Abstract
Orthogonal Matching Pursuit (OMP) plays an important role in data science and its applications such as sparse subspace clustering and image processing. However, the existing OMP-based approaches lack of data adaptiveness so that the information cannot be represented well enough and may lose the accuracy. This paper proposes a novel approach to enhance the data-adaptive capability for OMP-based sparse subspace clustering. In our method a parameter selection process is developed to adjust the parameters based on the data distribution for information representation. Our theoretical analysis indicates that the parameter selection process can efficiently coordinate with any OMP-based methods to improve the clustering performance. Also a new Self-Expressive-Affinity (SEA) ratio metric is defined to measure the sparse representation conversion efficiency for spectral clustering to obtain data segmentations. Experiments show that our approach achieves better performances compared with other OMP-based sparse subspace clustering algorithms in terms of clustering accuracy, SEA ratio and representation quality, and keeps the time efficiency and anti-noise ability.

1 Introduction
Clustering is a common task in the fields of data science, image processing, machine learning, and computer vision, etc. In the big data applications, the analysis and processing of large-scale high-dimensional data becomes an important research direction [Patwary et al., 2015; Vandal et al., 2015].

In general, to represent complex data, a reasonable assumption is that high-dimensional data lie in a union of multiple low-dimensional subspaces. Sparse subspace clustering (SSC) [Elhamifar and Vidal, 2009] algorithms have become popular to segment the high-dimensional data from different subspaces into essentially low-dimensional subspaces and reflect the underlying nature of the data. It works by creating representation coefficients of data in low-dimensional subspace for compression to build sparse representation, construct affinity matrix based on the representation coefficient matrix, and obtain the clustering label of the data by spectral clustering. [Kim et al., 2007; Yin et al., 2016; Zhang, 2011].

Orthogonal matching pursuit (OMP) and its improved algorithms play an important role in data science and its applications such as sparse subspace clustering [Dai and Milenkovic, 2009; You et al., 2016; Elhamifar and Vidal, 2013; Do et al., 2008; Patel et al., 2013; Dyer et al., 2013]. Different from the basic SSC algorithm which uses convex programming tools to obtain the representation matrix, the OMP-based algorithms can relax problem requirements, decompose each data on the union of all data points and orthogonalize all the selected atoms at each step during decompositon, achieve faster convergence and offer better trade-off between clustering accuracy and computational efficiency.

However, the existing OMP-based approaches lack of data adaptiveness. When OMP-based algorithms are used to solve SSC problems [Donoho et al., 2012; Needell and Vershynin, 2010; Zhao et al., 2018], the parameters are often set manually by experience or according to the evaluation results after enumerating a group of optional parameters. The parameters do not adjust accordingly to the meaningful knowledge from the underlying data and remain the same for every data point regardless of its nature. It is noted that in the OMP-based algorithms when the data lies on the boundary between different categories, its dictionary atoms may be chosen from several subspaces to represent it. Since property of the dictionary is important for sparse representation [Xu et al., 2014], in this case, the data may be mislabeled resulting in low representation quality, low clustering accuracy and waste of computation resources.

Theoretically, by adding a data-sensitive selection step, the nondata-adaptive methods could become data-adaptive [Esling and Agon, 2012; Hubbard et al., 2016; Barakat et al., 2013]. It is the motivation for this paper to improve accuracy and preserve time efficiency of OMP-based Sparse Subspace Clustering. In this paper a novel approach to enhance the data-adaptive capability for OMP-based sparse subspace clustering is proposed, also a parameter selection process to adjust parameters based on data distribution is developed to expand the existing framework. Meanwhile, a specific parameter preprocessing algorithm is proposed. After preprocessing parameters, the framework pays more attention to data that meets the potential laws of its belonged category and
reduce the consumption when processing data that is far away from its belonged category in the processing of high-dimensional multi-category data.

The contributions of this paper are as follows:

- We propose a new approach for OMP-based sparse subspace clustering which add a parameter selection process in front of the existing framework.
- We propose a parameter preprocessing algorithm in the parameter selection process, dealing with the important parameter of OMP-based algorithms: dictionary size.
- We define a new metric Self-Expressive-Affinity(SEA) ratio to measure the sparse representation conversion efficiency for spectral clustering to obtain data segmentations.

The rest of this paper is organized as follows: Section 2 briefly introduces the previous basic framework and proposes a new approach by adding a parameter selection process. Section 3 proposes a parameter preprocessing algorithm, and gives relevant explanations. Section 4 introduces some widely used evaluation metrics of SSC problems, and defines the SEA ratio to measure the sparse representation conversion efficiency from self-expressive matrix to affinity matrix. Section 5 verifies our theoretical analysis through experiments on real-world datasets, effectiveness, efficiency and anti-noise ability of our proposed approach and algorithm. Finally, Section 6 concludes the paper.

2 Our New Approach

In this section, we briefly introduce the previous sparse subspace clustering framework and proposes a new data-adaptive approach by adding a parameter selection process in front of the existing framework, then give an example to illustrate difference between previous framework and our approach.

2.1 Previous Framework

Previous basic framework of SSC can be described as: establish a subspace representation model for given data, search the representation coefficients of the data in low-dimensional subspace, then construct the affinity matrix according to the representation coefficient matrix, finally obtain the clustering results by spectral clustering method.

There are three major steps in the previous basic framework of sparse subspace clustering: data input, subspace sparse representation and spectral clustering. As shown in Figure 1, \( X = \{x_1, \ldots, x_N \} \) is the input data, \( \forall x_i \in X \) \((i = 1, \ldots, N)\), a set \( X_i' \) is selected from \( X - i \) due to \( X \) and preset parameters, where \( X - i \) denotes \( X \) with the point \( x_i \) removed (avoid trivial solution). If \( x_i \) belongs to a set \( S_i \) (low-dimensional subspace, denotes a category), ideally \( X_i' \subseteq S_i \), then the sparse representation of \( x_i \) is called subspace preserving, \( c_i \) denotes representation coefficients, comes from the projection of \( x_i \) onto the subspace spanned by points in \( X_i' \), define self-expressive coefficient matrix \( C = [c_1, \ldots, c_N] \), affinity matrix \( A \) is constructed according to \( C \), usually \( A = [C \cdot C^T] \). Finally, applying spectral clustering [von Luxburg, 2007] to \( A \) to obtain data segmentations.

Parameters in previous framework are often set manually. No matter how to set, parameters do not adjust accordingly to the meaningful knowledge from the underlying data and remain the same for every data point regardless of its nature. It is noted that in the OMP-based algorithms when the data lies on the boundary between different categories, its dictionary atoms may be chosen from several subspaces to represent it. Since property of the dictionary is important for sparse representation, in this case, the data may be mislabeled resulting in low representation quality, low clustering accuracy and waste of computation resources.

2.2 New Approach

In our proposed new approach, we add a parameter selection process in front of existing framework for OMP-based sparse subspace clustering, after the data input step and before the subspace sparse representation step (Figure 1). In anticipation of future algorithms that fit in our approach, we suppose they should preprocess the parameters they focus on by observing data \( X \)'s attributes such as distribution, and dealing with the originally “static” parameters, make parameters more in line with the underlying nature of data.

In previous framework, all data points are co-owners of parameters, each of the co-owner shares same parameters; in our approach, every data point is the owner of its exclusive parameters, the exclusiveness comes from parameter selection process based on data \( X \)'s underlying nature.
Now we give an example to illustrate difference between previous framework and our approach. As shown in Figure 2. (a) and (b), circles in different colors represent data points belong to different clusters, red dashed line represent the selection of each point choosing atoms to build dictionary for self-expression. Histogram on the lower right of Figure 2. (a) shows dictionary size of all points, under previous framework, dictionary size of all points are the same. Thus we observe some points are quite possible to be mislabeled due to their dictionary atoms are chosen from several subspaces, e.g. in the fuzzy region or lies on the boundary between different categories. Based on our parameter selection process, we make adaptive adjustment to dictionary size, histogram on the lower left of Figure 2. (b) shows dictionary size. When it is adjusted accordingly to the distribution of underlying data, each point choosing atoms more wisely and reduce the chances of making clustering error.

3 Our New Algorithm

In this section, we propose a parameter preprocessing algorithm in the parameter selection process, dealing with an important parameter of OMP-based algorithms: dictionary size.

The algorithm we propose for our approach is illustrated in Algorithm 1 [Pati et al., 1993]. Algorithm 1 needs 2 inputs: input data matrix $X = [x_1, \ldots, x_N]$ and dictionary size $K$. To begin with, $X' \times X$ ($X \in \mathbb{R}^{\text{dim} \times N}$) gives us a matrix $\text{DIST}$, the i-th row and j-th column of which is the inner product of $x_i$ and $x_j$, regarded as the distance between data points $x_i$ and $x_j$, $\text{dim}$ denotes data dimension, $N$ denotes scale of data. Then sort all rows of $\text{DIST}$ in descending order, take the 2 to k elements of each row. These elements represent the nearest $K-1$ points of every data point but itself (to avoid trivial solution), if all data points have been normalized to same length, then in our case by “nearest” we mean “have smallest angle between two points”. By calculating the mean of each row, we have a vector $\text{DIST} \in \mathbb{R}^N$ denotes mean distance from each data point to the nearest $K$-1 points. Then calculate MaxD and MinD as the max and min value among vector $\text{DIST}$ respectively. Using $K$, MaxD, MinD to normalize vector $\text{DIST}$ to a range, of mean and range(defined as max-min) to $K$. The output of Algorithm 1 is an array, calculated by adding an offset to $\text{DIST}$ to make its mean value approximately equal to $K$. The i-th element of the output array denotes the exclusive dictionary size of $x_i$.

```
Algorithm 1 Parameter-K Preprocessing

Input: matrix $X = [x_1, \ldots, x_N]$, dictionary size $K$
Output: an array of new $K$: $K_{\text{array}}$

1: $\text{DIST} \leftarrow X' \times X$  // matrix product of $X'$ and $X$
2: Sort all rows of $\text{DIST}$ in descending order
3: $\text{DIST} \leftarrow \text{DIST}(:, 2:K)$  // Take 2-$K$ columns of $\text{DIST}$
4: $\text{DIST} \leftarrow \text{mean of each row of } \text{DIST}$
5: MaxD $\leftarrow$ max value among $\text{DIST}$
6: MinD $\leftarrow$ min value among $\text{DIST}$
7: $\text{DIST} \leftarrow K \times (\text{DIST} - \text{MinD}) / (\text{MaxD} - \text{MinD})$
8: $K_{\text{array}} \leftarrow K - \text{round}((\text{mean}(\text{DIST})) + \text{round}(\text{DIST}))$
9: return $K_{\text{array}}$
```

The key idea of our algorithm is to let parameter-K do accordingly adjustments based on data distribution to enhance the data-adaptive capability (fits for the parameter selection process). When processing high-dimensional multi-category data, our new algorithm let SSC framework pays more attention to well-distributed data that meets the potential laws of its belonged category and let them choose more atoms to express themselves, to be fully functional. And pays less attention to data that lies on the boundary between different categories and is more likely to be mislabeled, downsize their dictionary, suppress their expression.

There are no loops in our algorithm, only includes some simple functions such as matrix product, digit rounding, mean/max/min value of an array, etc. That is to say our parameter preprocessing algorithm barely brings additional computation. Considering of the mean value of algorithm output is about the original dictionary size $K$, we believe that our algorithm preserves time efficiency.

Through comparison between step 2 in Algorithm 3 and step 2-3 in Algorithm 4, we can clearly see how our parameter preprocessing algorithm works. Algorithm 3 is SSC-OMP algorithm under previous framework and Algorithm 4 combines Parameter-K Preprocessing algorithm and SSC-OMP algorithm. Besides adding a piece of codes of parameter preprocessing step, there only make little change to current codes to upgrade the previous framework with our new approach. This is very convenient. In fact, our parameter selection process can efficiently coordinate with any OMP-based methods to improve the clustering performance, not only with SSC-OMP algorithm.
Algorithm 2 OMP
Input: matrix $X = [x_1, \ldots, x_n]$, vector $y$, $K$, threshold
Output: $X$’s self-expression $c^*$
1: Let $i = 0$, residual $q_0 = y$, support set $T_0 = \emptyset$.
2: while $k < K$ and $||q_k||_2 > \text{threshold}$ do
3: $T_{k+1} = T_k \cup \{i^*\}$, where $i^* = \arg \max_{i=1,\ldots,n} |x_i^*|^2 q_k$.
4: $q_{k+1} = (I - P_{T_{k+1}}) y$, where $P_{T_{k+1}}$ is the projection onto the span of the vectors $\{x_i\} \in T_{k+1}$.
5: $k \leftarrow k + 1$
6: end while
7: $c^* = \arg \min_{c : \supp(C) \in \mathbb{K}} \|y - Xc\|_2$.
8: return $c^*$

Algorithm 3 SSC-OMP
Input: matrix $X = [x_1, \ldots, x_n]$, $K$, threshold
Output: segmentation of data $X$
1: Normalize data $X$ to a range.  // usually 0~1
2: Compute $c^*_1$ from OMP($X_{-i}, x_i$, $K$, threshold) using Algorithm 2.
   // $X_{-i}$ denotes $X$ with the point $x_i$ removed
3: Set $C' = [c^*_1, \ldots, c^*_K]$ and $A = [c^*_1] + [c^*_2]$.
4: Compute segmentation from $A$ by spectral clustering.

Algorithm 4 Parameter-K Preprocessing + SSC-OMP
Input: matrix $X = [x_1, \ldots, x_n]$, $K$, threshold
Output: segmentation of data $X$
1: Normalize data $X$ to a range.  // usually 0~1
2: Compute $K^*$ from OMP($X$, $K$) using Algorithm 1.
3: Compute $c^*_i$ from OMP($X_{-i}, x_i$, $K^*(i)$, threshold) using Algorithm 2.
   // $X_{-i}$ denotes $X$ with the point $x_i$ removed
4: Set $C' = [c^*_1, \ldots, c^*_K]$ and $A = [c^*_1] + [c^*_2]$.
5: Compute segmentation from $A$ by spectral clustering.

4 Metrics
In this section, we introduce some widely used evaluation metrics of SSC problems, and defines a new metric Self-Expressive-Affinity (SEA) ratio to measure the sparse representation conversion efficiency from self-expressive matrix to affinity matrix for spectral clustering.

- **Clustering accuracy (ACCR%)**
  The percentage of correctly labeled data points.
  We believe that as number of clusters decreases, ACCR increases.

- **Running time (TIME)**
  Run all clustering tasks using @Matlab.
  We believe that as number of clusters, samples per cluster or dictionary size increases, TIME increases.

- **Connectivity (CONN)**
  The CONN denotes whether the data points in each cluster form a connected component of the graph. For an undirected graph with degree matrix $D$ and adjacency matrix $W$, we use the second smallest eigenvalue of the normalized Laplacian matrix $L = I - D^{-1/2}WD^{-1/2}$ to measure the connectivity of the graph. Theoretically, CONN $\in [0, n^{-1}]$, CONN = 0 if and only if the graph is not connected [Chung, 1997]. In our case, we compute the algebraic connectivity for each cluster and take the min value among them as the measure of connectivity.
  We believe that as number of clusters decreases, CONN increases.

- **Percentage of subspace-preserving representations (PERC%)**
  PERC is a direct measure of whether the solution is subspace-preserving or not, stands for the percentage of points whose representations are subspace-preserving. Ideally, PERC = 100, only if a subspace-preserving solution is given.
  We believe that as number of clusters decreases, PERC increases.

- **Subspace-preserving representation error (SSR%)**
  SSR measures how close the coefficients are from being subspace preserving. A subspace-preserving coefficient matrix gives SSR = 0.
  We believe that as number of clusters increases, SSR increases.

We use ACCR and TIME to evaluate the performance of subspace clustering methods, use CONN to evaluate the connectivity of undirected graph, and use PERC and SSR to evaluate the representation quality. Greater ACCR, CONN, PERC or smaller TIME, SSR gives better clustering results.

Next we develop a new evaluation metric to measure the conversion efficiency from self-expressive matrix to affinity matrix.

**Definition 1.** Let $G$ be a graph, denote the self-expressive matrix of $G$ by $C$, then use $A = |C| + |C^T|$ to denote the affinity matrix to apply spectral clustering. Finally, the Self-Expressive-Affinity (SEA) ratio, is defined as $\frac{\text{nnz}(A)}{2 \times \text{nnz}(C)}$, where $\text{nnz}(C)$ represents the number of non-zero elements in matrix $C$.

**Proposition 1.** $\text{SEA} \in [0, 1]$, $\text{SEA} = 0.5$ if and only if $C$ is symmetric, $\text{SEA} = 1$ if and only if $\forall i \neq j, C_{ij} \wedge C_{ji} \neq 1$, where $C_{ij}$ represents the element in $i$-th row and $j$-th column of matrix $C$, $\wedge$ stands for logical conjunction (AND).

**Proof.** Because $A = |C| + |C^T|$, hence $\text{nnz}(A) \geq \text{nnz}(C)$, by definition $\text{SEA} \geq 0.5$. $\text{SEA} = 0.5$ if and only if $\text{nnz}(A) = \text{nnz}(C)$, then $C$ must be symmetric. $A = |C| + |C^T|$ mean $\text{nnz}(A) \leq \text{nnz}(C) + \text{nnz}(C^T) = 2 \times \text{nnz}(C)$, by definition $\text{SEA} \leq 1$. $\text{SEA} = 1$ if and only if $\forall i \neq j, C_{ij} \neq 0$, then $C_{ij} = 0$. Considering in some cases $C_{ij} = C_{ji} = 0$, it is the equivalent of $\forall i \neq j, C_{ij} \wedge C_{ji} = 1$.

To avoid $C_{ij} \wedge C_{ji} = 1$, somehow alleviate redundant representation or wastes of representation chances for affinity matrix $A$, which means theoretically greater SEA gives better clustering results, thus ideally $\text{SEA} = 1$. 


In this section, we describe the datasets that we used for experiments and give details about the experimental settings, then evaluate performances on real-world datasets and verify our preceding theoretical analysis, show that our new approach and algorithm achieves better performances in terms of clustering accuracy (ACCR), running time (TIME), percentage of subspace-preserving representations (PERC), subspace-preserving representation error (SSR), connectivity (CONN), and Self-Expressive-Affinity (SEA) ratio compared with other OMP-based sparse subspace clustering algorithms. Our new approach also performs well in the anti-noise experiments.

Experimental hardware environment: Intel® Core™ i7-6700 CPU@3.40 GHz with Intel® HD Graphics 530, and Windows 10.0.17134 operating system. Software development platform: ®Matlab R2017b.

5.1 Experiments on Extended Yale B Dataset

Extended Yale B dataset [Georghiades et al., 2001] contains frontal face images of 38 individuals, each individual has about 64 different photos, taken from the same viewpoint under varying illumination conditions, each of size 192 × 168.

We have made the code available at: https://github.com/zwapimeow/ijcai2019.git

### Table 1: experiment results on Extended Yale B dataset

| (cluster) | SSC | OMP | ours + OMP | ROMP | ours + ROMP |
|-----------|-----|-----|------------|------|------------|
|            | 5   | 15  | 25         | 35   | 5          | 15  | 25  | 35   |
| clustering accuracy (%) | 73.98 | 55.42 | 42.99 | 38.30 | 6.80 | 52.27 | 123.57 | 207.08 |
| running time (sec.)       | 89.13 | 80.35 | 76.70 | 75.23 | 0.34 | 4.57  | 17.00  | 29.68  |
| self-expressive-affinity ratio (%) | 95.59 | 82.16 | 78.99 | 77.44 | 0.32 | 4.68  | 14.41  | 29.14  |
| connectivity (×10^2)      | 94.91 | 86.27 | 84.22 | 79.58 | 1.84 | 19.80 | 44.95  | 108.56 |
| percentage of subspace-preserving representations (%) | 95.86 | 89.91 | 84.90 | 80.38 | 1.88 | 16.44 | 44.88  | 87.52  |
| subspace-preserving representation error (%) | 61.62 | 64.53 | 63.90 | 63.59 | 3.88 | 2.14  | 0.40   | 0      |
| (cluster) | SSC | OMP | ours + OMP | ROMP | ours + ROMP |
|            | 5   | 15  | 25         | 35   | 5          | 15  | 25  | 35   |
| clustering accuracy (%) | 61.62 | 64.53 | 63.90 | 63.59 | 3.88 | 2.14  | 0.40   | 0      |
| running time (sec.)       | 89.27 | 91.22 | 92.09 | 92.59 | 7.94 | 3.75  | 1.09   | 0.08   |
| self-expressive-affinity ratio (%) | 90.63 | 92.09 | 92.77 | 93.14 | 7.81 | 2.59  | 0      | 0      |
| connectivity (×10^2)      | 90.68 | 91.94 | 92.60 | 93.07 | 13.14 | 5.65  | 1.22   | 0.24   |
| percentage of subspace-preserving representations (%) | 92.34 | 93.34 | 93.97 | 94.39 | 11.47 | 2.84  | 0      | 0      |
| subspace-preserving representation error (%) | 26.90 | 4.96  | 1.61  | 0.70  | 7.67  | 17.44 | 27.42  | 34.42  |
| (cluster) | SSC | OMP | ours + OMP | ROMP | ours + ROMP |
|            | 5   | 15  | 25         | 35   | 5          | 15  | 25  | 35   |
| clustering accuracy (%) | 2.05 | 0.28 | 0.07 | 0.05 | 12.80 | 18.15 | 20.94 | 22.60 |
| running time (sec.)       | 2.98 | 0.55 | 0.31 | 0.22 | 11.26 | 17.12 | 19.94 | 21.73 |
| self-expressive-affinity ratio (%) | 1.78 | 0.14 | 0.03 | 0.01 | 13.04 | 20.01 | 23.54 | 25.76 |
| connectivity (×10^2)      | 3.32 | 0.62 | 0.37 | 0.20 | 12.51 | 19.76 | 23.14 | 25.46 |

Figure 3: anti-noise experiments on Extended Yale B dataset

In experiments, all images are downsampled to 48 × 42 pixels and reshaped as vectors of size dim = 2016. We randomly pick number of clusters: n ∈ {5,15,25,35} individuals and take all the images as the data to be clustered, then compute 20 times for each n. Meanwhile we set threshold value: thr = 10^-6 and dictionary size: K = 8. Table 1 shows the evaluation performance of 5 methods on the face clustering problem: Sparse Subspace Clustering (convex optimization),
Experiments on USPS Dataset

In this part, we use a standard dataset for handwritten digit recognition, USPS\(^2\), consists of 8-bit grayscale images of number 0 through 9, so here we have \(n = 10\). All images in USPS dataset are \(16 \times 16\) grayscale pixels, we reshape each sample as vectors of size \(\text{dim} = 256\) and randomly pick sample \(\in \{250, 500, 800, 1100\}\) in each cluster, meanwhile set \(K \in \{8, 16, 24, 32\}\) and \(\theta_r = 10^{-6}\). When we randomly select 5 numbers for testing and take the averages after 20 times of calculation, we get results in Table 2. These results show the difference between Orthogonal Matching Pursuit (OMP) and our new algorithm+OMP (ours+OMP). The evaluation metrics are ACCR, TIME and SEA ratio.

We can observe that ours+OMP always performs better in terms of ACCR and SEA. When \(K\) or sample per cluster changes, the ACCR of ours+OMP always remain \(7\%\)~\(12\%\) higher than OMP, and SEA also remain higher than OMP. That is to say when \(K\) or sample per cluster changes, the advantages of our approach are unaffected. We also observe that TIME is barely changed, proves that our approach keeps computational efficiency.

Table 2: experiment results on USPS dataset

| (samples /cluster) | ACCR (%) | TIME (sec.) | SEA (%) |
|-------------------|----------|-------------|---------|
|                   | OMP      | ours + OMP  | OMP     | ours + OMP  | OMP     | ours + OMP  |
| K=8               |          |             |         |             |         |             |
| 250               | 58.39    | 66.40       | 0.59    | 0.56        | 92.16   | 92.69       |
| 500               | 57.48    | 65.68       | 2.28    | 2.33        | 93.70   | 94.08       |
| 800               | 51.06    | 60.03       | 9.54    | 10.27       | 94.55   | 94.81       |
| 1100              | 48.82    | 55.70       | 27.60   | 24.65       | 94.97   | 95.20       |
| K=16              |          |             |         |             |         |             |
| 250               | 56.18    | 67.06       | 1.21    | 1.17        | 92.16   | 94.24       |
| 500               | 57.14    | 65.72       | 4.29    | 4.25        | 95.29   | 95.73       |
| 800               | 49.96    | 59.78       | 18.48   | 18.77       | 96.14   | 96.47       |
| 1100              | 43.70    | 53.18       | 42.02   | 44.65       | 96.58   | 96.85       |
| K=24              |          |             |         |             |         |             |
| 250               | 57.70    | 66.80       | 2.05    | 2.00        | 94.05   | 94.75       |
| 500               | 57.05    | 65.73       | 7.01    | 4.47        | 95.88   | 95.88       |
| 800               | 47.72    | 57.92       | 27.60   | 27.45       | 96.73   | 97.07       |
| 1100              | 42.15    | 50.97       | 65.10   | 64.05       | 97.18   | 97.45       |
| K=32              |          |             |         |             |         |             |
| 250               | 58.16    | 65.42       | 3.47    | 3.52        | 94.22   | 94.99       |
| 500               | 54.47    | 66.00       | 9.41    | 9.39        | 96.16   | 96.65       |
| 800               | 47.09    | 58.44       | 34.80   | 38.82       | 97.04   | 97.40       |
| 1100              | 42.34    | 50.93       | 115.33  | 90.10       | 97.50   | 97.78       |

Orthogonal Matching Pursuit (OMP), Rotated Orthogonal Matching Pursuit (ROMP) [Zhong et al., 2018], our new algorithm+OMP (ours+OMP) and our new algorithm+ROMP (ours+ROMP). The evaluation metrics are: ACCR, TIME, PERC, SSR, CONN and SEA ratio.

In terms of ACCR and SEA, we can observe better performance when adding our algorithm in front of other methods in all cases, and it’s always much better than the performance of SSC. Though SSC has best performance of PERC and SSR, our method brings improvement and multiply PERC & reduce SSR of OMP and ROMP, especially when number of clusters gets greater. Although adding our algorithm increases the amount of computation, but it’s barely influence computational efficiency, when number of clusters gets greater, we can even observe the shortening of TIME. Since our algorithm specifically dispises outliers in data, resulting in slightly reduce CONN of subspaces, but basically it’s still higher than SSC.

In addition, we also carried out an anti-noise experiments studies ACCR. Set \(n = 5\) and \(K = 8\), using the imnoise function in Matlab to add Gaussian white noise of mean and variance default to 0 and 0.01 respectively to all images, while the noise rate: \(\sigma \in \{0, 0.1, ... , 0.9\}\). The results are plotted in Figure 3, when raising sigma, the ACCR of ours+OMP always remains \(1\%\)~\(5\%\) higher than ROMP, and when sigma < 0.5, the ACCR of ours+OMP always remains \(1\%\)~\(6\%\) higher than OMP. These results are pretty good, because when sigma \(\geq 0.6\), face images have low quality, generally people do not use such images in clustering problem.

5.2 Experiments on USPS Dataset

6 Conclusions

In the paper we propose a novel approach to enhance the data-adaptive capability for OMP-based sparse subspace clustering. A parameter selection process is proposed to expand the existing framework, and a specific parameter preprocessing algorithm is proposed to adjust parameters based on data distribution. Experiments verifies effectiveness, efficiency and anti-noise ability of our approach.

Acknowledgments

This work was supported in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing), in part by Shenzhen International cooperative research projects GJHZ20170313150021171, and in part by NSFC-Shenzhen Robot Jointed Founding (U1613215).

\(^2\) http://www.cs.nyu.edu/~roweis/data.html
References

[Patwary et al., 2015] M. M. A. Patwary, S. Byna, N. R. Satish, N. Sundaram, Z. Lukic, V. Roytershteyn, M. J. Anderson, Y. S. Yao, Prabhat, and P. Dubey. BD-CATS: Big Data Clustering at Trillion Particle Scale. Proceedings of Sc15: The International Conference for High Performance Computing, Networking, Storage and Analysis, 2015.

[Vandal et al., 2015] T. Vandal, D. McDuff, and R. El Kaliouby. Event Detection: Ultra Large-scale Clustering of Facial Expressions. 2015 11th Ieee International Conference and Workshops on Automatic Face and Gesture Recognition (Fg), Vol. 2, 2015.

[Elhamifar and Vidal, 2009] E. Elhamifar, and R. Vidal. Sparse Subspace Clustering. Cypr: 2009 Ieee Conference on Computer Vision and Pattern Recognition, Vols 1-4, 2782-2789, 2009.

[Kim et al., 2007] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An Interior-Point Method for Large-Scale l(1)-Regularized Least Squares. Ieee Journal of Selected Topics in Signal Processing, 1(4):606-617, Dec 2007.

[Yin et al., 2016] M. Yin, Y. Guo, J. B. Gao, Z. S. He, and S. L. Xie. Kernel Sparse Subspace Clustering on Symmetric Positive Definite Manifolds. 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cypr), 5157-5164, 2016.

[Zhang, 2011] T. Zhang. Sparse Recovery With Orthogonal Matching Pursuit Under RIP. Ieee Transactions on Information Theory, 57(9):6215-6221, Sep 2011.

[You et al., 2016] C. You, D. P. Robinson, and R. Vidal. Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit. 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cypr), 3918-3927, 2016.

[Elhamifar and Vidal, 2013] E. Elhamifar, and R. Vidal. Sparse Subspace Clustering: Algorithm, Theory, and Applications. Ieee Transactions on Pattern Analysis and Machine Intelligence, 35(11):2765-2781, Nov 2013.

[Do et al., 2008] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran. Sparsity Adaptive Matching Pursuit Algorithm for Practical Compressed Sensing. 2008 42nd Asilomar Conference on Signals, Systems and Computers, Vols I-4, 581-587, 2008.

[Patel et al., 2013] V. M. Patel, H. V. Nguyen, and R. Vidal. Latent Space Sparse Subspace Clustering. 2013 Ieee International Conference on Computer Vision (Iccv), 225-232, 2013.

[Dyer et al., 2013] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk. Greedy Feature Selection for Subspace Clustering. Journal of Machine Learning Research, 14(2487-2517), Sep 2013.

[Donoho et al., 2012] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck. Sparse Solution of Underdetermined Systems of Linear Equations by stagewise orthogonal matching pursuit. Ieee Transactions on Information Theory, 58(2):1094-1121, Feb 2012.

[Needell and Vershynin, 2010] D. Needell, and R. Vershynin. Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit. Ieee Journal of Selected Topics in Signal Processing, 4(2):310-316, Apr 2010.

[Zhao et al., 2018] C. C. Zhao, W. L. Hwang, C. L. Lin, and W. H. Chen. Greedy orthogonal matching pursuit for subspace clustering to improve graph connectivity. Information Sciences, 459(135-148), Aug 2018.

[Xu et al., 2014] J. Xu, H. He, and H. Man. Active Dictionary Learning in Sparse Representation Based Classification. Computer Vision and Pattern Recognition, 2014.

[Esl engaging and Agon, 2012] P. Esleng, and C. Agon. Time-Series Data Mining. Acm Computing Surveys, 45(1):Nov 2012.

[Hubbard et al., 2016] A. E. Hubbard, S. Kherad-Pajouh, and M. J. van der Laan. Statistical Inference for Data Adaptive Target Parameters. International Journal of Biostatistics, 12(1):3-19, May 2016.

[Barakat et al., 2013] M. Barakat, D. Lefebvre, M. Khalil, F. Druaux, and O. Mustapha. Parameter selection algorithm with self adaptive growing neural network classifier for diagnosis issues. International Journal of Machine Learning and Cybernetics, 4(3):217-233, Jun 2013.

[von Luxburg, 2007] U. von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4):395-416, Dec 2007.

[Pati et al., 1993] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, 40-44, 1993.

[Chung, 1997] F. Chung. Spectral graph theory. CBMS Regional Conference Series in Mathematics, 1997.

[Georghiades et al., 2001] 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 Transactions on Pattern Analysis and Machine Intelligence, 23(6):643-660, Jun 2001.

[Zhong et al., 2018] L. Zhong, Y. S. Zhu, and G. B. Luo. A New Sparse Subspace Clustering by Rotated Orthogonal Matching Pursuit. Ieee International Conference on Image Processing, 3853-3857, 7-10 Oct 2018.