

# Active Orthogonal Matching Pursuit for Sparse Subspace Clustering

Yanxi Chen, Gen Li, and Yuantao Gu\*

submitted June 20, 2017, accepted August 9, 2017, for publication in  
*IEEE Signal Processing Letters*

## Abstract

Sparse Subspace Clustering (SSC) is a state-of-the-art method for clustering high-dimensional data points lying in a union of low-dimensional subspaces. However, while  $\ell_1$  optimization-based SSC algorithms suffer from high computational complexity, other variants of SSC, such as Orthogonal Matching Pursuit-based SSC (OMP-SSC), lose clustering accuracy in pursuit of improving time efficiency. In this letter, we propose a novel Active OMP-SSC, which improves clustering accuracy of OMP-SSC by adaptively updating data points and randomly dropping data points in the OMP process, while still enjoying the low computational complexity of greedy pursuit algorithms. We provide heuristic analysis of our approach, and explain how these two active steps achieve a better tradeoff between connectivity and separation. Numerical results on both synthetic data and real-world data validate our analyses and show the advantages of the proposed active algorithm.

**Keywords:** Sparse subspace clustering, orthogonal matching pursuit, active algorithm, subspace detection property, connectivity

## 1 Introduction

In a big-data era, unsupervised learning plays a significant role in analyzing numerous unlabeled data. In many applications, such as motion segmentation and face clustering,

---

\*The authors are with the Department of Electronic Engineering and Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing 100084, China. The corresponding author of this work is Y. Gu (E-mail: gyt@tsinghua.edu.cn).

high-dimensional data samples are drawn approximately from a union of low dimensional subspaces. Subspace clustering (SC) [1] refers to the problem of clustering these data points into their original subspaces, and various methods have been developed to solve this problem [1–16]. Among them, Sparse Subspace Clustering (SSC), the first one to introduce sparse representation into SC problem, is a state-of-the-art method, with elegant formulation and theoretical guarantees to work under weak conditions, e.g., at the presence of noise [17] and outliers [18]. SSC algorithm solves SC problem by first finding a sparse self-representation for each data point, and then applying spectral clustering [19] to the similarity matrix. However, it is inefficient to solve an optimization problem for sparse representation (as the original SSC algorithm does) when analyzing large-scale data. One way to speed up SSC is replacing  $\ell_1$  optimization with greedy pursuit, e.g., Orthogonal Matching Pursuit (OMP) [20], for sparse self representation [21–23], which improves the time efficiency of SSC by several orders of magnitude for large-scale problem. Another way is to compress data points (i.e., to reduce dimension) before applying SSC algorithm, leading to Compressed Subspace Clustering algorithm [24–26], whose performance is guaranteed by works in random projection and Restricted Isometry Property (RIP) of subspaces [27, 28].

One motivation of this letter is the drawbacks of existing SSC algorithms. Traditional  $\ell_1$  optimization-based SSC ( $\ell_1$ -SSC) suffers from high computational complexity, while OMP-SSC loses clustering accuracy. Our goal is to develop a fast SSC algorithm with high clustering accuracy. We are also motivated by the insight that the final clustering accuracy depends on two major properties of the similarity matrix: *Subspace Detection Property* (SDP) [18], i.e., “no false connection”, among points from different subspaces, and *connectivity* among points from the same subspace. While there is rich literature on theoretical analysis of SDP, much fewer works concern about the connectivity problem [29]. [30] introduces a post-processing procedure, and [31] modifies the optimization target as a weighted sum of  $\ell_1$  norm and nuclear norm of the representation matrix in order to tradeoff between separation and connectivity.

In this letter, we proposed a novel algorithm, active OMP-SSC (A-OMP-SSC), which introduces two active-style processes, including adaptively updating and randomly dropping data points. We analyze how the proposed algorithm obtains a better tradeoff between connectivity and SDP, thus improving clustering accuracy while preserving advantages of greedy methods in time efficiency. Numerical results validate our analyses and show the

advantages of our algorithm.

## 2 Preliminary

Given data matrix  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ , where  $D$  is data dimension and  $N$  is total number of data points, the aim of SC is to cluster these data points into their original subspaces. In the preprocessing, all points are  $\ell_2$  normalized, i.e.,  $\|\mathbf{x}_i\|_2 = 1, \forall i$ .

### 2.1 SSC and OMP-SSC

The core idea behind SSC is that, each data point can be approximately represented as a sparse linear combination of other points, where the nonzero entries correspond to those from the same subspace. The first step of SSC is to find a sparse representation for each point by solving an optimization problem of

$$\min_{\mathbf{c}_i} \|\mathbf{c}_i\|_1 + \frac{\lambda}{2} \|\mathbf{x}_i - \mathbf{X}\mathbf{c}_i\|_2^2 \quad \text{s.t.} \quad c_{ii} = 0, \quad (1)$$

where  $\mathbf{c}_i$  denotes the representation coefficient vector for  $\mathbf{x}_i$  and  $\lambda$  is a balance parameter. The second step is to apply spectral clustering to the similarity matrix  $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^T$  and get the clustering labels.

OMP-SSC is a faster version of SSC. It finds the sparse representation by OMP instead. Given data point  $\mathbf{x}_i$  and dictionary composed of all other data points, OMP iteratively finds an atom that has the largest absolute inner product with residual, adds this atom to the neighbor set, projects  $\mathbf{x}_i$  onto the span of its current neighbors, and updates residual, until the iteration number reaches a certain value or the norm of residual is small enough.

### 2.2 Geometric Analysis

Most literature focuses theoretical analyses of SSC on SDP.

**Definition 1.** [18] Subspace Detection Property (SDP) holds if and only if it holds that for all  $i$ ,  $\mathbf{c}_i$  is non-trivial (i.e., not all-zeros), and  $c_{ij} \neq 0$  only when  $\mathbf{x}_i$  and  $\mathbf{x}_j$  lie in the same subspace.

A geometric framework for theoretical analysis of SDP was first proposed in [18], and has been thoroughly studied since then. In comparison, only a few works analyze the

connectivity problem of SSC. It is shown in [29] that, when subspace dimension exceeds three, there is no guarantee that data points from the same subspace will form a connected component even in the noiseless case.

### 3 Active OMP-SSC

#### 3.1 Intuition

As mentioned before, the final clustering accuracy of SSC depends on both connectivity and SDP of the similarity matrix obtained by the self-representation step. Actually, SDP holding true is not necessary for correct clustering results due to the robustness of spectral clustering. On the other hand, spectral clustering could fail if the connectivity of similarity matrix is so weak that data points from the same subspace are separated. The ideal case is that both connectivity and SDP are strong, which is unrealistic in practice at the presence of noise. Therefore, the problem of increasing clustering accuracy becomes to tradeoff between connectivity and SDP.

#### 3.2 Algorithm

The proposed A-OMP-SSC algorithm operates iteratively on all data points. For  $\mathbf{x}_i$ , we first find its sparse representation  $\mathbf{c}_i$  on the current dictionary, indexed by a set  $\mathcal{D}$ , using OMP,

$$\begin{aligned} \mathbf{c}_i = \arg \min_{\hat{\mathbf{c}}_i} \|\mathbf{x}_i - \mathbf{X}\hat{\mathbf{c}}_i\|_2, \quad \text{s.t. } \|\mathbf{c}_i\|_0 = d \text{ and} \\ c_{ij} = 0, \forall j \in (\{1, \dots, N\} \setminus \mathcal{D}) \cup \{i\}, \end{aligned} \quad (2)$$

where  $d$  denotes the iteration number of OMP. Since the goal of OMP here is not calculating exactly the representation coefficients, but finding a few neighbors reliably, the iteration number  $d$  is not very large, which will be specified later. We then calculate the representation residual

$$\mathbf{r}_i = \mathbf{x}_i - \mathbf{X}\mathbf{c}_i. \quad (3)$$

After the sparse representation step, we *update*  $\mathbf{x}_i$  by adding an offset term to it and keeping it  $\ell_2$  normalized

$$\mathbf{x}'_i = (\mathbf{x}_i + b\mathbf{r}_i) / \|\mathbf{x}_i + b\mathbf{r}_i\|_2, \quad (4)$$

---

**Algorithm 1** The Proposed A-OMP-SSC

---

**Input:** Data points  $\mathbf{X} \in \mathbb{R}^{D \times N}$ , dictionary index set  $\mathcal{D} = \{1, \dots, N\}$ , OMP iteration number  $d$ , modifier parameter  $b > 0$ , dropping probability  $p \in (0, 1)$ , cluster number  $k$ .

- 1: **for**  $i = 1, \dots, N$  **do**
- 2: Find  $\mathbf{c}_i$  using OMP (with  $d$  iterations) by (2).
- 3: Calculate residual  $\mathbf{r}_i$  by (3).
- 4: Update data point:  $\mathbf{x}_i \leftarrow \mathbf{x}'_i$  by (4).
- 5: Update dictionary:  $\mathcal{D} \leftarrow \mathcal{D} \setminus \{i\}$  with probability  $p$ .
- 6: **end for**
- 7: Apply spectral clustering (with number of clusters  $k$  given) to similarity matrix  $\mathbf{A}$ .

**Output:** Clustering labels.

---

where  $b > 0$  denotes a modifier parameter. Then, with certain probability  $p$  we *drop* this point from the future representation, i.e., removing index  $i$  from the dictionary index set. The process above is repeated sequentially for all data points. Finally, we apply spectral clustering to the similarity matrix, as done in SSC algorithms. Notice that OMP-SSC is a special case of A-OMP-SSC, with modifier parameter  $b = 0$  and dropping probability  $p = 0$ . The detailed procedure of the proposed algorithm is included in Algorithm 1.

### 3.3 Discussion

#### 3.3.1 Sequentially updating data points

From the construction of matrix  $\mathbf{A}$  comes the intuition that, a two-direction connection doesn't contribute to connectivity. If  $\mathbf{x}_i$  chooses  $\mathbf{x}_j$  ( $j > i$ ) as its neighbor, we should encourage  $\mathbf{x}_j$  to choose the data points other than  $\mathbf{x}_i$ . By the updating operation in (4), we push  $\mathbf{x}_i$  further away from its neighbors, as illustrated in Fig. 1.

This will be verified in an ideal case. Suppose (3) is a correct decomposition of  $\mathbf{x}_i$ , where  $\bar{\mathbf{x}}_i = \mathbf{X}\mathbf{c}_i$  lies in its ideal subspace and  $\mathbf{r}_i$  in its orthogonal subspace. Notice that the projection operation in OMP ensures that  $\mathbf{x}'_i{}^T \bar{\mathbf{x}}_i$  is nonnegative. After the updating of (4) we have

$$\mathbf{x}'_i{}^T \bar{\mathbf{x}}_i = \frac{(\bar{\mathbf{x}}_i + (b+1)\mathbf{r}_i)^T \bar{\mathbf{x}}_i}{\|\mathbf{x}_i + b\mathbf{r}_i\|_2} = \frac{\bar{\mathbf{x}}_i^T \bar{\mathbf{x}}_i}{\|\mathbf{x}_i + b\mathbf{r}_i\|_2} \leq \bar{\mathbf{x}}_i^T \bar{\mathbf{x}}_i = \mathbf{x}_i^T \bar{\mathbf{x}}_i. \quad (5)$$

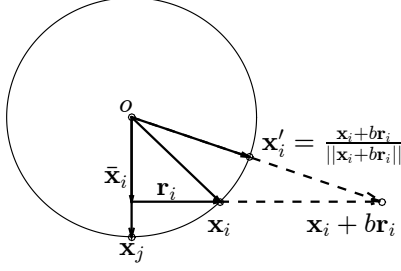


Figure 1: A visualization that updating data point  $\mathbf{x}_i$  may push it further away from its neighbor  $\mathbf{x}_j$ . Without loss of generality, we suppose  $\mathbf{x}_i^\top \mathbf{x}_j > 0$ .

The inequality in (5) comes from the fact that, when  $b \geq 0$ ,

$$\begin{aligned} \|\mathbf{x}_i + b\mathbf{r}_i\|_2^2 &= \|\bar{\mathbf{x}}_i + (b+1)\mathbf{r}_i\|_2^2 = \|\bar{\mathbf{x}}_i\|_2^2 + (b+1)^2\|\mathbf{r}_i\|_2^2 \\ &\geq \|\bar{\mathbf{x}}_i + \mathbf{r}_i\|_2^2 = 1, \end{aligned} \quad (6)$$

and the equality holds when  $b = 0$ .

From (6),  $\|\mathbf{x}_i + b\mathbf{r}_i\|_2$  is symmetric about  $b_0 = -1$ . In practice, since signal-to-noise ratio (SNR) of residual is generally lower than that of the data point itself, a positive  $b$  introduces less noise than the symmetric negative one in the updating step because it has a smaller absolute value, which is preferred. The choice of  $b$  reflects a tradeoff between improving connectivity and avoiding worsening SNR too much.

### 3.3.2 Randomly dropping data points

Dropping  $\mathbf{x}_i$  is more radical than updating it, as it is now impossible for  $\mathbf{x}_j, j > i$  to choose  $\mathbf{x}_i$  as its neighbor. However, dropping data points could be risky since it reduces the data density, which is undesirable in SSC as it worsens SDP.

### 3.3.3 OMP iteration number

Intuitively, in OMP process, a new edge improves connectivity if it is a true connection, but weakens SDP otherwise. For OMP, once an atom is added to the neighbor set at a certain iteration, it cannot be removed. Therefore, as iteration number  $d$  increases, connectivity increases but SDP decreases, as pointed out in [23]. Thus, it is not the best to choose  $d$  as the subspace dimension, as done in [22]. Choosing a smaller OMP iteration number can improve clustering accuracy and linearly decrease time consumption.

Another advantage of choosing a small OMP iteration number in our active algorithm is that, it allows more freedom for a correct choice of sparse representation coefficient satisfying SDP as  $\mathbf{c}_i$  has fewer non-zero entries. For the SC problem, the essential information from data is not the data points themselves, but their *relationship* instead. This inspires the updating and dropping steps in A-OMP-SSC, which seek to further exploit this relationship by changing the distribution of data points adaptively and thus changing the choice of representation coefficients.

### 3.3.4 When it benefits

A-OMP-SSC works well when data density is not too small. This comes from the fact that, the updating step changes the distribution of points, and the dropping step gradually decreases data density. When the starting data density is relatively high, there are still many near neighbors for later points, so SDP would not badly decrease.

### 3.3.5 Computational complexity

In OMP-SSC, each iteration requires  $N$  inner products with complexity  $\mathcal{O}(D)$ . Thus, for  $N$  points, each  $d$  iterations, the total complexity of self-representation step is  $\mathcal{O}(N^2dD)$ . For A-OMP-SSC, only an additional  $\mathcal{O}(D)$  updating step is required for each point, which is negligible. Moreover, the dropping step gradually decreases the number of points during the process, which improves time efficiency.

## 4 Numerical Experiments

In this section, we conduct numerical experiments to validate our analysis and illustrate the advantages of active OMP-SSC algorithm<sup>1</sup>.

### 4.1 Synthetic Experiments

We first conduct experiments to examine the effects of modifier parameter  $b$ , dropping probability  $p$ , and OMP iteration number  $d$ , respectively. Next, we compare  $\ell_1$ -SSC, OMP-SSC, and A-OMP-SSC algorithms, with respect to clustering error rate, connectivity, and SDP.

---

<sup>1</sup>The MATLAB codes for the proposed methods and all experiments are available at [http://gu.ee.tsinghua.edu.cn/codes/A\\_OMP\\_SSC.zip](http://gu.ee.tsinghua.edu.cn/codes/A_OMP_SSC.zip).

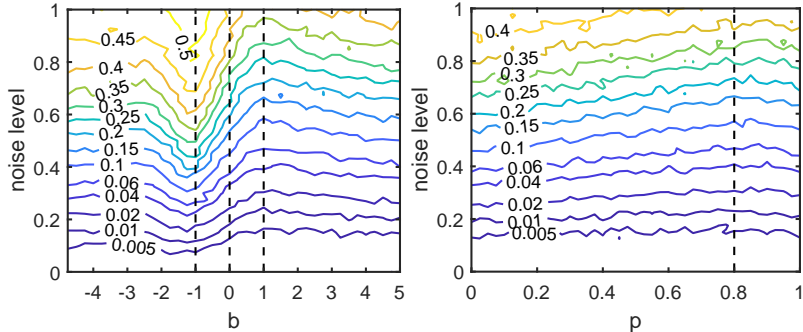


Figure 2: Clustering error rate versus noise level and parameter  $b$  (left) or  $p$  (right).

We also validate that A-OMP-SSC shows advantages over OMP-SSC when data density is not too small. Finally, the superior time efficiency of the proposed algorithm is verified. The generated synthetic samples are randomly permuted as we meet in practice.

#### 4.1.1 Modifier parameter $b$

We randomly generate 3 independent linear subspaces in 40-dimensional ambient space, each of which is of dimension 6 and has 45 data samples. The noise level, i.e., additive Gaussian noise strength of each sample, varies from 0 to 1. We set  $p = 0$  and  $d = 3$ . Clustering results with different choices of modifier parameter  $b$  are demonstrated in Fig. 2 (left), where each result is the average of 100 independent trials. According to Fig. 2 (left), we read that clustering error rate is the highest at  $b \approx -1$  and lowest at 1. As  $b$  increases from  $-1$ , clustering error rate first drops, leveraged by improvement in connectivity, but then rises due to a decrease in SNR and thus a reduction in SDP. (Due to limited space, results on connectivity and SDP are not shown here.) Also notice that a positive  $b$  is better than a negative one symmetric about  $-1$  thanks to higher SNR.

#### 4.1.2 Dropping probability $p$

The data is generated the same way as in the previous experiment. We set  $b = 0$  and  $d = 3$ . Clustering results with different choices of dropping probability  $p$  are demonstrated in Fig. 2 (right), which shows that clustering error rate reaches a minimum at  $p \approx 0.8$ . A large  $p$  up to 1 is a good choice in this experiment, because the subspace dimension and OMP iteration number are small while the number of data points per subspace is relatively large. Therefore, the decreasing data density would not be a big trouble.



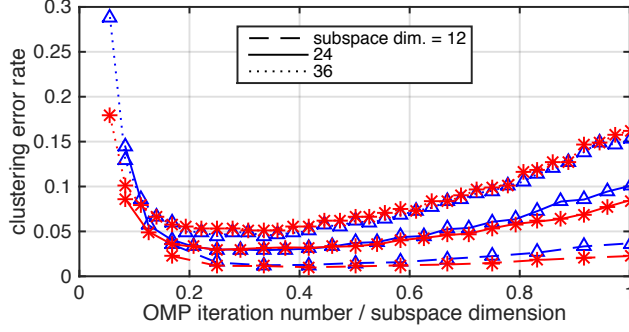


Figure 3: Effects of OMP iteration number on clustering error rate for OMP-SSC (triangle markers) and A-OMP-SSC (star markers).

#### 4.1.3 OMP iteration number $d$

We randomly generate 3 independent linear subspaces in  $\mathbb{R}^{120}$ , each of which has 60 data samples. For A-OMP-SSC,  $b = 1$ ,  $p = 0.3$ . This experiment is conducted under three conditions, with subspace dimension set to 12, 24, 36, and the corresponding noise level 0.9, 0.7, 0.5. Effects of OMP iteration number  $d$  are demonstrated in Fig. 3. As  $d$  increases from 1 to subspace dimension, clustering error rate first drops owing to increasing connectivity, but then rises due to decreasing SDP, just as we discussed. A reasonable choice of  $d$  is approximately one over three of subspace dimension. Notice that A-OMP-SSC doesn't perform well when subspace dimension is 36, which is a result of the small data density, as we will discuss later.

#### 4.1.4 Performance comparison

We take the second smallest eigenvalue of the normalized Laplacian of a cluster as a metric of connectivity (with range  $[0, 1]$ ); the larger it is, the stronger connectivity is [22]. We also use the percentage of points satisfying SDP as a metric of SDP (with range  $[0, 100]$ ); the larger it is, the stronger SDP is. The data is generated the same way as in the first experiment. Based on previous results, we set  $b = 1$ ,  $p = 0.8$ , and  $d = 3$ . A performance comparison of three algorithms is demonstrated in Fig. 4. Results show that A-OMP-SSC outperforms OMP-SSC in clustering accuracy, thanks to great improvement in connectivity, despite minor loss of SDP. A-OMP-SSC also outperforms  $\ell_1$ -SSC at slightly high noise level. This is because the number of chosen neighbors in the solution of  $\ell_1$ -SSC is generally around or larger than subspace dimension, therefore SDP percentage quickly decreases to zero as

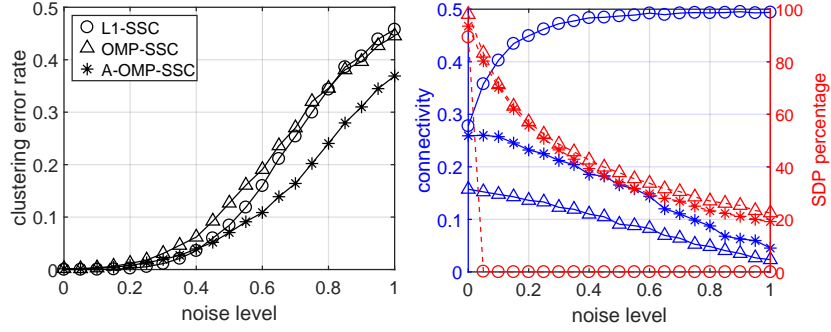


Figure 4: Comparison between  $\ell_1$ -SSC, OMP-SSC and A-OMP-SSC. Left: clustering error rate; right: connectivity (left axis, blue solid lines) and SDP percentages (right axis, red dashed lines).

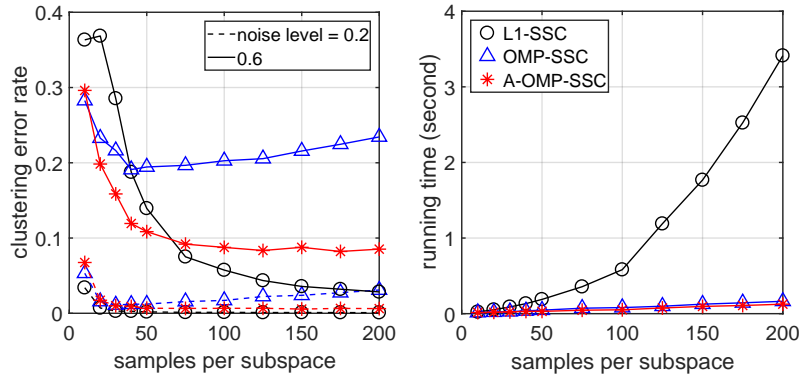


Figure 5: Algorithm comparisons under various problem scales. Left: clustering error rate; right: running time (in seconds).

noise level increases. This result demonstrates the robustness of A-OMP-SSC to white Gaussian noise.

#### 4.1.5 Data density

We use the same parameter set as the performance comparison experiment, except that the number of samples per subspace varies from 10 to 200. A comparison of clustering error rate between SSC algorithms under certain noise levels is demonstrated in Fig. 5 (left). OMP-SSC performs better when data density is small, while A-OMP-SSC possesses an increasing advantage as data density rises, which validates our previous analysis on the success condition of A-OMP-SSC.

We also compare the running time of SSC algorithms under the same simulation setting.

Table 1: EYaleB clustering error rate in percentage.

# clusters	2	3	5	8	13	21	38
OMP-SSC	1.11	2.64	5.52	8.43	14.4	19.8	28.7
A-OMP-SSC	<b>1.10</b>	<b>2.23</b>	<b>4.11</b>	<b>6.94</b>	<b>12.2</b>	<b>16.8</b>	<b>22.3</b>

The result is shown in Fig. 5 (right). Despite the disadvantage in clustering accuracy in some scenarios, a major advantage of OMP-SSC is that it is much faster than traditional  $\ell_1$ -SSC, which is consistent with the results from existing literature [22, 23]. Furthermore, A-OMP-SSC is even faster than OMP-SSC, because the operation of updating data points doesn't change the overall computational complexity, while the operation of dropping data points gradually reduces data density.

## 4.2 Face Clustering

We test our algorithm on EYaleB, a dataset of human-face pictures from 38 persons, each with around 64 photos under various illumination. In our experiment, the pictures are downsampled to  $48 \times 42$ . It is known that human-face images of a subject can be approximated by a 9-dimensional subspace [2]. We set  $b = 0.5$ ,  $p = 0.2$  and  $d = 3$ . A comparison of OMP-SSC and A-OMP-SSC on clustering error rate is demonstrated in Table 1. The result is the average of 200 trials. For each trial, given the number of clusters  $k$ , we randomly choose  $k$  subjects, mix all images of these subjects together, and permute them before applying clustering algorithms. The result shows that A-OMP-SSC outperforms its reference on human face clustering problem, especially in the scenario of multiple classes. Notice that the state-of-the-art clustering error rate of  $\ell_1$ -SSC in the case of all 38 subjects is 31.0% [22], while its running time is prohibitive. This highlights the fact that A-OMP-SSC enjoys both superior time efficiency and higher clustering accuracy in such large-scale problems.

## 5 Conclusion

We propose a fast and accurate A-OMP-SSC algorithm, which obtains a better tradeoff between connectivity and SDP via adaptively updating and randomly dropping data points. We heuristically explain the intuitions behind our algorithm and analyze the condition where it works. Numerical results validate our analyses, and demonstrate the advantages of our

algorithm over OMP-SSC and  $\ell_1$ -SSC. The proposed active mechanism may benefit many other SC algorithms.

## References

- [1] R. Vidal, “Subspace clustering,” *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 52–68, 2011.
- [2] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.
- [4] R. Vidal and P. Favaro, “Low rank subspace clustering (lrsc),” *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.
- [5] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, “Robust and efficient subspace segmentation via least squares regression,” *Computer Vision–ECCV 2012*, pp. 347–360, 2012.
- [6] C. Lu, J. Feng, Z. Lin, and S. Yan, “Correlation adaptive subspace segmentation by trace lasso,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1345–1352, 2013.
- [7] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin, “Correntropy induced l2 graph for robust subspace clustering,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1801–1808, 2013.
- [8] R. Heckel and H. Bölcskei, “Robust subspace clustering via thresholding,” *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.
- [9] Y. Zhang, Z. Sun, R. He, and T. Tan, “Robust subspace clustering via half-quadratic minimization,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3096–3103, 2013.

- [10] D. Park, C. Caramanis, and S. Sanghavi, “Greedy subspace clustering,” in *Advances in Neural Information Processing Systems*, pp. 2753–2761, 2014.
- [11] C. Yang, D. Robinson, and R. Vidal, “Sparse subspace clustering with missing entries,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2463–2472, 2015.
- [12] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, “Oracle based active set algorithm for scalable elastic net subspace clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3928–3937, 2016.
- [13] X. Peng, Z. Yu, Z. Yi, and H. Tang, “Constructing the l2-graph for robust subspace learning and subspace clustering,” *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2017.
- [14] H. Hu, Z. Lin, J. Feng, and J. Zhou, “Smooth representation clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3834–3841, 2014.
- [15] C.-G. Li, C. You, and R. Vidal, “Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework,” *IEEE Transactions on Image Processing*, 2017.
- [16] C. Peng, Z. Kang, M. Yang, and Q. Cheng, “Feature selection embedded subspace clustering,” *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 1018–1022, 2016.
- [17] Y.-X. Wang and H. Xu, “Noisy sparse subspace clustering,” in *Proc. Int. Conf. Mach. Learn.*, pp. 1–8, 2013.
- [18] M. Soltanolkotabi and E. J. Candes, “A geometric analysis of subspace clustering with outliers,” *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.
- [19] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [20] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

- [21] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, “Greedy feature selection for subspace clustering,” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.
- [22] C. You, D. Robinson, and R. Vidal, “Scalable sparse subspace clustering by orthogonal matching pursuit,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3918–3927, 2016.
- [23] M. Tschannen and H. Bölcskei, “Noisy subspace clustering via matching pursuits,” *arXiv preprint arXiv:1612.03450*, 2016.
- [24] X. Mao and Y. Gu, “Compressed subspace clustering: A case study,” in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pp. 453–457, IEEE, 2014.
- [25] R. Heckel, M. Tschannen, and H. Bölcskei, “Dimensionality-reduced subspace clustering,” *Information and Inference: A Journal of the IMA*, p. iaw021, 2017.
- [26] Y. Wang, Y.-X. Wang, and A. Singh, “A theoretical analysis of noisy sparse subspace clustering on dimensionality-reduced data,” *arXiv preprint arXiv:1610.07650*, 2016.
- [27] G. Li and Y. Gu, “Distance-preserving property of random projection for subspaces,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [28] G. Li and Y. Gu, “Restricted isometry property of gaussian random projection for finite set of subspaces,” *arXiv preprint arXiv:1704.02109*, 2017.
- [29] B. Nasihatkon and R. Hartley, “Graph connectivity in sparse subspace clustering,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2137–2144, IEEE, 2011.
- [30] Y. Wang, Y.-X. Wang, and A. Singh, “Graph connectivity in noisy sparse subspace clustering,” *CoRR abs/1504.01046*, 2016.
- [31] Y.-X. Wang, H. Xu, and C. Leng, “Provable subspace clustering: When lrr meets ssc,” in *Advances in Neural Information Processing Systems*, pp. 64–72, 2013.