
| LRR    | \( \lambda \in \{0.0001,0.01,0.05,0.1,0.2,0.5,1,2,5,8,10,15,20,50\} \) |
| LSR    | \( \lambda \in \{0.0001,0.01,0.05,0.1,0.2,0.5,1,2,5,8,10,15,20,50\} \) |
| BDR    | \( \lambda \in \{0.001,0.01,0.05,0.1,0.2,0.5,1,2,3,5,8,10,15,20,50\} \) |
| MR     | \( \lambda \in \{0.001,0.1,1,5,10\} \) |
| DSSC(DSSC) | \( \lambda \in \{0.01,0.25,1,2,5\} \) |
| DSSC(DSSC) | \( \eta \in \{0.01,0.05,0.1,0.2\} \) |
| IDR    | \( \lambda, \gamma \in \{0.0001,0.005,0.01,0.02,0.05,0.1,0.2,0.5,1,5,10,50,100,200\} \) |

2http://yann.lecun.com/exdb/mnist/
3http://images.ec.umist.ac.uk/danny/database.html
4We also provide the Matlab codes of MR and DSSC on https://github.com/weilyshmtu/Learning-idempotent-representation-for-subspace-segmentation. And the Matlab codes for SSC and LRR could be found on http://www.vision.jhu.edu/code/ and http://sites.google.com/site/guangcanliu/ respectively. The Matlab codes of LSR and BDR could be found on https://canyilu.github.io/code/
Fig. 2. The segmentation accuracies of each method versus variation in the range of corruption. (a) The best results of the evaluated algorithms, (b) the detailed results of IDR, BDR, MR, and DSSC, where BDR-Z denotes the results obtained by the reconstruction coefficient matrix $Z$ and BDR-B indicates the results obtained by the intermediate matrix introduced in BDR problem.

each algorithm versus variation of the percentage of corruption are reported in Fig. 2.

In addition, IDR and BDR can produce two coefficient matrices to compute the clustering results. By using different methods (SSC and LRR) to construct initial coefficient matrices, MR could obtain two different results. Based on different strategies, DSSC has two sub-models, namely JDSSC and ADSSC. Hence, we plot the accuracies of all the algorithms by selecting the better ones of the corresponding two results in Fig. 2(a). The detailed segmentation accuracies of IDR and BDR by using two different coefficient matrices and the results of MR based on SSC and LRR (denoted as MR-SSC and MR-LRR) as well as JDSSC and ADSSC are plotted in Fig. 2(b).

From Fig. 2(a), we can see that 1) IDR constantly achieves the best results; 2) the performances of LRR, LSR, MR, and DSSC are close to each other when the percentage of corruption is smaller than 50%; 3) when the percentage of corruption is larger than 50%, MR dominates LRR, LSR and DSSC; 4) SSC is inferior to other algorithms.

From Fig. 2(b), it can be seen that 1) the results obtained by two different coefficient matrices corresponding to IDR and BDR respectively are close to each other; 2) the performance of JDSSC and ADSSC is also similar to each other; 3) However, the results of MR-LRR are much better than those of MR-SSC. This means that the performance of MR relies on the initial coefficient matrices.

In order to show the sensitivity of IDR to its two parameters $\gamma$ and $\lambda$, we report the segmentation accuracies of IDR changed with the values of parameters. The sub-databases with $p = 10\%, 50\%, 90\%$ are now used. Then the mean of segmentation accuracies against the pairs of parameters is illustrated in Fig. 3.

From Fig. 3, we could see that 1) the performance of IDR is stable when the parameters varied in relatively large intervals; 2) when the corruption percentage is low, IDR is insensitive to $\gamma$. However, when the corruption percentage is high, small $\gamma$ and $\lambda$ could help IDR to achieve good results. We believe that when a data set is clean, a normalized membership reconstruction coefficient matrix is easy to get, so the idempotent constraint could also be satisfied. However, when most data samples in the data set are corrupted, the normalized membership reconstruction coefficient matrix is difficult to obtain. Hence, in such situations, the corresponding parameter $\gamma$ should be small.

C. Experiments on Hopkins 155 Data Set

Hopkins155 database is a well-known benchmark database to test the performances of subspace clustering algorithms. It consists of 120 sequences of two motions and 35 sequences of three motions. Each sequence is a sole clustering task and there are 155 clustering tasks in total. The features of each sequence are extracted and tracked along with the motion in all frames, and errors are manually removed for each sequence. We illustrate the sample images from Hopkins 155 database in Fig. 4.

We performed the experiments with the original data matrix and projected data matrix in $4s$—dimensional subspace obtained by using principal component analysis (PCA) [38]. Then the segmentation errors (i.e., $SE = 1 − SA$) of each evaluated algorithm are computed on each sequence.

First, we collect all the best results of each algorithm obtained on 155 sequences with the parameters changing in the given intervals. And the mean, median, and std. (standard variance)
The segmentation errors (%) and average computation time (sec.) of different algorithms on the Hopkins 155 database with the original data points are reported in Table III.

### Table III

| Methods | Average time (sec.) | 2 motions | median | std. | 3 motions | median | std. | All motions | median | std. |
|---------|---------------------|----------|--------|------|-----------|--------|------|-------------|--------|------|
| IDR-Z   | 9.15                | 0.25     | 0.18   |      | 1.14      | 0.20   | 2.11 | 0.46        | 0.18   | 1.47 |
| IDR-S   | 0.50                | 0.89     |        |      | 2.23      | 0.56   | 3.29 | 0.89        | 2.44   |      |
| SSC     | 2.98                | 1.66     | 5.13   |      | 5.29      | 1.46   | 7.35 | 2.48        | 5.88   |      |
| LRR     | 3.18                | 1.15     | 3.19   |      | 4.17      | 1.20   | 5.99 | 1.83        | 4.17   |      |
| LSR     | 0.03                | 0.56     | 2.18   |      | 1.94      | 0.21   | 4.12 | 0.87        | 2.79   |      |
| BDR     | 5.50                | 0.58     | 2.78   |      | 2.77      | 0     | 5.10 | 1.06        | 3.42   |      |
| BDR-Z   | 0.85                | 0.6      | 2.76   |      | 2.77      | 0     | 5.10 | 1.09        | 3.53   |      |
| MR-SSC  | 41.15               | 2.71     | 6.56   |      | 9.22      | 6.05   | 9.18 | 4.33        | 7.78   |      |
| MR-LRR  | 43.29               | 1.39     | 3.95   |      | 6.52      | 2.85   | 6.82 | 2.66        | 5.28   |      |
| JDSSC   | 16.29               | 12.51    | 11.45  | 10.54 | 24.09     | 25.06  | 11.62 | 15.13       | 14.48  | 11.8 |
| ADSSC   | 0.07                | 2.42     | 0      | 5.67 | 8.74      | 5.37   | 9.65 | 3.85        | 7.24   |      |

The best results of those algorithms are emphasized in bold.

The segmentation errors (%) and average computation time (sec.) of different algorithms on the 4s−dimensional data points by applying PCA are reported in Table IV.

### Table IV

| Methods | Average time (sec.) | 2 motions | median | std. | 3 motions | median | std. | All motions | median | std. |
|---------|---------------------|----------|--------|------|-----------|--------|------|-------------|--------|------|
| IDR-Z   | 9.13                | 0.30     | 1.24   |      | 1.20      | 0.56   | 3.27 | 2.48        | 0.56   | 5.88 |
| IDR-S   | 0.50                | 0.89     |        |      | 2.23      | 0.56   | 3.29 | 0.89        | 2.44   |      |
| SSC     | 2.98                | 1.66     | 5.13   |      | 5.29      | 1.46   | 7.35 | 2.48        | 5.88   |      |
| LRR     | 3.18                | 1.15     | 3.19   |      | 4.17      | 1.20   | 5.99 | 1.83        | 4.17   |      |
| LSR     | 0.03                | 0.56     | 2.18   |      | 2.77      | 0.89   | 5.10 | 1.09        | 3.53   |      |
| BDR     | 5.50                | 0.58     | 2.78   |      | 2.77      | 0     | 5.10 | 1.06        | 3.42   |      |
| BDR-Z   | 0.85                | 0.6      | 2.76   |      | 2.77      | 0     | 5.10 | 1.09        | 3.53   |      |
| MR-SSC  | 41.15               | 2.71     | 6.56   |      | 9.22      | 6.05   | 9.18 | 4.33        | 7.78   |      |
| MR-LRR  | 43.29               | 1.39     | 3.95   |      | 6.52      | 2.85   | 6.82 | 2.66        | 5.28   |      |
| JDSSC   | 16.29               | 12.51    | 11.45  | 10.54 | 24.09     | 25.06  | 11.62 | 15.13       | 14.48  | 11.8 |
| ADSSC   | 0.07                | 2.42     | 0      | 5.67 | 8.74      | 5.37   | 9.65 | 3.85        | 7.24   |      |

The best results of these algorithms are emphasized in bold.

of the results are reported in Tables III and IV. From the two tables, we could see that 1) IDR achieves the best results in these experiments; 2) BDR and LSR also achieve competitive results; 3) MR-LRR and MR-SSC do not conquer their corresponding classical methods SSC and LRR. This means the post-processing on the coefficient matrices in MR may not always enhance the performance of LRR and SSC; 5) JDSSC fails to achieve satisfying results.

Moreover, we also report the average computation time of each algorithm on the 155 motion sequences in Tables III and IV. Clearly, LSR and ADSSC are much more efficient than other algorithms. The average computation time of IDR is close to that of LRR. Hence the computation burden of IDR is acceptable.

Second, we analyze the experimental results of each algorithm in another way. For each algorithm, we present the percentage of motions with the obtained SEs are less than or equal to a given percentage of segmentation error in Fig. 5. We can see that the segmentation errors on all motions obtained by IDR are all less than 0.2. This indicates that IDR achieves good results on most of the sub-datasets.

Finally, we also test the sensitivity of IDR to the parameters on Hopkins 155 database. For a fixed pair of $(\gamma, \lambda)$, we compute the segmentation error for all 155 segmentation tasks, then the mean of 155 segmentation accuracies could be achieved. Then by changing the values of $\gamma$ and $\lambda$, we illustrate the performance of IDR against its parameters in Fig. 6.

Based on Fig. 6, we still could see that IDR is insensitive to its parameters in a large interval. And as we mentioned, errors are manually removed for each sequence, hence each sub-database in Hopkins 155 could be viewed as only slightly noise. Hence, we can see that IDR achieves good results even $(\gamma, \lambda)$ are relatively large.

D. Experiments on Two Face Image Data Sets

We now perform the experiments on two benchmark face image databases, i.e., ORL database [48] and AR database [49]. The brief information of the two databases is introduced as follows:

ORL database contains 400 face images (without noise) of 40 persons. Each individual has 10 different images. These
Fig. 6. The segmentation error against the variation of parameters of IDR on Hopkins 155 data set. Now the vertical axis in each figure denotes the subspace clustering error which is different from the experiments conducted on synthetic data sets. The two sub-figures in the left column show the segmentation errors obtained by $Z$ changed with parameters, and the two sub-figures in the right column record the segmentation errors obtained by $S$ changed with parameters. The first and second rows present the segmentation error of IDR on the original data sets and projected data sets respectively.

Fig. 7. Sample images from ORL database and AR database. Images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). In our experiments, each image is resized to $32 \times 32$ pixels.

AR database consists of over 4000 face images of 126 individuals. For each individual, 26 pictures were taken in two sessions (separated by two weeks) and each section contains 13 images. These images include a front view of faces with different expressions, illuminations, and occlusions. In our experiments, each image is resized to $50 \times 40$ pixels.

Moreover, the pixel value in each image belonging to the two databases lies in $[0,255]$. For efficient computation, we let each pixel value be divided by 255, so that the pixel value of each image fell into $[0,1]$. This will not change the distribution of the original data sets. Some sample images from ORL and AR databases are shown in Fig. 7.

We first randomly choose images of $q$ persons from the two databases. In ORL database, $q \in \{6, 12, 18, 24, 36\}$, and in AR database, $q \in \{4, 8, 12, 16, 20\}$. Then the performance of the evaluated methods is tested in these sub-databases. With the parameters varying, the highest clustering accuracy of each algorithm obtained on each sub-database is collected. These experiments run 10 trials, and the mean and standard variance of SAs obtained by each algorithm are reported in Tables V and VI respectively. Clearly, the two tables show that in most cases, IDR outperforms other algorithms on the two databases. Especially on AR database, IDR gets much better results than those of other evaluated algorithms.

We also compare the computation time of all the evaluated algorithms. For a face images database, on its sub-databases with a fixed $q$ (number of persons), we could compute the average computation time of each algorithm. Then the computation time of each algorithm changed with $q$ is illustrated in Fig. 8. Similar to the results obtained on Hopkins 155 databases, it could be seen that the computation time of IDR is acceptable. When $q$ is relatively small, the computation cost of IDR is close to that of LRR, when $q$ is relatively large, IDR is more efficient than LRR. However, JDSSC spends much more time than other algorithms. Moreover, the sensitivity verification of IDR to its parameters on the two databases is illustrated in Fig. 9. It still shows that IDR is stable and can get good results when $\gamma$ and $\lambda$ are relatively small.

E. Experiments on MNIST Data Set

MNIST database has 10 subjects, corresponding to 10 handwritten digits, namely ‘0’-‘9’. And each image is resized to $28 \times 28$ pixels. Some sample images from the database are illustrated in Fig. 10.

Then we followed the similar methodologies used in the above experiments. Here, we randomly chose images of $\{2, 4, 6, 8, 10\}$ digits from each subjects’ training data set to build sub-databases. We also run the experiments 10 trails and record the mean and standard variance of segmentation accuracies obtained by each algorithm in Table VII.

Form Table VII, we could find that IDR still dominates the other algorithms. Actually, IDR achieves much better results than those of other algorithms. In addition, we could see that the performance of other algorithms is close to each other.
Moreover, we plot the average computation time of each algorithm against the number of digits in Fig. 11(a) and show that the performances of IDR-Z and IDR-S changed with the values of parameters \( \gamma \) and \( \lambda \) in Fig. 11(b) and (c) respectively. For the visualization of IDR’s sensitivity, here we use 10 sub-databases with 10 digits. From Fig. 11, we could conclude that 1) the computation time of IDR is much less than MR and JDSSC; 2) the computation costs of MR-SSC and MR-LRR are much larger than those of other algorithms; 3) IDR could achieve better results with small \( \gamma \) and \( \lambda \). This coincides with the experiments provided above.

Based on these experiments, we could make the following summarizations: 1) IDR could get satisfying subspace clustering results on different kinds of databases; 2) Compared with the closely related algorithms, such as MR and DSSC, the computation cost of IDR is acceptable; 3) IDR is insensitive to its two parameters. However, small parameters could make IDR achieve better results.

### F. Experiments on More Data Sets

In order to evaluate the performance of IDR in more depth, more data sets including the whole ORL database (W-ORL), the whole AR database (W-AR), Umist, COIL20, and COIL40 are applied. Umist has 480 images with varied poses from 20 individuals. COIL20 dataset \[50\] is composed of 1440 images for 20 different objects. The images of each object are taken 5° apart as the object rotates on a turn table and each object has 72 images. COIL40 is similar to COIL20 which contains 2880 images for 40 different objects. The detailed information of these data sets is summarized in Table VIII. Meanwhile, some modern subspace clustering algorithms including LRSC \[5\],

---

**Table V**

| Methods | 4 | 8 | 12 | 16 | 20 |
|---------|---|---|----|----|----|
| IDR-Z   | 90.30 ± 9.70 | 85.66 ± 2.49 | 82.95 ± 2.23 | 80.91 ± 2.27 | 78.92 ± 2.30 |
| IDR-S   | 94.18 ± 1.67 | 94.04 ± 1.79 | 93.08 ± 1.22 | 90.54 ± 1.53 |
| SSC     | 85.29 ± 3.55 | 80.76 ± 2.40 | 73.56 ± 2.20 | 65.41 ± 2.30 |
| LRR     | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| BDR-Z   | 86.35 ± 2.02 | 85.76 ± 2.12 | 82.95 ± 2.23 | 80.91 ± 2.27 |
| BDR-R   | 85.29 ± 3.55 | 80.76 ± 2.40 | 73.56 ± 2.20 | 65.41 ± 2.30 |
| MR-SSC  | 85.10 ± 2.02 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| MR-LRR  | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| JDSSC   | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| ADSSC   | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |

**Table VI**

| Methods | 4 | 8 | 12 | 16 | 20 |
|---------|---|---|----|----|----|
| IDR-Z   | 90.30 ± 9.70 | 85.66 ± 2.49 | 82.95 ± 2.23 | 80.91 ± 2.27 | 78.92 ± 2.30 |
| IDR-S   | 94.18 ± 1.67 | 94.04 ± 1.79 | 93.08 ± 1.22 | 90.54 ± 1.53 |
| SSC     | 85.29 ± 3.55 | 80.76 ± 2.40 | 73.56 ± 2.20 | 65.41 ± 2.30 |
| LRR     | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| BDR-Z   | 86.35 ± 2.02 | 85.76 ± 2.12 | 82.95 ± 2.23 | 80.91 ± 2.27 |
| BDR-R   | 85.29 ± 3.55 | 80.76 ± 2.40 | 73.56 ± 2.20 | 65.41 ± 2.30 |
| MR-SSC  | 85.10 ± 2.02 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| MR-LRR  | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| JDSSC   | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |
| ADSSC   | 89.12 ± 2.01 | 85.45 ± 2.80 | 78.98 ± 2.30 | 72.45 ± 2.30 |

**Table VII**

| Methods | 2 | 4 | 6 | 8 | 10 |
|---------|---|---|---|---|----|
| IDR-Z   | 98.45 ± 3.59 | 96.44 ± 3.56 | 94.55 ± 3.53 | 92.66 ± 3.52 | 90.77 ± 3.51 |
| IDR-S   | 95.35 ± 3.54 | 93.56 ± 3.53 | 91.57 ± 3.52 | 89.68 ± 3.51 | 87.79 ± 3.50 |
| SSC     | 97.05 ± 2.34 | 95.54 ± 2.33 | 93.05 ± 2.32 | 90.54 ± 2.31 | 88.05 ± 2.30 |
| LRR     | 96.80 ± 5.15 | 95.26 ± 5.14 | 93.72 ± 5.13 | 91.28 ± 5.12 | 88.84 ± 5.11 |
| BDR-Z   | 95.16 ± 5.29 | 93.78 ± 5.28 | 92.40 ± 5.27 | 90.92 ± 5.26 | 89.44 ± 5.25 |
| BDR-R   | 93.20 ± 5.33 | 91.82 ± 5.32 | 90.44 ± 5.31 | 89.06 ± 5.30 | 87.68 ± 5.29 |
| MR-SSC  | 97.35 ± 2.78 | 96.05 ± 2.77 | 94.75 ± 2.76 | 92.45 ± 2.75 | 90.15 ± 2.74 |
| MR-LRR  | 96.25 ± 2.76 | 94.95 ± 2.75 | 92.65 ± 2.74 | 90.45 ± 2.73 | 88.15 ± 2.72 |
| JDSSC   | 97.75 ± 2.78 | 96.45 ± 2.77 | 95.15 ± 2.76 | 93.85 ± 2.75 | 91.55 ± 2.74 |
| ADSSC   | 97.75 ± 2.78 | 96.45 ± 2.77 | 95.15 ± 2.76 | 93.85 ± 2.75 | 91.55 ± 2.74 |

The best results (mean) of the algorithms are emphasized in bold.
Fig. 9. The segmentation accuracies of IDR against the variation of parameters on ORL and AR databases. Here, the sub-databases from ORL database contains 36 persons’ images and the sub-databases from AR database contains 20 persons’ images are used.

Fig. 10. Sample images from MNIST database.

Fig. 11. (a) The average computation time (sec.) of each algorithm versus the number of digits. (b) The segmentation accuracies of IDR-Z against the variation of parameters. (c) The segmentation accuracies of IDR-S against the variation of parameters.

SSCOMP [51], SSRC [52], FLSR [53] and GCSC [54] are used for comparison. Similar to the previous experiments, we record the best segmentation accuracies of the evaluated algorithm obtained on the data sets.

We summarize the experimental results in Table IX, where the best segmentation accuracies are emphasized in bold. From Table IX, we can see that compared with some modern methods, IDR still achieves the best results on these databases. Especially on Umist, COIL20, and COIL40 databases, IDR outperforms other algorithms by a large margin. This means that IDR could faithfully reveal the subspace structures of the three data sets.

G. Convergence Analysis

We test the convergence of optimization algorithm for solving IDR problem on all the real-world databases. When IDR is performed on a real-world database, the value of the optimization function (i.e., the Lagrangian function of IDR, Eq. (10)) is recorded in each iteration. And we fix the total number of iterations to be 300 and the number of iterations that satisfies the convergence condition is also recorded. Then the values of the optimization function versus the numbers of iterations on all the databases are plotted in Fig. 12. The red vertical line in a subfigure indicates the number of iterations when the convergence condition of Algorithm 1 is satisfied on a database.

From Fig. 12, we can see that the value of the optimization function constantly decreases and Algorithm 1 can achieve a stable solution after about 250 iterations on all the databases.
be a relatively clean data, we let $\gamma \geq (8) - (8)$, seeks a reconstruction coefficient matrix to show the differences between the coefficient matrices obtained by without idempotent constraint, doubly stochastic constraints, and trace constraint respectively. The last column show the two coefficient matrices obtained by IDR with all constraints.

**H. Ablation Study**

In this subsection, we discuss the necessities of the constraints used in IDR (i.e., (8)). In (8), the constraints imposed on the coefficient matrix could be divided into three groups: idempotent constraint ($\| S - S^2 \| = 0$), doubly stochastic constraints ($I_n^T S = I_n$, $S = S^T$, $S \geq 0$) and trace constraint ($Tr(S) = k$). First, if we abandon the idempotent constraint, IDR actually degenerates to JDSSC, and (8) seeks a reconstruction coefficient matrix satisfying doubly stochastic constraint and trace constraint. As we described in Section III, a reconstruction coefficient matrix satisfying doubly stochastic constraint may not be a normalized membership matrix that can not guarantee the dense connectedness in each block. Second, doubly stochastic constraints can prohibit solutions with all-zero rows or columns [36]. Hence, if the doubly stochastic constraint is not applied, we can not obtain an ideal reconstruction coefficient matrix. Third, the trace constraint can prevent the obtained reconstruction coefficient matrix to be an identity matrix.

We use a simple example to confirm the above analysis and demonstrate the role of different constraints. The data set has been used in Fig. 1 to show the differences between the coefficient matrices obtained by BDR and IDR (See Fig. 1). Then the comparative results are shown in Fig. 13. The first, second, and third columns of Fig. 13 show the two reconstruction coefficient matrices obtained by IDR without idempotent constraint, doubly stochastic constraints and trace constraint. The reconstruction coefficient matrices obtained by IDR with all constraints are shown in the last column. In all the experiments, $\lambda = 0.2$. To get the results of IDR without idempotent constraint, we let $\gamma = 0$. Because the data is clean, we let $\gamma$ be a relatively large value, $\gamma = 20$ for the rest experiments. The segmentation accuracy obtained by using the corresponding coefficient matrix is reported under each subfigure. Obviously, we can see that the corresponding reconstruction coefficient show the properties which we described above, and IDR with all constraint can get ideal coefficient matrices and achieve the best results.

**VI. CONCLUSION**

Spectral-type subspace clustering algorithms show their excellent performances in subspace clustering tasks. The classical spectral-type methods hope to use different norms of reconstruction coefficient matrices to seek coefficient matrices satisfying intra-subspace connectivity and inter-subspace sparse. In this paper, we design an idempotent constraint for reconstruction coefficient matrices based on the proposition that reconstruction coefficient vectors also obey the self-expressiveness property. By integrating double stochastic constraints, we present an idempotent representation (IDR) method for subspace clustering. Subspace clustering experiments conducted on both synthetic data sets and real-world data sets verify the effectiveness and efficiency of IDR.

**ACKNOWLEDGMENT**

The authors thank anonymous reviewers for their constructive comments on this paper.

**REFERENCES**

[1] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: A review,” SIGKDD Explor. Newsl., vol. 6, no. 1, pp. 90–105, 2004.

[2] R. Vidal, “Subspace clustering,” IEEE Signal Process. Mag., vol. 28, no. 2, pp. 52–68, Mar. 2011.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, L. M. Haas and A. Tiwary, Eds., Seattle, WA, USA, Jun. 2–4, 1998, pp. 94–105.

[4] H. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” ACM Trans. Knowl. Discov. Data, vol. 3, no. 1, pp. 1:1–1:58, 2009.

[5] R. Vidal and F. Favoro, “Low rank subspace clustering (LrSC),” Pattern Recognit. Lett., vol. 43, pp. 47–61, 2014.

[6] Y. Chu, Y. Chen, D. Yang, and M. Chen, “Reducing redundancy in subspace clustering,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 10, pp. 1432–1446, Oct. 2009.

[7] S. Yi, Z. He, Y. Cheung, and W. Chen, “Unified sparse subspace learning via self-contained regression,” IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 10, pp. 2537–2550, Oct. 2018.

[8] Y. Ma, H. Derksen, W. Hong, and J. Wright, “Segmentation of multivariate mixed data via lossy data coding and compression,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pp. 1546–1562, Sep. 2007.

[9] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” IEEE Trans Pattern Anal Mach Intell, vol. 35, no. 11, pp. 2765–81, Nov. 2013.

[10] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 171–184, Jan. 2013.

[11] K. Kanatani, “Motion segmentation by subspace separation and model selection,” in Proc. IEEE 8th Int. Conf. Comput. Vis., Vancouver, BC, Canada, Jul. 7–14, 2001, pp. 586–591.

[12] Y. Ma, A. Y. Yang, H. Derksen, and R. M. Fossum, “Estimation of subspace arrangements with applications in modeling and segmenting mixed data,” SIAM Rev., vol. 50, no. 3, pp. 413–458, 2008.

[13] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug. 2000.

[14] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, 2009, pp. 2790–2797.

[15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 2, pp. 220–237, Feb. 2009.

[16] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in Proc. 27th Int. Conf. Mach. Learn., Haifa, Israel, Jun. 21–24, 2010, pp. 663–670.

[17] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, “Robust and efficient subspace segmentation via least squares regression,” in Proc. 12th Eur. Conf. Comput. Vis., Florence, Italy, 2012, pp. 347–360.
[18] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, “Subspace clustering by block diagonal representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 2, pp. 487–501, Feb. 2019.

[19] J. Xu, K. Xu, K. Chen, and J. Ruan, “Reweighted sparse subspace clustering,” Comput. Vis. Image Understand., vol. 138, pp. 25–37, 2015.

[20] H. Zhang, J. Yang, F. Shang, C. Gong, and Z. Zhang, “LRR for subspace segmentation via tractable Schatten-p norm minimization and factorization,” IEEE Trans. Cybern., vol. 49, no. 5, pp. 1722–1734, May 2019.

[21] J. Xu et al., “Scaled simplex representation for subspace clustering,” IEEE Trans. Cybern., vol. 51, no. 3, pp. 1493–1505, Mar. 2021.

[22] C.-G. Li, C. You, and R. Vidal, “Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework,” IEEE Trans. Image Process., vol. 26, no. 6, pp. 2988–3001, Jun. 2017.

[23] Y. Panagakis and C. Kotropoulos, “Elliptic subspace clustering applied to pop/rock music structure analysis,” Pattern Recognit. Lett., vol. 38, pp. 46–53, 2014.

[24] C. You, C. Li, D. P. Robinson, and R. Vidal, “Oracle based active set algorithm for scalable elastic net subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 3928–3937.

[25] L. Zhuang et al., “Constructing a nonnegative low-rank and sparse graph with data-adaptive features,” IEEE Trans. Image Process., vol. 24, no. 11, pp. 3717–3728, Nov. 2015.

[26] K. Tang, R. Liu, and J. Zhang, “Structure-constrained low-rank representation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 12, pp. 2167–2179, Dec. 2014.

[27] X. Lu, Y. Wang, and Y. Yuan, “Graph-regularized low-rank representation for degrading of hyperspectral images,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 7, pp. 4009–4018, Jul. 2013.

[28] K. Tang, D. B. Dunson, Z. Su, R. Liu, J. Zhang, and J. Dong, “Subspace segmentation by dense block and sparse representation,” IEEE Trans. Image Process., vol. 75, pp. 66–76, 2016.

[29] L. Wei, F. J. H. Liu, R. Zhou, C. Z. and X. Zhang, “Subspace clustering via structured sparse relation representation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 9, pp. 4610–4623, Sep. 2022.

[30] Y. Su, G. Wang, and L. Zhang, “Sparse subspace clustering via low-rank structure propagation,” Pattern Recognit., vol. 95, pp. 261–271, 2019.

[31] C. Mathieu and W. Schudy, “Correlation clustering with noisy input,” in Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms, M. Charikar, Ed., Austin, Texas, USA, Jan. 17–19, 2010, pp. 712–720.

[32] C. Swamy, “Correlation clustering: Maximizing agreements via semidefinite programming,” in Proc. 15th Annu. ACM-SIAM Symp. Discrete Algorithms, J. I. Munro, Ed., New Orleans, LA, USA, Jan. 11–14, 2004, pp. 526–527.

[33] M. Lee, J. Lee, H. Lee, and N. Kwak, “Membership representation for detecting block-diagonal structure in low-rank or sparse subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Boston, MA, USA, Jun. 7–12, 2015, pp. 1648–1656.

[34] Z. Lin, M. Chen, and Y. Ma, “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices,” 2010, arXiv:1009.5055.

[35] R. Zass and A. Shashua, “A unifying approach to hard and probabilistic clustering,” in Proc. IEEE 10th Int. Conf. Comput. Vis., Beijing, China, Oct. 17–20, 2005, pp. 294–301.

[36] R. Zass and A. Shashua, “Doubly stochastic normalization for spectral clustering,” in Proc. Adv. Neural Inf. Process. Syst., B. Schölkopf, J. C. Platt, and T. Hofmann, Eds., Vancouver, BC, Canada, MIT Press, Dec. 4–7, 2006, pp. 1569–1576.

[37] C. Lu, J. Feng, Z. Lin, and S. Yan, “Correlation adaptive subspace segmentation by trace Laaso,” in Proc. IEEE Int. Conf. Comput. Vis., Sydney, Australia, Dec. 1–8, 2013, pp. 1345–1352.

[38] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.

[39] J. Mairal, “Optimization with first-order surrogate functions,” in Proc. 30th Int. Conf. Mach. Learn., ser. JMLR Workshop and Conference Proceedings, Atlanta, GA, USA, Jun. 16–21, 2013, pp. 783–790.

[40] D. X. Liu, T. M. Ma, and X. Yuan, “A note on the alternating direction method of multipliers,” J. Optim. Theory Appl., vol. 155, no. 1, pp. 227–238, 2012.

[41] D. Lim, R. Vidal, and B. D. Haeffele, “Doubly stochastic subspace clustering,” 2020, arXiv:2011.14859.

[42] S. Ma, “Alternating proximal gradient method for convex minimization,” J. Sci. Comput., vol. 68, no. 2, pp. 546–572, 2016.

[43] L. Wei, X. Wang, A. Wu, R. Zhou, and C. Zhu, “Robust subspace segmentation by self-representation constrained low-rank representation,” Neural Process. Lett., vol. 48, no. 3, pp. 1671–1691, 2018.

[44] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, “Is an affine constraint needed for affine subspace clustering?” in Proc. Int. Conf. Comput. Vis., 2019, pp. 9914–9923.

[45] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, “Rank-constrained spectral clustering with flexible embedding,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6073–6082, Dec. 2018.

[46] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, “One-step multiview spectral clustering,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 10, pp. 2022–2034, Oct. 2019.

[47] R. Tron and R. Vidal, “A benchmark for the comparison of 3-D motion segmentation algorithms,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Minneapolis, MN, USA, Jun. 18–23, 2007.

[48] F. Samaria and A. Harter, “Parameterisation of a stochastic model for human face identification,” in Proc. IEEE 2nd Work. Appl. Comput. Vis., Sarasota, FL, USA, Dec. 5–7, 1994, pp. 138–142.

[49] A. Martinez and R. Benavente, “The AR face database,” CVC, Tech. Rep. 24, 1998. [Online]. Available: https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[50] C. You, D. P. Robinson, and R. Vidal, “Scalable sparse subspace clustering by orthogonal matching pursuit,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 3918–3927.

[51] J. Xu et al., “Scaled simplex representation for subspace clustering,” IEEE Trans. Cybern., vol. 51, no. 3, pp. 1493–1505, Mar. 2021.

[52] Z. Ma, M. Zhang, G. Luo, L. Tian, and W. Chen, “Towards clustering-friendly representations: Subspace clustering via graph filtering,” in Proc. 28th ACM Int. Conf. Multimedia, Seattle, WA, USA, Oct. 12–16, 2020, pp. 3081–3089.

[53] Y. Cai, Z. Zhang, Z. Cai, X. Liu, X. Jiang, and Q. Yan, “Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 5, pp. 4191–4202, May 2021.
Changming Zhu received the PhD degree in computer science from the East China University of Science and Technology, Shanghai, China, in 2015. He is currently an associate professor with the College of Computer Science and Engineering, Shanghai Maritime University, Shanghai, China. His current research interests include pattern recognition and machine learning.

Jin Liu (Member, IEEE) received the PhD degree from Washington State University, Pullman, USA in 2007. He is currently a professor with the College of Information Engineering, Shanghai Maritime University, Shanghai, China. His research interests include multi-modal learning, the Internet of Things, and related information systems.