PAPER

Energy Minimization over *m*-Branched Enumeration for Generalized Linear Subspace Clustering

Chao ZHANG[†a)], ***Member***

SUMMARY In this paper, we consider the clustering problem of independent general subspaces. That is, with given data points lay near or on the union of independent low-dimensional linear subspaces, we aim to recover the subspaces and assign the corresponding label to each data point. To settle this problem, we take advantages of both greedy strategy and energy minimization strategy to propose a simple yet effective algorithm based on the assumption that an *m*-branched (i.e., perfect *m*-ary) tree which is constructed by collecting *m*-nearest neighbor points in each node has a high probability of containing the near-exact subspace. Specifically, at first, subspace candidates are enumerated by multiple *m*-branched trees. Each tree starts with a data point and grows by collecting nearest neighbors in the breadth-first search order. Then, subspace proposals are further selected from the enumeration to initialize the energy minimization algorithm. Eventually, both the proposals and the labeling result are finalized by iterative re-estimation and labeling. Experiments with both synthetic and real-world data show that the proposed method can outperform state-of-the-art methods and is practical in real application.

key words: *general subspace clustering, energy minimization, multi-branch enumeration*

1. Introduction

Linear subspace clustering is a classic and important problem widely studied in computer vision [1], [2], computer graphics [3], [4], and data mining [5], [6] communities. Considering data points/features $Q = \{q_i\}$ drawn from an unknown union of linear subspaces $\mathcal{S} = \{S_j\}$, the problem can be described in the form of matrix factorization as

$$Q = BX^T, \tag{1}$$

where the columns of B constitute a basis for the column space of Q. The matrix X contains coefficients for forming the columns of Q from the basis (a low-dimensional representation). Based on the assumption that Q lie on or near an unknown union of subspaces, it can be partitioned into clusters,

$$Q = [Q_1, Q_2, \dots, Q_W]. \tag{2}$$

Our purpose is to represent given Q in the union-of-subspace form by recovering each cluster which can be explained by a low-dimensional subspace,

$$Q = [S_1X_1^T, S_2X_2^T, \dots, S_WX_W^T], \tag{3}$$

and the models of the subspaces are unknown (i.e., subspace parameters are not given), which requires estimation of general subspaces. Posing assumptions on the subspace can improve the clustering result if the assumptions are correct. However, in practice, the data points could be drawn from multiple sources, and the latent subspaces of the data points can be different. For example, 1D subspaces with data sets in different number of dimensions can have different geometric parameters.

Among sizable works, many state-of-the-art methods belong to either of the following two categories: *nearest neighbor (NN) based methods* and *energy based methods*. Regarding the former category, methods find the nearest neighbor point to expand the current subspace, and select output subspaces typically by maximizing the number of inliers within some fixed threshold [7]. For the latter category, if the models of the subspaces can be parametrically represented, clustering can be conducted by optimizing the subspace parameters with a global energy function to describe the quality of each candidate solution [8]. Nevertheless, these techniques still suffer from many challenges: (1) Both the data segmentation and the subspace parameters are unknown in many practical situations. (2) The distribution of data within each subspace is generally unknown. (3) When subspaces are close to each other, the problem becomes harder because there are points which can belong to either of the subspaces. To alleviate these difficulties, both NN or energy-based methods usually follow a two-step strategy. In the first step, subspace proposals are randomly or greedily enumerated and evaluated. In the second step, subspace proposals and labeling results are iteratively updated by either inlier maximization or energy minimization. Such a strategy can be conducted in different ways with different limitations, as concluded as follows:

- Limitation of exhaustive methods: Assuming point set Q_A among Q lay on a certain d-dimensional subspace $\mathcal{S}$, there exist $C_{|Q|-1}^{d-1}$ candidate subspaces through each point $q_i \in Q$. Investigating all the candidates exhaustively can be very expensive and impractical.
- Limitation of greedy methods: Expanding each $q_i \in Q$ to a subspace iteratively by collecting the neighbor with maximum projection distance can result in $|Q|$ subspace candidates. However, greedy search, which makes a locally optimal choice, will easily be misled by points of other subspaces, especially when the points of two subspaces are very close.

Manuscript received May 23, 2019.
Manuscript publicized September 19, 2019.

[†]The author is with the University of Fukui, Fukui-shi, 910–8507 Japan.
a) E-mail: zhang@u-fukui.ac.jp

DOI: 10.1587/transinf.2019EDP7138

- Limitation of energy based methods: Unlike the exhaustive or greedy methods which select proposals from a large number of candidates, the accuracy of energy based methods largely depends on the initial random subspace proposals. Also, optimizing model parameters by minimizing the energy, which includes the sum of geometric errors, is applicable in affine or homography models but difficult in general cases.

In this paper, as the main technical contribution, we take advantages of both greedy methods and energy-based methods to propose a three-step method. First, subspace candidates are greedily enumerated by m-branched trees through each data point, which is likely to include appropriate subspace proposals. Secondly, subspace proposals are estimated/reestimated from the candidates by maximizing the number of inliers. Thirdly, each data point is assigned with a label (i.e., each proposal can be treated as a label) to calculate the value of a pre-designed energy function. The optimal labeling result at the current iteration can be achieved by minimizing the energy function. The second and the third steps can be iteratively run until the energy value stops decreasing.

2. Related Work

Motivated by the explosion of multi-source data in real-world applications, various algorithms have been proposed in the last decades. Many recent works focus to settle the following two challenges due to the attribute of the data: (1) clustering with prohibitively high-dimensional data [5]. With the increase of data dimension, the distance metrics between data points become less discriminative, and it is a common phenomenon that all the data are nearly equidistant from each other when the dimension is prohibitively large. Moreover, as the segmentation of the data is unknown, dimension reduction and feature selection methods can not perform well. (2) clustering with noisy [9] or missing data [10]. Nuisances, including noise, outliers, and missing entries can confuse the algorithms by involving irrelevant or excluding relevant attributes, which will further require the robustness of distance metrics and possibly lead to a different clustering result. In this paper, rather than the above two challenges, we focus on the challenge caused by the general structure of linear subspace (i.e., the geometric structure of the clusters is not specified). Especially, when two subspaces are very close, or subspaces include outliers, the clustering problem becomes harder because both the structure of subspaces and the distribution of data points is unknown, the assumption that a point and its nearest neighbors likely belong to the same subspace no longer holds for some data points.

Regarding linear subspace clustering, matrix factorization based algorithms have been extensively applied to noise-free or noiseless data. Given a data matrix M, one can find the potential subspaces by equivalently estimating a permutation of M's approximation X, which is a multiplication of two matrices. One of them has columns as orthogonal basis and the other one is consist of coefficients to approximate M from basis. Classic algorithms, take Costeira and Kanade algorithm [11] as an example, find X using singular value decomposition (SVD). In a more recent work [12], a global constraint of the relationship between subspaces is exploited, which restricts multiple lower-dimensional subspaces to lay in a higher-dimensional subspace by further factorizing the data points in each subspace with a shared global matrix. GPCA [13], [14] methods can deal both independent and dependent linear subspaces by clustering normals of data points. Derivating multiple polynomials can calculate normals for each data point. The combination of polynomials fit the union of subspaces. Instead of linear dependent subspaces, we aim to cluster general independent subspaces.

There are also several works that are capable of dealing with general subspace structure. Sparse Subspace Clustering (SSC) [15], which is one of the spectral clustering-based methods, proposes that an arbitrary data point in the data set can be represented by the combination of any other points. Finding which subspace the point belongs corresponds to finding a sparse combination, which has a minimum number of nonzero coefficients. As SSC does no rely on finding neighbor points with certain distance metrics, it can deal with data nuisances well. The drawback is, since the sparsest combination is obtained by solving a sparse optimization problem, the solution becomes less confident when the number of points per dimension is small. Both the advantage and drawback can be observed in the experiment of this paper. SSC Orthogonal Matching Pursuit (SSC-OMP) [16] proposes to replace the optimization procedure in SSC [15] by OMP algorithm, which greedily and incrementally select the dictionary elements that are most likely to be correct at current. Low-Rank Representation (LRR) [17] solves the problem in a similar way of SSC, which finds a low-rank representation instead of sparse representation by nuclear norm minimization. Furthermore, Low-Rank and Sparse Subspace Clustering (LRSSC) [18] combines both L1 norm in [15] and the nuclear norm in [18]. Thresholding based Subspace Clustering (TSC) [19] is also based on spectral clustering, which constructs the adjacency matrix based on calculating inner products between data points. Nearest Subspace Neighbor (NSN) [7] constructs a neighbor matrix at first and followed by clustering step with Greedy Subspace Recovery (GSR) or spectral clustering. For each point, neighbors are sequentially collected by finding the largest norm of the projection onto the subspace spanned by the correct neighbors collected so far. NSN+GSR guarantees exact clustering performance, especially in noise-less conditions. However, when the data set involves data nuisances, the performance decreases because the algorithm may fail in collecting correct neighbors in the early stage, and lead to wrong subspace estimation. The procedure of our subspace enumeration is inspired by [7].

Subspace clustering can be recast as a geometric model fitting problem if the model of the subspaces is given.

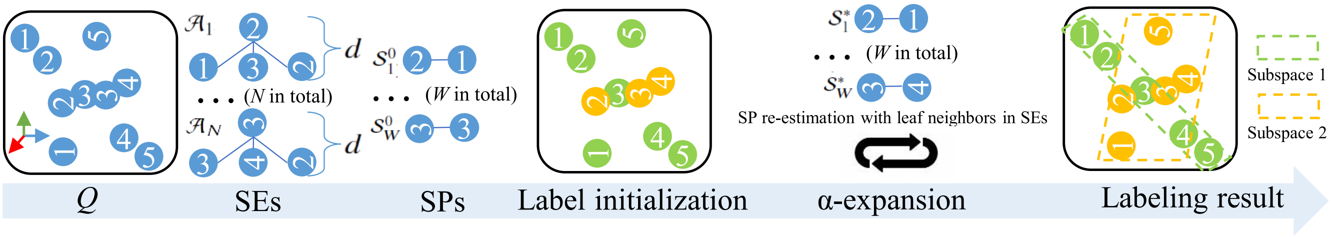


Fig. 1 Overview of the proposed method. The right arrow indicates the execution order of each step. Our method starts with a given data set Q of data points lay on or near different subspaces (the points are in 3D and the subspace is 2D here as an example). Upright and rotated numbers indicate the points on/near two different subspaces respectively.

RANSAC [20] and its variants [21], [22] belongs to a well-known category which aims to estimate the parameters of the model that with the largest number of inliers (i.e., maximum consensus). It performs well when the data only follow one model. In the multi-model/multi-subspace case, [8] proposes PEARL based on α-expansion [23], which formulates the clustering task as a optimal labeling problem, and the optimal solution can be achieved by minimizing a global energy function. Random cluster models (RCM) [24] is proposed to generate hypotheses based on larger-than-minimal subsets to improve the quality of the hypotheses and construct a graph cuts based method for labeling. The procedure of our energy minimization is based on [8], which finds optimal labeling with enumerated subspace proposals instead of randomly initialized proposals.

Table 1 Notation and description.

NOTATION	DESCRIPTION
m	Parameter for constructing m-branched tree.
N	Number of total data points/SEs.
n	Number of data points per subspace.
Q	Input data set.
q_i	i th data point in Q.
$\mathcal{A}_i$	SE with respect to q_i.
p	Ambient dimension.
d	Subspace dimension (same as the depth of each SE).
a_i	Subspace spanned by i th root-to-leaf path.
T	Number of enumerated subspaces in one SE.
$\mathcal{S}^0, \mathcal{S}^1, \ldots, \mathcal{S}^*$	Estimated union of subspaces after each iteration. $\mathcal{S}^0$ is the initial and $\mathcal{S}^*$ is the final result.
$\mathcal{S}$	A single general subspace.
W	Number of estimated subspaces.
$\mathcal{L}^0, \mathcal{L}^1, \ldots, \mathcal{L}^*$	Labeling result after each iteration. $L_i = \mathcal{S}$: assign q_i to $\mathcal{S}$. $L_i == \mathcal{S}$: is true if $\mathcal{S}$ is assigned to q_i.

3. Our Approach

3.1 Overview and Notation

For easy understanding, we first illustrate the whole procedure of our method in Fig. 1 as an overall description, then the detailed explanation for each step will be given in the following subsections. At first, given a set of N points $Q = \{q_1, \ldots q_i, \ldots q_N\}, q_i \in \mathbb{R}^p$ ($p = 3$ in the illustration), points lay on or near a union of subspaces (two 2D subspaces in the illustration). For each q_i, a subspace enumeration (SE) $\mathcal{A}_i$ is generated, which is an m-branched tree with each root-to-leaf path being a subspace candidate, will be explained in Sect. 3.2. Secondly, subspace proposals (SPs) $\mathcal{S}^0 = \{\mathcal{S}_1^0, \ldots, \mathcal{S}_W^0\}$ are selected from SEs for labeling, where the superscript indicates the iteration number. The selection procedure will be introduced in Sect. 3.3. Thirdly, each q_i is assigned with a label from SPs to maximize the number of inliers according to a threshold, and the labeling result is the initial input to the energy minimization part (α-expansion). Then, the labeling result is iteratively updated by α-expansion followed by a SP re-estimation procedure, which will be introduced in Sects. 3.4 and 3.5. Throughout the paper, we denote the set of N indices by $[N] = \{1, 2, \ldots, N\}$, and span$\{\cdot\}$ is to denote a subspace spanned by a set of vectors/points. For example, span$\{v_1, \ldots, v_N\} = \{v : v = \sum_{i=1}^{N} \alpha_i v_i, \alpha_1, \ldots, \alpha_N \in \mathbb{R}\}$. $\text{Proj}_{\mathcal{U}} y$ denotes the projection of y onto the subspace $\mathcal{U}$ (i.e., $\text{Proj}_u y = \arg\min_{u \in \mathcal{U}} \|y - u\|_2$). $\mathbb{I}\{\cdot\}$ is an indicator function to turn the true/false into 1/0. All the important notations are summarized in Table 1 for clarity.

3.2 Subspace Enumeration

Greedy method [7] collects neighbors sequentially at each data point by maximizing the number of inliers, assuming that the current collected points span a correct subspace. This assumption can easily fail when the data set incorporates outliers or the data points are close with each other, in which case the nearest neighbor is not necessarily on the same subspace. On the other hand, spanning subspace from the points in the data set can result in a nearly infinite number of combinations, and the probability of "clean" subsets of points decreases exponentially with the increase in the size of the subsets [25]. To alleviate the influence brought by the outliers, we enumerate subspace candidates in a tree structure, to keep the top-m points at each stage of local optimum selection, instead of selecting only the top point as the optimal choice. As shown in Alg. 1, given a data point q_t, T subspace candidates through q_t, denoted by $\mathcal{A}_t$ will be enumerated. In *Line* 1, all the vectors (points) are normalized to unit vectors for simpler projection calculation. As the linear subspaces are closed under scalar multiples, this will not change the estimation result. A perfect m-ary tree

Algorithm 1 Subspace Enumeration (SE)

```
Input: A data point $q_t$, data point set $Q = \{q_1, \dots q_N\}$, subspace dimension
    $d$, branch number $m$
Output: An SE with respect to $q_t$: $\mathcal{A}_t = \{a_1, \dots, a_T\}, T = m^{d-1}, a_t \in \mathbb{R}^d$
 1: $q_i \leftarrow q_i / \|q_i\|_2, \forall i \in [N]$                // Normalize magnitudes
 2: Build an empty perfect $m$-ary tree with depth $d$
 3: Root node $\leftarrow t$
 4: for each internal node in depth-first order do
 5:     $\mathcal{U} \leftarrow \mathrm{span}\{q_k | k \in \mathcal{P}\}$
 6:     // $\mathcal{P}$ includes nodes on the path from the root to the current node
 7:     $\mathcal{I} \leftarrow \mathcal{P}$
 8:     for $i = 1, \dots, m$ do
 9:         $\hat{k} = \arg\max_{k \in [N] \setminus \mathcal{I}} \mathrm{Proj}_{\mathcal{U}} q_k$
10:                                    // $q_{\hat{k}}$ is the closest point to $\mathcal{U}$
11:         Assign data index $\hat{k}$ to $i$ th child of current node
12:         $\mathcal{I} \leftarrow \mathcal{P} \cup \hat{k}$
13:     end for
14: end for
15: for each leaf node $a_j$ from left to right do
16:     $a_j \leftarrow \mathrm{span}\{q_k | k \in \mathcal{P}\}$
17:                         // Points with indices in each leaf node span an ES
18: end for
```

is then built with each node to restore indices of the data points. The root node is initialized by t and in each root-to-leaf path, the neighbor points are sequentially collected in a greedy fashion. In *Line* 5, points in root-to-current node path span a subspace $\mathcal{U}$, and the point which is closest to $\mathcal{U}$ is newly collected in *Line* 9. The $\mathcal{U}$ in each node is represented by a matrix, with whose columns are the orthonormal basis calculated by Gram-Schmidt orthonormalization. Therefore, $\mathrm{Proj}_{\mathcal{U}} q_k$ can be calculated by $\|\mathcal{U}^\top q_k\|_2$. In *Line* 16, over each root-to-leaf path, a subspace candidate is spanned by all the points on the path for further subspace proposal and re-estimation.

3.3 Subspace Proposal

After generating subspace enumerations through each data point, we are ready to select subspace proposals. Although the number of subspace candidates has been limited to a certain scale after enumeration, it is still impractical to test them all. To select high-quality candidates as proposals via a fixed selection criteria, the number of points which are close to each subspace are counted. It is natural because many points should lay on or near a true subspace. In *Line* 2 of Alg. 2, points with their projections larger than a certain threshold, which mean inlier points, are counted to select the best subspace candidate $\hat{A}_i$ from each $\mathcal{A}_i$. From *Line* 7, subspace proposals are further selected from N subspace candidates. *Line* 8 is to find the index of the best subspace candidate which can maximize the number of inliers among the N subspace candidates achieved by running *Line* 1 ~ *Line* 4. In *Line* 10, note that the points for evaluating a certain subspace can only be used once. That is, if the points have already assigned to previous subspaces, they can no longer be used for evaluating other subspaces. The output of Alg. 2 is used for labeling in Sect. 3.4, and can also be updated by Alg. 3. As pointed out by [24] that the quality of proposals is central to clustering, which will affect not only the convergence speed but also the clustering accuracy. Although it has been claimed in [8] that the PEARL algorithm may converge to good subspaces even from a small set of rough proposals, selected "guesses" rather than random "guesses" may yield in better results.

Algorithm 2 Subspace Proposal (SP)

```
Input: SE set $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_N\}$, data point set $Q = \{q_1, \dots q_N\}$
Output: Union of subspace proposals $\mathcal{S}^0 = \{\mathcal{S}^0_1, \dots, \mathcal{S}^0_W\}, W < N$
 1: for $i = 1, \dots N$ do
 2:     $\hat{A}_i = \arg\max_{a_j \in \mathcal{A}_i} \sum_{t=1}^N \mathbb{I}\{\mathrm{Proj}_{a_j} q_t \geq 1 - \epsilon\}$
 3:     // Select the ES $\hat{A}_i$ from each $\mathcal{A}_i$ which maximizes the number of
        inliers
 4: end for
 5: $\mathcal{I} \leftarrow [N]$
 6: $w \leftarrow 1$
 7: while $\mathcal{I} \neq \emptyset$ do
 8:     $\hat{i} = \arg\max_{i \in \mathcal{I}} \sum_{t=1}^N \mathbb{I}\{\mathrm{Proj}_{\hat{A}_i} q_t \geq 1 - \epsilon\}$
 9:     $\mathcal{S}^0_w = \hat{A}_{\hat{i}}$   // Select $W$ subspaces as proposals (i.e., input of Alg. 3)
10:     $\mathcal{I} \leftarrow \mathcal{I} \setminus \{t | \mathrm{Proj}_{\mathcal{S}^0_w} q_t \geq 1 - \epsilon\}$
11:                                // Ensure the proposals distribute sparsely
12:     $w \leftarrow w + 1$
13: end while
```

3.4 Label Assignment

We utilize PEARL [8] for labeling by treating each SP as a label. The main difference is, in [8], initial proposals are generated by randomly sampling points following a geometric model, while our SPs are generated by Alg. 1 and 2 without any prior knowledge. The set of SPs $\mathcal{S}^0$ is treated as a set of current labels, and our task is to assign SPs (labels) to each data point, i.e., labeling. The optimal labeling result is achieved by minimizing the following energy function,

$$\begin{aligned} \mathrm{E}(\mathcal{L}) = & \sum_{i \in [N]} \|q_i - L_i(L_i^\top q_i)\|^2 \\ & + \lambda \sum_{(q_i, q_j) \in \mathcal{N}} \exp\left(-\frac{\|q_i - q_j\|^2}{\delta^2}\right) \mathbb{I}\{L_i \neq L_j\}, \end{aligned} \tag{4}$$

where $\mathcal{L} = \{L_t | t \in [N]\}$ is the labeling result. Each L_t is assigned with a subspace. The first *data term* measures the geometric error over all the points. The squared perpendicular distance between a point and its label (subspace) is used. Data term gets smaller when appropriate labels are assigned to points. The second term is the *smooth term* which measures the smoothness of points locally (i.e., the labels of points are assumed to be the same with a high probability for each pair of neighboring points). Here, we follow the suggestion of [8] to adopt Potts model [23]: increase the smooth term when a point and its neighbor are not assigned with the same label. The penalty is exponential, with a constant parameter σ to control the smoothness of the exp function. The minimization is conducted by running α-expansion [23] until the energy function stops decreasing, which can be intuitively understood as a procedure to balance between the number of inliers and smoothness. Of course, the labeling result relies largely on the quality of the SPs. For better labeling result, we use currently estimated inliers to update the SPS and rerun the label assignment algorithm repeatedly until the labeling result converges.

3.5 Subspace Re-Estimation

Once the labeling result is calculated with current SPs, the SPs can be re-estimated for running a new round of label assignment. We combine label assignment to explain the re-estimation in Alg. 3. $\mathcal{S}^0 \rightarrow \ldots \rightarrow \mathcal{S}^i \rightarrow \ldots \rightarrow \mathcal{S}^*$ is the refinement procedure of subspace clustering, with $\mathcal{S}^i$ denoting the clustering result after i-th iteration of label assignment and subspace re-estimation. $\mathcal{S}^*$ and $\mathcal{L}^*$ are the final clustering and labeling results. From *Line* 1 to *Line* 4, the labeling result is initialized by assigning the closest subspace (label) to each point. *Line* 7 is the label assignment procedure introduced in Sect. 3.4, and in *Line* 8, SPs are re-estimated by solving the following equation,

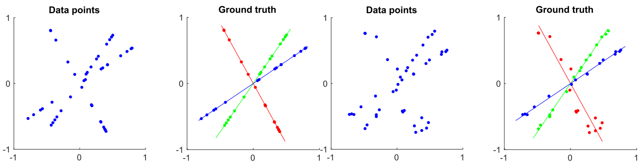
$$\hat{\mathcal{S}}^i_w = \underset{\mathcal{S} \in \text{Leaves}(\mathcal{S}^i_w)}{\arg\max} \sum_{L^i_t == \mathcal{S}^i_w} \text{Proj}_{\mathcal{S}} q_t, \tag{5}$$

where Leaves(·) is a function to return subspaces spanned by root-to-leaf paths in the SE that the input subspace belongs. The sum term is to calculate the projection over the points that are assigned with the input subspace (label). In other words, the re-estimation here is to find a neighboring subspace candidate in the m-branched tree that can better fit the points assigned with the same subspace (label).

The re-estimation procedure can often be found in geometric multi-model fitting problems but can be hardly applied to clustering tasks with general structures. The reason is that the geometric models have parameters to control the subspace, and the models can be updated to better fit the points by optimizing the parameters, while subspaces with general structures can not achieve better SPs in this way. Instead of optimizing the geometric parameters, our method solves this problem by searching the SP which better fits the points from an m-branched enumeration. Both the *Line* 7 and *Line* 8 of Alg. 3 can decrease the energy Eq. (4), we can iterate over these two steps until convergence (e.g., we can stop the iterations when a new round of α-expansion does not change the labeling result).

Algorithm 3 Iteration of Label Assignment and Subspace Re-estimation

```
Input: Union of subspace proposals $\mathcal{S}^0 = \{\mathcal{S}^0_1, \ldots, \mathcal{S}^0_W\}$
Output: Estimated union of subspaces $\mathcal{S}^* = \{\mathcal{S}^*_1, \ldots, \mathcal{S}^*_W\}$, labeling result
    $\mathcal{L}^* = \{L^*_1, \ldots, L^*_N\}, L^*_t \in \mathcal{S}^*$ is a subspace (label) assigned to a given
    data point $q_t$
 1: $i \leftarrow 0$
 2: for each $q_t$ do
 3:     $L^i_t = \arg\max_{\mathcal{S} \in \mathcal{S}^0} \text{Proj}_{\mathcal{S}} q_t, L^i_t \in \mathcal{L}^i$        // Label initialization
 4: end for
 5: repeat
 6:     $i \leftarrow i + 1$
 7:     Run $\alpha$-expansion with Eq. (4) to update the labels $\mathcal{L}^i$
 8:     Solve Eq. (5) to re-estimate subspace proposals
 9: until increase of energy converges
10: $\mathcal{S}^* = \mathcal{S}^i$
11: $\mathcal{L}^* = \mathcal{L}^i$
```



(a) $\theta = 0, N = 45, p = 2, d = 1$ (b) $\theta = 0.1, N = 45, p = 2, d = 1$

Fig. 2 A example of data generation without/with inlier range. Parameter θ controls the range of inliers. Each color denotes different 1D linear subspace.

4. Experiment

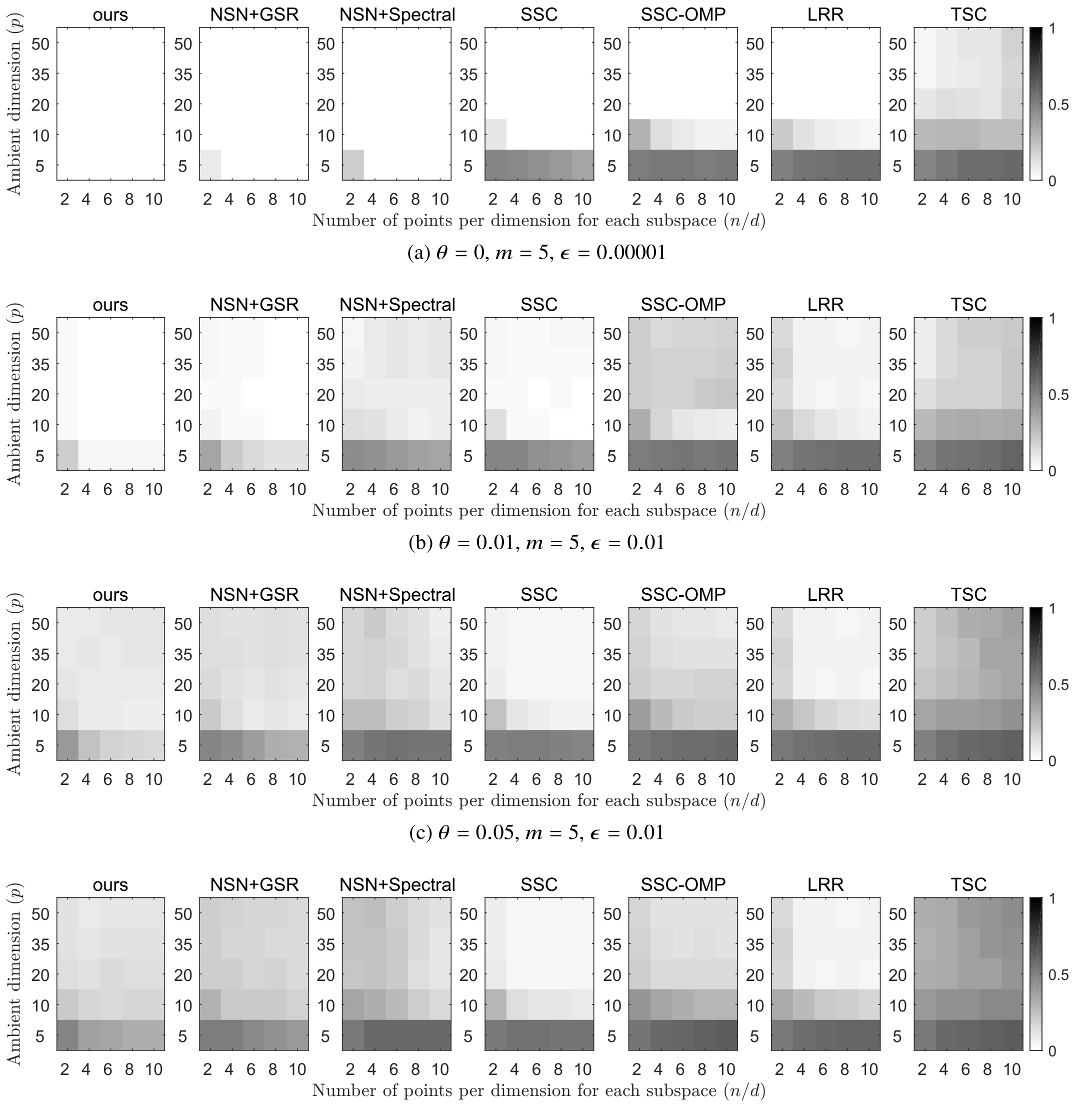
In this Section, we compare our method against several existing methods NSN+GSR [7], NSN+Spectral [7], SSC [15], SSC-OMP [16], LRR [17], and TSC [19] with both synthetic data and further against k-means and k-flats with real-world data. The number of replicates of K-means, K-flats, and the k-means used in the spectral clustering is fixed to 10. The implementations of all the compared methods are provided by the authors, which are publicly available. We use clustering error (CE) as the criteria for evaluating the accuracy defined as following, which is also suggested by [7],

$$\text{(CE)} = \min_{\pi \in \Pi_L} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(L^{gt}_i \neq \pi(L^*_i)\right), \tag{6}$$

where Π_L is the permutation space of [L]. For example, if we have two labels A and B, the permutation space is $\{(A, B), (B, A)\}$. Function π is to convert the order of labeling assignment into according label order. L^{gt}_i is the ground truth label of point q_i. For example, if we have four points and two labels, with their ground truth labels as (A, B, A, B), the correct clustering result can be (A, B, A, B) or (B, A, B, A). That is, CE is the proportion of incorrectly labeled data points, which ignores the order of label assignment (i.e., the labeling result is invariant up to permutation of label indices).

4.1 Statistical Results with Synthetic Date

To validate the effectiveness of our proposed method, we show statistical CE performance with randomly generated synthetic data over 100 trials. To provide a comprehensive comparison, we introduce inliers vary within a certain range (i.e., points which approximately can be fitted to a subspace) during the data generation, instead of inliers just lay on the subspaces. Specifically, points lay on the ground truth subspaces are first generated and then randomly shifted with respect to each dimension, according to a given range pa-



(a) $\theta = 0$, $m = 5$, $\epsilon = 0.00001$

(b) $\theta = 0.01$, $m = 5$, $\epsilon = 0.01$

(c) $\theta = 0.05$, $m = 5$, $\epsilon = 0.01$

(d) $\theta = 0.1$, $m = 5$, $\epsilon = 0.01$

Fig. 3 Comparison of average CE over 100 random trials. The first column shows the results of our method and others are compared methods. Here, $d = 3$ and the number of subspaces is fixed to five. Basically, with smaller p and smaller n/d, the clustering task becomes more difficult, and lower CE indicates better clustering performance. Also, the first row is with inliers lay exactly on the subspaces, and from the second row, with the increase of θ, the inliers distribute more and more irregularly.

rameter θ. For example, in Fig. 2 (a), the data points are generated exactly on the subspaces while in Fig. 2 (b), the data points are generated by $\theta = 0.1$. That is, x-axis and y-axis values of points are summed by a value randomly generated from $[-0.1, 0.1]$. For the evaluation data, at first, five d-dimensional subspaces in $\mathbb{R}^p$ space are generated randomly. Then, points are generated iid uniformly on each subspace. Each point is then shifted by parameter θ. We run the comparative experiment with fixed $d = 3$, #subspaces=5, and vary p and n/d to obtain statistical results. From Fig. 3, we can observe that in most cases, the performance increases with the increase of p and n/d. Our method outperforms others totally when $\theta = 0$ or $\theta = 0.01$, while it is not always the case when $\theta = 0.05$ or $\theta = 0.1$. This well demonstrates the advantages and limitations of our proposed method, which can be concluded as follows,

Advantages:

- Our method is power for exact clustering task (i.e., data points are lying exactly on the subspaces). In Fig. 3 (a), our method even achieves 0 error.
- Our method can deal with the situation when both p and d are close and small. This is important for many real-world applications (e.g., motion segmentation). For example, when $p = 5$, we can observe from Fig. 3 that the CE of our method decreases most with the increase of n/d, while some compared methods even get higher CE.

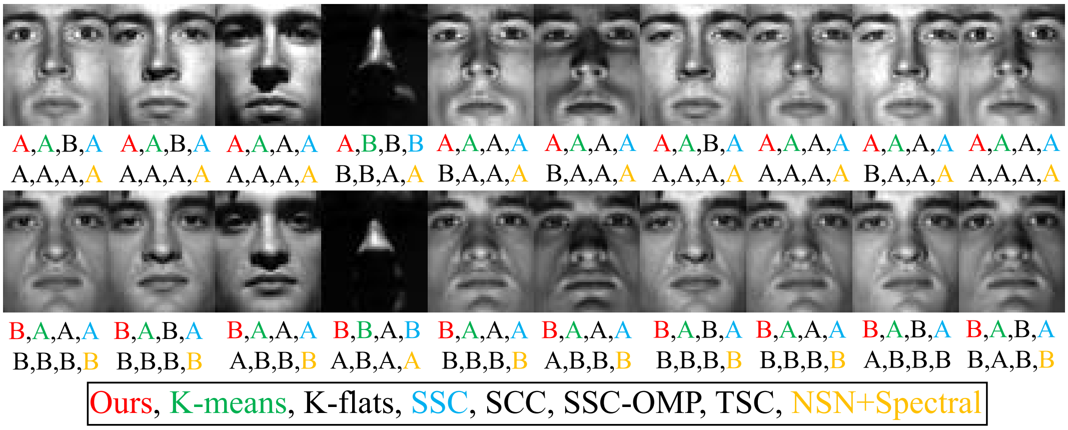

A,A,B,A A,A,B,A A,A,A,A A,B,B,B A,A,A,A A,A,A,A A,A,B,A A,A,A,A A,A,A,A A,A,A,A
A,A,A,A A,A,A,A A,A,A,A B,B,A,A B,A,A,A B,A,A,A A,A,A,A A,A,A,A B,A,A,A A,A,A,A
B,A,A,A B,A,B,A B,A,A,A B,B,A,B B,A,A,A B,A,A,A B,A,B,A B,A,A,A B,A,B,A B,A,B,A
B,B,B,B B,B,B,B A,B,B,B A,B,A,A B,B,B,B A,B,B,B B,B,B,B B,B,B,B A,B,B,B B,A,B,B
Ours, K-means, K-flats, SSC, SCC, SSC-OMP, TSC, NSN+Spectral

Fig. 4 Comparative results with face clustering task. We only show two subjects under 10 different illumination conditions for illustration. Images within the same row belong to the same subject. A and B are two labels to distinguish two subjects (labeling results of each method can be represented in A-B or B-A order, which are equivalent). Colored A/B represents labeling result of competitive methods, which are in the order of the legend. Best viewed in color.

Limitations:

- Unlike SSC or LRR, low CE can not be achieved when the θ is large. As we will describe in Sect. 4.3, to deal with larger data nuisances, larger scale of enumeration need to be generated and an appropriate inlier thresold is necessary, which is hard to be estimated from input data without any prior knowledge.
- The scale of m-branched enumeration grows exponentially with the increase of d, which practically limits the support to large data nuisances.

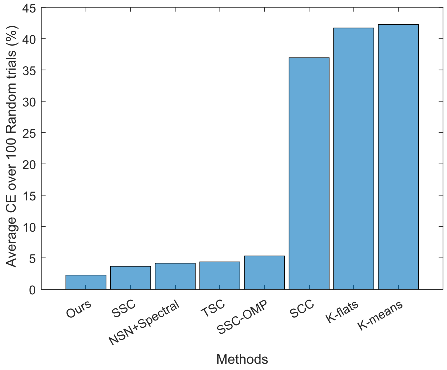
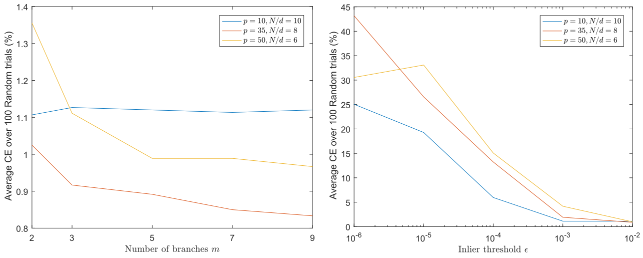


Fig. 5 Comparative result of average CE. Two subjects are randomly selected over 38 subjects for 100 trials to calculate the average CE.

4.2 Qualitative and Quantitative Results with Real Application of Face Clustering

To further show the usefulness of the proposed method, we compare some existing methods with respect to the face clustering task. We use the extended Yale B dataset [26], [27], which contains 2432 images with 38 subjects under 64 illumination conditions. Each image is cropped to 192 by 168 pixels and downscaled to 48 by 42 pixels. For better visualization, we randomly choose 10 illumination conditions as shown in Fig. 4. Among the eight different methods, only our method correctly cluster the two subjects (subspace). A common failure of K-means and SSC is that they cluster the images with respect to the light intensity: two images in the fourth column are clustered as one subspace and the left images form another cluster. Figure 5 shows the statistical result, in which the proposed method has the lowest average CE to show the effectiveness.



(a) Branch number m vs. CE. (b) Inlier threshold ϵ vs. CE.

Fig. 6 Effect of hyper parameters m and ϵ by synthetic test.

4.3 Effect of Hyper Parameters

There are two main parameters m and ϵ to affect the global performance of the proposed method. To study their effects, we give out the curves of average CE with respect to three different parameter settings in Fig. 6. From Fig. 6 (a), we can observe that the average CE decreases with the increase of m, except when the p is small. From Fig. 6 (b), we can observe that closer the ϵ is to the range of ground truth inlier ($\theta = 0.01$), lower average CE can be achieved. These observations well reveal the limitations described in Sect. 4.1: the performance depends on “reasonable guess” of θ and m has a trade-off between accuracy and processing time.

5. Conclusion

In this paper, we presented a subspace clustering algorithm with general subspace structures. The main contribution is to take advantages of both greedy strategy and energy minimization strategy to propose a hybrid solution. Extensive experiments demonstrate that our method outperforms the state-of-the-art methods in both synthetic and real-world situations. Despite the robustness of our method, it still has a few limitations. The performance is likely to depend on the initial setting of inlier threshold, and its computational cost can grow exponentially with the increase of the enumeration parameter. One potential way to solve these problems is to predetermine the threshold through a learning-based method and run the enumeration procedure in parallel. As the future work, we would like to develop effective subspace proposal evaluation methods for more real-world applications.

Acknowledgments

This work is partly supported by JSPS KAKENHI Grant Number JP18K17823.

References

[1] F. Lauer and C. Schnörr, "Spectral clustering of linear subspaces for motion segmentation," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.678–685, 2009.

[2] R. Basri and D.W. Jacobs, "Lambertian reflectance and linear subspaces," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.2, pp.218–233, 2003.

[3] R. Hu, L. Fan, and L. Liu, "Co-segmentation of 3D shapes via subspace clustering," Comput. Graph. Forum, vol.31, no.5, pp.1703–1713, 2012.

[4] J. Zhang, J. Cao, X. Liu, J. Wang, J. Liu, and X. Shi, "Point cloud normal estimation via low-rank subspace clustering," Comput. Graph., vol.37, no.6, pp.697–706, 2013.

[5] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," ACM SIGKDD Explorations Newsletter, vol.6, no.1, pp.90–105, 2004.

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

[7] D. Park and S. Caramanis, Constantine Sanghavi, "Greedy subspace clustering," Proc. Advances in Neural Information Processing Systems (NIPS), pp.2753–2761, 2014.

[8] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," Int. J. Comput. Vision., vol.97, no.2, pp.123–147, 2012.

[9] Y.X. Wang and H. Xu, "Noisy sparse subspace clustering," Journal of Machine Learning Research (JLMR), vol.17, no.1, pp.320–360, 2016.

[10] C. Yang, D. Robinson, and R. Vidal, "Sparse subspace clustering with missing entries," Proc. International Conference on Machine Learning (ICML), pp.2463–2472, 2015.

[11] J.P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," Int. J. Comput. Vision., vol.29, no.3, pp.159–179, 1998.

[12] V. Larsson and C. Olsson, "Compact matrix factorization with dependent subspaces," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.280–289, 2017.

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

[14] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.12, pp.1945–1959, 2005.

[15] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," IEEE Trans. Pattern Anal. Mach. Intell., vol.35, no.11, pp.2765–2781, 2013.

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

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

[18] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When LRR meets SSC," Proc. Advances in Neural Information Processing Systems (NIPS), pp.64–72, 2013.

[19] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," IEEE Trans. Inf. Theory, vol.61, no.11, pp.6320–6342, 2015.

[20] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol.24, no.6, pp.381–395, 1981.

[21] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC–full experimental evaluation," Proc. British Machine Vision Conference 2012, pp.95.1–95.11, 2012.

[22] T.J. Chin, P. Purkait, A. Eriksson, and D. Suter, "Efficient globally optimal consensus maximisation with tree search," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2413–2421, 2015.

[23] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," Proc. Seventh IEEE International Conference on Computer Vision, pp.377–384, 1999.

[24] T.T. Pham, T.-J. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting," IEEE Trans. Pattern Anal. Mach. Intell., vol.36, no.8, pp.1658–1671, 2014.

[25] P. Meer, "Robust techniques for computer vision," Emerging topics in computer vision, pp.107–190, 2004.

[26] 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 Trans. Pattern Anal. Mach. Intell., vol.23, no.6, pp.643–660, 2001.

[27] K.-C. Lee, J. Ho, and D.J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.5, pp.684–698, 2005.

Chao Zhang received his Ph.D. at Iwate University (Japan) in March 2017. He is now a full-time assistant professor at graduate school of engineering, University of Fukui (Japan). His research interests include computer vision and graphics, mainly focused on feature matching and vision-based optimization problems. He is a member of the IEEE computer society, IEEE signal processing society, ACM and IEICE.