Subspace Clustering via Good Neighbors

Jufeng Yang, Jie Liang, Kai Wang, Paul L. Rosin, Ming-Hsuan Yang

Abstract—Finding the informative subspaces of high-dimensional datasets is at the core of numerous applications in computer vision, where spectral-based subspace clustering is arguably the most widely studied method due to its strong empirical performance. Such algorithms first compute the affinity matrix to construct a self-representation for each sample using other samples as a dictionary. Sparsity and connectivity of the self-representation play important roles in effective subspace clustering. However, simultaneous optimization of both factors is difficult due to their conflicting nature, and most existing methods are designed to address only one factor. In this paper, we propose a post-processing technique to optimize both sparsity and connectivity by finding good neighbors. Good neighbors induce key connections among samples within a subspace and not only have large affinity coefficients but are also strongly connected to each other. We reassign the coefficients of the good neighbors and eliminate other entries to generate a new coefficient matrix. We show that the few good neighbors can effectively recover the subspace, and the proposed post-processing step of finding good neighbors can be complementary to most existing subspace clustering algorithms. Experiments on five benchmark datasets show that the proposed algorithm performs favorably against the state-of-the-art methods with negligible additional computation cost.

Index Terms—Spectral-based subspace clustering, post-processing, good neighbors, sparsity, graph connectivity.

1 Introduction

Modeling high-dimensional data is a critical issue in computer vision [1]–[4]. High-dimensional data is usually distributed in multiple low-dimensional subspaces, and numerous subspace clustering methods [5] based on iterative optimization, algebraic operators, statistical analysis and spectral clustering [6] have been developed. This paper proposes a post-processing technique that refines the recent spectral-based subspace clustering (SBSC) methods.

Traditional SBSC algorithms aim to disentangle a specified number of low-dimensional subspaces of the data space. They first calculate the similarity among all instances and then conduct spectral clustering on the affinity graph [6]. The key step of SBSC methods is to compute the coefficient matrix $\boldsymbol{Z}$ by solving the following self-representation optimization problem:

$$\min_{\boldsymbol{Z}} \; L(\boldsymbol{X}\boldsymbol{Z}, \boldsymbol{X}) + \lambda \|\boldsymbol{Z}\|_{\xi} \quad \text{s.t. } \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0}. \tag{1}$$

where $\boldsymbol{X}$ denotes the dataset, and λ is the trade-off parameter. Here, the first term reconstructs each sample from other samples, whereas the second term introduces regularization of the coefficient matrix. Different ξ's (including ℓ_0, ℓ_1, ℓ_2, ℓ_∞ or the nuclear norm) lead to different properties of the reconstruction, such as sparsity (subspace-preserving) and connectivity [6]–[8]. Such a coefficient matrix can be interpreted as a "similarity matrix", where each entry reflects the similarity of two samples. Therefore, a block-diagonal structure [9] is expected so that the intra-subspace samples are densely connected in the affinity graph and the inter-subspace samples are disconnected.

However, it is hard to achieve both benefits within a single model. ℓ_2 or nuclear norm-based regularization leads to a dense coefficient matrix [10]. Although the connectivity within subspaces is guaranteed, the coefficients of the inter-subspaces are usually non-zero. Thus, the subspace-preserving property, *i.e.*, no connections among inter-subspace samples, is not satisfied [11]. In contrast, the ℓ_0 and ℓ_1 norms lead to a sparse reconstruction [10]. While a sparse construction enforces the subspace-preserving property of a data matrix, the connectivity within each subspace cannot be guaranteed [12]. To optimize both properties, several recent algorithms use a mixed norm for the regularization term in (1), *e.g.*, trace Lasso [11] and elastic net [10], to interpolate between the ℓ_1 and ℓ_2 norms adaptively. Nevertheless, these schemes do not perform consistently well on different applications.

To achieve a better trade-off between the sparsity and connectivity constraints than that in previous work, we propose a post-processing technique for the coefficient matrix derived from recent state-of-the-art methods [5], [10], [13]–[16]. We prune the weak connections in the self-representation system based on robust heuristics, which substantially increases the sparsity while preserving the connectivity. Peng *et al*. [17] show that representations based on linear projections with different regularizations (*e.g.*, ℓ_1, ℓ_2, ℓ_∞ and nuclear norms) preserve the property of intra-subspace projection dominance (IPD), *i.e.*, the coefficients for the intra-subspace points are larger than those for inter-subspace points. However, simply preserving larger coefficients of $\boldsymbol{Z}$ does not guarantee the connectivity of each subspace [12], [18]. Therefore, in this paper, we propose using the *good neighbors* with key connections to guarantee the latent connectivity with as few connections as possible. Here, a good neighbor is a neighbor (with high similarity) with several common neighbors inside the local neighborhood.

Accordingly, in this paper, we propose to find good neighbors based on the coefficient matrix derived from recent SBSC methods. Figure 1 shows the main steps of the proposed approach for finding good neighbors in the subspace clustering (FGNSC). We first generate the initial coefficient matrix by means of an off-the-shelf grouping algorithm that preserves the invariance of the projection. That is, when two data points are close, the computed similarity is strong. Next, we find good neighbors for each sample

- *J. Yang, J. Liang and K. Wang are with the College of Computer Science, Nankai University, Tianjin 300350, China. E-mail: yangjufeng@nankai.edu.cn; liang27jie@163.com; wangk@nankai.edu.cn*
- *Paul L. Rosin is with the School of Computer Science and Informatics, Cardiff University, Wales, UK. E-mail: Paul.Rosin@cs.cf.ac.uk*
- *M.-H. Yang is with the School of Engineering, University of California, Merced, CA 95344 USA. E-mail: mhyang@ucmerced.edu*

- *Code available: https://github.com/JLiangNKU/FGNSC*

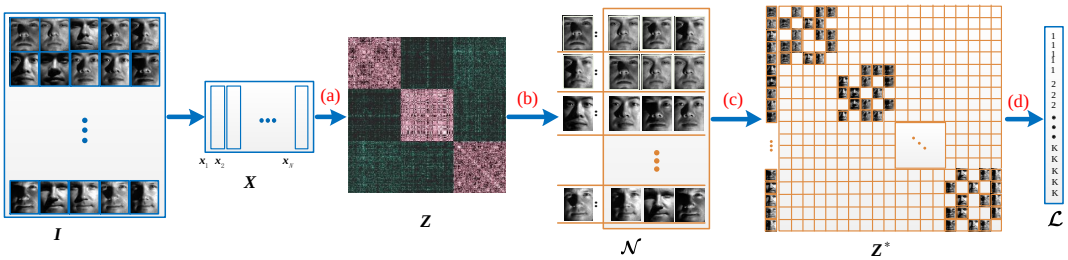


Fig. 1. Main steps of the proposed FGNSC algorithm. We first organize the data matrix $\boldsymbol{X}$ from input image dataset $\boldsymbol{I}$. Then, we compute the initial coefficient matrix $\boldsymbol{Z}$ from $\boldsymbol{X}$ according to (1). Using Algorithm 1 with $\boldsymbol{Z}$, we obtain the matrix of good neighbors $\mathcal{N}$. Then, $\boldsymbol{Z}^*$ is computed by assigning new coefficients to the good neighbors and eliminating the other values. Here, $\boldsymbol{Z}^*$ maintains both the sparsity and connectivity, as it preserves few non-zero elements with strong connections. Lastly, we apply the classic spectral clustering for the final segmentation result $\mathcal{L}$. (a) Calculating (1). (b) Finding good neighbors, as shown in Algorithm 1. (c) Generating $\boldsymbol{Z}^*$ by computing (7). (d) Conducting spectral clustering.

that induce the key connections within the subspace. Given the collection of good neighbors, we update the coefficient matrix by updating all the connections among good neighbors and pruning all other connections. In this manner, we obtain a new sparse and subspace-preserving coefficient matrix and theoretically prove that the newly constructed graph has satisfactory connectivity. Similar to the existing methods [6], [19], the spectral clustering algorithm [20] is used for segmentation in the final step.

The contributions of this paper are summarized as follows: 1) We find good neighbors for each sample and define a novel metric for evaluation. We theoretically prove the subspace-preserving property and the connectivity of the updated graph. 2) Based on the good neighbors, we propose FGNSC, which is a post-processing technique that can be integrated with recent subspace clustering methods. 3) Extensive experimental results on five benchmarks demonstrate that the FGNSC algorithm performs favorably compared to the state-of-the-art methods.

2 Related Work

Data clustering, especially SBSC, has been extensively studied in past decades [21]–[27]. The main characteristic among SBSC algorithms [28], [29] is the self-representation optimization, which learns the coefficient matrix with different properties, *i.e.*, sparsity and connectivity, based on different regularizations. While some schemes exploit sparsity by applying ℓ_1 or ℓ_0 minimization [30], [31] in (1), others produce strong connectivity with a dense coefficient matrix using ℓ_2 or nuclear regularization [7]. Closely related to this work are the methods that aim to bridge the gap between both properties [10], [11], [18] using mixed norms. Recently, various post-processing methods [17], [32] are proposed to further enhance the sparsity, but do not guarantee connectivity.

Sparsity. ℓ_0 and ℓ_1-based subspace clustering methods aim to compute a sparse and subspace-preserving representation for each data sample. Elhamifar and Vidal [5] propose the sparse representation by incorporating the ℓ_1 norm in (1). The generated coefficient matrix recovers the subspace-preserving property but may not satisfy graph connectivity if the dimension of the subspace is greater than three [12]. Yang et al. [8] 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 difficulties due to non-convexity, *e.g.*, orthogonal matching pursuit [33] and nearest subspace neighbor [31]. The scalable sparse subspace clustering by orthogonal matching pursuit (SSC-OMP) method [15] heuristically determines a given number of positions in the coefficient matrix that should be non-zero and then calculates the entry based on self-representation among subsets. However, this general pairwise relationship does not accurately reflect the sample correlation, especially for data pairs in the intersection of subspaces [34]. As a result, these positions may be incorrectly assigned, and the connectivity within each subspace cannot be guaranteed.

Connectivity. An excessively sparse coefficient matrix leads to unsatisfactory clustering results if the non-zero elements do not contain sufficient connections within each subspace [12]. Therefore, numerous methods have been developed for preserving more correlation information but with less sparsity [9], [35], [36].

Low-rank clustering methods [16], [37], [38] incorporate the nuclear norm in (1) to generate a block-diagonal solution with dense connections. However, the nuclear norm does not enforce subset selection well when noise exists, and the generated coefficient matrix is too dense to be an efficient representation. In [13], the least squares regression (LSR) method is used to model highly correlated data by minimizing the Frobenius norm. Furthermore, the smooth representation (SMR) [14] scheme focuses on the invariance of the projection from the data space to the representations. As the LSR and SMR methods lack sparsity, the dense coefficient matrix retains the connections of the inter-clusters, which affects the clustering results.

Bridging the Gap. Both sparsity and connectivity play important roles in spectral clustering. Therefore, several methods have been developed to address both issues in a single model [18], [39]–[41]. You *et al.* [10] have recently proposed to balance the subspace-preserving and connectivity properties using elastic net regularization. Both a theoretical justification and a geometric interpretation of the trade-off between subspace-preservation and connectivity are presented [10]. Similarly, the correlation adaptive subspace segmentation method proposed by Lu *et al.* [11] takes data correlation into account by using the mixed norm of trace Lasso, which adaptively interpolates between the ℓ_1 norm and the ℓ_2 norm of $\boldsymbol{Z}$. Nevertheless, the structure of the data correlations depends on the data matrix, and trace Lasso is not effective for model selection.

Post-Processing. The L2-graph method proposed by Peng *et al.* [17] preserves the given number of top values of each column in the coefficient matrix according to the intra-subspace projection dominance. Although the sparsity and subspace-preserving properties can be guaranteed, some samples may be assigned to other subspaces due to the lack of connections. Pan *et al.* [42] propose to robustify the shape interaction matrix (RSIM) using

a normalization operation based on singular vectors. Similarly, the reweighted sparse subspace clustering (RSSC) method [32] approximates the ℓ_0 norm minimization problem by using the ℓ_1 norm and iteratively reducing the penalty on large coefficients. Although the RSSC method improves the sparsity by relaxing the ℓ_0 norm, intra-subspace samples may not be connected in the affinity graph. In this paper, we preserve the connections over good neighbors, which not only retain high correlations but also induce key connections among samples within each subspace. The proposed algorithm optimizes the latent connectivity among each subspace while satisfying the subspace-preserving property.

3 Proposed Algorithm

Subspace clustering assumes the self-representation property, *i.e.*, each sample can be reconstructed by a linear combination of other points in the dataset [43]. As a result, $\boldsymbol{x}_i$ can be represented as $\boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{z}_i$, s.t., $z_{ii} = 0$, where the data matrix $\boldsymbol{X}$ is considered to be a self-representative dictionary, and $\boldsymbol{z}_i = [z_{1i}, z_{2i}, \cdots, z_{Ni}]$ contains the coefficients of the reconstruction. The goal of the self-representation system is to represent each sample using only the intra-subspace samples.

However, given an arbitrary subspace $\boldsymbol{\mathcal{S}}$, the number of samples ($n_{\mathcal{S}}$) is always larger than the intrinsic dimension ($d_{\mathcal{S}}$) of the subspace. Therefore, to reconstruct each sample, the model must select $d_{\mathcal{S}}$ samples from $n_{\mathcal{S}}$ candidates, which is not unique in general [43]. Meanwhile, the samples can be distributed near the intersection of multiple subspaces in real applications, which may introduce redundant or incorrect connections.

In this paper, we propose a post-processing technique to optimize both the sparsity (subspace-preserving property) and the connectivity of the self-representation system. Please find a summary of all the notations in the supplementary material.

Definition 1. (Subspace-Preserving Property) *The coefficient matrix $\boldsymbol{Z}$ derived from (1) satisfies the subspace-preserving property if and only if $z_{ij} = 0$ for all $\boldsymbol{x}_i \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{S}}$, where $\boldsymbol{\mathcal{S}}$ denotes an arbitrary subspace.*

Definition 2. *(**Connectivity**) A graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ is connected if we have:*

$$\forall \boldsymbol{v}_i, \boldsymbol{v}_j : \ \mathbf{1}_{\boldsymbol{v}_i \leftrightarrow \boldsymbol{v}_{i_1}} \times \prod_{i=1}^{l} \mathbf{1}_{\boldsymbol{v}_{i_l} \leftrightarrow \boldsymbol{v}_{i_{l+1}}} \times \mathbf{1}_{\boldsymbol{v}_{i_l} \leftrightarrow \boldsymbol{v}_j} = 1 \tag{2}$$

where $\boldsymbol{i} \in \mathbb{R}^{l-2}$ contains the indices of the samples that lie on the path between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, l denotes the length (number of samples) of the path, $\mathbf{1}$ is an indicator function and $\leftrightarrow$ indicates that the two vertexes are connected [44].

Note if we have $l = 2$, the graph $\mathcal{G}$ is fully connected. The connectivity property among a set of intra-subspace samples ensures a connected component in the affinity graph $\mathcal{G}$ [12].

Problem 1. (Pruning Erroneous Connections) *Assume that the coefficient matrix $\boldsymbol{Z}$ satisfies the intra-subspace projection dominance property [17]. The goal of the pruning process is to preserve the fewest connections in the affinity graph while satisfying both properties in Definition 1 and 2.*

3.1 Good Neighbors

Peng *et al.* [17] have proved the intra-subspace projection dominance property of the self-representation in (1) for $\xi = 1, 2, \infty$, *etc.*, which indicates that the coefficients over intra-subspace samples are likely to be larger than those over inter-subspace samples. However, $\boldsymbol{Z}$ may not be symmetric. Specifically, $\boldsymbol{x}_j$ may not necessarily choose $\boldsymbol{x}_i$ in its sparse representation, even if $\boldsymbol{x}_i$ is represented by a linear combination of points including $\boldsymbol{x}_j$ [43]. Therefore, we introduce the symmetric nonnegative affinity matrix $\boldsymbol{W}$ as

$$\boldsymbol{W} = \frac{1}{2}(|\boldsymbol{Z}| + |\boldsymbol{Z}|^\top), \tag{3}$$

to guarantee that the nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are symmetrically connected. Similarly to [17], [28], we first collect the samples with the top η coefficients in $\boldsymbol{w}_i$ for $\boldsymbol{x}_i$, where the parameter η is empirically determined from the experiments discussed in Section 4.2.

Definition 3. (η-neighbors) *For each sample $\boldsymbol{x}_i$, its η-neighbors (constructing the set $\boldsymbol{N}_\eta(\boldsymbol{x}_i) \in \mathbb{R}^{1\times\eta}$) are defined as follows:*

$$\boldsymbol{N}_\eta(\boldsymbol{x}_i) = \{\boldsymbol{x}_j\}_{j=1}^{\eta} = \arg\max_{\boldsymbol{x}_j} \sum_{j=1}^{\eta} |w_{ij}|. \tag{4}$$

The definition of η-neighbors is similar to the work [45], which preserves the max γ entries in $\boldsymbol{w}_i$ ($\gamma < \eta < n_{\mathcal{S}}$, where n is the number of samples in subspace $\boldsymbol{\mathcal{S}}$). Since a larger w_{ij} usually reflects a higher similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and $d \leq \gamma < n_{\mathcal{S}}$, such a pruning process guarantees the sparsity and subspace-preserving properties of the self-representation. However, it does not consider the connectivity property, where intra-subspace samples may not form a connected component in the affinity graph [10]. Simply considering the max γ edges for each sample may fail to handle the noise corrupted data where wrong connections are preserved since the max edges are sensitive to noise and outliers. Therefore, the model in [45] may over-segment the samples.

To achieve both the sparsity and connectivity properties, in this paper, we define the good neighbors of each sample that induce the key latent connections in a graph. For each sample, we preserve γ good neighbors from the η-neighbors ($\eta > \gamma$) rather than the γ largest entries of $\boldsymbol{w}_i$, where the connectivity within each subspace is enhanced (see Section 3.4).

Definition 4. (Good Neighbor) *Given the η-neighbors of $\boldsymbol{x}_i$,* i.e., *$\boldsymbol{N}_\eta(\boldsymbol{x}_i)$, $\boldsymbol{x}_j$ is a good neighbor of $\boldsymbol{x}_i$ if there are μ samples $\{\boldsymbol{x}_{j_l}\}_{l=1}^{\mu} \subset \boldsymbol{N}_\eta(\boldsymbol{x}_j)$ that satisfy:*

$$\prod_{l=1}^{\mu} \mathbf{1}_{\boldsymbol{x}_i \in \boldsymbol{N}_\eta(\boldsymbol{x}_{j_l})} = 1 \tag{5}$$

where $\mu \leq \eta$ and $\boldsymbol{i} \in \mathbb{R}^\mu$ is the set of indices.

Basically, a good neighbor is defined as a neighbor with at least μ common neighbors inside the local neighborhood. It is considered as a maximally sparse and connected neighborhood relationship where the sparsity (subspace-preserving property) and connectivity are analyzed in Section 3.3 and Section 3.4, respectively. The exploration of good neighbors provides complementary information when handling the noise corrupted data, which generates a more robust self-representation with both sparsity and connectivity properties.

In the remainder of this paper, we use $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{\gamma\times N}$ to determine the collection of good neighbors, where N is the number of data samples. γ and μ are used to control the sparsity and connectivity, respectively. If there are fewer than γ candidates that satisfy the condition in (5), we consider the candidates with the largest coefficients as the relaxed good neighbors.

Algorithm 1 : FGN

Input: $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{N\times N}$, γ, η, μ.

Compute the affinity matrix $\boldsymbol{W}$ by (3);
Initialize $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{\gamma\times N}$ with zeros;
for $i = 1 : N$ **do**
  Compute the η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i) = \{\boldsymbol{x}_{\boldsymbol{j}_l}\}_{l=1}^{\eta}$ by (4);
  $m = 1$;
  for $l = 1 : \eta$ **do**
    Compute s_{il} for $\boldsymbol{x}_{\boldsymbol{i}_l} \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ using (6);
    if $s_{il} > \mu$ & $m < \gamma$ **then**
      $\boldsymbol{\mathcal{N}}_i = \boldsymbol{\mathcal{N}}_i \cup \boldsymbol{x}_{\boldsymbol{i}_l}$;
      $m = m + 1$;
    end if
  end for
  if $m < \gamma$ **then**
    $\boldsymbol{\mathcal{N}}_i = \boldsymbol{\mathcal{N}}_i \cup \{\tilde{\boldsymbol{x}}\}_{\gamma-m}$ where $\{\tilde{\boldsymbol{x}}\}_{\gamma-m}$ have the largest $\gamma - m$ similarity in $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$.
  end if
end for

Output: Good neighbor matrix $\boldsymbol{\mathcal{N}}$.

Algorithm 2 : FGNSC

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

Generate $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$ via (1).
Compute the good neighbor matrix $\boldsymbol{\mathcal{N}}$ Using Algorithm 1.
for $i = 1 : N$ **do**
  Compute z^*_{ij} in $\boldsymbol{Z}^*$ using (7).
end for
Let $\boldsymbol{W}^* = \frac{1}{2}(|\boldsymbol{Z}^*| + |\boldsymbol{Z}^*|^\top)$ and compute the segmentation from $\boldsymbol{W}^*$ by spectral clustering [20].

Output: Labels of samples $\boldsymbol{\mathcal{L}} \in \mathbb{R}^N$.

3.2 Finding Good Neighbors for Subspace Clustering

Based on the good neighbor matrix $\boldsymbol{\mathcal{N}}$ derived from Definition 4, we propose the FGNSC method for clustering. The main steps for finding good neighbors (FGN) are summarized in Algorithm 1. Given the collection of N data samples $\{\boldsymbol{x}_i\}_{i=1}^N$ that lie in a union of K subspaces, the FGN method first computes the coefficient matrix $\boldsymbol{Z}$ via an off-the-shelf optimization function in (1). Then, a symmetric nonnegative affinity matrix $\boldsymbol{W}$ is obtained via (3).

For each sample $\boldsymbol{x}_i$, $i \in [1, 2, \cdots, N]$, we aim to find its γ good neighbors that satisfy Definition 4. We first generate its η-neighbors $\boldsymbol{N}_\eta(\boldsymbol{x}_i) \in \mathbb{R}^\eta$ by (4), where we set $\eta > \gamma$ for the pruning process. The samples in $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ are the candidates for the γ good neighbors of $\boldsymbol{x}_i$.

To determine whether $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ is a good neighbor of $\boldsymbol{x}_i$, we design a greedy scoring scheme by traversing $\boldsymbol{N}_\eta(\boldsymbol{x}_i)$ as a relaxation of (5), which is NP-hard. We instantiate the process by setting $\mu = 1$ and requiring that the path between $\boldsymbol{x}_i$ and one of its good neighbors must contain one different neighbor of $\boldsymbol{x}_i$. Given $\boldsymbol{x}_j \in \boldsymbol{N}_\eta(\boldsymbol{x}_i)$ and $\boldsymbol{N}_\eta(\boldsymbol{x}_j) = \{\boldsymbol{x}_{\boldsymbol{j}_m}\}_{m=1}^{\eta}$, we compute the connection score s_{ij} between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ using the following equation:

$$s_{ij} = \sum_{\mathrm{m}=1}^{\eta} \mathbf{1}_{\boldsymbol{x}_i \in \boldsymbol{N}_\eta(\boldsymbol{x}_{\boldsymbol{j}_m})}. \tag{6}$$

We consider $\boldsymbol{x}_j$ to be a good neighbor of $\boldsymbol{x}_i$ whenever $s_{ij} > \mu$, where μ is the required common neighbors for a good neighbor. Then, each row of $\boldsymbol{\mathcal{N}}$ is obtained by finding the first γ candidates from the corresponding η-neighbors that satisfy the aforementioned requirement.

Spectral clustering may wrongly focus on keeping the stronger connections in a graph [43], [46]. Therefore, based on the good neighbor matrix generated from Algorithm 1, we normalize the self-representation system where $0 \le z^*_{ij} < 1$ and the sum of each row equals 1. Specifically, we calculate a new coefficient matrix $\boldsymbol{Z}^*$ where each entry z^*_{ij} is computed by:

$$z^*_{ij} = \begin{cases} (w_{ij}) \,/\, \left(\sum_j w_{ij}\right), & \text{if } \boldsymbol{x}_j \in \boldsymbol{\mathcal{N}}_i; \\ 0, & \text{if } \boldsymbol{x}_j \notin \boldsymbol{\mathcal{N}}_i. \end{cases} \tag{7}$$

Here, w_{ij} is the $ij-$th entry of $\boldsymbol{W}$, and $\boldsymbol{W}$ is the affinity matrix computed by (3).

We then infer the subspace structure using the sparse representations in $\boldsymbol{Z}^*$. Specifically, we first compute the new affinity matrix by $\boldsymbol{W}^* = \frac{1}{2}(|\boldsymbol{Z}^*| + |\boldsymbol{Z}^*|^\top)$. Therefore, each node $\boldsymbol{x}_i$ connects itself to one of its good neighbors $\boldsymbol{x}_j \in \boldsymbol{\mathcal{N}}_i$ with weight $\frac{1}{2}(|z^*_{ij}| + |z^*_{ji}|)$. We construct a new affinity graph according to $\boldsymbol{W}^*$. Subsequently, the normalized cut method [20] is applied to the graph in a way similar to that in [9], [36], [47] to generate the final segmentation results. The main procedures of FGNSC are summarized in Figure 1 and Algorithm 2.

The proposed FGNSC algorithm is a general post-processing module that can be complementary to other spectral-based clustering algorithms, especially those developed based on dense coefficient matrices. The FGNSC algorithm transforms the coefficient matrix $\boldsymbol{Z}$ into a sparser matrix $\boldsymbol{Z}^*$ with subspace-preserving property, while simultaneously preserving the connectivity within each subspace based on the good neighbors of each sample. As a result, $\boldsymbol{Z}^*$ is more robust to noise, outliers or samples lying on the intersection between two subspaces, which simultaneously prunes erroneous connections and avoids over-segmentation.

3.3 Subspace-Preserving Property

The intrinsic requirement for the success of the SBSC methods is that the optimization process recovers a linear representation of each sample [43]. Specifically, the non-zero entries of the representation $\boldsymbol{z}_i$ should be related only to the intra-subspace samples of $\boldsymbol{x}_i$.

Definition 5. *(Intra-subspace projection dominance, IPD [17]) The IPD property of a coefficient matrix $\boldsymbol{Z}$ indicates that for all $\boldsymbol{x}_p, \boldsymbol{x}_q \in \boldsymbol{\mathcal{S}}$ and $\boldsymbol{x}_k \notin \boldsymbol{\mathcal{S}}$, we have $z_{pq} \geq z_{pk}$.*

In other words, the coefficients among intra-subspace samples are always larger than those among inter-cluster samples. Peng *et al.* [17] have proved the IPD property of the coefficient matrix $\boldsymbol{Z}$ derived by the ℓ_1, ℓ_2 or nuclear norm-based linear projections, where any noise or outliers are regarded as inter-subspace samples when representing $\boldsymbol{x}$. Based on their results, we have the following proposition with the same assumptions.

Proposition 1. *The $\boldsymbol{Z}^*$ derived from the proposed FGNSC satisfies the subspace-preserving property, as defined in Definition 1.*

Note that the subspace-preserving property holds for the set of η-neighbors which is a superset of the proposed good neighbors, please find the detailed proof in the supplementary material. As a result, the coefficient matrix $\boldsymbol{Z}^*$ derived by the proposed FGNSC algorithm satisfies the subspace-preserving property when combined with most existing linear representation schemes.

3.4 Connectivity

Conventional subspace clustering methods ensure the connectivity of each subspace by preserving compact connections in the reconstruction [10], [18], [48]. However, a dense graph may also contain incorrect connections between the inter-subspace samples. In this paper, we construct a connected graph satisfying Definition 2 for each subspace with the least edges, where each edge induces multiple latent connections among intra-subspace samples. In addition, the pruning of edges between samples where $\boldsymbol{x}_j \notin \boldsymbol{\mathcal{N}}_i$ also eliminates the erroneous connections among inter-subspace samples.

Note the proposed method achieves a trade-off between the sparsity and connectivity properties of the self-representation except for the following two extreme and rare cases, *i.e.*, 1) All the $N\times\gamma$ good neighbors of $\boldsymbol{\mathcal{N}}$ cannot be found; 2) The model requires $\max(\gamma,\mu) = n-1$ where all samples are compactly connected. For the first case where each sample cannot find any good neighbors, we choose an equal number of η-neighbors derived by (4) for relaxation. As a result, the model is relaxed to the sparsity-based one. For the second case, all entries of $\boldsymbol{Z}^*$ derived from (7) are non-zero. The model is thus relaxed to the approaches aiming to enhance the connectivity using dense connections.

To sum up, the new self-representation $\boldsymbol{Z}^*$ reflects a balance between the inter-subspace separation and the intra-subspace connectivity. Since the subsequent steps of spectral clustering segment the graph by correcting the erroneous connections in the affinity graph [21], [46], which relies on both properties, the proposed algorithm is expected to improve the clustering performance.

4 Experimental Results

4.1 Setup

Evaluated methods. We compare the proposed FGNSC algorithm with the state-of-the-art subspace clustering methods including SSC (with the ADMM solver) [5], [43], spatSC [35], L2-Graph [17], LSR [13], SMR [14], SSC-OMP [15], ORGEN [10], RSSC [32], NSN [31], iPursuit [28] and RSIM [42]. We tune the parameters for these methods to achieve the best results.

Datasets. We conduct the experiments on the extended Yale B (EYaleB [49]), COIL-20 [50], MNIST [51], USPS [52] and AR [53] datasets. The images are resized to $p \times q$ pixels to form the data vectors $\boldsymbol{x}_i \in \mathbb{R}^{pq}$, and concatenated together to form $\boldsymbol{X}$. Please refer to the supplementary material for more details.

Metrics. We evaluate all the methods using two widely used metrics in clustering: clustering accuracy (ACC) and normalized mutual information (NMI). In addition, we define the error rate $\mathcal{N}_e$ of the good neighbors. Let $\boldsymbol{g}_i$ be the set of intra-subspace samples of $\boldsymbol{x}_i$ in the ground-truth, and let $\boldsymbol{n}_{ij}$ be the j-th good neighbor of $\boldsymbol{x}_i$ ($j \in [1:\gamma]$, where γ denotes the number of good neighbors for each sample). We define $\mathcal{N}_e$ as

$$\mathcal{N}_e = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{\gamma} \mathbf{1}_{\boldsymbol{n}_{ij}\in\boldsymbol{g}_i}}{N\times\gamma}, \tag{8}$$

where $\mathbf{1}$ is an indicator function. A smaller value of $\mathcal{N}_e$ indicates better performance.

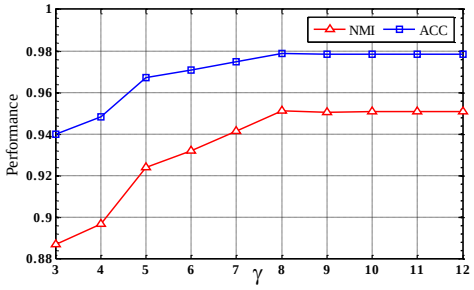


Fig. 2. Clustering performance with different γ on the extended Yale B dataset of 8 classes.

4.2 Parameters

For computational efficiency, we set $\mu = 1$ in the rest of this paper since the connections between samples become weaker as the path length increases, which means that $\boldsymbol{x}_j$ is considered to be a good neighbor of $\boldsymbol{x}_i$ if $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ have one common neighbor (or equally, $s_{ij} \geq 1$).

To optimize the connectivity of each subspace, we evaluate the effect of γ on a subset of 8 subjects from the extended Yale B dataset. For each experiment, we assign the value of $\gamma \in [3:12]$ and obtain the curves, as shown in Figure 2. For $\gamma = [1,2]$, ACC$= [0.3073, 0.8281]$ and NMI$= [0.1148, 0.7528]$, respectively, which we omit from the figure for clarity.

The curves in Figure 2 indicate that the FGNSC algorithm does not perform well with γ=$\{3,4\}$. When γ is increased, the FGNSC algorithm achieves better performance until reaching peak performance at γ=8. Therefore, we use γ=8 and μ=1 for the rest of the experiments. Accordingly, we set $\eta = 20$ for (4) in consideration of the balance between the computational efficiency and the effectiveness for selecting γ good neighbors.

4.3 Matrix of Good Neighbors

The coefficient matrices derived by the SSC-OMP [15] and iPursuit [28] methods are computed based on the correlation of neighboring data points in the data space. In this section, we analyze the effects of $\boldsymbol{\mathcal{N}}$ by comparing the SSC-OMP, iPursuit and FGNSC methods.

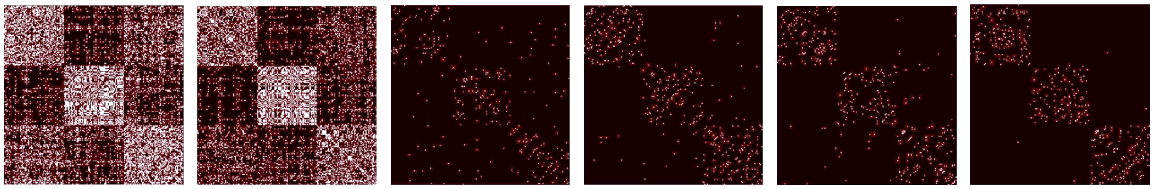
We select images of N_c subjects from the extended Yale B dataset with $N_c \in \{3,5,8,10,15,20,25,30,35,38\}$ and $\gamma = 8$ for all three methods. Table 1 shows the experimental results. Overall, the FGNSC algorithm achieves an accuracy of more than 94% on all subsets, with an error $\mathcal{N}_e$ of less than 9%. By contrast, SSC-OMP does not perform well. It achieves a clustering accuracy of less than 60% on the set of 38 subjects with an error rate of the neighbor matrix of 39.84%. The NMI scores show similar performance with other metrics.

4.4 Affinity Matrix

We evaluate six clustering methods in terms of the affinity matrix. Figure 3 shows the affinity matrices derived by the evaluated methods. We conduct the experiments on the first three digits of MNIST for a clear presentation. For each digit, we choose the first 300 images to construct the subset. Figure 3(f) shows the affinity matrix derived by the FGNSC algorithm. This affinity matrix is generated from the original coefficient matrix in Figure 3(b), which is computed by the SMR [14] method. Note that the affinity matrix in Figure 3(f) has better sparsity and the block-diagonal

TABLE 1
Clustering results by the SSC-OMP, iPursuit and FGNSC methods on subsets of the extended Yale B dataset using $\mathcal{N}_e$ and ACC. Each column shows the results on subsets with different numbers of subspaces. The FGNSC method achieves higher accuracy with lower $\mathcal{N}_e$ on all subsets.

Metrics	Methods	3	5	8	10	15	20	25	30	35	38
$\mathcal{N}_e \downarrow$	SSC-OMP [15]	20.09	25.73	23.44	31.23	30.46	37.12	38.74	37.32	41.30	39.84
	iPursuit [28]	9.10	10.63	10.70	10.42	11.04	11.76	11.80	12.10	12.52	12.82
	FGNSC	**3.29**	**4.89**	**6.04**	**6.61**	**7.28**	**7.65**	**8.03**	**8.21**	**8.44**	**8.52**
ACC $\uparrow$	SSC-OMP [15]	94.32	85.03	78.41	72.22	71.29	59.13	54.74	61.62	51.84	57.44
	iPursuit [28]	97.19	96.19	95.98	95.03	95.47	93.41	92.77	91.55	91.62	83.68
	FGNSC	**99.34**	**98.61**	**97.86**	**97.44**	**96.60**	**96.09**	**95.72**	**95.45**	**95.41**	**94.24**



(a) (b) (c) (d) (e) (f)

Fig. 3. Affinity matrix $\boldsymbol{W}$ derived by six methods on the first three digits of the MNIST dataset. (a) Affinity matrix by LSR [13] with an accuracy of 64.89%. (b) Affinity matrix by SMR [14] with an accuracy of 83.11%. (c) Affinity matrix by SSC-OMP [15], which is sufficiently sparse enough but does not have the block-diagonal property. (d) Affinity matrix by ORGEN [10] with an accuracy of 53.22% and an NMI of 0.4912. (e) A variant of (b) that simply preserves the top γ entries for each column, resulting in an accuracy of 81.33% and an NMI of 0.4835. (f) Affinity matrix by FGNSC with an accuracy of 98.11% and an NMI of 0.9140.

property, both of which are important for spectral clustering, leading to an accuracy of 98.11% and an NMI of 0.9140. In Figure 3(e), we preserve the top γ entries for each column of the affinity matrix in Figure 3(b) to compare the FGNSC algorithm with the L2-Graph in [17]. More erroneous connections are introduced in Figure 3(e) than in Figure 3(f) since several digits may lie at the intersection of the subspaces, and incorrect combinations for the representations may be generated.

4.5 Comparison to the State-of-the-Art Methods

Table 2 shows the run-time of the FGNSC algorithm and other methods using subsets from the extended Yale B dataset on a machine with a 2.93GHz CPU and 32 GB RAM. Table 3 shows the clustering results in terms of the average ACC and NMI. Overall, the proposed FGNSC consistently performs favorably compared to the state-of-the-art methods on five benchmarks.

On the extended Yale B dataset, the FGNSC algorithm performs well in terms of ACC and NMI. The ACC of the FGNSC algorithm is more than 94%, and the NMI is more than 0.9 on all subsets. The SMR method achieves an accuracy of 96% with $N_c = 8$, but the performance declines as the dataset increases in size. The main reason for the performance degradation is that the coefficient matrix $\boldsymbol{Z}$ derived by the SMR method has many redundant connections, so the spectral clustering performs poorly. The iPursuit method also achieves an NMI of more than 0.9 on all subsets; however, it requires more than 400 seconds on the whole dataset. The FGNSC algorithm takes 90 seconds to process the entire extended Yale B dataset with 38 classes, but most of the run-time is spent computing the initial $\boldsymbol{Z}$ matrix by the SMR method, which uses 75 seconds.

The FGNSC method performs well on the COIL-20 dataset, especially for $N_c = 5$, with an average accuracy of more than 99% and an NMI of more than 0.98. For the COIL-20 dataset, the intra-subspace samples are close (as a result of dense sampling with clean backgrounds), while the inter-subspace samples are distinct (as a result of diverse object classes). This structure of the dataset benefits the proposed post-processing technique.

The FGNSC algorithm performs well on the MNIST dataset, especially on the subset with three digits. However, the average accuracy decreases to 75% for the whole dataset when $N_c = 10$. This decrease can be attributed to the fact that some handwritten digits are similar (*e.g.*, 3 and 8; 1 and 7). We note that the SSC-OMP scheme performs worse, with the accuracy dropping from 97.64% ($N_c = 3$) to 53.30% ($N_c = 10$).

On the USPS dataset, the FGNSC algorithm exhibits a similar performance trend to that on the MNIST set. The ORGEN scheme performs better than the FGNSC method on the USPS dataset, mainly because the performance of the SMR method is poor in terms of the NMI (approximately 50% on all subset scales).

The accuracy of the ORGEN method is less than 26% on the AR dataset, whereas the FGNSC algorithm achieves an accuracy of more than 84%. The AR dataset is difficult to cluster because it contains real-world face images. While the background is simple, the face images are not aligned well. Most clustering algorithms based on sparsity, *e.g.*, SSC-OMP and SSC, do not perform well, as the connections between samples are not extracted properly. Overall, good neighbors can effectively be generated by the FGNSC algorithm because the subspace structure can be accurately reconstructed. This is also the main reason why the FGNSC method performs well on all the evaluated datasets.

4.6 Combination with Other Subspace Models

In this section, we combine the proposed FGN method with other subspace clustering methods, *i.e.*, LSR [13], OSC [19], spatSC [35] and LRR [16], rather than SMR [14]. In each experiment, we first compute the initialized coefficient matrix $\boldsymbol{Z}$ by each subspace scheme. The FGN method is then used to generate $\boldsymbol{Z}^*$. Finally, spectral clustering is performed on both $\boldsymbol{Z}$ and $\boldsymbol{Z}^*$. Table 4 shows the clustering results for all the evaluated methods.

The proposed post-processing module substantially enhances the performance of the evaluated subspace clustering algorithms with different representation terms. For instance, low rank representation (LRR) [16]) minimizes the nuclear norm $\|\boldsymbol{Z}\|_*$ to generate a dense coefficient matrix $\boldsymbol{Z}$, which results in an accuracy

TABLE 2
Run-time in seconds (s) on the subsets of the extended Yale B database. On the entire extended Yale B dataset with 38 subjects, the FGNSC algorithm takes 90 seconds, most of which is spent computing the initial coefficient matrix (75 seconds). The best and second best results are shown in bold and underline, respectively.

# of subjects	SSC [5]	spatSC [35]	L2-Graph [17]	LSR [13]	SMR [14]	SSC-OMP [15]	ORGEN [10]	RSSC [32]	NSN [31]	iPursuit [28]	RSIM [42]	FGNSC
8	38.30	0.20	**0.08**	0.98	0.64	0.36	0.99	6.92	0.45	12.87	0.79	0.70
15	118.50	0.84	**0.28**	2.02	3.28	1.06	1.92	17.44	1.29	26.51	5.12	3.71
30	658.52	4.90	**2.24**	9.08	30.61	3.17	4.69	83.29	7.93	256.59	56.23	37.45
38	1239.90	8.66	**5.76**	17.93	75.24	4.88	7.17	147.39	13.78	422.35	122.32	91.73

TABLE 3
Clustering results of the comparative methods on five datasets: extended Yale B [49], COIL-20 [50], MNIST [51], USPS [52] and AR [53], where each result is an average of fifty trails. The best and second best results are shown in bold and underline, respectively.

Dataset	Scale	Metric	SSC [5]	spatSC [35]	L2-Graph [17]	LSR [13]	SMR [14]	SSC-OMP [15]	ORGEN [10]	RSSC [32]	NSN [31]	iPursuit [28]	RSIM [42]	FGNSC
EYale B	8	ACC	59.61	20.84	27.53	73.46	96.43	78.41	66.78	73.42	89.43	95.98	96.03	**97.86**
		NMI	0.5518	0.1179	0.1927	0.6332	0.9270	0.5873	0.6333	0.6550	0.8013	0.9254	0.9318	**0.9511**
	15	ACC	48.46	19.62	20.55	58.88	92.15	71.29	58.61	62.98	84.56	95.47	91.93	**96.60**
		NMI	0.5594	0.2590	0.2127	0.5447	0.8885	0.6372	0.6166	0.6137	0.7767	0.9202	0.8869	**0.9284**
	30	ACC	36.71	18.02	16.73	58.62	89.48	61.62	54.53	58.31	78.66	91.55	86.32	**95.45**
		NMI	0.5211	0.3704	0.2603	0.5855	0.8774	0.6243	0.6291	0.6104	0.7705	0.9163	0.8236	**0.9186**
	38	ACC	32.52	20.23	20.81	58.26	87.91	57.44	53.66	57.81	77.55	83.68	79.29	**94.24**
		NMI	0.5466	0.3806	0.3692	0.5917	0.8559	0.6054	0.6388	0.6152	0.7711	0.9072	0.7734	**0.9084**
COIL-20	5	ACC	66.01	69.97	49.18	72.51	85.43	82.39	84.73	68.11	85.81	85.42	84.21	**99.23**
		NMI	0.6688	0.7296	0.4903	0.6639	0.7225	0.4996	0.8854	0.6323	0.8452	0.8378	0.8542	**0.9814**
	10	ACC	56.46	59.44	31.93	63.70	80.83	77.54	75.07	64.55	85.32	76.21	82.13	**96.13**
		NMI	0.6233	0.7293	0.4175	0.6402	0.6993	0.6632	0.8588	0.6555	0.8905	0.7883	0.8016	**0.9414**
	20	ACC	50.39	51.25	23.75	57.43	75.49	67.36	75.56	59.72	81.94	73.47	76.96	**90.07**
		NMI	0.7051	0.7082	0.3511	0.6296	0.5949	0.6552	0.8822	0.6563	0.8734	0.7788	0.7685	**0.8901**
MNIST	3	ACC	71.39	79.10	73.35	77.44	89.11	97.64	78.76	38.68	38.65	37.82	78.75	**97.96**
		NMI	0.5475	0.5260	0.5190	0.5275	0.6671	0.8961	0.6622	0.0097	0.0092	0.0066	0.7266	**0.9046**
	7	ACC	19.70	57.13	29.65	62.00	78.09	71.13	68.92	19.29	19.54	18.94	72.31	**85.88**
		NMI	0.5336	0.5045	0.2755	0.5326	0.5345	0.5647	0.6790	0.0125	0.0145	0.0117	0.6803	**0.7551**
	10	ACC	40.16	41.83	30.50	57.80	68.87	53.30	61.50	15.30	15.50	14.50	69.30	**75.40**
		NMI	0.4872	0.4631	0.2516	0.5197	0.5195	0.4823	0.6463	0.0212	0.0168	0.0138	**0.6642**	0.6629
USPS	3	ACC	65.81	84.49	48.05	81.05	85.64	92.74	89.86	57.50	58.62	57.95	73.42	**94.12**
		NMI	0.4435	0.6129	0.2401	0.6377	0.5164	0.7676	**0.8242**	0.2594	0.3253	0.2659	0.6892	0.7953
	7	ACC	18.60	50.57	47.14	62.58	76.90	59.01	65.06	31.09	31.17	30.23	63.07	**80.13**
		NMI	0.5295	0.4486	0.4785	0.5219	0.5160	0.3857	0.6625	0.3067	0.3002	0.2609	0.5438	**0.6927**
	10	ACC	32.34	44.73	37.37	60.40	68.83	47.33	63.57	23.15	24.40	23.30	59.02	**75.37**
		NMI	0.4814	0.4693	0.4233	0.5096	0.5136	0.3272	**0.6782**	0.2721	0.2628	0.2540	0.5548	0.6390
AR	30	ACC	32.91	32.92	47.14	76.21	81.26	30.37	25.25	68.09	23.12	66.88	77.75	**87.24**
		NMI	0.5241	0.5504	0.6029	0.7913	0.6021	0.3759	0.4313	0.7598	0.3696	0.7336	0.7042	**0.8566**
	60	ACC	25.58	32.70	27.35	75.19	80.63	25.21	23.09	71.63	21.78	67.44	72.57	**86.04**
		NMI	0.5162	0.5976	0.4373	0.8068	0.5602	0.4161	0.4825	0.8128	0.4321	0.7769	0.6843	**0.8601**
	90	ACC	23.87	30.41	23.43	72.53	80.39	23.52	21.81	71.56	20.29	67.46	68.04	**84.22**
		NMI	0.5367	0.6173	0.4240	0.7983	0.5090	0.4404	0.5036	0.8278	0.4668	0.7891	0.6330	**0.8522**
	100	ACC	25.54	29.19	25.58	74.15	79.85	23.12	20.88	76.31	20.08	65.88	62.94	**83.31**
		NMI	0.5393	0.6394	0.4920	0.8145	0.4755	0.4434	0.5054	**0.8554**	0.4756	0.7880	0.5875	0.8493

of less than 60% and an NMI of 0.6331 on the extended Yale B dataset with 38 subjects. When combined with the FGN module, this algorithm achieves an ACC of 82.85% and an NMI of 0.8891. As the proposed FGN method preserves only the key connections and eliminates noisy connections, it performs well with different representation schemes.

5 Conclusions

In this work, we propose a post-processing technique FGNSC manipulating the self-representations of subspace clustering to exploit both the sparsity and connectivity properties within each subspace. We find good neighbors for each sample by utilizing the correlation information contained in the initial affinity matrix rather than the input data space. The relationship of good neighbors requires not only direct connections derived by pairwise correlations but also latent connections induced by other samples on the connected path. We then reassign the coefficients of the selected good neighbors and eliminate other values such that the good neighbors have greater opportunity to be segmented into the same cluster. While recent methods focus on balancing the sparsity and connectivity via different norms, the proposed FGNSC simply refines their solution. Extensive experimental results demonstrate the effectiveness and efficiency of the proposed algorithm compared to state-of-the-art methods.

6 Acknowledgements

This work was supported by the NSFC (NO.61876094), 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] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE TPAMI*, vol. 41, no. 2, pp. 487–501, 2019. 1

TABLE 4
More results from combining FGN with other linear representation schemes on the extended Yale B dataset (*e.g.*, FGN-LSR is the combination of the FGN in Algorithm 1 and the linear representation module in LSR [13]). The best results are shown in boldface.

Method	Metric	8	15	30	38
LSR [13]	ACC	73.46	58.88	58.62	58.26
	NMI	0.6332	0.5447	0.5855	0.5917
FGN-LSR	ACC	**96.11**	**90.80**	**89.85**	**90.30**
	NMI	**0.9195**	**0.8778**	**0.8779**	**0.8824**
OSC [19]	ACC	24.53	20.16	16.60	15.34
	NMI	0.1315	0.1760	0.2211	0.2424
FGN-OSC	ACC	**91.19**	**80.00**	**75.66**	**75.12**
	NMI	**0.8549**	**0.7603**	**0.7399**	**0.7412**
spatSC [35]	ACC	20.84	19.62	18.02	20.23
	NMI	0.1179	0.2590	0.3704	0.3806
FGN-spatSC	ACC	**76.02**	**73.40**	**69.80**	**70.72**
	NMI	**0.7265**	**0.7544**	**0.7411**	**0.7512**
LRR [16]	ACC	68.10	61.70	61.69	59.54
	NMI	0.6078	0.6161	0.6410	0.6331
FGN-LRR	ACC	**93.77**	**85.80**	**82.97**	**82.85**
	NMI	**0.9152**	**0.8794**	**0.8839**	**0.8891**

[2] J. Liang, J. Yang, H.-Y. Lee, K. Wang, and M.-H. Yang, "Sub-GAN: An unsupervised generative model via subspaces," in *ECCV*, 2018. 1

[3] 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. 1

[4] 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. 1

[5] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009. 1, 2, 5, 7

[6] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011. 1, 2

[7] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE TPAMI*, vol. 38, no. 3, pp. 504–517, 2016. 1, 2

[8] Y. Yang, J. Feng, N. Jojic, J. Yang, and T.-S. Huang, "ℓ^0-sparse subspace clustering," in *ECCV*, 2016. 1, 2

[9] 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. 1, 2, 4

[10] 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. 1, 2, 3, 5, 6, 7

[11] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace Lasso," in *ICCV*, 2013. 1, 2

[12] B. Nasihatkon and R. Hartley, "Graph connectivity in sparse subspace clustering," in *CVPR*, 2011. 1, 2, 3

[13] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *ECCV*, 2012. 1, 2, 5, 6, 7, 8

[14] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *CVPR*, 2014. 1, 2, 5, 6, 7

[15] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *CVPR*, 2016. 1, 2, 5, 6, 7

[16] 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. 1, 2, 6, 8

[17] 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, 2016. 1, 2, 3, 4, 5, 6, 7

[18] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," in *NIPS*, 2013. 1, 2, 5

[19] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *CVPR*, 2014. 2, 6, 8

[20] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000. 2, 4

[21] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE TPAMI*, vol. 27, no. 12, pp. 1945–1959, 2005. 2, 5

[22] Z. Li, S. Yang, L.-F. Cheong, and K.-C. Toh, "Simultaneous clustering and model selection for tensor affinities," in *CVPR*, 2016. 2

[23] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *ECCV*, 2018. 2

[24] 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

[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," *International Journal of Computer Vision*, 2018. 2

[26] 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

[27] 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. 2

[28] M. Rahmani and G. Atia, "Innovation pursuit: A new approach to the subspace clustering problem," in *ICML*, 2017. 2, 3, 5, 6, 7

[29] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *CVPR*, 2017. 2

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

[31] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *NIPS*, 2014. 2, 5, 7

[32] J. Xu, K. Xu, K. Chen, and J. Ruan, "Reweighted sparse subspace clustering," *CVIU*, vol. 138, pp. 25–37, 2015. 2, 3, 5, 7

[33] E.-L. Dyer, A.-C. Sankaranarayanan, and R.-G. Baraniuk, "Greedy feature selection for subspace clustering," *JMLR*, vol. 14, no. 1, pp. 2487–2517, 2013. 2

[34] 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. 2

[35] 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. 2, 5, 6, 7, 8

[36] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *CVPR*, 2014. 2, 4

[37] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *CVPR*, 2015. 2

[38] 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. 2

[39] H. Lai, Y. Pan, C. Lu, Y. Tang, and S. Yan, "Efficient k-Support matrix pursuit," in *ECCV*, 2014. 2

[40] 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. 2

[41] E. Kim, M. Lee, and S. Oh, "Robust Elastic-Net subspace representation," *IEEE TIP*, vol. 25, no. 9, pp. 4245–4259, 2016. 2

[42] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data," in *ICCV*, 2015. 2, 5, 7

[43] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE TPAMI*, vol. 35, no. 11, pp. 2765–2781, 2013. 3, 4, 5

[44] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972. 3

[45] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *AAAI*, 2015. 3

[46] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. 4, 5

[47] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *CVPR*, 2017. 4

[48] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014. 5

[49] 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. 5, 7

[50] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (COIL-20)," *Technical Report CUCS-005-96*, 1996. 5, 7

[51] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 5, 7

[52] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer Series in Statistics New York, 2001. 5, 7

[53] A.-M. Martinez, "The AR face database," *CVC Technical Report*, 1998. 5, 7