Simultaneous Subspace Clustering and Cluster Number Estimating based on Triplet Relationship

Jie Liang, Jufeng Yang*, Ming-Ming Cheng, Paul L. Rosin, Liang Wang

***Abstract*—In this paper we propose a unified framework to discover the number of clusters and group the data points into different clusters using subspace clustering simultaneously. Real data distributed in a high-dimensional space can be disentangled into a union of low-dimensional subspaces, which can benefit various applications. To explore such intrinsic structure, state-of-the-art subspace clustering approaches often optimize a self-representation problem among all samples, to construct a pair-wise affinity graph for spectral clustering. However, a graph with pairwise similarities lacks robustness for segmentation, especially for samples which lie on the intersection of two subspaces. To address this problem, we design a hyper-correlation based data structure termed as the *triplet relationship*, which reveals high relevance and local compactness among three samples. The triplet relationship can be derived from the self-representation matrix, and be utilized to iteratively assign the data points to clusters. Based on the triplet relationship, we propose a unified optimizing scheme to automatically calculate clustering assignments. Specifically, we optimize a model selection reward and a fusion reward by simultaneously maximizing the similarity of triplets from different clusters while minimizing the correlation of triplets from same cluster. The proposed algorithm also automatically reveals the number of clusters and fuses groups to avoid over-segmentation. Extensive experimental results on both synthetic and real-world datasets validate the effectiveness and robustness of the proposed method.**

***Index Terms*—Subspace clustering, triplet relationship, estimating the number of clusters, hyper-graph clustering**

I. Introduction

With the ability to disentangle latent structure of data in an unsupervised manner [2]–[4], subspace clustering is regarded as an important technique in the data mining community and for various computer vision applications [5]–[8]. Traditional subspace clustering methods approximate a set of high-dimensional data samples into a union of lower-dimensional linear subspaces [9], where each subspace usually contains a subset of the samples.

In past two decades, spectral-based subspace clustering methods have achieved state-of-the-art performance, taking a two-step framework as follows. First, by optimizing a self-representation problem [4], [10], a similarity matrix (and also a similarity graph) is constructed to depict the relationship (or connection) among samples. Second, spectral clustering [11], [12] is employed for calculating the final assignment based on eigen-decomposition of the affinity graph. Note that in practice, both the number of subspaces and their dimensionalities are always unknown [4], [13]. Hence, the goals of subspace clustering include finding the appropriate number of clusters and grouping data points into them [14], [15].

Nevertheless, it is challenging to estimate the number of clusters in a unified optimization framework, since the definition of clusters is subjective, especially in **a** high-dimensional ambient space [14]. In addition, for samples which are of different clusters but near the intersection of two subspaces, they may be even closer than samples from same cluster. This may lead to a wrong estimation with several redundant clusters, namely over-segmentation problem. Therefore, most of the spectral-based subspace clustering algorithms depend on a manually given and fixed number of clusters, which cannot be generalized for multiple applications [4].

Most clustering schemes group similar patterns into the same cluster by jointly minimizing the inter-cluster similarity and the intra-cluster dissimilarity [16]. Considering the complexity of the high-dimensional ambient data space, an effective way for estimating the number of clusters is to first map the raw samples into an intrinsic correlation space, namely similarity matrix, followed by an iterative optimization according to the local and global similarity relationships derived from the projection. Elhamifar *et al*. [17] propose that the permuted similarity matrix can be block-diagonal, where the number of blocks is identical to the number of clusters. Moreover, Peng *et al*. [3] verify the intra-subspace projection dominance (IPD) of such a similarity matrix, which can be applied to self-representation optimizations with various kinds of regularizations. The IPD theory says that for two arbitrary samples from the same subspace and one from another, the generated similarities between the former samples are always larger than between latter ones in a noise-free system.

Accordingly, considering an affinity graph derived from the similarity matrix [18], [19], where the vertices denote data samples and the edge weights denote similarities, an automatic sub-graph segmentation can be greedily conducted via the following two steps inspired by the density based algorithms [20]: 1) constructing a proper number of initialized cluster centers by minimizing the weighted sum of all inter-cluster connections and maximizing the intra-cluster ones; 2) merging the remaining samples to an existing cluster by

Manuscript received September 03, 2018; revised February 12, 2019; accepted February 19, 2019.
J. Liang, J. Yang, and M.-M. Cheng are with College of Computer Science, Nankai University, Tianjin 300350, China (E-mail: liang27jie@163.com; yangjufeng@nankai.edu.cn; cmm@nankai.edu.cn).
P. L. Rosin is with School of Computer Science and Informatics, Cardiff University, Wales, UK. (E-mail: Paul.Rosin@cs.cf.ac.uk).
L. Wang is with the National Laboratory of Pattern Recognition, CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (E-mail: wangliang@nlpr.ia.ac.cn).
A preliminary version of this work appeared at AAAI [1].
∗ : Corresponding author.

maximizing the weighted connections between the sample and the cluster.

Yet, there are also difficulties for these greedy iterative schemes. Since the ambient space can also be very dense [21], any two points which are close by evaluating the pairwise distance may not belong to the same subspace, especially for samples near the intersection of two subspaces. Consequently, a hypergraph where each edge can be connected to more than two samples [22], [23] is proposed to solve the problem in traditional pairwise graphs. In this paper, we further introduce a novel data structure termed as the *triplet relationship*, to explore the local geometry structure in projected space with hyper-correlations. Each triplet consists of three points and their correlations, which are considered as a meta-element for clustering. We require that all correlations are large enough, which indicates that the three points are strongly connected according to the IPD property [3].

In contrast to evaluating similarities using pairwise distances, the proposed triplet relationship demonstrates favorable performance due to the following two reasons. On one hand, it is more robust when partitioning the samples near the intersection of two subspaces since the mutual relevance among multiple samples can provide complementary information when calculating the local segmentation. On the other hand, the triplet evokes a hyper-similarity by efficiently counting the frequency of intra-triplet samples, which enables a greedy way to calculate the assignments.

Based on the newly-defined triplet relationship, in this paper, we further propose a unified framework termed as the *autoSC* to jointly estimate the number of clusters and group the samples by exploring the local density derived from triplet relationships. Specifically, we first calculate the self-representation for each sample via an off-the-shelf optimization scheme, followed by extracting the triplet relationship for all samples. Then, we greedily initialize a proper number of clusters via optimizing a new model selection reward, which is achieved by maximizing inter-cluster dissimilarity among triplets. Finally, we merge each of the remaining samples into an existing cluster to maximize the intra-cluster similarity by optimizing a new fusion reward. We also fuse groups to avoid over-segmentation.

The main contributions of this paper are summarized as follows:

- First, we define a hyper-dimensional triplet relationship which ensures a high relevance and density among three samples to reflect their local similarity. We also validate the effectiveness of triplets and distinguish them against the standard pairwise relation.
- Second, we design a unified framework, *i.e.*, autoSC, based on the intrinsic geometrical structures depicted by our triplet relationships. The proposed autoSC can be used for simultaneous estimating the number of clusters and subspace clustering in a greedy way.

Extensive experiments on benchmark datasets indicate that our autoSC outperforms the state-of-the-art methods in both effectiveness and efficiency.

This paper is an extended version of our earlier conference paper [1], to which we **have enriched** the contributions in the following five aspects: (1) We have added detailed analysis of the proposed algorithm to distinguish it from comparative methods, for example, we have added analysis and experimental validation on the computational complexity. (2) We have provided a visualized illustration of the proposed autoSC for clear presentation. (3) We have designed a relaxation termed as the neighboring based autoSC (autoSC-N), which directly calculates the neighborhood relationship from raw data space and is more efficient than autoSC. (4) We have conducted experiments on evaluating the influence of the parameter m (number of preserved neighbors for each sample). (5) We have experimentally evaluated our method on a real-world application, *i.e.*, motion segmentation, which also demonstrates the benefits of the proposed method.

II. Related Work

Automatically approximating samples in high-dimensional ambient space by a union of low-dimensional linear subspaces is considered to be a crucial task in computer vision [24]–[32]. In this section, we review the related contributions in the following three aspects, *i.e.*, self-representation calculation, estimating the number of clusters and hyper-graph clustering.

A. Calculating Self-Representation

To separate a collection of data samples which are drawn from a high-dimensional space according to the latent low-dimensional structure, traditional self-expressiveness based subspace clustering method calculates a linear representation for each sample using the remaining samples as a basis set or a dictionary [33], [34]. Subspace clustering assumes that the set of data samples are drawn from a union of multiple subspaces, which can best fit the ambient space [17]. There are numerous real applications satisfying this assumption with varying degrees of exactness [35], *e.g.*, face recognition, motion segmentation, *etc*.

By solving an optimization problem with self-representation loss and regularizations, subspace clustering [36], [37] calculates a similarity matrix where each entry indicates the relevance between two samples. Different regularizing schemes with various norms of the similarity matrix, *e.g.*, ℓ_1 [9], ℓ_2 [38], elastic net [39] or nuclear norm [40], can explore different intrinsic properties of the neighborhood space. There are mainly three types of the regularization terms, including sparse-oriented, densely-connected and mixed norms.

Algorithms based on sparse-type norms [10], [41], *e.g.*, ℓ_0 and ℓ_1 norms, eliminate most of the non-zero values in the similarity matrix to ensure that there are no connections between samples from different clusters. Elhamifar and Vidal [9] propose the sparse representation based on ℓ_1 norm optimization. The obtained similarity matrix recovers a sparse subspace representation but may not satisfy the graph connectivity if the dimension of the subspace is greater than three [18]. In addition, the ℓ_0 based subspace clustering methods aim to compute a sparse and subspace-preserving representation for each data sample. Yang *et al*. [10] present a sparse clustering method with a regularizer based on the ℓ_0 norm by using the proximal gradient descent method. Numerous alternative methods have been proposed for ℓ_0 minimization while avoiding non-convex

problems, *e.g.*, orthogonal matching pursuit [42] and nearest subspace neighbor [43]. The scalable sparse subspace clustering by orthogonal matching pursuit (SSC-OMP) method [33] compares elements in each column of the dot product matrix to determine which positions of the similarity matrix should be non-zero. However, this general pairwise relationship does not reflect the sample correlation well, especially for data pairs in the intersection of two subspaces [44].

In contrast, dense connection based methods, such as smooth representation [38] with ℓ_2 norm and low rank representation with nuclear norm based methods [45], [46], propose to preserve many non-zero values in the similarity matrix to ensure the connectivity among intra-cluster samples [47]–[49]. For these densely connected frameworks [50], [51], the similarity matrix is interpreted as a projected representation of raw samples. Each column of the matrix is considered as the self-representation of a sample, and should be dense for mapping invariance (also termed as the grouping effect [38], [52]). Low-rank clustering methods [53], [54] solve a nuclear norm based optimization problem with the aim of generating a block diagonal solution with dense connections. However, the nuclear norm does not enforce subset selection well when noise exists, and the self-representation is too dense to be an efficient feature.

Neither a sparse nor dense similarity matrix reveals a comprehensive correlation structure among samples due to their conflicted nature [55]–[57]. Consequently, to achieve trade-off between sparsity and the grouping effect, numerous mixed norms, *e.g.*, trace Lasso [58] and elastic net [39], have been integrated into the optimization function. Nevertheless, the structure of the data correlations depends on the data matrix, and the mixed norm is not effective for structure selection. Therefore, this method does not perform consistently well on different applications.

Recently, many frameworks that incorporate various constraints into the optimization function have been proposed to detect different intrinsic properties of the subspace [40], [59], [60]. For instance, to handle sequential samples, Guo *et al.* [61] explore the neighboring relationship by incorporating a new penalty, *i.e.*, a lower triangular matrix with -1 on the diagonal and 1 on the second diagonal, to force consecutive columns in the similarity matrix to be closer. In this paper, based on the intrinsic neighboring relevance and geometrical structures depicted in the similarity matrix, we calculate triplet relationships to form a hyper-correlation constraint of the clustering system. We validate the robustness of the proposed triplet relationship on top of different similarity matrices with various intrinsic properties.

B. Estimating the Number of Clusters

Most of the real applications in computer vision require estimating the number of clusters, according to the latent distribution of data samples [9]. To solve this problem, three main techniques exists: singular-based Laplacian matrix decomposition, density-based greedy assignment and hyper-graph based segmentation.

Singular-based Laplacian matrix decomposition is common in subspace clustering [62] and spectral clustering [63] due to the availability of the similarity matrix. Liu *et al.* [64] propose a heuristic estimator inspired by the block-diagonal structure of the similarity matrix [48]. Specifically, they estimate the number of clusters by counting the small singular values of a normalized Laplacian matrix which should be smaller than a given cut-off threshold. These singular based methods [9], [65] are dependent on a large gap between singular values, which is limited to applications in which the subspaces are sparsely distributed in the ambient space. Meanwhile, the matrix decomposition process is time-consuming when extended to large scale problems. Recently, Li *et al.* propose SCAMS [35], [66] which estimates the number of clusters by minimizing the rank of a binary relationship matrix encoding the pairwise relevance among all data samples. Simultaneously, they incorporate a penalty term on the clustering cost by minimizing the Frobenius inner product of the similarity matrix and binary relationship matrix.

Density based methods [67] greedily discover both the optimal number of clusters and the assignments of data to the clusters according to the local and global densities which are calculated by the pairwise distances in ambient space. Rodriguez *et al.* [20] automatically cluster samples based on the assumption that each cluster center is characterized by a higher density in the weight space than all its neighbors, while different centers should be far apart enough to avoid redundancy. Specifically, for each sample, its Euclidean-based local density and the distance to any points with higher densities are iteratively calculated and updated. In each iteration, the algorithm finds a trade-off between the density of cluster centers and the inter-cluster distance to update the assignments. Wang *et al.* [13] employ the Bayesian nonparametric method based on a Dirichlet process, and propose DP-space, which exploits a trade-off between data fitness and model complexity. DP-space is more tolerate to noisy and outlier values than the alternative algebraic and geometric solutions. Recently, correlation clustering (CC) [68] first constructs an undirected graph with positive and negative edge weights, followed by minimizing the sum of cut weights during the segmenting process. Sequentially, the clustering assignments can be optimized with a greedy scheme. Nevertheless, most of these density based algorithms are limited to pairwise correlation when evaluating the similarity of data samples, which is not robust for densely distributed subspaces.

C. Hyper-graph Clustering

To tackle the limitations of the pairwise relation based methods, the hyper-graph relation [69]–[71] is proposed and the related literature follows two different directions. Some transform the hyper-correlation into a simpler pairwise graph [22], [72], followed by a standard graph clustering method, *e.g.*, normalized cut [12], to calculate the assignments. In addition, other methods [14], [45] explore a generalized way of extending the pairwise graph to the hyper-graph or hyper-dimensional tensor analysis. For instance, Li *et al.* [14] propose a tensor affinity variant of SCAMS, *i.e.*, SCAMSTA, which exploits a higher order mathematical structure by providing multiple groups of nodes in the binary matrix derived from an outer product operation on multiple indicator vectors. However,

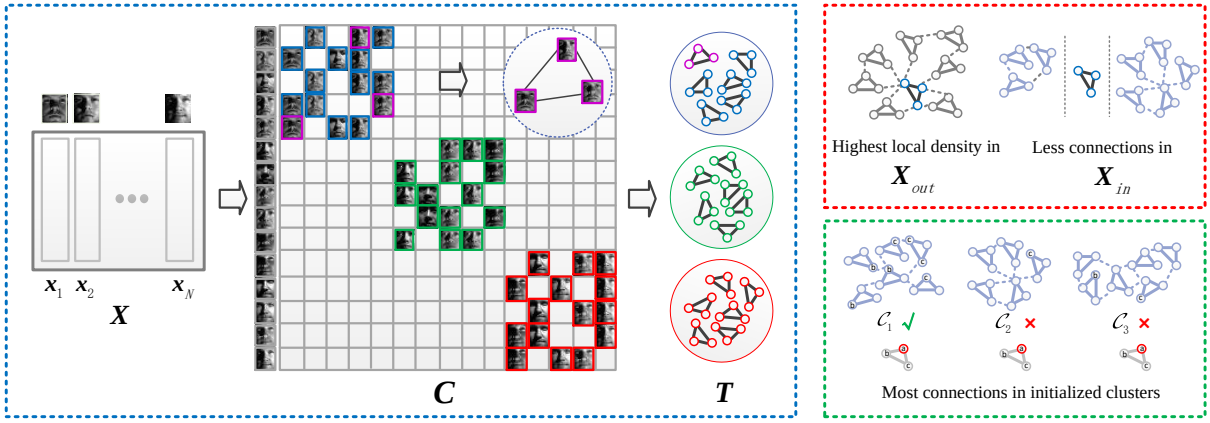


Fig. 1. Overview of the proposed autoSC. The algorithm is composed of three steps, *i.e.*, calculating triplet relationships $\boldsymbol{T}$ (blue dashed box), estimating the number of clusters via model selection reward (black) and finishing the clustering assignment via fusion reward (green). Given similarity matrix $\boldsymbol{C}$ derived from self-representation schemes, we illustrate an example of the triplet which is composed of the three samples shown with magenta frames. These samples induce a high local correlation and should be grouped into the same cluster. By optimizing the fusion reward, sample 'a' is assigned into $\mathcal{C}_1$ since $\mathcal{C}_1$ has most connections to the triplet which involves 'a'.

estimating the number of clusters from the rank of affinity matrix only works well for the ideal case, and can hardly be extended to complex applications since the noise can have a significant impact on the rank of affinity matrix.

In this paper, we estimate the number of clusters by initializing the cluster centers with maximum inter-cluster dissimilarities and also maximum local densities. We calculate the initialization according to the local correlations reflected by the proposed triplet relationships, where each of them depicts a hyper-dimensional similarity among three samples and easily-evaluated relevances to other triplets. Both theoretical analysis on triplets as well as the experimental results demonstrate the effectiveness of the proposed method.

III. Methodology

A. Preliminary

The main notations in the manuscript and the corresponding descriptions are shown in Table I. Given a set of N data samples $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$ lying in K subspaces $\{S_i\}_{i=1}^K$ where D denotes the dimensionality of each sample, spectral-based subspace clustering usually takes a two-step approach to calculate the clustering assignment. First, it learns a self-representation for each sample to disentangle the subspace structure. The algorithm then employs spectral clustering [12] on the learned similarity graph derived from $\boldsymbol{C}$ for final assignments. Note that in practice, both the number of subspaces K and their dimensions $\{d_j\}_{j=1}^K$ are always unknown [4], [13]. Hence, the goals of subspace clustering include finding the appropriate K and assigning data points into K clusters [14], [17].

In this paper, inspired by the block-diagonal structure of the similarity matrix [73], we propose to simultaneously estimate the number of clusters and assign samples into each cluster in a greedy manner. We design a novel meta-sample that we call a *triplet relationship*, followed by optimizing both a model selection reward and a fusion reward for clustering.

B. Learning the Self-Representation

To explore the neighboring relationship in the ambient space $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$, typical subspace clustering methods first optimize a linear representation of each data sample using the remaining dataset as a dictionary. Specifically, spectral-based subspace clustering calculates a similarity matrix $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ by solving a self-representation optimization problem as follows:

$$\min_{\boldsymbol{C}} \; L(\tilde{\boldsymbol{X}}\boldsymbol{C}, \tilde{\boldsymbol{X}}) + \lambda \|\boldsymbol{C}\|_\xi, \tag{1}$$

where $\tilde{\boldsymbol{X}}$ indicates the data matrix composed of the samples in $\boldsymbol{X}$, $L(\cdot,\cdot) : \mathbb{R}^{N\times D} - \mathbb{R}^{N\times D} \to \mathbb{R}^+$ denotes the reconstruction loss, λ is the trade-off parameter and $\|\cdot\|_\xi$ denotes the regularization term where different ξ's lead to various norms [4], [10], *e.g.*, $\|\boldsymbol{C}\|_1$ [9], [17], $\|\boldsymbol{C}\|_*$ [48], [64], $\|\boldsymbol{C}\|_F^2$ [52], or many kinds of mixed norms like trace Lasso [58] or elastic net [39].

The $\boldsymbol{C}$ in (1) can be explained as a new representation of $\tilde{\boldsymbol{X}}$, where each sample $\boldsymbol{x}_i \in \mathbb{R}^D$ is mapped to $\boldsymbol{c}_i \in \mathbb{R}^N$. Furthermore, $\boldsymbol{C}$ is a pairwise distance matrix where each entry c_{ij} reflects the similarity between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Nevertheless, the pairwise distance reflects poor discriminative capacity on partitioning samples near the intersection of two subspaces. To handle this problem, in this paper, we explore a higher-dimensional similarity called Triplet relationship, which is based on a greedy combination of pairwise distances reflected by $\boldsymbol{C}$.

C. Discovering Triplet Relationships

Given the similarity matrix $\boldsymbol{C}$ where each entry c_{ij} reflects the pairwise relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we propose to find the neighboring structure in a greedy way. For each data sample $\boldsymbol{x}_j \in S$, subspace clustering algorithms calculate a projected adjacent space based on the self-expressive property, *i.e.*, each data sample can be reconstructed by a linear combination of other points in the dataset $\boldsymbol{X}$ [17], [74]. Therefore, $\boldsymbol{x}_j$ is represented as

$$\boldsymbol{x}_j = \tilde{\boldsymbol{X}}\boldsymbol{c}_j, \quad s.t., c_{jj} = 0, \tag{2}$$

TABLE I
SUMMARY OF MAIN NOTATIONS IN THE MANUSCRIPT AND CORRESPONDING DESCRIPTIONS.

Notation	Description
$\boldsymbol{x}, \boldsymbol{X}$	data sample, a set of samples
$\tilde{\boldsymbol{X}}$	data matrix composed of samples in $\boldsymbol{X}$
$\boldsymbol{X}_{-\mathcal{N}}$	samples in $\boldsymbol{X}$ without $\boldsymbol{x} \in \mathcal{N}$
S	a subspace in the ambient space
$\boldsymbol{\tau}, \boldsymbol{T}$	triplet, a set of triplets
d, D	dimensionality of subspace, dimensionality of sample
n, N	number of triplets, number of samples
$K, \widetilde{K}, \widehat{K}$	real number of clusters, initialized number of cluster centers, estimated number of clusters
$\boldsymbol{C}$	similarity matrix derived from subspace representation method
$\boldsymbol{C}^*$	binary similarity matrix preserving the top m values in each row of $\boldsymbol{C}$ and modifying them to 1
c_{ij}	ij-th entry of the similarity matrix $\boldsymbol{C}$
$\boldsymbol{c}_i$	i-th column of $\boldsymbol{C}$
$\boldsymbol{N}_m(\boldsymbol{x})$	set of m nearest neighbors of $\boldsymbol{x}$
$\mathcal{C}$	cluster center which contains part of samples in a subspace
$\boldsymbol{T}_{in}^I, \boldsymbol{T}_{out}^I$	set of triplets which are already/not assigned into clusters in the I-th iteration
$\boldsymbol{X}_{in}^I, \boldsymbol{X}_{out}^I$	set of samples which are already/not assigned into clusters in the I-th iteration, preserving the frequency
$R_m(\mathcal{C})$	model selection reward of $\mathcal{C}$
$R_f^i(\mathcal{C}_i\|\boldsymbol{x})$	fusion reward that $\boldsymbol{x}$ being fused into $\mathcal{C}_i$
s_{ij}	connection score of $\boldsymbol{x}_i$ toward $\boldsymbol{x}_j$
$\rho(\boldsymbol{\tau}, \boldsymbol{X})$	local density of $\boldsymbol{\tau}$ against $\boldsymbol{X}$
$\mathcal{G}_i, \boldsymbol{\mathcal{G}}$	one of the result groups, set of the result groups
NC_e	deviation rate between the estimated $\widehat{K}$ and K
$\mathcal{A}$	error rate of the triplets

where the data matrix $\tilde{\boldsymbol{X}}$ is considered as a self-expression dictionary for representation. In addition, $\boldsymbol{c}_j = [c_{1j}, c_{2j}, \cdots, c_{Nj}]$ records the coefficients of such combination system. With the regularization from various well-designed norms on $\boldsymbol{c}_j$, the optimized result of (2) is capable of preserving only linear combinations of samples in S while eliminating others. Inspired by [3], [43], for each sample $\boldsymbol{x}_j$, we first collect its m nearest neighbors, i.e. those with the top m coefficients in $\boldsymbol{c}_j$. The m nearest neighbors are defined as follows.

Definition 1. (m Nearest Neighbors) *Let $\boldsymbol{N}_m(\boldsymbol{x}_j) \in \mathbb{R}^{1\times m}$ denote the m nearest neighbors for data point $\boldsymbol{x}_j$. Let:*

$$\boldsymbol{N}_m(\boldsymbol{x}_j) = \arg\max_{\{\boldsymbol{x}_{\boldsymbol{i}_l}\}} \sum_{l=1}^{m} |c_{\boldsymbol{i}_l,j}|. \tag{3}$$

where $\boldsymbol{i}_l$ denotes the set of indices for the nearest neighbors, and $c_{\boldsymbol{i}_l,j}$ denotes the coefficient between $\boldsymbol{x}_{\boldsymbol{i}_l}$ and $\boldsymbol{x}_j$.

According to Definition 1, we obtain $\boldsymbol{N}_m(\boldsymbol{x}_j)$ for $\boldsymbol{x}_j$ which contains the m samples with the m largest coefficients in $\boldsymbol{c}_j$. The number of preserved neighbors, *i.e.*, the parameter m, reflects the intrinsic dimension of the low-dimensional subspaces [75], [76], which we empirically evaluated in the experiment section. Based on the m nearest neighbors, we define the triplet relationship to explore the local hyper-correlation among samples.

Definition 2. (Triplet Relationship) *A triplet $\boldsymbol{\tau}$ includes three samples,* i.e.*, $\boldsymbol{\tau} = \{\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k\}$, and their relationships, if and only if $\boldsymbol{x}_i, \boldsymbol{x}_j$ and $\boldsymbol{x}_k$ satisfy:*

$$\mathbf{1}_{\boldsymbol{x}_i \in \boldsymbol{N}_m(\boldsymbol{x}_j)} \times \mathbf{1}_{\boldsymbol{x}_j \in \boldsymbol{N}_m(\boldsymbol{x}_k)} \times \mathbf{1}_{\boldsymbol{x}_k \in \boldsymbol{N}_m(\boldsymbol{x}_i)} = 1, \tag{4}$$

where $\mathbf{1}_{\boldsymbol{x}\in\boldsymbol{N}_m}$ denotes the indicator function which equals 1 if $\boldsymbol{x} \in \boldsymbol{N}_m$ and 0 otherwise.

Based on Definition 2, we obtain n triplets where we always have $n > N$, *i.e.*, each sample is included in multiple triplet relationships. For clarity of presentation, we define a triplet matrix $\boldsymbol{T} \in \mathbb{R}^{n\times 3}$ for data samples $\boldsymbol{X}$, where each row of $\boldsymbol{T}$ records the indices of a samples in a triplet $\boldsymbol{\tau} = \{\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k\}$.

Compared against the traditional pairwise relationship evoked from $\boldsymbol{C}$, the triplet incorporates complementary using the constraint in (4), which shows more robust capacity in partitioning samples near the intersection of two subspaces. Each triplet depicts a local geometrical structure which enables a better performance to estimate the density of each sample. Furthermore, the overlapped samples in multiple triplets reflect a global hyper-similarity among each other, which can be measured efficiently. Therefore, based on the triplet relationship, we can jointly estimate the number of subspaces and calculate the clustering assignment in a greedy manner.

D. Modeling Clustering Rewards

Given $\boldsymbol{X}$, we iteratively group data samples into clusters, *i.e.*, $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$, where $\widehat{K}$ denotes the estimated number of subspaces. According to the greedy strategy, in the I-th iteration, the triplet $\boldsymbol{T}$ is divided into two subsets, *i.e.*, "in-cluster" triplets $\boldsymbol{T}_{in}^I$ which are already assigned into clusters, and "out-of-cluster" triplets $\boldsymbol{T}_{out}^I$ which are still to be assigned in the subsequent iterations. For clear presentation, we reshape both matrices $\boldsymbol{T}_{in}^I \in \mathbb{R}^{p\times 3}$ and $\boldsymbol{T}_{out}^I \in \mathbb{R}^{q\times 3}$ to vectors $\boldsymbol{X}_{in}^I \in \mathbb{R}^{3p}$ and $\boldsymbol{X}_{out}^I \in \mathbb{R}^{3q}$. In each iteration, we propose to optimize two new rewards, *i.e.*, the model selection and the fusion reward, to simultaneously estimate the number of clusters and merge samples into respective cluster.

Definition 3. (Model Selection Reward) *Given $\boldsymbol{X}_{in}^I$ and $\boldsymbol{X}_{out}^I$ in the I-th iteration, the model selection reward $R_m(\mathcal{C})$ for each initialized cluster $\mathcal{C}$ in $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$ is defined as:*

$$R_m(\mathcal{C}) = \sum_i \sigma(\mathcal{C}_i|\boldsymbol{X}_{out}^I) - \lambda_m \sum_i \sigma(\mathcal{C}_i|\boldsymbol{X}_{in}^I), \tag{5}$$

where $\sigma(\mathcal{C}|\boldsymbol{X})$ is a counting function on the frequency that $\boldsymbol{x} \in \boldsymbol{X}$ for all $\boldsymbol{x} \in \mathcal{C}_i$, λ_m denotes the trade-off parameter.

By maximizing the model selection reward $R_m(\mathcal{C})$, we generate the initialized cluster $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$ which has the following two advantages, where $\widehat{K}$ is the estimated number of clusters (see Fig. 1 for visualization). Firstly, the local density of sample $\boldsymbol{x} \in \mathcal{C}$ is high, *i.e.*, $\boldsymbol{x}$ has a large amount of correlated samples in $\boldsymbol{X}_{out}$, which enables many to be merged in the next iteration. Secondly, each $\mathcal{C}$ has little correlation with samples in $\boldsymbol{X}_{in}$, which eliminates the overlap of any inter-clusters. Consequently, we can simultaneously estimate $\widehat{K}$ and initialize the clusters by optimizing the model selection reward R_m.

Algorithm 1 : Automatic Subspace Clustering (autoSC)

Input: $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$, m.

Normalize the magnitudes by $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i/\|\boldsymbol{x}\|_2$;
Calculate the similarity matrix $\boldsymbol{C}$ by (1);
for $i = 1 : N$ **do**
    Calculate the m Nearest Neighbors $\boldsymbol{N}_m(\boldsymbol{x}_i)$ by (3);
end for
Generate the triplet matrix $\boldsymbol{T} \in \mathbb{R}^{n\times 3}$ by (10);
Reshape $\boldsymbol{T}$ to $\boldsymbol{X}_{out} \in \mathbb{R}^{3n}$, $\boldsymbol{X}_{in} = \varnothing$;
$\widetilde{K} = 1$;
Calculate $\boldsymbol{\tau}_{ini}^{\widetilde{K}}$ by (11);
while $\rho(\boldsymbol{\tau}_{ini}^{\widetilde{K}}, \boldsymbol{X}_{out}) > \rho(\boldsymbol{\tau}_{ini}^{\widetilde{K}}, \boldsymbol{X}_{in})$ **do**
    $\mathcal{C}_{\widetilde{K}} = \boldsymbol{\tau}_{ini}^{\widetilde{K}}$;
    $\mathcal{C}_{\widetilde{K}} = \mathcal{C}_{\widetilde{K}} \cup \{\boldsymbol{\tau}^*\}$ where $\boldsymbol{\tau}^*$ is calculated by (12);
    $\boldsymbol{X}_{in} = \boldsymbol{X}_{in} \cup \boldsymbol{\tau}_{ini}^{\widetilde{K}} \cup \{\boldsymbol{\tau}^*\}$;
    $\boldsymbol{X}_{out} = \boldsymbol{X}_{out}/(\boldsymbol{\tau}_{ini}^{\widetilde{K}} \cup \{\boldsymbol{\tau}^*\})$;
    $\widetilde{K} = \widetilde{K} + 1$;
    Calculate $\boldsymbol{\tau}_{ini}^{\widetilde{K}}$ by (11);
end while
Merge $\mathcal{C}_i$ and $\mathcal{C}_j$ if we have (14); Get $\widehat{K}$ clusters;
for $j = 1 : |\boldsymbol{X}_{out}|$ **do**
    Calculate $\mathcal{C}^*$ for $\boldsymbol{x}_j$ by (15);
end for

Output: The cluster assignment $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$.

Definition 4. (Fusion Reward) *Given the initialized clusters $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$, the fusion reward is defined as the probability that $\boldsymbol{x}_j \in \boldsymbol{X}_{out}$ is assigned into $\mathcal{C}_i$:*

$$R_f^i(\mathcal{C}_i|\boldsymbol{x}_j \in \boldsymbol{X}_{out}) = \sigma(\boldsymbol{x}_j|\mathcal{C}_i) + \lambda_f \sigma(\boldsymbol{N}_m(\boldsymbol{x}_j)|\boldsymbol{N}_m(\mathcal{C}_i)), \tag{6}$$

where $\boldsymbol{N}_m(\boldsymbol{x}_j)$ denotes the m nearest neighbors of $\boldsymbol{x}_j$ and $\boldsymbol{N}_m(\mathcal{C}_i)$ denotes the set of m nearest neighbors of samples in $\mathcal{C}_i$, λ_f denotes the trade-off parameters.

In the optimization procedure, we calculate $\widehat{K}$ fusion rewards $\{R_f^i\}_{i=1}^{\widehat{K}}$ for each $\boldsymbol{x}_j$, which represent the probabilities that $\boldsymbol{x}_j$ is assigned into clusters $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$, respectively. We then merge $\boldsymbol{x}_j$ into the cluster with the largest fusion reward, and move $\boldsymbol{x}_j$ from $\boldsymbol{X}_{out}$ to $\boldsymbol{X}_{in}$.

E. Automatic Subspace Clustering Algorithm

The first triplet for initializing a new cluster is chosen to have maximal local density. The local density ρ is defined as follows.

Definition 5. (Local Density) *The local density ρ of the triplet $\boldsymbol{\tau}$ regarding to the $\boldsymbol{X}_{out}$ is defined as follows:*

$$\rho(\boldsymbol{\tau}, \boldsymbol{X}_{out}) = \sum_{j=1}^{|\boldsymbol{n}|} \sigma(\boldsymbol{x}_{\boldsymbol{n}_j}|\boldsymbol{X}_{out}), \tag{7}$$

where $\boldsymbol{x}_{\boldsymbol{n}_j}$ denotes the sample in the current triplet $\boldsymbol{\tau}$ and $\boldsymbol{n}$ is the set of their indexes, $|\boldsymbol{n}|$ denotes the scale of $\boldsymbol{n}$.

Also, to measure the hyper-similarity between samples and determine the optimal triplet to merge into the initialized clusters, we define the connection score s as follows.

Definition 6. (Connection Score) *The connection score s between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as:*

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = f\left(\boldsymbol{x}_i \middle| \bigcap_{k=1}^{n'} \left(\mathbf{1}_{\boldsymbol{x}_j \in \boldsymbol{\tau}_k} \times \boldsymbol{\tau}_k\right)\right), \tag{8}$$

where $\mathbf{1}_{\boldsymbol{x}_j \in \boldsymbol{\tau}_k}$ is equal to 1 when $\boldsymbol{x}_j \in \boldsymbol{\tau}_k$ and 0 otherwise, n' is the number of all triplets in $\boldsymbol{T}_{out}$.

We greedily optimize the proposed model selection reward R_m and fusion reward R_f in autoSC to simultaneously estimate the number of clusters and generate the segmentation among samples:

$$\begin{aligned} &\max_{\boldsymbol{\mathcal{G}}, \widehat{K}} \sum_{k=1}^{\widehat{K}} R_m(\mathcal{G}_k) + \lambda \sum_{k=1}^{\widehat{K}} R_f(\mathcal{G}_k|\boldsymbol{X}), \\ &s.t.\ \mathcal{G}_k \bigcap \mathcal{G}_{k'\neq k} = \varnothing, \bigcup_{k=1}^{\widehat{K}} \mathcal{G}_k = [1, \cdots, N], \end{aligned} \tag{9}$$

where $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_1, \cdots, \mathcal{G}_{\widehat{K}}\}$ denotes the set of the result groups, $\widehat{K}$ is the estimated number of clusters and $[1, \cdots, N]$ denotes the universal ordinal set of samples.

We present the proposed autoSC in Fig. 1 and Algorithm 1. Specifically, the optimization includes three steps: 1) generating the triplet relationships $\boldsymbol{T}$ from the similarity matrix $\boldsymbol{C}$; 2) estimating the number of clusters $\widehat{K}$ and initializing the clusters $\mathcal{C}$; 3) assigning the samples $\boldsymbol{x} \in \boldsymbol{X}_{out}$ into proper cluster.

Calculating Triplets: The similarity matrix $\boldsymbol{C}$ reflects the correlations among samples [9], where larger values demonstrate stronger belief for the correlation between samples. For instance, $c_{ij} > c_{ik}$ indicates a larger probability for $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ being in the same cluster over $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$. Accordingly, we explore the intrinsic local correlations among samples by the proposed triplets derived from $\boldsymbol{C}$.

Many subspace representations guarantee the mapping invariance via a dense similarity matrix $\boldsymbol{C}$. However, the generation of triplets relies only on the strongest connections to avoid the wrong assignment. Therefore, for each column of $\boldsymbol{C}$, *i.e.*, $\boldsymbol{c}_i$, we preserve only the top m values which are then modified to 1 for a new binary similarity matrix $\boldsymbol{C}^*$.

Then, we extract each triplet from $\boldsymbol{C}^*$ by the following function:

$$\begin{aligned} &\boldsymbol{\tau} = \{\boldsymbol{x}_{\boldsymbol{n}_1}, \boldsymbol{x}_{\boldsymbol{n}_2}, \boldsymbol{x}_{\boldsymbol{n}_3}\} \in \boldsymbol{T}, \\ &\text{if and only if}: \quad c^*_{\boldsymbol{n}_1\boldsymbol{n}_2} \times c^*_{\boldsymbol{n}_2\boldsymbol{n}_3} \times c^*_{\boldsymbol{n}_3\boldsymbol{n}_1} = 1, \end{aligned} \tag{10}$$

where c^*_{xy} denotes the xy-th value of $\boldsymbol{C}^*$. Note each sample $\boldsymbol{x}$ can appear in many triplets. Therefore, we consider each $\boldsymbol{\tau}$ as a meta-element in the clustering, which improves the robustness due to the complementarity constraints.

Initializing Clusters: In the I-th iteration, we first determine an initial triplet (termed as $\boldsymbol{\tau}_{ini}^I$) from $\boldsymbol{T}_{out}$ to initialize the cluster $\mathcal{C}$, followed by merging the most correlated samples of $\boldsymbol{\tau}_{ini}^I$ into each $\mathcal{C}$.

Following [20], we initialize a new cluster using $\boldsymbol{\tau}_{ini}^{I}$ with highest local density:

$$\boldsymbol{\tau}_{ini}^{I} = \arg\max_{\boldsymbol{\tau}} \quad \rho(\boldsymbol{\tau}, \boldsymbol{X}_{out}^{I}), \tag{11}$$

where ρ calculates the local density defined in Definition 5. The high local density of the triplet reflects the most connections between $\boldsymbol{\tau}_i$ and other triplets, which produces the most connections between $\boldsymbol{x}_{\boldsymbol{n}_j^i}$ and other samples in $\boldsymbol{X}_{out}^{I}$.

Once the initialized triplet $\boldsymbol{\tau}_{ini}^{I}$ is determined, we iteratively extend the initialized cluster $\mathcal{C}$ by fusing the most confident triplets. For each triplet $\boldsymbol{\tau}_i$ in $\boldsymbol{T}_{out}$, we calculate the sum of the connection score regarding the samples in $\mathcal{C}$ to greedily determine whether the samples in $\boldsymbol{\tau}_i$ should be assigned into $\mathcal{C}$ or not:

$$\begin{aligned} &\boldsymbol{\tau}^* = \arg\max_{\boldsymbol{\tau}} \sum_{j=1}^{3}\sum_{\kappa}^{|\boldsymbol{m}|} s_{\boldsymbol{n}_j \boldsymbol{m}_\kappa}, \\ &s.t. \sum_{j=1}^{3}\sum_{\kappa}^{|\boldsymbol{m}|} s_{\boldsymbol{n}_j \boldsymbol{m}_\kappa} > 1; \{\boldsymbol{x}_{\boldsymbol{n}_j}\}_{j=1}^{3} \in \boldsymbol{\tau}; \{\boldsymbol{x}_{\boldsymbol{m}_\kappa}\}_{\kappa=1}^{|\boldsymbol{m}|} \in \mathcal{C}, \end{aligned} \tag{12}$$

where $\boldsymbol{n}, \boldsymbol{m}$ denote the set of indexes for the samples in $\boldsymbol{\tau}$ and $\mathcal{C}$, respectively. We iteratively update the auxiliary sets $\boldsymbol{T}_{out}^{I}$, $\boldsymbol{T}_{in}^{I}$, $\boldsymbol{X}_{out}^{I}$ and $\boldsymbol{X}_{in}^{I}$ in the iterations.

Terminating: We terminate the process of estimating the number of clusters and get $\widetilde{K}$ clusters if and only if $\boldsymbol{\tau}_{ini}^{\widetilde{K}+1}$ satisfies:

$$\rho(\boldsymbol{\tau}_{ini}^{\widetilde{K}+1}, \boldsymbol{X}_{out}^{\widetilde{K}+1}) \leq \rho(\boldsymbol{\tau}_{ini}^{\widetilde{K}+1}, \boldsymbol{X}_{in}^{\widetilde{K}+1}). \tag{13}$$

Specifically, if the samples in $\boldsymbol{\tau}_{ini}^{\widetilde{K}+1}$ are of high frequency in $\boldsymbol{X}_{in}^{\widetilde{K}+1}$, *i.e.*, the triplet with the highest local density in $\boldsymbol{T}_{out}^{\widetilde{K}+1}$ is already contained in $\boldsymbol{T}_{in}^{\widetilde{K}+1}$, we consider that the clusters are sufficient for modeling the intrinsic subspaces.

Avoiding Over-Segmentation: We also introduce an alternative step to check the redundancy among initialized clusters $\{\mathcal{C}_i\}_{i=1}^{\widetilde{K}}$ to avoid over-segmentation. We calculate the connection scores s for small-scale clusters against others, and merge the highly correlated clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ if we have

$$s_{ij} > \min(|\mathcal{C}_i|, |\mathcal{C}_j|), \tag{14}$$

where $|\mathcal{C}|$ denotes the number of samples in $\mathcal{C}$. We then get the initialized clusters $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$, where $\widehat{K}$ is the estimated number of clusters and $\widehat{K} \leq K$.

Assigning Rest Samples: Given $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$, we assign each of the remaining samples into $\mathcal{C}$ which evokes an optimal fusion reward. For $\boldsymbol{x}_j$, we find its optimal cluster $\mathcal{C}^*$ by the following equation:

$$\mathcal{C}^* = \arg\max_{\mathcal{C}_i} \; R_f(\mathcal{C}_i|\boldsymbol{x}_j), \quad i \in \{1, 2, \cdots, \widehat{K}\}, \tag{15}$$

where $R_f(\mathcal{C}|\boldsymbol{x})$ is the fusion reward defined by (6).

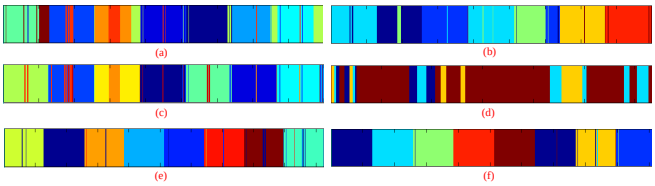


Fig. 2. Visualization of the clustering assignments for six methods, *i.e.*, (a) SCAMS, (b) DP, (c) SVD, (d) DP-space, (e) autoSC-N and (f) autoSC. The experiments are conducted on the extended Yale B dataset with 8 subjects. SCAMS over-segments the samples, while DP-space assigns the majority into one cluster. The proposed autoSC-N and auto-SC do not suffer from these problems.

Algorithm 2 : Neighboring based autoSC (autoSC-N)

Input: $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \in \mathbb{R}^{D\times N}$, m.

1: Normalize the magnitudes by $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i/\|\boldsymbol{x}\|_2$;
2: Initialize the overall neighbor matrix $\mathcal{N}$ with $\mathcal{N} = \varnothing$;
3: **for** $j = 1 : N$ **do**
4: Initialize the spanned subspace $\mathcal{S}_j$ with $\mathcal{S}_j = \boldsymbol{x}_j$;
5: Initialize the neighbor matrix $\mathcal{N}_j$ with $\mathcal{N}_j = \varnothing$;
6: **for** $I = 1 : m$ **do**
7: Calculate the similarity s_{jk} between $\boldsymbol{x}_j$ and $\boldsymbol{x}_k \in \boldsymbol{X}_{-\mathcal{N}}$ using (16);
8: Update the spanned subspace $\mathcal{S}_j$ using (17);
9: $\mathcal{N}_j^{I+1} \leftarrow \mathcal{N}_j^{I} \cup \arg\max_{\boldsymbol{x}_k \in \boldsymbol{X}_{-\mathcal{N}}} s_{jk}$;
10: **end for**
11: $\mathcal{N} \leftarrow \mathcal{N} \cup \mathcal{N}_j$;
12: **end for**
13: Let $\mathcal{N}$ replace the m Nearest Neighbors and conduct the steps from Step 6 to Step 21 in Algorithm 1.

Output: The cluster assignment $\{\mathcal{C}_i\}_{i=1}^{\widehat{K}}$.

F. An Extension: Neighboring based AutoSC Algorithm

In Definition 1, we collect m nearest neighbors according to the magnitude of similarities between sample $\boldsymbol{x}_j$ and all other samples in $\boldsymbol{X}$. These similarities are depicted in $\boldsymbol{C}$ by optimizing (1) which is composed of a reconstruction loss term and a regularization term. In this subsection, we extend the autoSC with an alternative technique to find $\boldsymbol{N}_m(\boldsymbol{x}_j)$ for each $\boldsymbol{x}_j$ based on greedy search.

For each data sample $\boldsymbol{x}_j$, we let $\mathcal{S}_j^I$ be the subspace spanned by $\boldsymbol{x}_j$ and its neighbors in the I-th iteration, where the neighbor set $\mathcal{N}_j$ is initialized as $\mathcal{N}_j^0 = \varnothing$, and $\mathcal{S}_j^I \in \mathbb{R}^{D\times \dim(\mathcal{N}_j)}$. In each iteration, we measure the projected similarity between $\boldsymbol{x}_j$ and other non-neighbor samples by calculating the orthonormal ordinates in the spanned subspace. For example, to calculate the similarity between $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ in the I-th iteration, we have

$$s_{jk} = \|(S_j^I)^\top \boldsymbol{x}_k\|_F^2, \tag{16}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm and $\boldsymbol{x}_k \in \boldsymbol{X}_{-\mathcal{N}}$. Consequently, for $\boldsymbol{x}_j$ in the I-th iteration, we find the closest neighbor and update $\mathcal{S}_j$ as follows:

$$\mathcal{S}_j^{I+1} \leftarrow \mathcal{S}_j^I \cup \arg\max_{\boldsymbol{x}_k \in \boldsymbol{X}_{-\mathcal{N}}} s_{jk}. \tag{17}$$

Here, we find one neighbor in each iteration and update the spanned subspace accordingly. The newly spanned subspace

reflects more local structure of the ambient space which is assumed to cover the current sample. The neighbor set $\mathcal{N}_j$ is also updated by adding the new neighbor which is found in the I-th iteration. Finally, with m iterations for each sample, we get an alternative m nearest neighbor set $\mathcal{N}$.

Given the neighbor matrix $\mathcal{N}$, we propose the neighboring based autoSC algorithm (autoSC-N) to directly discover the triplet relationship among data samples, followed by optimizing both model selection and fusion rewards for clustering. The main steps of autoSC-N are summarized in Algorithm 2.

G. Computational Complexity Analysis

In traditional subspace clustering system, the calculation of self-representation requires solving N convex optimization problems over $D \times N$ constraints [21]. Spectral clustering is based on an eigen-decomposition operation on the Laplacian matrix followed by conducting K-means on the eigenvectors, both of which are time-consuming, involving a complex algebraic decomposition and iterative optimization, respectively [74], [77]. The overall computational complexity can be more than $O(N^3)$. For the proposed autoSC, it takes $O(Nm^2)$ to collect the triplet relationships for N samples in the space spanned by the m nearest neighbors. Here, since we have $m \ll N$, the complexity of collecting the triplet relationships is $O(N)$. The optimization of both model selection and fusion rewards takes $O(Nn)$ where the number of triplets n has the same order of magnitude as N. Specifically, we have $n \approx N \times \frac{m}{2}$, and thus the complexity of clustering is $O(N^2)$.

For the extension, *i.e.*, autoSC-N, collecting the neighbor matrix takes $O(Nm)$ where the basic operation is a dot product of the D-dimensional tensors. This avoids the calculation of any convex optimization problem.

IV. Experiments

A. Experimental Setup

In the experiments[1], we compare the automatic methods on the benchmark datasets, *i.e.*, the extended Yale B [78] and the COIL-20 [79] dataset, followed by verifying the robustness of the proposed method to different $\boldsymbol{C}$ derived from various self-representation schemes along with combinations of different methods for estimating the number of clusters and segmenting the samples. We design comprehensive evaluation metrics to validate the clustering performance, *i.e.*, the error rate of the number of clusters and the triplets. For all experiments on subsets, the reported results are the average of 50 trials. We also conduct experiments on a motion segmentation task using the Hopkins 155 dataset.

1) Datasets: The extended Yale B [78] dataset is a widely used face clustering dataset which contains face images with different illumination of 38 subjects, each subject has 64 images.

The COIL-20 [79] dataset consists of 20 different real subjects, including cups, bottles and so on. For each subject, there are 72 images with different camera viewpoints.

The Hopkins 155 dataset [80] consists of 155 video sequences. For each video sequence, there are 2 or 3 motions.

[1] Code available at https://github.com/JLiangNKU/autoSC.

2) Comparative Methods: We make comparisons with the following methods: SCAMS [14], [35], a density peak based method (DP) [20], a singular value decomposition based method (SVD) [64] and DP-space [13]. Besides, we utilize the following subspace representation methods to generate different coefficient matrices $\boldsymbol{C}$: LRR [45], CASS [58], LSR [52], SMR [38] and ORGEN [39]. The similarity matrix $\boldsymbol{C}$ is then used to calculate the triplet relationships for autoSC.

3) Evaluation Metrics: To evaluate the performance of the proposed triplets, we define the error rate $\mathcal{A}$ as follows:

$$\mathcal{A} = \frac{1}{n}\sum_{i=1}^{n}\frac{3-\sigma(\boldsymbol{\tau}_i|\boldsymbol{g}_i^*)}{2}, \tag{18}$$

where n denotes the number of the triplets and $\sigma(\boldsymbol{\tau}|\boldsymbol{g}^*)$ is the counting function on the frequency that $\boldsymbol{x} \in \boldsymbol{g}^*$ for all $\boldsymbol{x} \in \boldsymbol{\tau}$. Here the output of f ranges from 0 to 2. The dynamic set $\boldsymbol{g}_i^*$ consists of samples in one subspace S according to the ground truth, where S contains as many samples in $\boldsymbol{\tau}_i$ as possible.

We introduce the error rate of the number of clusters (NC_e) as the primary evaluation metric for the clustering methods which estimate the number of clusters $\widehat{K}$ automatically:

$$\mathrm{NC}_e = \frac{1}{M}\sum_{i=1}^{M}|\widehat{K}_i - K|, \tag{19}$$

where K is the real number of clusters, M is the number of trials and $\widehat{K}_i$ is the estimated number of clusters in the i-th trial. We also use the standard normalized mutual information (NMI) [81] to measure the similarity between two clustering distributions, *i.e.*, the prediction and the ground truth. With respect to NMI, the entropy illustrates the nondeterminacy of one clustering to the other, and the mutual information quantifies the amount of information that one variable obtains from the other.

4) Parameter m: The parameter m in Definition 1, *i.e.*, the number of preserved neighbors for each sample, is related to the intrinsic dimension of the subspaces. We empirically evaluate the influence of m on both extended Yale B and COIL-20 datasets with 15 subjects. Besides, we use subspace representations derived from SMR. The results are shown in Table III. As shown in the table, the proposed method achieves best performance when we have $m = 8$ for most cases. Actually, the parameter m is robust since the performance is stable when $m > 8$.

B. Comparisons among Automatic Clustering

We conduct experiments on the extended Yale B and COIL-20 datasets with different numbers of subjects, and compare four methods with the proposed autoSC and autoSC-N on the metrics of NC_e and NMI. For SCAMS [14], [35], DP [20], SVD [64] and our autoSC, the optimization module in SMR [38] is employed to generate the similarity matrix $\boldsymbol{C}$. The DP-space method simultaneously estimates $\widehat{K}$ and finds the subspaces without the requirement of a similarity matrix. All parameters of the contrasted methods are tuned to provide the best performance.

Fig. 2 and Table II report the performance. As shown in Table II, when combining SMR, the averaged NC_e of autoSC

TABLE II

OVERALL COMPARISON AMONG COMPARATIVE METHODS ON SUBSETS OF THE EXTENDED YALE B AND COIL-20 DATASETS. THE SIMILARITY MATRIX $\boldsymbol{C}$ CALCULATED BY BOTH SMR AND LSR IS UTILIZED AS THE CORRELATION MATRIX OF SCAMS, DP, SVD AND THE PROPOSED AUTOSC. THE BEST RESULTS ARE IN BOLD FONT WHILE * INDICATES THE SECOND BEST PERFORMANCE.

$\boldsymbol{C}$	Clustering	Metrics	extended Yale B					COIL-20			
			8	15	25	30	38	5	10	15	20
LSR [52]	SCAMS [14], [35]	NC_e	5.21	14.00	17.12	21.25	23.00	4.36	9.00	18.32	21.00
		NMI	0.1652	0.0643	0.1544	0.3821	0.4236	0.2435	0.1524	0.1124	0.1728
SMR [38]	SCAMS [14], [35]	NC_e	9.26	23.60	41.39	76.22	81.00	8.48	19.72	32.40	37.00
		NMI	0.7183	0.7272	0.6992	0.7266	0.7425	0.5885	0.6527	0.6668	0.6712
LSR [52]	DP [20]	NC_e	7.90	98.38	127.92	308.00	341.00	10.90	14.70	301.05	228.00
		NMI	0.7060	0.6067	0.6245	0.6516	0.6611	0.7060	0.4984	0.6516	0.5283
SMR [38]	DP [20]	NC_e	3.06	7.84	14.62	24.76	29.00	2.22	5.30	9.72	11.00
		NMI	0.6196	0.5026	0.4391	0.2166	0.2384	0.6864	0.4467	0.3643	0.3547
LSR [52]	SVD [64]	NC_e	7.00	9.42	21.04	41.23	44.00	2.76	9.00	12.05	14.00
		NMI	0.2412	0.4304	0.5567	0.6523	0.6726	0.6210	0.1302	0.4092	0.4125
SMR [38]	SVD [64]	NC_e	2.40	9.06	11.65	24.00	28.00	0.48	2.58	8.36	12.00
		NMI	0.7078	0.4993	0.3739	0.2808	0.2766	0.7024	0.7127	0.7224	0.7035
-	DP-space [13]	NC_e	2.08	8.96	15.75	23.92	26.00	0.78	4.78	9.38	14.00
		NMI	0.0343	0.0226	0.0432	0.0406	0.0525	0.0904	0.0829	0.0718	0.0834
-	**autoSC-N**	NC_e	0.87*	3.16*	4.32*	7.68*	9.00*	0.75*	2.21*	2.42	5.00
		NMI	0.8306*	0.7328	0.7165*	0.6566*	0.6871*	0.7933*	0.6216*	0.7895*	0.7126
LSR [52]	**autoSC**	NC_e	1.08	3.32	5.79	10.50	12.00	1.50	3.40	2.00*	4.00*
		NMI	0.8251	0.7375*	0.6871	0.5972	0.5833	0.7786	0.5581	**0.8670**	**0.8239**
SMR [38]	**autoSC**	NC_e	**0.76**	**2.08**	**3.15**	**4.98**	**4.00**	**0.38**	**1.18**	**0.80**	**2.00**
		NMI	**0.9062**	**0.8589**	**0.8432**	**0.8287**	**0.7943**	**0.8315**	**0.7701**	0.7266	0.7568*

TABLE III

CLUSTERING PERFORMANCE OF THE PROPOSED AUTOSC WITH DIFFERENT m (THE FIRST ROW) ON BOTH THE EXTENDED YALE B (EYALEB) AND COIL-20 (COIL-20) DATASETS WITH 15 SUBJECTS EACH. THE SUBSPACE REPRESENTATION IS DERIVED FROM SMR. BASED ON THE RESULTS, WE SET $m = 8$ IN THE REST OF THE PAPER.

Datasets	Metrics	5	6	7	8	9	10	11
EYaleB	NC_e	4.52	3.68	3.13	**2.08**	2.12	2.29	2.18
	NMI	0.7125	0.7736	0.8047	**0.858**	0.8551	0.8423	0.8536
Coil-20	NC_e	1.68	1.29	0.92	0.80	0.88	**0.76**	0.98
	NMI	0.5647	0.6157	0.6774	**0.7266**	0.7107	0.7211	0.7120

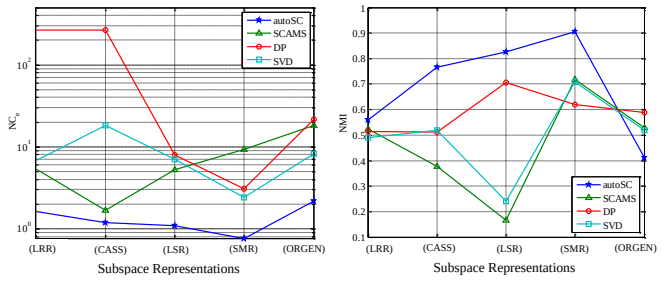


Fig. 3. Clustering results using different self-representation schemes on the extended Yale B dataset with 8 subjects. The left figure denotes the comparison of four methods using the NC_e metric while the right one uses NMI. Each point in the curves is derived by the combination of different clustering methods and self-representation schemes. The proposed autoSC achieves consistent performance on the evaluation of NC_e.

is smaller than other comparative methods on all experimental configurations, indicating that it gives a close estimation on the number of clusters. For example, the estimated $\widehat{K}$ on the extended Yale B with 8 subjects has a deviation of less than 1, and produces a NMI higher than 0.9. For each of the proposed triplet relationships, three samples are encouraged to be highly correlated and are considered as a meta-element during clustering, which can introduce more accurate information for segmenting two densely distributed subspaces than the conventional pair-wise methods. In addition, the operation for avoiding over-segmentation as shown in (14) also induces a better performance on determining the number of clusters. However, combining the similarity matrix derived from LSR [52] can hardly outperform autoSC-N which calculates $\boldsymbol{N}_m(\boldsymbol{x}_j)$ using greedy search in data space. Although it outperforms the others without the triplet relationships, the results still demonstrate a dependence of the proposed method on tuning the similarity among samples. The autoSC-N gets the second best performance on most configurations, which demonstrates the effectiveness of both triplet relationship and reward optimization.

In contrast, SVD achieves comparable results on the small-scale configuration of each dataset, but the performance becomes poor when the number of samples increases. It is mainly because the largest gap between the pair of singular values decreases when the number of clusters becomes larger. When combining SMR, SCAMS performs comparably according to NMI on both datasets, however, as is illustrated in Fig. 2 (a) and Table II, it provides a much larger $\widehat{K}$ than the ground truth, *e.g.*, $\widehat{K} > 100$ when $K = 30$ on the extended Yale B dataset. NMI does not strongly penalize over-segmentation, making the metric NC_e be the primary evaluation of the SCAMS method. The DP-space performs well on NC_e, but has poor performance on the NMI. This is because most samples are assigned into

TABLE IV
ERROR RATES OF TRIPLETS ($\mathcal{A}$) FOR THE PROPOSED AUTOSC ON SUBSETS OF THE EXTENDED YALE B (EYALEB) AND COIL-20 DATASETS. FOR EACH ROW, WE UTILIZE THE SIMILARITY MATRIX $\boldsymbol{C}$ DERIVED FROM ONE SELF-REPRESENTATION METHOD IN THE FIRST COLUMN. THE AUTOSC ACHIEVES CONSISTENCY ON THE CALCULATION OF THE TRIPLETS.

$\boldsymbol{C}$	extended Yale B					COIL-20			
	8	15	25	30	38	5	10	15	20
LRR [45]	0.0155	0.0147	0.0158	0.0176	0.0169	0.0185	0.0252	0.0224	0.0231
CASS [58]	0.0158	0.0148	0.0140	0.0157	0.0162	0.0195	0.0198	0.0203	0.0193
LSR [52]	0.0144	0.0148	0.0162	0.0181	0.0172	0.0188	0.0188	0.0212	0.0199
SMR [38]	0.0135	0.0149	0.0154	0.0181	0.0161	0.0175	0.0182	0.0196	0.0202
ORGEN [39]	0.0166	0.0145	0.0151	0.0177	0.0169	0.0196	0.0210	0.0215	0.0220

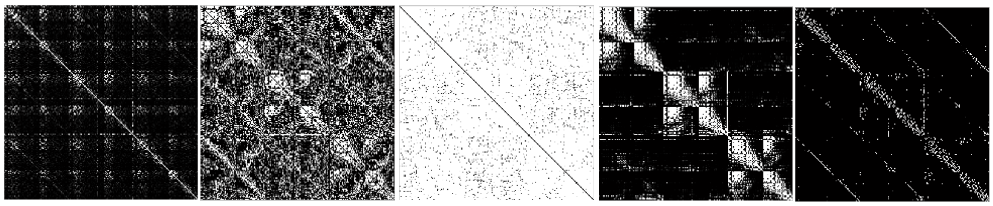


Fig. 4. Visualization of the similarity matrix $\boldsymbol{C}$ derived from different self-representation schemes on the extended Yale B dataset with 3 subjects. The white regions denote the locations with non-zero coefficients. Different methods, *i.e.*, (a) LRR, (b) CASS, (c) LSR, (d) SMR and (e) ORGEN, produce $\boldsymbol{C}$ with different characteristics, *e.g.*, (d) derived from SMR is block-diagonal, while (e) derived from ORGEN is sparse.

one cluster, and the other clusters are small. In addition, when combining LSR, as shown in Table II, the performance of all methods decrease, while the proposed autoSC still achieves best performance on most configurations against other state-of-the-art methods. It demonstrates the generalization ability of our autoSC. The main reason is that the three samples in any triplet relationships are complementary and can be more robust on wrong similarities to some extent. It is also proved by the consistently high accuracy of triplets as shown in Table IV, where the potential errors of similarities derived by LSR are compensated or corrected during the extraction of triplet relationship.

C. Robustness to Self-Representations Schemes

The methods including SCAMS [14], [35], DP [20], SVD [64] and the proposed autoSC require the similarity matrix $\boldsymbol{C}$ as input. Also, for DP [20], the distance among samples needs to be calculated. We calculate the distance d_{ij} between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by $d_{ij} = \frac{1}{c_{ij}}$ rather than the simple Euclidean distance. To verify the robustness of the proposed autoSC regarding various subspace representations, we calculate the similarity matrix $\boldsymbol{C}$ using 5 subspace representation modules, followed by the combinations with the 4 methods which automatically estimate the number of clusters and segment the samples.

Table IV shows the evaluation results of $\mathcal{A}$ on both datasets with the combinations of 5 subspace representations, while the NC_e and NMI on the extended Yale B dataset with 8 subjects are reported in Fig. 3. Moreover, we visualize the similarity matrix $\boldsymbol{C}$ derived from 5 subspace representation modules in Fig. 4. We can see from Fig. 3 that the SCAMS, DP and SVD methods are sensitive to the choice of the subspace representation module. For example, DP estimates $\widehat{K}$ as a relatively close value to the ground truth when combined with SMR ($\text{NC}_e = 3.06$), but generates a totally wrong estimation when combined with LRR ($\text{NC}_e = 265.60$). Different subspace representation modules generate coefficient matrices with various intrinsic properties [4], thus the parameter for truncation error ϵ needs to be tuned carefully.

For the proposed autoSC, it is stable on different combinations considering the metric of NC_e and $\mathcal{A}$, which demonstrates the complementary ability of the proposed method. For all combinations, the error rate of the triplets obtained from (10) is less than 2%, which guarantees the consistency of the proposed autoSC with different kinds of $\boldsymbol{C}$. It is also the main reason that the proposed method is robust on combining different self-representation schemes. Furthermore, it shows better performance when combined with CASS, LSR and SMR than other combinations on both metrics in Fig. 3. The reason lies on the guarantee of the mapping invariance which is termed as the grouping effect [38], [52], [58], together with the filtering of weak connections and the self-constraint among samples within triplets. As shown in Fig. 4 (b), (c), (d), the coefficient matrices are dense while it shows block-diagonal structure in Fig. 4 (d) and each block corresponds to one cluster. Therefore, the nearest neighbors which are used to generate the triplets can be chosen precisely. The performance decreases when combined with ORGEN since the similarity matrix $\boldsymbol{C}$ derived from ORGEN is sparse with less locations for constructing effective triplets.

D. Time Efficiency

Table V shows the run-time of comparative methods using subsets from the extended Yale B dataset. The experiments are conducted on a machine with a 2.93GHz CPU and 32GB RAM. AutoSC-N requires the least run-time compared to all

TABLE V
COMPARISON OF TIME-CONSUMPTION (IN SECONDS) ON THE EXTENDED YALE B DATASET. THE BEST RESULTS ARE IN BOLD FONT WHILE * INDICATES THE SECOND BEST PERFORMANCE.

Subjects	SCAMS	DP	SVD	DP-space	autoSC-N	autoSC
8	12.45	6.92	17.32	9.52	**1.69**	2.02*
15	30.12	13.04	44.13	18.05	**4.72**	5.79*
25	125.66	33.76	146.78	36.94	**10.28**	19.81*
30	175.80	59.02	225.93	67.89	**16.33**	36.06*
38	267.07	97.69	314.28	104.50	**29.45**	62.35*

TABLE VI
MOTION SEGMENTATION PERFORMANCE ON HOPKINS 155 DATASET. THE ROW OF 'TIME' REPORTS THE AVERAGE TIME-CONSUMPTION ON HANDLING EACH VIDEO SEQUENCE. THE BEST RESULTS ARE IN BOLD FONT WHILE * INDICATES THE SECOND BEST PERFORMANCE. THE PROPOSED AUTOSC OUTPERFORMS OTHER COMPARATIVE METHODS REGARDING BOTH THE ESTIMATION OF K AND CLUSTERING.

Metrics	SCAMS	DP	SVD	DP-space	autoSC-N	autoSC
NC_e	3.67	2.12	1.29	2.97	0.52*	**0.18**
NMI	0.7892	0.8233	0.8670	0.7921	0.9155*	**0.9871**
Time/s	2.68	1.22	2.45	1.56	**0.26**	0.32*

comparative methods due to the following two reasons. First, autoSC-N explores the neighborhood relationship in the raw data space rather than solving a convex optimization problem. Second, it employs a greedy optimization scheme to estimate the number of clusters and calculate the clustering assignment rather than a complex optimizing method such as computing the singular value decomposition. Note the proposed autoSC method achieves the second best result among comparative methods.

E. Real World Application: Motion Segmentation

Motion segmentation refers to the task of segmenting multiple video sequence. The candidate video is composed of multiple foreground objects, which are rapidly moving and required to be clustered into spatiotemporal regions corresponding to specific motions. Following the traditional scheme [17], we consider the Hopkins 155 dataset [80] and solve the motion segmentation problem by first extracting a set of feature points for each frame followed by clustering them based on the motions. Table VI reports the comparison against four automatic clustering methods. For SCAMS [14], [35], DP [20] and SVD [64], the SMR [38] is firstly conducted to calculate the similarity matrix. As shown in the table, the proposed autoSC achieves best performance on both metrics, indicating that the autoSC is effective at both estimating the number of motions (about 0.18 error rate) and segmenting the feature points (obtains NMI of more than 0.98). In addition, it shows favorable efficiency on the motion segmentation task. The autoSC-N is the most efficient method ($0.26s$ per sequence) with second best performance on NC_e and NMI. The SVD method obtains the best result among other comparative methods, but it consumes much more time (about more than $2.5s$ per sequence) due to the singular value decomposition process.

V. CONCLUSION

In this paper, we present to jointly estimate the number of clusters and segment the samples accordingly. We first design a hyper-correlation oriented meta-element termed as the triplet relationship based on the similarity matrix derived from state-of-the-art subspace clustering methods. The calculation of triplets is considered as a mapping from pairwise correlation to hyper-correlations, which is more robust than pairwise relationships when partitioning samples near the intersection of two subspaces due to the complementarity of mutual restrictions. The proposed autoSC and autoSC-N are verified to be effective mainly due to the robustness and effectiveness of the triplet relationship. Experimental results also show generalization capacity of the proposed method on various schemes when calculating the similarity matrix. It would be interesting to explore this effective triplet relationship in other hot topics which rely on modeling the relationship among samples, such as metric learning [82] and zero-shot learning [83]. *etc*. In addition, it is also a potential topic to combine the triplet relationship with graph neural networks (GNN, [84]) since GNN is a powerful tool on relational reasoning by learn the relationship between nodes and can also benefit from hyper-correlations.

VI. ACKNOWLEDGEMENT

This work was supported by the NSFC (NO.61876094, 61620106008, 61572264), Natural Science Foundation of Tianjin, China (NO.18JCYBJC15400, 18ZXZNGX00110), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] J. Yang, J. Liang, K. Wang, Y.-L. Yang, and M.-M. Cheng, "Automatic model selection in subspace clustering via triplet relationships," in *AAAI*, 2018. 1, 2
[2] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *CVPR*, 2017, pp. 923–931. 1
[3] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE TCYB*, vol. 47, no. 4, pp. 1053–1066, 2017. 1, 2, 5
[4] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011. 1, 4, 10
[5] H. Jia and Y.-M. Cheung, "Subspace clustering of categorical and numerical data with an unknown number of clusters," *IEEE TNNLS*, vol. 29, no. 8, pp. 3308–3325, 2018. 1
[6] X. Xu, Z. Huang, D. Graves, and W. Pedrycz, "A clustering-based graph laplacian framework for value function approximation in reinforcement learning," *IEEE TCYB*, vol. 44, no. 12, pp. 2613–2625, 2014. 1
[7] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Subspace clustering guided unsupervised feature selection," *Pattern Recognition*, vol. 66, pp. 364–374, 2017. 1
[8] X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh, "Constrained multi-view video face clustering," *IEEE TIP*, vol. 24, no. 11, pp. 4381–4393, 2015. 1
[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009. 1, 2, 3, 4, 6
[10] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "l_0-sparse subspace clustering," in *ECCV*, 2016. 1, 2, 4

[11] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NeurIPS*, 2002. 1

[12] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000. 1, 3, 4

[13] Y. Wang and J. Zhu, "DP-space: Bayesian nonparametric subspace clustering with small-variance asymptotics," in *ICML*, 2015. 1, 3, 4, 8, 9

[14] Z. Li, S. Yang, L. F. Cheong, and K. C. Toh, "Simultaneous clustering and model selection for tensor affinities," in *CVPR*, 2016. 1, 3, 4, 8, 9, 10, 11

[15] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Background-foreground modeling based on spatiotemporal sparse subspace clustering," *IEEE TIP*, vol. 26, no. 12, pp. 5840–5854, 2017. 1

[16] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, "A hybrid approach to clustering in big data," *IEEE TCYB*, vol. 46, no. 10, pp. 2372–2385, 2016. 1

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE TPAMI*, vol. 35, no. 11, pp. 2765–2781, 2013. 1, 2, 4, 11

[18] B. Nasihatkon and R. Hartley, "Graph connectivity in sparse subspace clustering," in *CVPR*, 2011. 1, 2

[19] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE TCYB*, no. 99, pp. 1–9, 2017. 1

[20] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014. 1, 3, 7, 8, 9, 10, 11

[21] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *CVPR*, 2013. 2, 8

[22] S. Gao, I. W. Tsang, and L. T. Chia, "Laplacian sparse coding, hypergraph laplacian sparse coding, and applications," *IEEE TPAMI*, vol. 35, no. 1, pp. 92–104, 2013. 2, 3

[23] S. Kim, D. Y. Chang, S. Nowozin, and P. Kohli, "Image segmentation usinghigher-order correlation clustering," *IEEE TPAMI*, vol. 36, no. 9, pp. 1761–1774, 2014. 2

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *ICCV*, 2015. 2

[25] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *IJCV*, 2018. 2

[26] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE TIP*, vol. 28, no. 1, pp. 265–278, 2019. 2

[27] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE TIP*, vol. 26, no. 6, pp. 2988–3001, 2017. 2

[28] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao, "Flexible multi-view dimensionality co-reduction," *IEEE TIP*, vol. 26, no. 2, pp. 648–659, 2017. 2

[29] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE TCSVT*, vol. 28, no. 10, pp. 2473–2483, 2018. 2

[30] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE TCYB*, vol. 48, no. 11, pp. 3171–3183, 2017. 2

[31] J. Liang, J. Yang, H.-Y. Lee, K. Wang, and M.-H. Yang, "Sub-GAN: An unsupervised generative model via subspaces," in *ECCV*, 2018. 2

[32] J. Yang, J. Liang, H. Shen, K. Wang, P. L. Rosin, and M.-H. Yang, "Dynamic match kernel with deep convolutional features for image retrieval," *IEEE TIP*, vol. 27, no. 11, pp. 5288–5302, 2018. 2

[33] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *CVPR*, 2016. 2, 3

[34] Y. Cheng, Y. Wang, M. Sznaier, and O. Camps, "Subspace clustering with priors via sparse quadratically constrained quadratic programming," in *CVPR*, 2016. 2

[35] Z. Li, L.-F. Cheong, S. Yang, and K.-C. Toh, "Simultaneous clustering and model selection: Algorithm, theory and applications," *IEEE TPAMI*, vol. 40, no. 8, pp. 1964–1978, 2018. 2, 3, 8, 9, 10, 11

[36] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Ordered subspace clustering with block-diagonal priors," *IEEE TCYB*, vol. 46, no. 12, pp. 3209–3219, 2016. 2

[37] C. G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *CVPR*, 2015. 2

[38] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *CVPR*, 2014. 2, 3, 8, 9, 10, 11

[39] C. You, C. G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *CVPR*, 2016. 2, 3, 4, 8, 10

[40] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Robust semi-supervised subspace clustering via non-negative low-rank representation," *IEEE TCYB*, vol. 46, no. 8, pp. 1828–1838, 2016. 2, 3

[41] M. Rahmani and G. Atia, "Innovation pursuit: A new approach to the subspace clustering problem," in *ICML*, 2017. 2

[42] E.-L. Dyer, A.-C. Sankaranarayanan, and R.-G. Baraniuk, "Greedy feature selection for subspace clustering," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013. 3

[43] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *NeurIPS*, 2014. 3, 5

[44] P. Purkait, T.-J. Chin, A. Sadri, and D. Suter, "Clustering with hypergraphs: the case for large hyperedges," *IEEE TPAMI*, vol. 39, no. 9, pp. 1697–1711, 2017. 3

[45] H. Liu, L. Latecki, and S. Yan, "Robust clustering as ensembles of affinity relations," in *NeurIPS*, 2010. 3, 8, 10

[46] C.-G. Li and R. Vidal, "A structured sparse plus structured low-rank framework for subspace clustering and completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 24, pp. 6557–6570, 2016. 3

[47] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for drill hole spectral data," *Journal of Applied Remote Sensing*, vol. 8, no. 1, p. 083644, 2014. 3

[48] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *CVPR*, 2014. 3, 4

[49] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Product Grassmann manifold representation and its LRR models," in *AAAI*, 2016. 3

[50] B. Liu, X.-T. Yuan, Y. Yu, Q. Liu, and D.-N. Metaxas, "Decentralized robust subspace clustering," in *AAAI*, 2016. 3

[51] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *CVPR*, 2015. 3

[52] 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 *ECCV*, 2012. 3, 4, 8, 9, 10

[53] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for LRR," *IEEE TPAMI*, vol. 38, no. 3, pp. 417–430, 2016. 3

[54] C. Xu, Z. Lin, and H. Zha, "A unified convex surrogate for the Schatten-p norm." in *AAAI*, 2017. 3

[55] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *NeurIPS*, 2013. 3

[56] H. Lai, Y. Pan, C. Lu, Y. Tang, and S. Yan, "Efficient k-Support matrix pursuit," in *ECCV*, 2014. 3

[57] E. Kim, M. Lee, and S. Oh, "Robust Elastic-Net subspace representation," *IEEE TIP*, vol. 25, no. 9, pp. 4245–4259, 2016. 3

[58] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by Trace Lasso," in *CVPR*, 2015. 3, 4, 8, 10

[59] J. Xu, K. Xu, K. Chen, and J. Ruan, "Reweighted sparse subspace clustering," *CVIU*, vol. 138, pp. 25–37, 2015. 3

[60] A. A. Abin, "Querying beneficial constraints before clustering using facility location analysis," *IEEE TCYB*, vol. 48, no. 1, pp. 312–323, 2018. 3

[61] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for hyperspectral data segmentation," in *International Conference on Digital Information Processing and Communications*, 2013. 3

[62] J. Wang, X. Wang, F. Tian, C. H. Liu, and H. Yu, "Constrained low-rank representation for robust subspace clustering," *IEEE TCYB*, vol. 47, no. 12, pp. 4534–4546, 2017. 3

[63] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE TCYB*, vol. 45, no. 5, pp. 1083–1094, 2015. 3

[64] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE TPAMI*, vol. 35, no. 1, pp. 171–184, 2013. 3, 4, 8, 9, 10, 11

[65] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *CVPR*, 2011. 3

[66] Z. Li, L. F. Cheong, and S. Z. Zhou, "SCAMS: Simultaneous clustering and model selection," in *CVPR*, 2014. 3

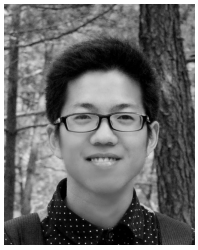
[67] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, 1996. 3

[68] T. Beier, F. A. Hamprecht, and J. H. Kappes, "Fusion moves for correlation clustering," in *CVPR*, 2015. 3

[69] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *CVPR*, 2016. 3

[70] P. Purkait, T. J. Chin, H. Ackermann, and D. Suter, "Clustering with hypergraphs: The case for large hyperedges," *IEEE TPAMI*, vol. 39, no. 9, pp. 1697–1711, 2017. 3

[71] X. Li, G. Cui, and Y. Dong, "Graph regularized non-negative low-rank matrix factorization for image clustering," *IEEE TCYB*, vol. 47, no. 11, pp. 3840–3853, 2017. 3
[72] B. Schölkopf, J. Platt, and T. Hofmann, "Learning with hypergraphs: Clustering, classification, and embedding," in *NeurIPS*, 2006. 3
[73] M. Lee, J. Lee, H. Lee, and N. Kwak, "Membership representation for detecting block-diagonal structure in low-rank or sparse subspace clustering." in *CVPR*, 2015. 4
[74] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NeurIPS*, 2002. 4, 8
[75] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *NeurIPS*, 2011. 5
[76] C. Li, J. Guo, and H. Zhang, "Learning bundle manifold by double neighborhood graphs," in *ACCV*, 2009. 5
[77] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. 8
[78] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE TPAMI*, vol. 23, no. 6, pp. 643–660, 2001. 8
[79] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," *Columbia Universty, Tech. Rep. CUCS-005-96*, 1996. 8
[80] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *CVPR*, 2007, pp. 1–8. 8, 11
[81] M. Li, X. Chen, X. Li, and B. Ma, "Clustering by compression," in *IEEE International Symposium on Information Theory*, vol. 51, no. 4, 2003, pp. 1523–1545. 8
[82] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *NeurIPS*, 2003. 11
[83] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *NeurIPS*, 2009. 11
[84] T. Kawamoto, M. Tsubaki, and T. Obuchi, "Mean-field theory of graph neural networks in graph partitioning," in *NeurIPS*, 2018. 11



Jie Liang is currently a Master student with the College of Computer Science, Nankai University. His current research interests include computer vision, machine learning, pattern recognition and optimization.

Jufeng Yang is an associate professor in the College of Computer Science, Nankai University. He received the PhD degree from Nankai University in 2009. From 2015 to 2016, he was working at the Vision and Learning Lab, University of California, Merced. His research falls in the field of computer vision, machine learning and multimedia.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests includes computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program, *etc*.

Paul L. Rosin is a professor at the School of Computer Science & Informatics, Cardiff University. His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, medical and biological image analysis, mesh processing, non-photorealistic rendering and the analysis of shape in art and architecture.

Liang Wang received the Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CAS), China, in 2004. He was a Research Assistant with the Imperial College London, U.K., and Monash University, Australia, and a Research Fellow with the University of Melbourne, Australia. He was a Lecturer with the Department of Computer Science, University of Bath, U.K. Currently, he is a Professor of the Hundred Talents Program of CAS with the Institute of Automation, CAS. His major research interests include machine learning, pattern recognition, computer vision, and data mining.