석사 학위논문

Master's Thesis

대용량 부분 공간 클러스터링 반복 알고리즘에 대한 연구

Scalable Iterative Algorithm for Robust Subspace Clustering: Convergence and Initialization

전 상 혁 (全 相 赫 Chun, SangHyuk)

전기 및 전자공학부

School of Electrical Engineering

KAIST

2016

MEE
20143583

전 상 혁. Chun, SangHyuk. Scalable Iterative Algorithm for Robust Subspace Clustering: Convergence and Initialization. 대용량 부분 공간 클러스터링 반복 알고리즘에 대한 연구. School of Electrical Engineering . 2016. 24p. Advisor Prof. Jinwoo Shin. Text in English.

ABSTRACT

Subspace Clustering (SC), a generalized task of Principle Component Analysis (PCA), has been used extensively for dimensionality reduction of high-dimensional data. Recently, several methods have been proposed to enhance the robustness of PCA and SC, but most of them are computationally expensive, especially for high-dimensional large-scale data. In this paper, we develop a much faster algorithm for optimizing the NP-hard SC objective using a sum of the α-th power of ℓ_2-norm with $0 < \alpha \leq 2$, where $\alpha = 2$ is the standard choice and $\alpha < 2$ enhances the robustness of SC. The known implementations achieving a local optimum of the objective would be costly due to the alternation of two separate tasks: optimal cluster-membership assignment and optimal subspace selection, while the substitution of one process to a faster surrogate can cause failure in convergence. Furthermore, such an alternating method has been often criticized due to the sensitivity of initialization. To address the issues, our proposed algorithm has the following key features: (a) release nested robust PCA loops for subspace update, (b) use a simplified singular value decomposition that only requires a few matrix-vector multiplications instead of solving an expensive eigenvector problem and (c) initialize carefully to avoid poor clustering. We prove that it monotonically converges to a local minimum of the SC objective for any $0 < \alpha \leq 2$ and finds the true clustering under a statistical assumption on data. In our experiments, it is shown to converge at an order of magnitude faster than the standard implementation optimizing the same objective, and outperforms known SC methods in the literature for MNIST handwritten digit dataset.

Contents

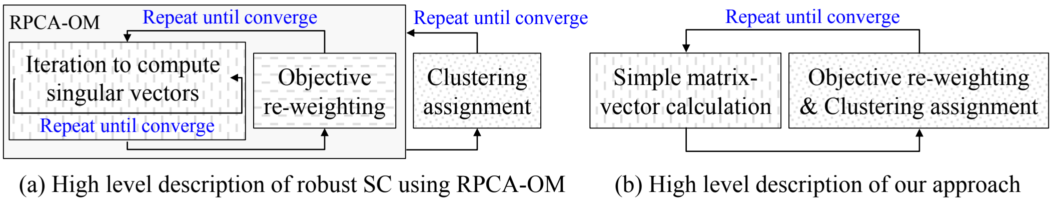
List of Tables

List of Figures

Chapter 1. Introduction

The investigation of regularity in high-dimensional data generally seeks a low-dimensional representation with a few simple rules that explains dominant properties of the system [1, 2]. *Subspace Clustering* (SC) is one of popular methods for dimensionality reduction, where it generalizes *Principal Component Analysis* (PCA) [3] by finding multiple low-dimensional subspaces instead of single one. SC has found numerous applications in computer vision [4, 5, 6, 7], DNA microarray analysis [8, 9], text-mining [10, 11], recommendation systems [12, 13] and system identification [14]. We refer the survey paper [7] on the SC literature for interested readers.

However, the true low-dimensional structure often cannot be retrieved due to noise. In particular from a few large anomalies or the complete corruption of elements, even when most data follow the regularity in the low-dimensional subspace. To alleviate the effect of such noise, an algorithm needs to incorporate robustness by using less information from data abnormally separated from the majority of data. Canonical methods for incorporating such robustness have appeared in improving PCA. The *squared* ℓ_2-norm in the traditional PCA objective has been replaced by different ones to enhance the robustness, e.g., ℓ_1-norm [15, 16, 17, 18, 12] and *non-squared* ℓ_2-norm [19, 20, 21]. In general, the optimization with ℓ_1-norm uses non-analytic methods, whereas that with (squared or non-squared) ℓ_2-norm can use analytic solutions such as *Singular Value Decomposition* (SVD) that makes the algorithm interpretable and scalable with appropriate approximation. The additional benefit of ℓ_2-norm is providing rotational invariance [19].



(a) High level description of robust SC using RPCA-OM (b) High level description of our approach

Figure 1.1: Comparison of (a) EM implementation of subspace clustering for $\alpha = 1$ and (b) our approach for subspace clustering for $0 < \alpha \leq 2$

In this paper, we study robust SC methods via optimizing a sum of the α-th power of ℓ_2-norm with $0 < \alpha \leq 2$, under the assumption that data are composed of m clusters of different low-dimensional affine subspaces. The classical non-squared and squared ℓ_2-norms correspond to $\alpha = 1$ and $\alpha = 2$, respectively, where we study more general setups and smaller α that can be used to further enhance robustness. This optimization task is NP-hard and generally performed via alternating two steps of Expectation Maximization (EM) style: optimal cluster-membership assignment and affine subspace selection for each cluster. In particular for $\alpha = 1$, a sum of non-squared ℓ_2-norm objective, the affine subspace selection, called *Robust PCA with Optimal Mean* (RPCA-OM), was recently studied [21], while that of linear subspace selection was developed much earlier [19]. The authors of RPCA-OM propose an alternating procedure between SVD iterations and objective re-weighting. High-level description of the EM implementation using RPCA-OM is in Figure 1.1 (a). Here, RPCA-OM becomes a major bottleneck, and the naive back-and-forth optimization is infeasible for large-scale high-dimensional data.

In particular, due to expensive computation of SVD and slow convergence of the nested RPCA-OM loops.

Contribution: To optimize the SC objective using the α-th power of ℓ_2-norm, we first design a generalized version of RPCA-OM that works for any $0 < \alpha \leq 2$, which we call α-PCA where 1-PCA coincides with RPCA-OM. Then, we modify the optimization in α-RPCA for subspace selection and dramatically reduce the running time over the naive EM scheme. Instead of performing exact SVD calculations in α-PCA, we use a simplified procedure requiring a few matrix-vector multiplications for subspace update, where it is a few iterations of the subspace-iteration method (also called *orthogonal-iteration*) [22] using QR decomposition. Here, the main assumption is that the matrix-vector multiplication can be done efficiently. We further release nested α-PCA loops, i.e., perform the clustering assignment without waiting for the convergence of subspace update. Our approach is summarized in Figure 1.1 (b).

Despite these modified efficient procedures, we prove that the proposed iterative algorithm monotonically converges to a local optimum like the EM scheme. As we reported in Chapter 5, our proposed algorithm computes a solution of the same quality with those computed by the EM scheme at an order of magnitude faster. Such a monotone convergence is not clear even for $m = 1$ and $\alpha = 1$, where the authors of RPCA-OM recently established it. The cases $\alpha \neq 1$ or $m \neq 1$ provide additional technical challenges in establishing the monotone convergence, where our proof utilizes structural properties of QR decompositions and certain monotonicity in terms of α. The spirit of our idea "Don't wait convergence for convergence" for such efficient subspace selection is similar, for example, to contrastive divergence learning [23] and inexact augmented Lagrange multiplier [24]. However we emphasize that their correct convergences do not hold in general.

Moreover, we also propose an initialization method to avoid poor clustering, where iterative optimization-based SC algorithms without careful initialization have been often criticized in the literature [7]. The proposed initialization can be thought as a randomized version of that used in [25]. The randomness enhances the robustness against outliers as we show via experiments, and moreover we prove that the initialization indeed recovers the true clustering under a statistical assumption on data points.

We perform extensive experiments comparing the performances of the proposed scheme and other prior SC methods. They show that it converges at an order of magnitude faster than known algorithms optimizing the same SC objective, and outperforms prior SC methods in accuracy for MNIST handwritten digit datasets. As we evidenced in experiments, our proposed scheme is in particular attractive for large-scale datasets due to its low complexity, and we believe that it would find numerous applications involving large-scale robust dimensionality reduction.

Organization: Chapter 2 introduces the α-SC optimization task, and explain the EM scheme to solve it. In Chapter 3, we propose an efficient iterative algorithm optimizing the α-SC objective and its initialization method to avoid poor clustering. The proof on theoretical results of the algorithm and the initialization method is presented in Chapter 4. We report our experimental results in Chapter 5.

Chapter 2. Preliminaries

Let $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$ denote a data matrix of n data points in d dimensional space. PCA is the problem of finding a low-dimensional subspace representation that best describes high-dimensional data points. There have been extensive efforts in developing more robust PCA methods in the literature [19, 12, 20, 21] by replacing the squared ℓ_2-norm by the ℓ_1-norm or the non-squared ℓ_2-norm in the optimizing objective. However, most works on this line ignored the center parameter in their optimization tasks, and robust PCA which jointly considers the center and subspace parameters was recently studied [21] for optimizing the non squared ℓ_2-norm objective. In this paper, we study a more generalized setup by introducing parameter $0 < \alpha \le 2$ as follows:

$$\alpha\textbf{-PCA:} \quad \min_{b,U:\, U^\top U = I} \sum_{i=1}^{n} \|(I - UU^\top)(x_i - b)\|_2^\alpha, \tag{2.1}$$

where $b \in \mathbb{R}^d$ and $U \in \mathbb{R}^{d\times r}$ $(r \le d)$ are center and basis parameters respectively. The prior PCAs with non-squared and squared ℓ_2-norms correspond to $\alpha = 1$ and $\alpha = 2$, respectively, where smaller α can be used to enhance its robustness. The above PCA optimization (2.1) is naturally generalized to the following optimization for subspace clustering:

$$\alpha\textbf{-SC:} \quad \min_{\substack{[w_{ij}],[b_j],[U_j]: \\ U_j^\top U_j = I,\, \sum_{j=1}^m w_{ij} = 1}} \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} \|(I - U_j U_j^\top)(x_i - b_j)\|_2^\alpha, \tag{2.2}$$

where b_j, U_j indicate center and basis variables for the j-th subspace and $w_{ij} \in \{0, 1\}$ does a cluster membership. We assume that the number m of clusters and the dimension r of subspaces are given.

Since the optimization task (2.2) is NP-hard, it is impossible to compute a global optimum in general. The popular approximation algorithm is an alternating method of EM type: alternatively update w_{ij} via taking the best cluster j for each data point x_i and b, U from a PCA solver in each cluster, which is formally described as follows.

```
SC-EM  EM implementation for $\alpha$-SC
Input: Data $X \in \mathbb{R}^{d\times n}$, number of clusters $m$ and dimension of subspace $r$.
Output: Cluster membership variable $w_{ij}$, center vector $b_j$ and low dimensional subspace $U_j$.
Initialize: Initialize $w_{ij}$ randomly
repeat
  for all j do
    1. Update $b_j, U_j$ using current data points in cluster $j$ by solving $\alpha$-PCA.
  end for
  for all i, j do
    2. Update clustering information $w_{ij}$ by finding the best cluster for each data point.
  end for
until converge
```

If $\alpha = 2$, an optimal solution of α-**PCA** can be computed by Singular Value Decomposition (SVD). Using this observation, one can easily prove that **SC-EM** converges to a local optimum of (2.2) when $\alpha = 2$. On the other hand, if $\alpha \neq 2$, it is far from being clear how to design a α-**PCA** solver. An iterative re-weighting algorithm was proposed [21] for $\alpha = 1$ case, and the authors prove that it converges monotonically to a local optimum of the α-**PCA** objective (2.1). We generalize the algorithm for any $0 < \alpha \leq 2$ and call it **PCA-IR** which is exactly same as standard PCA using SVD when $\alpha = 2$ and same as RPCA-OM [21] when $\alpha = 1$. In **PCA-IR**, $\mathbf{1}$ denotes the column vector with entire elements being one.

```
PCA-IR  Iterative re-weighting algorithm for $\alpha$-PCA
  Input: Data $X \in \mathbb{R}^{d\times n}$, dimension of subspace $r$.
  Output: Center vector $b$ and low dimensional subspace $U$.
  Initialize: Set $D$ as an identity matrix.
  repeat
    1. Update $b$ as $\frac{XD\mathbf{1}}{\mathbf{1}^\top D\mathbf{1}}$ and $U$ as the $r$ largest singular vectors of $XD - \frac{D\mathbf{1}\mathbf{1}^\top D}{\mathbf{1}^\top D\mathbf{1}}$.
    2. Update the diagonal matrix $D$ as its $i$-th element is set by $\frac{\alpha}{2}\|(I - UU^\top)(x_i - b)\|_2^{(\alpha-2)}$.
  until converge
```

The main idea of **PCA-IR** is that a sum of the squared ℓ_2-norm objective is relatively easy while solving a sum of the α-th power of ℓ_2-norm objective is difficult. In each iteration of **PCA-IR**, instead of solving original objective (2.1), we solve much easier objective

$$\min_{b,U:\,U^\top U=I} \sum_{i=1}^{n} d_i \|(I - UU^\top)(x_i - b)\|_2^2.$$

where $d_i = \frac{\alpha}{2}\|(I - UU^\top)(x_i - b)\|_2^{(\alpha-2)}$. After update b and U using SVD, we update d_i and alternatively update b, U and d_i until the objective is converged.

We remark that the monotone and local convergence proof [21] does not directly generalize for **PCA-IR**. In this paper, we do not prove the monotone and local convergence property of **PCA-IR**, but one can easily prove it using similar approach to the proof of Theorem 1. If one use **PCA-IR** as a subroutine of **SC-EM**, it might be slow for high-dimensional (large d) and large-scale (large n) data because they require many SVD calls until its nested **PCA-IR** converges. The goal of this paper is to develop a more efficient algorithm for optimizing the α-**SC** objective (2.2) in addition to designing a careful initialization method to avoid poor clusterings.

Chapter 3. Scalable Iterative Algorithm for Robust Subspace Clustering

3.1 Algorithm description and monotone convergence

In this section, we describe our proposed iterative algorithms for optimizing the α-**SC** optimization (2.2) for any choice of $\alpha \in (0, 2]$.

SC-SI Scalable iterative algorithm for α-**SC**

```
Input: Data $X \in \mathbb{R}^{d\times n}$, number of clusters $m$, dimension of subspace $r$, a positive integer $k$.
Output: Cluster membership variable $w_{ij}$, center vector $b_j$ and low dimensional subspace $U_j$.
Initialize: Initialize $b_j, U_j, w_{ij}$ and the diagonal matrix $D_j$ using SC-IN.
repeat
  for all j do
    1. Update $b_j \leftarrow \frac{XD_jW_j\mathbf{1}}{\mathbf{1}^\top D_jW_j\mathbf{1}}$ where $W_j$ is the diagonal matrix whose $i$-th element is $w_{ij}$.
    2. Update $U_j$ by the following steps where $k$ is some positive integer.
      repeat
        $U_j \leftarrow XH_jH_j^\top X^\top U_j$ where $H_j = D_jW_j - \frac{D_jW_j\mathbf{1}\mathbf{1}^\top D_jW_j}{\mathbf{1}^\top D_jW_j\mathbf{1}}$.
        $U_j \leftarrow$ Orthonomalization of $U_j$ by QR decomposition.
      until $k$ times
  end for
  for all i, j do
    3. Update the $i$-th element of diagonal matrix $D_j$ by $\frac{\alpha}{2}\|\left(I - U_jU_j^\top\right)(x_i - b_j)\|_2^{(\alpha-2)}$.
    4. Update $w_{ij} \leftarrow 1$ if $j \in \arg\min_\ell \|\left(I - U_\ell U_\ell^\top\right)(x_i - b_\ell)\|_2^\alpha$ and $w_{ij} \leftarrow 0$ otherwise.
  end for
until converge
```

Each iteration of **SC-SI** is designed to solve the following optimization instead of (2.2) directly:

$$\min_{\substack{[b_j],[U_j],[w_{ij}]:\\ \sum_{j=1}^m w_{ij}=1; U_j^\top U_j = I}} \sum_{i=1}^n \sum_{j=1}^m d_{ij} w_{ij} \|\left(I - U_jU_j^\top\right)(x_i - b_j)\|_2^2, \tag{3.1}$$

where d_{ij} is the i-th diagonal element of D_j that appears in Step 3 of the algorithm. By taking the derivative with respect to b_j and setting it to zero, one can obtain the optimal form of b_j that appears in Step 1 of the algorithm. Substituting this form of b_j into (3.1) and by using $\|A\|_F^2 = \mathtt{tr}(AA^\top)$ where $\|A\|_F$ and $\mathtt{tr}(A)$ denotes Frobenius norm and trace of matrix A respectively, we have

$$\max_{[U_j],[w_{ij}]:U_j^\top U_j = I} \sum_j \mathtt{tr}\left(U_j^\top XH_jH_j^\top X^\top U_j\right), \tag{3.2}$$

where H_j is defined in Step 2 of the algorithm. The optimal U_j of the above optimization can be computed by SVD of XH_j, which might be computationally expensive. Instead, **SC-SI** update U_j by using inexact SVD performs only k iterations of the 'subspace-iteration method' [22] which involves a few matrix-vector multiplications in Step 2. Furthermore, it uses U_j at the previous iteration as the initial point at the next iteration. Hence, if one chooses small k, e.g., $k = 1$, the computational cost of each iteration is much smaller compared to that performing SVD exactly, in particular for high dimensional data sets. Step 3 is designed for re-weighting the squared ℓ_2-norm in (3.1) to the α-th power of ℓ_2-norm in (2.2), and Step 4 rearranges data points to clusters using updated parameters. We note that for $\alpha = 2$, the classical setup, Step 3 is not necessary because D is always the identity matrix.

At a high level, the algorithm couples intermediate steps of three computational tasks: inexact SVD (Step 1, 2), objective re-weighting (Step 3) and clustering assignments (Step 4). Compared to **SC-EM**, **SC-SI** has much faster subspace selections because of (a) **SC-SI** uses a cheap inexact SVD procedure in Step 2 and (b) **SC-SI** does not waits until the nested PCA procedure converges. Moreover, since the algorithm only require simple matrix-vector multiplication, **SC-SI** requires much less memory than **SC-EM**. Despite such computational efficiency, we prove that **SC-SI** also has the same desired monotone convergence property stated as follows.

Theorem 1. *For any $\alpha \in (0, 2]$,* **SC-SI** *monotonically decreases the objective value of (2.2) at each iteration. Furthermore, it converges to a local minimum of (2.2).*

The proof of Theorem 1 is presented in Section 4.1. We remark that even if **SC-SI** converges monotonically, it does not mean the convergence to a local optimum of (2.2) because it optimizes a different objective in (3.1). Our proof strategy is similar to that in [21], but establishing the monotone convergence of **SC-SI** imposes additional issues due to the parameter α and the modified procedures that do not exist in [21]. To address the issues, we utilize some structural properties of QR decomposition and certain monotonicity in terms of α.

3.2 Initialization

The clustering performance of **SC-SI** algorithm is sensitive to how U_j, w_{ij} and D_j are initialized. In this section, we aim for designing careful initialization techniques to avoid poor clustering. To this end, we propose the initialization method, named **SC-IN**. It is intuitively natural: it iteratively finds next center vectors with further distances probabilistically, where the distance is calculated using the previously chosen subspaces and center vectors. A similar idea for the linear subspace setting was used in [25], but the authors deterministically chooses next furthest centers as opposed to ours. The probabilistic nature in our initialization makes it more robust to outliers. Specifically, the parameter β decides the trade-off between robustness and accuracy, where a choice of lower $\beta > 0$ provide more robustness of the proposed initialization. This is similar to the role of α in the α-**SC** objective. For practical purpose, we choose β via cross-validation.

SC-IN Initialization method for **SC-SI**

```
Input: Data $X \in \mathbb{R}^{d \times n}$, number of clusters $m$, dimension of subspace $r$, $\beta > 0$, $R_c > 0$ and $\widehat{N} \geq r$.
Output: $\widehat{b}_j, \widehat{U}_j, \widehat{w}_{ij}$ and $\widehat{D}_j$.
for j = 1 : $m$ do
    1. If j = 1, then choose the first center vector $\widehat{b}_1$ uniformly random from data points $X$. Otherwise, choose $\widehat{b}_j$ from data points as $\widehat{b}_j = x_i$ with probability $\frac{f_D(x_i,\beta)}{\sum_{i'} f_D(x_{i'},\beta)}$, where
       $$f_D(x,\beta) = \min_{1 \leq j' \leq j-1} \|(I - U_{j'} U_{j'}^\top)(x - b_{j'})\|_2^\beta.$$
    2. Let $\widehat{X}_j$ be randomly chosen $\widehat{N}$ points from the set of data whose distance from $\widehat{b}_j$ is less than $R_c$.
    3. Update $\widehat{b}_j$ as $\widehat{b}_j \leftarrow \frac{1}{\widehat{N}_j} \widehat{X}_j \mathbf{1}$.
    4. Update $\widehat{U}_j$ as $\widehat{U}_j \leftarrow r$ largest singular vectors of $\widehat{X}_j - \widehat{b}_j \mathbf{1}^\top$.
end for
for all i, j do
    5. Set the $i$-th element of diagonal matrix $\widehat{D}_j$ by $\frac{\alpha}{2} \| \left(I - U_j U_j^\top\right)(x_i - b_j) \|_2^{(\alpha-2)}$.
    6. Set $\widehat{w}_{ij} \leftarrow 1$ if $j \in \arg\min_\ell \| \left(I - U_\ell U_\ell^\top\right)(x_i - b_\ell) \|_2^\alpha$ and $\widehat{w}_{ij} \leftarrow 0$ otherwise.
end for
```

Note that Step 2 of **SC-IN** is not practical but necessary for theoretical guarantee. In practice, we suggest to modify the algorithm by setting $\widehat{X}_j$ as N_c number of nearest neighbors of b_j instead of selecting data whose distance between b_j is less than R_c. In other words, Step 2 of modified practical initialization algorithm looks like the following: 'Let $\widehat{X}_j$ be randomly chosen $\widehat{N}$ data points from N_c nearest neighbors of $\widehat{b}_j$'.

Modified initialization method works well in practice but Theorem 2 is not satisfied any more, because it does not guarantee uniform randomness of each column of $\widehat{X}_j$. Detailed experimental setting is described in Chapter 5.

We obtain the following provable performance guarantee of the initialization method under a statistical assumption on data points.

Theorem 2. *Suppose there are m number of clusters where the j-th cluster has an optimal subspace $U_j \in \mathbb{R}^{d\times r}$ and a center $b_j \in \mathbb{R}^d$. Suppose all U_j are independent. i.e., $U_\ell^\top U_j = U_j^\top U_\ell = 0$ for all $\ell \neq j$. Also suppose all b_j satisfy $4\sqrt{r(1+\varepsilon^2)} \leq \|b_j - b_\ell\|_2$ for all $j \neq \ell$. Let $U_{j\perp} \in \mathbb{R}^{d\times(d-r)}$ be the orthogonal subspace of U_j, i.e.,*

$$U_jU_j^\top + U_{j\perp}U_{j\perp}^\top = I \text{ and } U_j^\top U_{j\perp} = U_{j\perp}^\top U_j = 0.$$

For the j-th cluster, draw $\frac{n-n_o}{m}$ points and each point is independently randomly generated as

$$U_js + U_{j\perp}e + b_j$$

where s and e uniformly chosen in $[-1,1]^r$ and $\left[-\sqrt{\frac{r}{d-r}}\varepsilon, \sqrt{\frac{r}{d-r}}\varepsilon\right]^{d-r}$ respectively. Let $\mathcal{C}_j$ be the set of points in the j-th cluster and $\mathcal{C}_o$ be the set of data points not in clusters, namely, it is a set of outliers. We suppose that number of outliers is n_o where every $x_o \in \mathcal{C}_o$ and $x \notin \mathcal{C}_o$ satisfies $\|x_o - x\|_2 > 2\sqrt{r(1+\varepsilon^2)}$ for all j. Also we assume that there exists d_o such that $d_o \geq \|x_o\|_2$ for all $x_o \in \mathcal{C}_o$. Suppose that $\frac{n-n_o}{m}$ is asymptotically large number and $d \gg r$. Let $\widehat{C}_j$ be the set of points in the j-th cluster which is found by **SC-IN**. *Then when $d \to \infty$ and $\widehat{N}/d \to \lambda \in (0,1)$, if we choose $R_c = 2\sqrt{r(1+\varepsilon^2)}$ for* **SC-IN** *algorithm, $\widehat{C}_1, \ldots, \widehat{C}_m$ and $\mathcal{C}_1, \ldots, \mathcal{C}_m$ are isomorphic with probability at least*

$$\frac{n-n_o}{n}\prod_{j=1}^{m-1}\frac{(m-j)(\alpha_2)^\beta}{j(\alpha_1)^\beta + (m-j)(\alpha_2)^\beta + \frac{m}{n-n_o}n_od_o^\beta}$$

where $\zeta = 2\varepsilon\sqrt{\frac{r}{d-r}}\frac{\sqrt{\widehat{N}}+\sqrt{d-r}}{\sqrt{\widehat{N}}-\sqrt{r}}$, $\alpha_1 = \zeta\sqrt{r} + \varepsilon\sqrt{r}$ and $\alpha_2 = \sqrt{|1-\zeta^2|}\left(4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}\right)$.

The proof of Theorem 2 is presented in Section 4.2. Theorem 2 guarantees the performance of our initialization method. Approximately, α_1, α_2 and d_o indicate how noise, signal and outliers are strong respectively. According to Theorem 2, smaller ε and d_o, in other words less noise and outliers, makes the lower bound more tighter. One interesting observation is when α_2 is larger than α_1 and d_o, larger β enhance overall performance. However, if α_2 is smaller than others, then proper small β enhance robustness of the initialization method. In the real world, a power of noises is much smaller than a power of signals while a power of outliers is much larger than a power of signals in many cases. Therefore, one can use large β when there only exists noise and small β when there exists any outliers.

Moreover, assume that $n_o = 0$ and $\varepsilon = 0$ which deduce $\zeta = 0$, $\alpha_1 = 0$, $\alpha_2 = 3\sqrt{r}$ and $d_o = 0$. This observation means the lower bound of Theorem 2 becomes 1 which means we always can recover true clustering by using **SC-IN** and so that **SC-SI** recovers true clustering. However, even if under the same condition, random initialization cannot guarantee true clustering for every initialization. For example, if we choose $b_1, \ldots, b_m$ in the same cluster, with high probability, **SC-SI** cannot recover true clustering.

Chapter 4. Proofs of Theorems

4.1 Proof of Theorem 1

4.1.1 Proof of the monotone convergence of SC-SI

In this subsection, we show that the **SC-SI** algorithm monotonically decreases the objective (2.2). As we discuss in Chapter 3.1, it is clear that w_{ij} and b_j are updated to decreases the objective (3.1). On the other hand, it is not clear that U_j is updated to decrease the objective in (3.1) since the algorithm runs only a few iterations of the subspace-iteration method instead of SVD. To address this issue, we first state the following key lemmas whose proofs are given in Section 4.1.3.

Lemma 3. *Consider the following subspace-iteration for a positive semi-definite matrix* $A \in \mathbb{R}^{d\times d}$

$$U_A(1) \leftarrow \textit{Orthonomalization of } AU(0) \textit{ by QR decomposition},$$

where $U(0) \in \mathbb{R}^{d\times d}$ *is an orthogonal matrix. Then, for any* $r \le d$*, it follows that*

$$\sum_{i=1}^{r} U(0)_i^\top A U(0)_i \le \sum_{i=1}^{r} U_A(1)_i{}^\top A U_A(1)_i,$$

where $U(0)_i, U_A(1)_i$ *denote the* i*-th column vectors of* $U(0), U_A(1)$*, respectively.*

Lemma 4. *For any* $x, y \ge 0$ *and* $\alpha \in (0, 2]$,

$$x^\alpha - \frac{\alpha x^2}{2y^{2-\alpha}} \le y^\alpha - \frac{\alpha y^2}{2y^{2-\alpha}}.$$

Now we are ready to complete the monotone decreasing property of **SC-SI**. Let W, D, B, U and $\widehat{W}, \widehat{D}, \widehat{B}, \widehat{U}$ be the current and updated (after one iteration) values of **SC-SI**. From Lemma 3 for positive semi-definite matrix $XH_jH_j^\top X^\top$, it follows that

$$\sum_j \mathtt{tr}(\widehat{U}_j^\top XH_jH_j^\top X^\top \widehat{U}_j) \ge \sum_j \mathtt{tr}(U_j^\top XH_jH_j^\top X^\top U_j).$$

Combining the above inequality and the optimality of $\widehat{B}$ leads to

$$\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}d_{ij}\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_i - \widehat{b}_j)\|_2^2 \le \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}d_{ij}\|(I - U_jU_j^\top)(x_i - b_j)\|_2^2. \tag{4.1}$$

By using the definition of d_{ij}, (4.1) reduces to

$$\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\frac{\alpha\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_i - \widehat{b}_j)\|_2^2}{2\|(I - U_jU_j^\top)(x_i - b_j)\|_2^{(2-\alpha)}} \le \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\frac{\alpha\|(I - U_jU_j^\top)(x_i - b_j)\|_2^2}{2\|(I - U_jU_j^\top)(x_i - b_j)\|_2^{(2-\alpha)}}. \tag{4.2}$$

We further use Lemma 4 to obtain

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{j=1}^{m}\left[w_{ij}\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_i - \widehat{b}_j)\|_2^\alpha - w_{ij}\frac{\alpha\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_i - \widehat{b}_j)\|_2^2}{2\ \|(I - U_jU_j^\top)(x_i - b_j)\|_2^{2-\alpha}}\right] \\ &\quad\le \sum_{i=1}^{n}\sum_{j=1}^{m}\left[w_{ij}\|(I - U_jU_j^\top)(x_i - b_j)\|_2^\alpha - w_{ij}\frac{\alpha\|(I - U_jU_j^\top)(x_i - b_j)\|_2^2}{2\ \|(I - U_jU_j^\top)(x_i - b_j)\|_2^{2-\alpha}}\right]. \end{aligned} \tag{4.3}$$

Finally, from (4.2), (4.3) and the updating rule of w_{ij}, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\widehat{w}_{ij}\|(I-\widehat{U}_j\widehat{U}_j^\top)(x_i-\widehat{b}_j)\|_2^\alpha \le \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\|(I-U_jU_j^\top)(x_i-b_j)\|_2^\alpha .$$

This completes the proof of the monotone decreasing property of **SC-SI**.

4.1.2 Proof of the convergence of SC-SI to a local optimum

The previous subsection guarantees the convergence of **SC-SI**. In this subsection, we show that the **SC-SI** algorithm converges to a local optimum of (2.2), i.e., the convergence point satisfies the KKT condition of (2.2).

First, the Lagrangian function of (2.2) is the following

$$\begin{aligned}&\mathcal{L}_1([w_{ij}],[b_j],[U_j],[\Lambda_j],[\lambda_j],[\mu_{ij}])\\ &=\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\|(I-U_jU_j^\top)(x_i-b_j)\|_2^\alpha-\sum_{j=1}^{m}\mathtt{tr}\big((U_j^\top U_j-I)\Lambda_j\big)\\ &-\sum_{i=1}^{n}\big(\lambda_i\sum_{j=1}^{m}(w_{ij}-1)\big)-\sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}w_{ij}(w_{ij}-1),\end{aligned}$$

where Λ,λ and μ is Lagrange multipliers. Taking derivative with respect to w_{ij}, U_j and b_j respectively and setting them to zero, one can obtain the KKT condition of (2.2) as the follows

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial w_{i'j'}}&=\|(I-U_{j'}U_{j'}^\top)(x_{i'}-b_{j'})\|_2^\alpha-\lambda_{i'}-\mu_{i'j'}=0\\ \frac{\partial\mathcal{L}_1}{\partial U_{j'}}&=\sum_{i=1}^{n}\alpha\ w_{ij'}\frac{(I-U_{j'}U_{j'}^\top)(x_i-b_{j'})(b_{j'}-x_i)^\top U_{j'}}{\|(I-U_{j'}U_{j'}^\top)(x_i-b_{j'})\|_2^{2-\alpha}}-U_{j'}\Lambda_{j'}=0\\ \frac{\partial\mathcal{L}_1}{\partial b_{j'}}&=\sum_{i=1}^{n}\alpha\ w_{ij'}\frac{(I-U_{j'}U_{j'}^\top)(b_{j'}-x_i)}{\|(I-U_{j'}U_{j'}^\top)(x_i-b_{j'})\|_2^{2-\alpha}}=0\end{aligned}$$

On the other hand, the Lagrangian function of (3.1) is the following

$$\begin{aligned}&\mathcal{L}_2([w_{ij}],[b_j],[U_j],[\Lambda_j],[\lambda_j],[\mu_{ij}])\\ &=\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}w_{ij}\|(I-U_jU_j^\top)(x_i-b_j)\|_2^2-\sum_{j=1}^{m}\mathtt{tr}\big((U_j^\top U_j-I)\Lambda_j\big)\\ &-\sum_{i=1}^{n}\big(\lambda_i\sum_{j=1}^{m}(w_{ij}-1)\big)-\sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}w_{ij}(w_{ij}-1).\end{aligned}$$

Again, taking derivative with respect to w_{ij}, U_j and b_j respectively and setting them to zero, one can obtain the KKT condition of (3.1)

$$\begin{aligned}\frac{\partial\mathcal{L}_2}{\partial w_{i'j'}}&=d_{i'j'}\|(I-U_{j'}U_{j'}^\top)(x_{i'}-b_{j'})\|_2^2-\lambda_{i'}-\mu_{i'j'}=0\\ \frac{\partial\mathcal{L}_2}{\partial U_{j'}}&=\sum_{i=1}^{n}2d_{ij'}w_{ij'}(I-U_{j'}U_{j'}^\top)(x_i-b_{j'})(b_{j'}-x_i)^\top U_{j'}-U_{j'}\Lambda_{j'}=0\\ \frac{\partial\mathcal{L}_2}{\partial b_{j'}}&=\sum_{i=1}^{n}2d_{ij'}w_{ij'}(I-U_{j'}U_{j'}^\top)(b_{j'}-x_i)=0\end{aligned}$$

By substitute definition of $d_{ij}=\frac{\alpha}{2}\|\left(I-U_jU_j^\top\right)(x_i-b_j)\|_2^{(\alpha-2)}$ into the above conditions (since the algorithm converges), the above equations are equivalent to the KKT condition of (2.2). This concludes that the **SC-SI** algorithm converges to a local minimum of (2.2).

4.1.3 Proofs of Lemma 3 and Lemma 4

Proof of Lemma 3. To begin with, the subspace-iteration can be summarized as follows

$$U_A(1)R_A = A\,U(0),$$

where R_A is an upper triangular matrix, namely, it is a QR decomposition of $A\,U(0)$. Since A is positive semi-definite, there exists a unique positive semi-definite square-root matrix $B \in \mathbb{R}^{d\times d}$, or simply, $A = B^2$. Let us define $U_B(0) = U(0)$ and consider the two steps of subspace-iterations for B

$$U_B(1)R_B(1) = BU_B(0), \qquad U_B(2)R_B(2) = BU_B(1),$$

which implies that $U_B(2)R_B(2)R_B(1) = A\,U(0)$. Hence, without loss of generality, one can conclude that

$$U_B(2) = U_A(1). \tag{4.4}$$

Furthermore, let us define $B(k) := U_B(k)^\top BU_B(k)$, $Q_B(k+1) := U_B(k)^\top U_B(k+1)$ and observe that

$$\begin{aligned}
&B(0) = U_B(0)^\top BU_B(0) = U_B(0)^\top U_B(1)R_B(1) = Q_B(1)R_B(1),\\
&B(1) = U_B(1)^\top BU_B(1) \;= Q_B(1)^\top U_B(0)^\top BU_B(0)Q_B(1) = Q_B(1)^\top B(0)Q_B(1) \;= R_B(1)Q_B(1).
\end{aligned}$$

Therefore, since $B(0), B(1)$ are positive semi-definite, we have

$$B(0)^2 = R_B(1)^\top R_B(1), \qquad B(1)^2 = R_B(1)R_B(1)^\top.$$

Now we let $\mathtt{sub}[M]$ be the r-th principal submatrix of a matrix $M \in \mathbb{R}^{d\times d}$ which is obtained from M by removing the $(r+1)$-th to d-th rows and columns. Then, observe that

$$\mathtt{tr}\left(\mathtt{sub}\left[B(k)^2\right]\right) = \mathtt{tr}\left(\mathtt{sub}\left[U_B(k)^\top B^2 U_B(k)\right]\right) = \sum_{i=1}^{r} U_B(k)_i^\top B^2\, U_B(k)_i = \sum_{i=1}^{r} U_B(k)^\top A\, U_B(k)_i, \tag{4.5}$$

Since $R(1)$ is an upper triangular matrix, it follows that

$$\begin{aligned}
&\mathtt{tr}(\mathtt{sub}\left[B(1)^2\right]) - \mathtt{tr}(\mathtt{sub}\left[B(0)^2\right]) = \mathtt{tr}(\mathtt{sub}\left[R(1)R(1)^\top\right]) - \mathtt{tr}(\mathtt{sub}\left[R(1)^\top R(1)\right])\\
&= \sum_{i=1}^{r}\sum_{j=i}^{d}(R(1))_{ij}^2 - \sum_{i=1}^{r}\sum_{j=i}^{r}(R(1))_{ij}^2 = \sum_{i=1}^{r}(\sum_{j=i}^{d}(R(1))_{ij}^2 - \sum_{i=1}^{r}(R(1))_{ij}^2) = \sum_{i=1}^{r}\sum_{j=r+1}^{d}(R(1))_{ij}^2 \geq 0,
\end{aligned} \tag{4.6}$$

where $(M)_{ij}$ is the (i,j)-th element of a matrix M. Similarly, one can show that

$$\mathtt{tr}(\mathtt{sub}\left[B(2)^2\right]) - \mathtt{tr}(\mathtt{sub}\left[B(1)^2\right]) \geq 0. \tag{4.7}$$

Combining (4.4), (4.5), (4.6) and (4.7) leads to the conclusion of Lemma 3. □

Proof of Lemma 4. To begin with, let us define $f(x,y,\alpha)$ as the following

$$f(x,y,\alpha) := x^2 - \frac{2}{\alpha}x^\alpha y^{2-\alpha} + \left(\frac{2}{\alpha} - 1\right)y^2.$$

We will prove that $f(x,y,\alpha) \geq 0$ is satisfied for $\forall x,y \geq 0$ and $\forall \alpha \in (0,2]$ and that leads to the proof of the lemma. Taking derivative $f(x,y,\alpha)$ w.r.t. α, we obtain

$$\frac{\partial f(x,y,\alpha)}{\partial \alpha} = \frac{2y^2}{\alpha^2}\left[\left(\frac{x}{y}\right)^\alpha\left(1 + \ln\left(\frac{y}{x}\right)^\alpha\right) - 1\right].$$

Let $z := \left(\frac{y}{x}\right)^\alpha$, $C := \frac{2y^2}{\alpha^2}$ and substitute z, C into the above equation then we obtain

$$\frac{\partial f(x,y,\alpha)}{\partial \alpha} = \frac{C}{z}\left(1 + \ln z - z\right).$$

Since $C/z \geq 0$ for $\forall x, y, \alpha \geq 0$ and $1 + \ln z - z \leq 0$ for $\forall z \geq 0$, $\frac{\partial f(x,y,\alpha)}{\partial \alpha} \leq 0$ for $\forall x, y, \alpha \geq 0$. Therefore $f(x,y,\alpha)$ is decreasing function on α and it is easy to show that $f(x,y,\alpha) = 0$ when $\alpha = 2$. Thus $f(x,y,\alpha) \geq 0$ for $\alpha \in (0,2]$. □

4.2 Proof of Theorem 2

4.2.1 Proof of performance guarantee of SC-IN

In this section, we show that **SC-IN** guarantees successful clustering provably under a statistical assumption on data points. We first state the following key lemmas whose proof is given in Section 4.2.2. For Lemma 5 and 6, we define $\zeta = 2\varepsilon\sqrt{\frac{r}{d-r}}\frac{\sqrt{\widehat{N}}+\sqrt{d-r}}{\sqrt{\widehat{N}}-\sqrt{r}}$, $\alpha_1 = \zeta\sqrt{r} + \varepsilon\sqrt{r}$ and $\alpha_2 = \sqrt{|1-\zeta^2|}\left(4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}\right)$.

Lemma 5. *Under the same setting of Theorem 2, Let $\widehat{b}_j$ be a point of cluster j whose optimal center vector is b_j and optimal subspace is U_j. Suppose we run Step 2 to 4 of* **SC-IN** *while fixing $\widehat{b}_j$. Also suppose that we choose $R_c = 2\sqrt{r(1+\varepsilon^2)}$, $\widehat{N} > 0$ and $\beta > 0$. Let $\widehat{U}_j$ and $\widehat{b}_j$ be outputs of the algorithm. Then when $d \to \infty$ and $\widehat{N}/d \to \lambda \in (0,1)$, following inequality is satisfied.*

$$\|U_j^\top \widehat{U}_{j\perp}\|_2 \leq \zeta.$$

Lemma 5 shows that 'distance' between an optimal subspace U_j and a reconstructed subspace $\widehat{U}_j$ by **SC-IN** is upper bounded in terms of SNR ε, number of sampled data $\widehat{N}$, dimension of the data d and dimension of subspaces r.

Lemma 6. *Under the same setting of Lemma 5, following inequalities hold for all j and $\ell \neq j$.*

$$\begin{aligned}\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_{ij} - \widehat{b}_j)\|_2 &\leq \alpha_1,\\ \|(I - \widehat{U}_j\widehat{U}_j^\top)(x_{i\ell} - \widehat{b}_j)\|_2 &\geq \alpha_2.\end{aligned}$$

The first inequality of Lemma 6 shows that any data point in same cluster with current center vector has distance less or equal than α_1, while the second inequality shows that any data point in different cluster with current center vector has distance greater or equal than α_2. Therefore, by using Lemma 6, we can calculate the probability to choose the next center vector in different clusters which do not include already chosen center vectors.

Now we are ready to complete proof of Theorem 2. Suppose we randomly choose first center vector b_1 from not in $\mathcal{C}_o$ which happens with probability $(n - n_o)/n$ Without loss of generality, let's say b_1 is chosen from cluster 1.

Then a data point x is chosen as b_2 by the probability of $\frac{f_D(x,\beta)}{\sum_i f_D(x_i,\beta)}$. If x is in the same cluster with b_1, then $f_D(x,\beta)$ is upper bounded by the first inequality of Lemma 6. Also, if x is in the different cluster with b_1, then $f_D(x,\beta)$ is bounded by the second inequality of the same Lemma. Finally, by the condition of Theorem 2, $f_D(x_{io},\beta) \leq d_o^\beta$ for all x_{io} in $\mathcal{C}_o$.

Using these observations, we have the lower bound of probability to choose next center vector b_2 from another cluster as the following

$$\begin{aligned}\text{Prob}(b_2 \text{ from cluster not } 1) &= \frac{\sum_{\ell\neq 1}\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i\ell},\beta)}{\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i1},\beta) + \sum_{\ell\neq 1}\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i\ell},\beta) + \sum_{i=1}^{n_o} f_D(x_{io},\beta)} \\ &\geq \frac{(m-1)\frac{n-n_o}{m}(\alpha_2)^\beta}{\frac{n-n_o}{m}(\alpha_1)^\beta + (m-1)\frac{(n-n_o)}{m}(\alpha_2)^\beta + n_o d_o^\beta}.\end{aligned}$$

Here, we emphasize that even in denominator term, lower bound of the probability to choose b_2 in the different cluster with b_1. It is because, the following function is a decreasing function on a

$$\frac{a}{a+b} = 1 - \frac{b}{a+b}.$$

Therefore, the following inequality is satisfied:

$$\frac{a}{a+b} \geq \frac{\min a}{\min a + \min b}.$$

Suppose we choose all j number of center vectors in different clusters but not in outliers. Once b_ℓ is chosen from $\mathcal{C}_\ell$, $f_D(x_{i\ell},\beta) \leq \alpha_1^\beta$ while $f_D(x,\beta) \geq \alpha_2^\beta$ for x in unchosen clusters. Therefore, the lower bound of probability to choose b_{j+1} from one of $(m-j)$ clusters which are not chosen before is given by

$$\begin{aligned}&\text{Prob}(b_{j+1} \text{ from cluster not } 1,\ldots,j) \\ &\qquad = \frac{\sum_{\ell=j}^{m}\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i\ell},\beta)}{\sum_{\ell=1}^{j-1}\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i\ell},\beta) + \sum_{\ell=j}^{m}\sum_{i=1}^{\frac{n-n_o}{m}} f_D(x_{i\ell},\beta) + \sum_{i=1}^{n_o} f_D(x_{io},\beta)} \\ &\qquad \geq \frac{(m-j)\frac{n-n_o}{m}(\alpha_2)^\beta}{j\frac{n-n_o}{m}(\alpha_1)^\beta + (m-j)\frac{(n-n_o)}{m}(\alpha_2)^\beta + n_o d_o^\beta}.\end{aligned}$$

By the mathematical Induction, the lower bound of probability to choose center vectors in different clusters is given by

$$\frac{n-n_o}{n}\prod_{j=1}^{m-1}\frac{(m-j)\frac{n-n_o}{m}(\alpha_2)^\beta}{j\frac{n-n_o}{m}(\alpha_1)^\beta + (m-j)\frac{n-n_o}{m}(\alpha_2)^\beta + n_o d_o^\beta}$$

This completes the proof of Theorem 2.

4.2.2 Proofs of Lemma 5 and Lemma 6

Proof of Lemma 5. To begin with, we first show that any data point x not in outliers, all data points whose distance between x is less than $2\sqrt{r(1+\varepsilon^2)}$ are all in same cluster.

For cluster j, each i-th point x_{ij} is independently randomly generated as $x_{ij} = U_j s_{ij} + U_{j\perp}\varepsilon_{ij} + b_j$ where s and e uniformly chosen in $[-1,1]^r$ and $\left[-\sqrt{\frac{r}{d-r}}\varepsilon, \sqrt{\frac{r}{d-r}}\varepsilon\right]^{d-r}$ respectively. Therefore, for all j, Euclidean distance between x_{ij} and b_j is always less or equal than $\sqrt{r(1+\varepsilon^2)}$ while minimum distance between b_j and b_ℓ for all $\ell \neq j$ is $4\sqrt{r(1+\varepsilon^2)}$. Therefore, the minimum distance between data points in different cluster is larger than $2\sqrt{r(1+\varepsilon^2)}$. In addition to, since we assume that $\|x_o - x\|_2 > 2\sqrt{r(1+\varepsilon^2)}$ for all j, $x_o \in \mathcal{C}_o$ and $x \notin \mathcal{C}_o$, if we assume that the center vector $\widehat{b}_j$ is chosen not in outliers, the set of data points whose distance between $\widehat{b}_j$ is less than $2\sqrt{r(1+\varepsilon^2)}$ is exactly same as $\mathcal{C}_j$. In the algorithm, $\widehat{X}_j$ is generated by uniformly randomly choosing $\widehat{N}$ number of data points from $\mathcal{C}_j$. From the observation, $\widehat{X}_j$ can be represented as the following.

$$\widehat{X}_j = U_j\widehat{S}_j + U_{j\perp}\widehat{E}_j + b_j\mathbf{1}^\top,$$

where each column vector of $\widehat{S}_j \in \mathbb{R}^{r\times\widehat{N}}$ and $\widehat{E}_j \in \mathbb{R}^{(d-r)\times\widehat{N}}$ are in $[-1,1]^r$ and $\left[-\sqrt{\frac{r}{d-r}}\varepsilon, \sqrt{\frac{r}{d-r}}\varepsilon\right]^{d-r}$ respectively.

Let's run Step 3 of the algorithm. Then now $\widehat{b}_j$ is updated as the mean of $\widehat{X}_j$, or $\widehat{b}_j = \frac{1}{\widehat{N}_j}\widehat{X}_j\mathbf{1}$. Let $\widehat{Y}_j = \widehat{X}_j - \widehat{b}_j\mathbf{1}^\top$. Then by the optimality of $\widehat{U}_j$, we have

$$\|\big(I-\widehat{U}_j\widehat{U}_j^\top\big)\widehat{Y}_j\|_2 \le \|\big(I-U_jU_j^\top\big)\widehat{Y}_j\|_2. \tag{4.8}$$

Also, by triangular inequality property of norm, we have

$$\|U_jU_j^\top\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2 - \|U_jU_j^\top\widehat{Y}_j - \widehat{Y}_j\|_2 \le \|\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2. \tag{4.9}$$

Combining (4.8) and (4.9), it follows that

$$\|U_jU_j^\top\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2 \le 2\|\big(I-U_jU_j^\top\big)\widehat{Y}_j\|_2 = 2\|U_{j\perp}U_{j\perp}^\top\widehat{Y}_j\|_2 = 2\sigma_1\big(U_{j\perp}^\top\widehat{Y}_j\big). \tag{4.10}$$

Since $U_{j\perp}^\top\widehat{Y}_j$ is a $r\times r$ matrix and $\widehat{U}_{j\perp}^\top\widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j = 0$,

$$\begin{aligned}
\|\widehat{U}_{j\perp}^\top U_j\|_2\sigma_r(U_j^\top\widehat{Y}_j) &\le \|\widehat{U}_{j\perp}^\top U_jU_j^\top\widehat{Y}_j\|_2\\
&= \|\widehat{U}_{j\perp}^\top U_jU_j^\top\widehat{Y}_j - \widehat{U}_{j\perp}^\top\widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2\\
&\le \|\widehat{U}_{j\perp}^\top\|_2\|U_jU_j^\top\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2\\
&= \|U_jU_j^\top\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|_2
\end{aligned}$$

From the above inequality and (4.10), we have

$$\|U_j^\top\widehat{U}_{j\perp}\|_2 \le \frac{\|U_jU_j^\top\widehat{Y}_j - \widehat{U}_j\widehat{U}_j^\top\widehat{Y}_j\|}{\sigma_r\big(U_j^\top\widehat{Y}_j\big)} \le 2\frac{\sigma_1\big(U_{j\perp}^\top\widehat{Y}_j\big)}{\sigma_r\big(U_j^\top\widehat{Y}_j\big)}. \tag{4.11}$$

From the definition of $\widehat{Y}_j$, we have

$$\begin{aligned}
U_j^\top\widehat{Y}_j &= U_j^\top\big(U_jS_j + U_{j\perp}E_j + (b_j-\widehat{b}_j)\mathbf{1}^\top\big) = S_j + U_j^\top(b_j-\widehat{b}_j)\mathbf{1}^\top\\
U_{j\perp}^\top\widehat{Y}_j &= U_{j\perp}^\top\big(U_jS_j + U_{j\perp}E_j + (b_j-\widehat{b}_j)\mathbf{1}^\top\big) = E_j + U_{j\perp}^\top(b_j-\widehat{b}_j)\mathbf{1}^\top.
\end{aligned} \tag{4.12}$$

Since each data point of $\widehat{X}_j$ is generated independently and identically distributed, $\widehat{Y}_j$ is a $d\times\widehat{N}$ random matrix whose entries are independent and identically distributed. Note that if we do not choose $\widehat{N}$ as $\frac{n-n_o}{m}$, entries of $\widehat{Y}_j$ is not independent because in **SC-IN** algorithm, we choose $\widehat{X}_j$ by distance from b_j not uniformly random in cluster j. Therefore, we can find upper bound and lower bound of singular value of $\widehat{Y}_j$ by using random matrix theory.

From Theorem 2 in [26], the smallest and largest singular value of a random matrix $A \in \mathbb{R}^{N\times n}$ whose entries are i.i.d. subgaussian random variable with zero mean and unit variance can be easily deduced by

$$\sigma_1(A) \to \sqrt{N}+\sqrt{n}, \sigma_n(A) \to \sqrt{N}-\sqrt{n}, \text{ as } n\to\infty \text{ and } n/N \to \lambda \in (0,1). \tag{4.13}$$

In addition, by Chernoff Bound we can bound $\|b_j - \widehat{b}_j\|_2$ as the following. Let's define $\widehat{s}_{ij}^{(k)}, \widehat{e}_{ij}^{(k)}$ as i-th entry of k-th element of $\widehat{E}_j$ and $\widehat{S}_j$ respectively. Then we have

$$\widehat{b}_j = b_j + \frac{1}{\widehat{N}}\sum_{k=1}^{\widehat{N}}(U_j\widehat{s}_{ij}^{(k)} + U_{j\perp}\widehat{e}_{ij}^{(k)}).$$

Let b_{ij} and $\widehat{b}_{ij}$ be i-th entry of b_j and $\widehat{b}_j$ respectively. Since $\widehat{s}_{ij}^{(k)}, \widehat{e}_{ij}^{(k)}$ are random variable, $\widehat{b}_{ij}$ is also random variable whose mean is b_{ij}. By applying Chernoff Bound to $\widehat{b}_{ij}$, we have

$$\mathbb{P}\left[|\widehat{b}_{ij} - b_{ij}| > \delta b_{ij}\right] > 1 - \exp(-\frac{5}{6}\delta^2 b_{ij}).$$

Let $\delta = \frac{1}{\widehat{N}}$, then we have

$$\mathbb{P}\left[|\widehat{b}_{ij} - b_{ij}| > \frac{b_{ij}}{\widehat{N}}\right] > 1 - \exp(-\frac{5}{6}\frac{b_{ij}}{\widehat{N}^2}).$$

Therefore as $\widehat{N} \to \infty$, $|\widehat{b}_{ij} - b_{ij}|$ goes to 0. Then $\|\widehat{b}_j - b_j\|_\infty = \max_i |\widehat{b}_{ij} - b_{ij}|$ goes to 0. Since ℓ_2 norm is always smaller than ℓ_∞ norm, we have

$$\|b_j - \widehat{b}_j\|_2 \leq \|b_j - \widehat{b}_j\|_\infty = \max_i |\widehat{b}_{ij} - b_{ij}| \to 0, \text{ as } \widehat{N} \to \infty.$$

From the above inequality, we have

$$\|(b_j - \widehat{b}_j)\mathbf{1}^\top\|_2 \leq \|b_j - \widehat{b}_j\|_2 \cdot \|\mathbf{1}^\top\|_2 \to 0, \text{ as } \widehat{N} \to \infty. \tag{4.14}$$

From (4.12), (4.13) and (4.14), when $d \to \infty$ and $\widehat{N}/d \to \lambda \in (0,1)$ we have

$$\begin{aligned} &\sigma_1\left(U_{j\perp}^\top \widehat{Y}_j\right) \leq \sqrt{\frac{r}{d-r}}\varepsilon\left(\sqrt{\widehat{N}} + \sqrt{d-r}\right) + \sigma_1\left((b_j - \widehat{b}_j)\mathbf{1}^\top\right) \simeq \sqrt{\frac{r}{d-r}}\varepsilon\left(\sqrt{\widehat{N}} + \sqrt{d-r}\right) \\ &\sigma_r\left(U_j^\top \widehat{Y}_j\right) \geq \sigma_r(S_j) - \sigma_1\left((b_j - \widehat{b}_j)\mathbf{1}^\top\right) \simeq \sigma_r(S_j) \geq \sqrt{\widehat{N}} - \sqrt{r}. \end{aligned} \tag{4.15}$$

In the first inequality, $\sqrt{\frac{r}{d-r}}\varepsilon$ is multiplied because from (4.12), we have $U_{j\perp}^\top \widehat{Y}_j \simeq E_j$ where entries of E_j are in $\left[-\sqrt{\frac{r}{d-r}}\varepsilon, \sqrt{\frac{r}{d-r}}\varepsilon\right]$ while (4.13) is for matrix whose entries are in $[-1,1]$.

By combining (4.11) and (4.15), the proof is completed. □

Proof of Lemma 6. To begin with, we prove the first inequality. Let $x_{ij} = U_j s_{ij} + U_{j\perp} e_{ij} + b_j$, then by the triangular inequality and Chernoff Bound (4.14) what we proved in Lemma 5, we have

$$\begin{aligned} \|(I - \widehat{U}_j\widehat{U}_j^\top)(x_{ij} - \widehat{b}_j)\|_2 &= \|\widehat{U}_{j\perp}\widehat{U}_{j\perp}^\top(x_{ij} - \widehat{b}_j)\|_2 = \|\widehat{U}_{j\perp}^\top(x_{ij} - \widehat{b}_j)\|_2 \\ &= \|\widehat{U}_{j\perp}^\top(U_j s_{ij} + U_{j\perp} e_{ij} + b_j - \widehat{b}_j)\|_2 \\ &\leq \|\widehat{U}_{j\perp}^\top U_j s_{ij}\|_2 + \|\widehat{U}_{j\perp}^\top U_{j\perp} e_{ij}\|_2 + \|\widehat{U}_{j\perp}^\top(b_j - \widehat{b}_j)\|_2 \\ &\simeq \|\widehat{U}_{j\perp}^\top U_j s_{ij}\|_2 + \|\widehat{U}_{j\perp}^\top U_{j\perp} e_{ij}\|_2 \end{aligned}$$

Since e_{ij} is uniformly chosen in $\left[-\sqrt{\frac{r}{d-r}}\varepsilon, \sqrt{\frac{r}{d-r}}\varepsilon\right]^{d-r}$, we have

$$\|\widehat{U}_j^\top U_{j\perp} e_{ij}\|_2 \leq \|e_{ij}\|_2 \leq \varepsilon\sqrt{r}.$$

Also from the property of norm and s_{ij} is uniformly chosen in $[-1,1]^r$, we have

$$\|\widehat{U}_{j\perp}^\top U_j s_{ij}\|_2 \leq \|\widehat{U}_{j\perp}^\top U_j\|_2 \|s_{ij}\|_2 \leq \|\widehat{U}_{j\perp}^\top U_j\|_2\sqrt{r}.$$

From the above inequalities and by Lemma 5, we can prove the first inequality.

$$\|\widehat{U}_{j\perp}^\top U_j s_{ij}\|_2 + \|\widehat{U}_{j\perp}^\top U_{j\perp} e_{ij}\|_2 \leq \|\widehat{U}_{j\perp}^\top U_j\|_2\sqrt{r} + \varepsilon\sqrt{r} \leq \zeta\sqrt{r} + \varepsilon\sqrt{r}.$$

Now we prove the last inequality for the Lemma. By the property of norm, Chernoff Bound (4.14) and $\|\widehat{U}_j^\top U_{j\perp} e_{ij}\|_2 \leq \varepsilon\sqrt{r}$, we have

$$\begin{aligned}\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_{i\ell} - \widehat{b}_j)\|_2 &= \|\widehat{U}_{j\perp}^\top(x_{i\ell} - \widehat{b}_j)\|_2 \\ &= \|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + U_{\ell\perp}e_{i\ell} + b_\ell - \widehat{b}_j)\|_2 \\ &\geq \|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - \widehat{b}_j)\|_2 - \|\widehat{U}_{j\perp}^\top U_{\ell\perp}e_{i\ell}\|_2 \\ &\geq \|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - \widehat{b}_j)\|_2 - \varepsilon\sqrt{r} \\ &= \|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - b_j + (b_j - \widehat{b}_j))\|_2 - \varepsilon\sqrt{r} \\ &\simeq \|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - b_j)\|_2 - \varepsilon\sqrt{r}\end{aligned}$$

$\|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - b_j)\|_2$ denotes size of projection of $U_\ell s_{i\ell} + b_\ell - b_j$ to the subspace $\widehat{U}_{j\perp}$. Also the minimum singular value of $\widehat{U}_{j\perp}^\top U_\ell$ denotes the minimum value of projection of unit vector spanned by U_ℓ to the subspace $\widehat{U}_{j\perp}^\top$. Therefore the lower bound of this projection can be computed by

$$\|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - b_j)\|_2 \geq \sigma_r(\widehat{U}_j^\top U_\ell)\|U_\ell s_{i\ell} + b_\ell - b_j\|_2.$$

From the condition of Theorem 2, lower bound of $\|U_\ell s_{i\ell} + b_\ell - b_j\|_2$ is given by

$$\|U_\ell s_{i\ell} + b_\ell - b_j\|_2 \geq \|b_j - b_\ell\|_2 - \|U_\ell s_{i\ell}\|_2 \geq 4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}.$$

By the condition that U_j, U_ℓ are independent subspaces with rank r and by the definition of $U_{j\perp}$,

$$\sigma_r(\widehat{U}_{j\perp}^\top U_\ell) \geq \sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp}).$$

Therefore, we have

$$\|\widehat{U}_{j\perp}^\top(U_\ell s_{i\ell} + b_\ell - b_j)\|_2 \geq \sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp})\left(4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}\right).$$

$\sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp})$ denotes the minimum value of projection of unit vector $u_{j\perp}$ spanned by $U_{j\perp}$ to the subspace $\widehat{U}_{j\perp}$. Let p_j be the projected vector of $u_{j\perp}$ to the subspace $U_{j\perp}$ and $\widehat{p}_j$ be projected vector of $u_{j\perp}$ to the subspace $\widehat{U}_{j\perp}$. Then because $U_jU_j^\top + U_{j\perp}U_{j\perp}^\top = I$, $\|p_j\|_2^2 + \|\widehat{p}_j\|_2^2 = 1$. Note that by Lemma 5, the upper bound of $\|\widehat{p}_j\|_2$ is given by ζ while the lower bound of $\|p_j\|_2$ is $\sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp})$. Therefore we have,

$$\sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp}) \geq \sqrt{|1-\zeta^2|}.$$

Therefore,

$$\|(I - \widehat{U}_j\widehat{U}_j^\top)(x_{i\ell} - \widehat{b}_j)\|_2 \geq \sigma_{d-r}(\widehat{U}_{j\perp}^\top U_{j\perp})\left(4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}\right) \geq \sqrt{|1-\zeta^2|}\left(4\sqrt{r(1+\varepsilon^2)} - \sqrt{r}\right).$$

□

Chapter 5. Experimental Results

5.1 Experiment setup

In this chapter, we report experimental results for the **SC-SI** algorithm to verify how our algorithm is efficient and accurate than naive EM-style algorithm and other popular subspace clustering algorithms. This chapter consists of two sections.

In the first section, we will show that how our proposed initialization method, **SC-IN**, affects objective value convergence and clustering performance of our algorithm **SC-SI** and the naive EM algorithm **SC-EM** mentioned in Chapter 2 on MNIST handwritten digit dataset. Also, we will report relationship between value of α, β and robustness to noise and outliers of **SC-SI** algorithm.

In the last section, we will compare **SC-SI** and other clustering algorithms including median k-flat algorithm (MKF) [25], local best-fit flats (LBF), spectral LBF (SLBF), their heuristic versions (LBF-MS, SLBF-MS) [27], sparse subspace clustering (SSC) [28], low rank subspace clustering (LRSC), [29], Structured Subspace Clustering [30] and robust subspace clustering (RSC) [31] on MNIST handwritten digit dataset.

Performance Measure: Evaluating performance of clustering result for labeled data is not trivial. One of naive approach to measure performance of an algorithm is using classification error by matching original class and each cluster appropriately. However, since matching cluster and class becomes more difficult when number of cluster is large, this kind of approach cannot measure clustering performance exactly when number of cluster goes large.

To alleviate the issue, we consider the following clustering error which is based on binary classification error. Define a binary variable $e_{ij} = \mathbf{1}_{cluster_i = cluster_j}$. In other words, e_{ij} is 1 if data points i and j are in same cluster, and 0 otherwise. Therefore, we can consider true positive and false positive of e_{ij}. True positive occurs when i, j are in the same cluster both in fact and in the result of a cluster algorithm, Similarly, false positive, true negative and false negative can be also argued.

We use *Jaccard index* measure of how well a binary classification test correctly works, or simply, it measures the performance of a clustering algorithm. Let TP, TN, FP and FN be the number of true positives, true negatives, false positives and false negatives respectively. Then, the Jarccard index (JI) are defined as the following.

$$JI = \frac{TP}{TP + FP + FN}.$$

The Jarccard index has value between 0 and 1 and higher score means better clustering performance. Obviously, we expect to achieve better performance than random label assignment. In standard classification problem, performance of the random clustering is $\frac{1}{m}$ if all class has equal number of data points. However, in our setting, *Jaccard index* of random clustering archives

$$JI(\text{random clustering}) = \frac{\frac{1}{m}}{\frac{m-1}{m} + \frac{m-1}{m} + \frac{1}{m}} = \frac{1}{2m-1},$$

where m is the number of clusters. For example, if $m = 10$, then baseline performance of clustering algorithms is about 0.05.

Note that our evaluation method should compare every n^2 pairs to compute exact performance measure where n is number of total data points from data set X. However, if n goes large, $O(n^2)$ complexity is too painful to evaluation method. To reduce the computation complexity of evaluation method, we uniformly sample n_s number of data points from the data set X and compute JI only using n_s^2 pairs for evaluation. We sample 10% of data points for performance measure to MNIST dataset.

Detailed setting for SC-SI: In all experiments below, we choose $k = 1$ for **SC-SI**, which means we only run a single iteration of the subspace iteration in Step 2 of **SC-SI** algorithm. The dimension of low rank spaces r and β for **SC-IN** is determined by cross validation. We specify detailed value of hyper-parameters for every dataset.

As we mentioned in Chapter 3.2, in the practice, it is difficult to choose appropriate R_c in Step 2 of **SC-IN**. Instead of choosing $\widehat{X}_j$ using R_c, we choose $\widehat{N}$ number of data points from N_c number of nearest neighbors of each b_j. However, we again emphasize that this modified method cannot guarantee true clustering such as Theorem 2 because this breaks uniform randomness of $\widehat{X}_j$.

If detailed values of $\widehat{N}$ and N_c are not specified, we choose $N_c = n/m^2$ and $\widehat{N} = 0.9 \times N_c$.

5.2 Convergence and Robustness of SC-SI

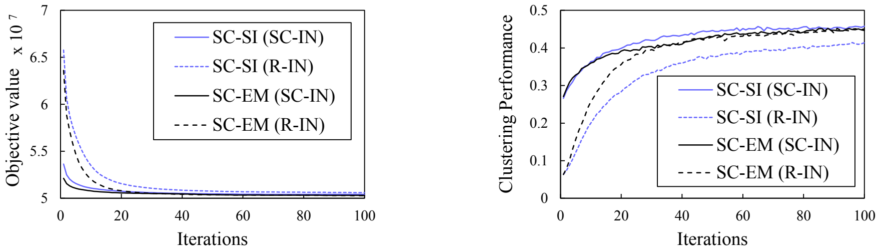
Convergence comparison between SC-SI and SC-EM: We show objective value and clustering performance for each iteration of **SC-SI** and **SC-EM** on MNIST dataset whose dimension is 784 and the number of data points is 60,000. We set $\alpha = 1$, $\beta = 10$, $r = 20$, $\widehat{N} = 600$, $N_c = 1000$ for the experiment. We repeat 10 times for each algorithms and report their averages. The result is reported in Figure 5.1.

As reported in Figure 5.1, **SC-SI** initialized by **SC-IN** is the fastest among other algorithms. When randomly initialized, EM algorithm seems converged much faster than **SC-SI**. However if algorithms initialized by **SC-IN**, **SC-SI** converges faster than **SC-EM**.

Required time per iterations comparison between SC-SI and SC-EM: We observe how required time per each iteration of **SC-SI** and **SC-EM** changes among the total number of data n, the dimension of data d and subspace r when $\alpha = 1$. We generate uniformly random data whose each entry uniformly chosen in $[-1, 1]$. We run 10 iterations for each random dataset with random initialization. For each setting, we repeated our experiment 10 times. i.e., in Figure 5.2 each point is the average of 100 iterations. Since the computation complexity of algorithm is affected by n, d and r jointly, we observe the relationship between each variable and the runtime of algorithms while fixing other variables. The detailed setting of each experiment is described in Figure 5.2.

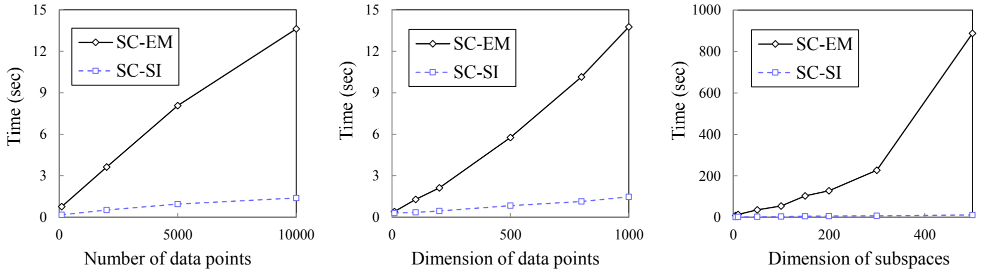
The result shows that, each iteration of **SC-SI** is much more faster than **SC-EM**. Moreover, computation complexity of **SC-EM** seems super-linear on dimension d and r. It is because complexity of the SVD computation, the major bottleneck of the EM algorithm, is $O(nd^2)$ when $n > d$ while **SC-SI** theoretically and practically requires only linear computation complexity.

Robustness of SC-SI among various α, β: We show the clustering performance of **SC-SI** with various α and β to noisy MNIST dataset. We add the addictive white Gaussian noise (AWGN) to given dataset. Magnitude of noise is specified by SNR in dB scale. If SNR = 0dB, powers of noise and signal are same. Note that larger SNR means less noise.



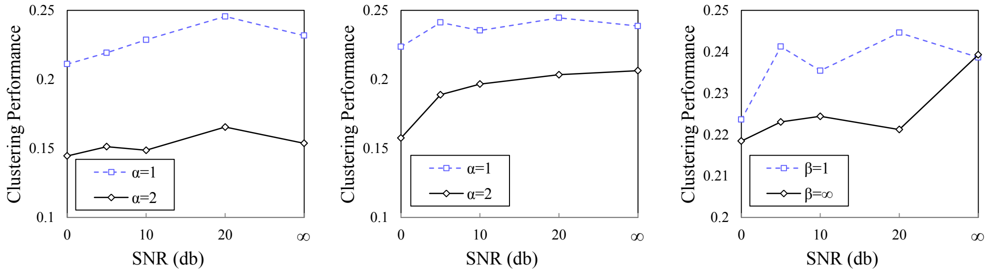
(a) (b)

Figure 5.1: Performance comparisons in (a) convergence and (b) clustering performance measured by *Jaccard Index* between **SC-SI** and **SC-EM**. **SC-IN** and **R-IN** denotes different initializations, first one is **SC-IN** in Section 3.2, and a purely random one, respectively.



(a) (b) (c)

Figure 5.2: Required running time per each iteration comparison between **SC-SI** and **SC-EM**. (a) running time and n while $d = 1,000$ and $r = 10$, (b) running time and d while $n = 10,000$ and $r = 10$, (c) running time and r while $n = 10,000$ and $d = 1,000$. In (a), (b), (c) each point is obtained from averaging over 10 iterations for different 10 randomly generated samples.



(a) (b) (c)

Figure 5.3: Performance comparison of **SC-SI** among different α, β and SNR (Signal-to-Noise) ratio in dB. (a) $\alpha = 1, 2$ for $\beta = 0$, i.e., it is random initialization (b) $\alpha = 1, 2$ for $\beta = 1$ (c) $\beta = 1, \infty$ for $\alpha = 1$.

For the experiment, we uniformly randomly sample 5% of data from MNIST training dataset to reduce entire runtime. We set $\widehat{N} = 30$, $N_c = 50$ for the experiment. The result is reported in Figure 5.3.

The result shows that regardless of initialization, smaller α is much robust. Moreover, **SC-IN** is more robust than random initialization while smaller β makes **SC-IN** is more robust.

However in some cases, especially number of data points is small and data have little noise, larger α and β may perform much better than smaller one. We will discuss this issue later.

5.3 Performance comparison between SC-SI and other algorithms

We also compare the clustering accuracy of **SC-SI** with those of other spectral subspace clustering algorithms including MKF [25], LBF, SLBF, their heuristic versions (LBF-MS, SLBF-MS) [27], SSC [28], LRSC, [29], Structured Subspace Clustering [30], and RSC [31] for MNIST dataset.

To observe robustness of each algorithm, we add AWGN to the dataset with dB scale. Also we add random outliers whose each entry is uniformly randomly chosen from $[0, 255]$. Since spectral methods require to build a huge affinity matrix whose size is the number of data points by the number of data points, we didn't try spectral algorithms for entire MNIST training dataset which require 3.6 billion entries to affinity matrix (If each entry is 4 byte float data type, we require more than 130GB memory for affinity matrix). Instead of running algorithm to entire dataset, we randomly sample 5% of data and run algorithms.

As we mentioned in Section 3.2, Theorem 2 shows that we need smaller β when we have outliers. To enhance robustness to outliers, we set $\alpha = 1$ and $\beta = 1$ when we add outliers to data points while we set $\beta = 10$ for non-outliers setting. For all algorithms, we choose a dimension of subspace r as 20. Also for sampled dataset, to enhance performance of our algorithm, we repeatedly run SC-SI 3 times and choose one which has the minimum converged objective value. The repeated result is reported as SC-SI (repeated) in Table 5.2. The overall experimental results are reported in Table 5.1 and 5.2 For iterative algorithms, we repeat algorithms 10 times for each data settings. Also when we sample data, we repeatedly sample data 10 times and run algorithms. Therefore, in Table 5.2, each result of iterative algorithms are average of 100 times experiments. In Table 5.1 and 5.2, we denote $n_m = n/m$, i.e., the number of data points in each cluster. The results shows that (a) **SC-SI** outperforms other iterative algorithms in accuracy but require more times to converge (b) **SC-SI** is significantly faster than other spectral algorithms while archive much better accuracy.

Table 5.1: Performance comparisons in the average clustering performance (measured by *Jaccard index*) for the entire MNIST training data.

Algorithm	Baseline	SNR (dB)					Outliers/n_m			
		20	10	5	1	0	50%	100%	150%	200%
SC-SI	0.42	0.42	0.40	0.43	0.38	0.43	0.43	0.34	0.36	0.32
K-means	0.13	0.13	0.11	0.09	0.09	0.09	0.13	0.13	0.13	0.12
MKF	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
LBF	0.21	0.20	0.24	0.26	0.26	0.25	0.20	0.20	0.20	0.21
LBF-MS	0.21	0.19	0.21	0.21	0.19	0.19	0.20	0.19	0.19	0.19
Structured SC	0.30	0.29	0.22	0.15	0.11	0.09	0.29	0.29	0.28	0.29

Table 5.2: Performance comparisons in the average clustering performance (measured by *Jaccard index*) for the 5% sampled MNIST training data.

Algorithm	Baseline	SNR (dB)					Outliers/n_m			
		20	10	5	1	0	50%	100%	150%	200%
SC-SI	0.32	0.31	0.32	0.30	0.28	0.30	0.29	0.28	0.29	0.28
SC-SI (Repeated)	0.37	0.34	0.34	0.33	0.32	0.32	0.32	0.31	0.31	0.31
K-means	0.14	0.14	0.11	0.09	0.09	0.09	0.14	0.13	0.13	0.13
MKF	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
LBF	0.28	0.27	0.31	0.29	0.27	0.24	0.27	0.29	0.27	0.25
LBF-MS	0.23	0.24	0.30	0.28	0.27	0.26	0.24	0.24	0.22	0.22
Structured SC	0.31	0.28	0.22	0.16	0.11	0.09	0.28	0.27	0.28	0.27
SLBF	0.31	0.31	0.30	0.26	0.22	0.20	0.28	0.28	0.25	0.26
SLBF-MS	0.26	0.29	0.28	0.24	0.21	0.20	0.27	0.27	0.25	0.25
SSC	0.23	0.23	0.24	0.23	0.24	0.23	0.23	0.23	0.23	0.22
LRSC	0.30	0.25	0.18	0.12	0.08	0.07	0.26	0.26	0.27	0.25
RSC	0.20	0.29	0.20	0.21	0.22	0.21	0.20	0.19	0.23	0.23

Chapter 6. Conclusion

Various methods have addresses to enhance the robustness of dimensionality reduction techniques. However, they are very computationally expensive compared to non-robust ones. In this paper, we design and analyze an efficient iterative SC algorithm and its initialization method.The algorithm is desinged via optimizing a sum of the α-th power of ℓ_2-norm objective with $0 < \alpha \leq 2$, where it is particularly attractive for high-dimensional and large-scale data. Considering the growing popularity of dimensionality reduction techniques in the machine learning problems, we believe that our proposed algorithm would find numerous applications in various domains.

References

[1] S. Lall and J. E. Marsden. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal on Robust and Nonlinear Control*, 12:519–535, 2002.

[2] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on*, 26(1):17–32, 1981.

[3] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[4] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–11. IEEE, 2003.

[5] W. Hong, J. Wright, K. Huang, and Y. Ma. Multiscale hybrid linear models for lossy image representation. *Image Processing, IEEE Transactions on*, 15(12):3655–3671, 2006.

[6] A. Y Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.

[7] R. Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2010.

[8] Q. Wang, Y. Ye, J. Z. Huang, and S. Feng. Fuzzy soft subspace clustering method for gene co-expression network analysis. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2010 IEEE International Conference on*, pages 47–50. IEEE, 2010.

[9] B. McWilliams and G. Montana. Subspace clustering of high-dimensional data: a predictive approach. *Data Mining and Knowledge Discovery*, 28(3):736–772, 2014.

[10] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

[11] L. Jing, M. K. Ng, J. Xu, and J. Z. Huang. Subspace clustering of text documents with feature weighting k-means algorithm. In *Advances in Knowledge Discovery and Data Mining*, pages 802–812. Springer, 2005.

[12] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[13] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari. Guess who rated this movie: Identifying users through subspace clustering. *arXiv preprint arXiv:1208.1544*, 2012.

[14] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 167–172. IEEE, 2003.

[15] A Baccini, Ph Besse, and A De Falguerolles. Al1-norm pca and a heuristic approach. 1996.

[16] Q. Ke and T. Kanade. Robust subspace computation using l1 norm. 2003.

[17] Q. Ke and T. Kanade. Robust l 1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.

[18] F. De La Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[19] C. Ding, D. Zhou, X. He, and H. Zha. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006.

[20] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[21] F. Nie, J. Yuan, and H. Huang. Optimal mean robust principal component analysis. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1062–1070, 2014.

[22] G. H. Golub and Charles F. Van L. *Matrix computations*, volume 3. JHU Press, 2012.

[23] R. Salakhutdinov and G. Hinton. An efficient learning procedure for deep boltzmann machines. *Neural computation*, 24(8):1967–2006, 2012.

[24] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[25] T. Zhang, A. Szlam, and G. Lerman. Median k-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 234–241. IEEE, 2009.

[26] Z Bai and YQ Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. *The annals of Probability*, 21:1276–1294, 1993.

[27] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012.

[28] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[29] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43:47–61, 2014.

[30] Benjamin Haeffele, Eric Young, and Rene Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 2007–2015, 2014.

[31] M. Soltanolkotabi, E. Elhamifar, E. J. Candes, et al. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.