Scalable Iterative Algorithm for Robust Subspace Clustering

Sanghyuk Chun ∗ Yung-Kyun Noh † Jinwoo Shin ‡

March 6, 2015

Abstract

Subspace clustering (SC) is a popular method for dimensionality reduction of high-dimensional data, where it generalizes Principal Component Analysis (PCA). Recently, several methods have been proposed to enhance the robustness of PCA and SC, while most of them are computationally very expensive, in particular, for high-dimensional large-scale data. In this paper, we develop much faster iterative algorithms for SC, incorporating robustness using a non-squared ℓ_2-norm objective. The known implementations for optimizing the objective would be costly due to the alternative optimization of two separate objectives: optimal cluster-membership assignment and robust subspace selection, while the substitution of one process to a faster surrogate can cause failure in convergence. To address the issue, we use a simplified procedure requiring efficient matrix-vector multiplications for subspace update instead of solving an expensive eigenvector problem at each iteration, in addition to release nested robust PCA loops. We prove that the proposed algorithm monotonically converges to a local minimum with approximation guarantees, e.g., it achieves 2-approximation for the robust PCA objective. In our experiments, the proposed algorithm is shown to converge at an order of magnitude faster than known algorithms optimizing the same objective, and have outperforms prior subspace clustering methods in accuracy and running time for MNIST dataset.

1 Introduction

The investigation of regularity from high-dimensional data generally seeks a low-dimensional representation of the regularity, with which the dominant properties of the system are explained with a few simple rules [13, 16]. However, true low-dimensional structure of the system often cannot be retrieved due to noise, in particular from a few large anomalies or the complete corruption of elements, even when most data faithfully follow the regularity in the low-dimensional space. To alleviate the effect of such noise, algorithms need 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 Principal Component Analysis (PCA) [20]. The sum of squared ℓ_2-norm in traditional PCA is replaced by different ones: sum of ℓ_1-norm [2, 10, 11, 3, 26] and sum of non-squared ℓ_2-norm [4, 27, 18], which are more robust to anomalies. In general, optimization with ℓ_1-norm uses non-analytic methods, whereas with non-squared ℓ_2-norm, the optimization can use analytic solutions such as Singular Value Decomposition (SVD), which makes the algorithm interpretable and scalable with appropriate approximation. The additional benefit of non-squared ℓ_2-norm is providing rotational invariance [4].

∗Dept. of Electrical Engineering, Korea Advanced Institute of Science & Technology, Email: sanghyuk.chun@kaist.ac.kr
†Dept. of Mechanical & Aerospace Engineering, Seoul National University, Email: nohyung@snu.ac.kr
‡Dept. of Electrical Engineering, Korea Advanced Institute of Science & Technology, Email: jinwoos@kaist.ac.kr
In this paper, we study robust methods for subspace clustering (SC) via optimizing a sum of non-squared $\ell_2$-norm objective, where SC generalizes PCA by assuming that data are composed of clusters of different low-dimensional affine subspaces. In this problem, the optimization is generally performed via alternating two steps of Expectation Maximization (EM) style: optimal cluster-membership assignment and affine subspace selection for each cluster. The robust affine subspace selection, called Robust PCA with Optimal Mean (RPCA-OM), under a sum of non-squared $\ell_2$-norm objective was recently studied [18], while the robust linear subspace selection was developed much earlier [4]. The authors of RPCA-OM propose an alternating procedure between SVD iterations and norm normalizations. Its high-level description of robust SC using RPCA-OM is in Figure 1. Here, RPCA-OM becomes a major bottleneck, and the naive back-and-forth optimization is inefficient for large-scale high-dimensional data, in particular, due to the expensive SVD steps.

Contribution. In our approach, we modify the optimization in RPCA-OM for subspace selection and dramatically reduce the running time over the naive EM scheme described in Figure 1. Instead of SVD calculations in RPCA-OM, we use a simplified procedure requiring a few matrix-vector multiplications for subspace update, where it is one iteration of the subspace-iteration method (also called orthogonal-iteration) [6] using Gram-Schmidt process. The main assumption for our method is that the matrix-vector multiplication can be done efficiently. Furthermore, we release nested robust PCA loops, i.e., perform the clustering assignment without waiting the convergence of subspace update. Our approach is summarized in Figure 2.

Despite of these simplified procedures, we prove that the proposed iterative algorithm monotonically converges to local optima as like the EM scheme. As we reported in Section 5, it computes a solution of same quality with those computed by EM and RPCA-OM at an order of magnitude faster. In spirit, our idea for such efficient computation is similar, for example, to contrastive divergence learning [21] and inexact augmented Lagrange multiplier [14], but their convergences do not hold in general. Our proof of monotone convergence utilizes structural properties of the subspace-iteration method under its equivalence to QR iterations. In addition, we propose a special initialization inspired by [4] and [1] so that the proposed algorithm converges to a local minimum with approximation guarantees, e.g., it achieves 2-approximation for the robust PCA objective.

Subspace clustering has been applied to numerous applications in computer vision [23, 8, 7, 29], DNA microarray analysis [15, 25], music analysis [12], text-mining [19, 9], recommendation systems [26, 30] and system identification [24] to name just a few, and various methods have been proposed for them. For many synthetic data and MNIST dataset, our experimental results show that the proposed algorithm outperforms other prior subspace clustering algorithms in both running time and accuracy.
Organization. Section 2 introduces the robust PCA and SC optimization problems. In Section 3, we propose an efficient iterative algorithm for both purposes, and we prove the convergence of the proposed method in Section 4. We report our experimental results in Section 5.

2 Preliminaries

2.1 PCA and Subspace Clustering

Given a data matrix $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$, where $d$ is the dimension of data and $n$ is the number of total data points. PCA is the problem of finding a subspace representation that best describes data points taken from a high-dimensional space. It can be addressed as the following optimization task using $\ell_2$-norm $\|\cdot\|_2$:

$$\text{PCA: } \min_{b, U, Y: U^\top U = I} \sum_{i=1}^{n} \|x_i - b - Uy_i\|_2^2,$$

where $b \in \mathbb{R}^d$ and $U \in \mathbb{R}^{d \times r}$ ($r \leq d$) are center and basis parameters, respectively, and $Y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{r \times n}$ is a low dimensional representation of original data $X$. Subspace clustering is a generalized version of PCA where it finds multiple clustered subspace representations:

$$\text{SC: } \min_{w_{ij}, b_j, U_j, Y: U_j^\top U_j = I, \sum_{j=1}^{m} w_{ij} = 1} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \|x_i - b_j - U_j y_i\|_2^2,$$

where $w_{ij} \in \{0, 1\}$ indicates a cluster membership and now we have center and basis variables $b_j, U_j$ for the $j$-th subspace. If there exists one subspace, i.e., $m = 1$, then (2) reduces to (1). We assume that the number $m$ of clusters and the dimension $r$ of subspaces are given.

It is well known that the PCA optimization (1), i.e., (2) with $m = 1$, has a closed-form SVD solution. However, the SC optimization (2) with $m \geq 2$ is a mixed integer programming and not easy to compute a global optimum. The popular method is an alternating algorithm of EM type: alternatively update $w_{ij}$ via taking the best cluster $j$ for each data point $x_i$ and $b, U$ from a SVD solver in each cluster, where one can prove that it converges to a local optimum of (2). In this paper, we aim for developing an efficient algorithm for solving ‘robust’ versions of (1) and (2).

2.2 Robust Subspace Clustering

There have been extensive efforts for developing robust PCA methods in the literature [4, 26, 27, 18], where they study certain optimization tasks similar to (1) via replacing the squared $\ell_2$-norm by $\ell_1$-norms or non-squared $\ell_2$-norms. However, most works on this line ignored the center parameter $b$ in their optimization task, and recently, the following optimization jointly considering the center parameter was first studied [18]:

$$\text{RPCA: } \min_{b, U, Y: U^\top U = I} \sum_{i=1}^{n} \|x_i - b - Uy_i\|_2.$$

The non-squared $\ell_2$-norm gives significant challenges to optimize, i.e., it is not clear whether (3) can be reduced to a convex optimization as like the case of squared $\ell_2$-norm. To address the issue, the authors [18] observe that (3) reduces to the following optimization:

$$\min_{b, U: U^\top U = I} \sum_{i=1}^{n} \|(I - UU^\top)(x_i - b)\|_2.$$
Then, they propose the following algorithm for solving (4): alternates between SVD and norm normalizations.

**RPCA-OM 1 Iterative re-weighting algorithm [18]**

**Input:** Data $X \in \mathbb{R}^{d \times n}$, dimension of subspace $r$.

**Initialize:** Set $D$ as an identity matrix.

**repeat**
1. Update $b \leftarrow \frac{XD}{1+D}$. 
2. Update $U$ by the $r$ largest singular vectors of $XD - D11^T D^2$. 
3. Update the diagonal matrix $D$ as its $i$-th diagonal element is set by $\frac{1}{2\| (I-UU^T)(x_i-b) \|^2}$.
**until** converge

In the above description, 1 denotes the column vector with entire elements being one. The main idea underlying the above algorithm is following: it first solves the squared $\ell_2$ version of (4) with additional weighting parameter $D$ via SVD and then updates $D$ to normalize the squared $\ell_2$-norm to the non-squared $\ell_2$-norm. This is summarized in Figure 1. Although RPCA-OM minimizes the squared $\ell_2$ version of (4) in each iteration, [18] prove that it monotonically decreases the original objective in (4) and furthermore converges a local optimum of (4).

The robust PCA optimization (3) is naturally generalized to the following optimization for robust subspace clustering:

**RSC:**

$$\min_{[w_{ij}], [b_j], [U_j], Y; U_j^T U_j = I, \sum_{j=1}^m w_{ij} = 1} \sum_{i=1}^n \sum_{j=1}^m w_{ij} \| x_i - b_j - U_j y_{i} \|^2.$$  

(5)

Again, it is equivalent to solve the following optimization:

$$\min_{[w_{ij}], [b_j], [U_j], Y; U_j^T U_j = I, \sum_{j=1}^m w_{ij} = 1} \sum_{i=1}^n \sum_{j=1}^m w_{ij} \| (I - U_j U_j^T)(x_i - b_{j}) \|^2.$$  

(6)

Now, the above optimization task for robust subspace clustering becomes much harder than that of robust PCA due to integer clustering variables $w_{ij}$. Nevertheless, one can design the following algorithm of EM style for solving (6).

**RSC-EM 2 EM algorithm for RSC**

**Input:** Data $X \in \mathbb{R}^{d \times n}$, number of clusters $m$ and dimension of subspace $r$.

**Initialize:** Set $w_{ij}$ arbitrarily.

**repeat**
1. Update $b_j, U_j$ using current data points in cluster $j$ by RPCA-OM.
2. Update clustering information $w_{ij}$ by finding the best cluster for each data point
**end for**
**until** converge

Using the convergence guarantee established by [18] for RPCA-OM, one can also prove that the above algorithm also monotonically decreases the objective in (6) and furthermore converges its local optimum.

However, both RPCA-OM and RSC-EM might be slow for high-dimensional data (i.e. large $d$) since they require many iterations of a SVD method. The issue is even severe for the robust subspace clustering.
case since RSC-EM keeps solving SVD until its nested RPCA-OM converges. The goal of this paper is to develop more efficient algorithm for both robust PCA and robust subspace clustering.

3 Scalable Iterative Algorithm for Robust Subspace Clustering

3.1 Algorithm Description and Convergence

In this section, we propose a new iterative algorithm for solving both RPCA (4) and SC (6). We only describe our algorithm for the latter (robust subspace clustering), where its special case \( m = 1 \) corresponds to the former (robust PCA). In particular, we propose an efficient iterative algorithm, called \( \text{RSC-SI} \) that is described in what follows.

RSC-SI 3 Scalable iterative algorithm for RSC

| Input: Data \( X \in \mathbb{R}^{d \times n} \), number of clusters \( m \), dimension of subspace \( r \). |
| Output: Cluster membership variable \( w_{ij} \), center vector \( b_j \in \mathbb{R}^d \) and low dimensional subspace \( U_j \in \mathbb{R}^{d \times r} \).
| Initialize: Set \( U_j, w_{ij} \) arbitrarily and the diagonal matrix \( D_j \) as an identity matrix \( \in \mathbb{R}^{n \times n} \). |

repeat
  for all \( j \) do
    1. Update \( b_j \leftarrow XD_jW_j \frac{1}{1+D_jW_j} \) where \( W_j \) is the diagonal matrix whose \( i \)-th element is \( w_{ij} \).
    2. Update \( U_j \) by following steps where \( k \) is some positive integer.
       repeat
       \( U_j \leftarrow XH_jH_j^\top X^\top U_j \)
       \( U_j \leftarrow \text{Gram-Schmidt Orthonormalization of } U_j \)
       until \( k \) times
       where \( H_j = D_jW_j - \frac{D_jW_j}{1+D_jW_j} \).
  end for
  for all \( i, j \) do
    3. Update the diagonal matrix \( D_j \) as its \( i \)-th diagonal element is set by \( \frac{1}{2\| (I-U_jU_j^\top) (x_i-b_j) \|_2} \).
    4. Update \( w_{ij} \leftarrow 1 \) if
       \( j \in \arg \min_\ell \| (I-U_\ell U_\ell^\top) (x_i-b_\ell) \|_2 \)
       and \( w_{ij} \leftarrow 0 \) otherwise.
  end for
until converge

Each iteration of \( \text{RSC-SI} \) is designed to solve following optimization instead of (6):

\[
\min_{b_j, U_j, w_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} w_{ij} \| (I-U_jU_j^\top) (x_i-b_j) \|_2^2, \tag{7}
\]

where \( d_{ij} \) is the \( i \)-th diagonal element of \( D_j \). It is easy to show that the optimal \( b_j \) of the optimization is following:

\[
b_j = XD_jW_j \frac{1}{1+D_jW_j}, \tag{8}
\]
where it appears in Step 1 of the algorithm. Substituting \( b_j \) into the problem (7), we have
\[
\max_{U_j, H_j} \sum_j \text{tr} \left( U_j^T X H_j H_j^T X^T U_j \right).
\] (9)

The optimal \( U_j \) of the above optimization can be computed by SVD of \( XH_j \), which can be computationally expensive. Instead, the proposed algorithm performs only \( k \) iterations of the ‘subspace-iteration method’ [6], i.e., inexact cheap SVD, in Step 2. Furthermore, it uses \( U_j \) at the previous iteration, as the initial point at the next iteration. Hence, if \( k \) is small, e.g., \( k = 1 \), the computational complexity of each iteration is much smaller compared to performing SVD directly. Step 3 is designed for normalizing squared \( \ell_2 \)-norm in (7) to non-squared one in (6), and Step 4 rearrange data points to clusters using updated parameters. We note that when \( m = 1 \), i.e., robust PCA, Step 4 is not necessary in RSC-SI.

At a high level, the algorithm couples intermediate steps of three computational tasks: SVD, clustering assignments and robust-norm normalizations, which are solved separately in RPCA-OM and RSC-EM. In case \( m = 1 \), i.e., robust PCA, the computational gain of the proposed algorithm compared to RPCA-OM is from the inexact SVD procedure in Step 2. In case \( m \geq 2 \), i.e., robust subspace clustering, its computational gain compared to RSC-EM is much more significant because the latter waits until the nested robust PCA procedure converges at each iteration, while the former does not.

The authors [18] show that RPCA-OM monotonically decreases the desired objective value at each iteration and converges to a local minimum. Such properties are well known for the EM algorithm, i.e., RSC-EM. Despite of the computational efficiency of our proposed algorithm, we prove that it also has the same desired monotone convergence properties stated as follows.

**Theorem 1** RSC-SI monotonically decreases the objective value of (6) at each iteration. Furthermore, it converges to a local minimum of (6).

Once RSC-SI converges, its convergence to a local optimum of (6) can be derived in a relatively straightforward manner using same proof arguments in [18]. However, for establishing the monotone convergence of RSC-SI, we need new proof techniques due to our simplified procedures that do not exist in [18], where we use certain structural properties of Gram-Schmidt process. In other words, other orthonormalization might not guarantee the monotone convergence. We present the proof of Theorem 1 in Section 4.

### 3.2 Approximation Guarantee via Initialization

Now we aim for establishing the quality of a local minimum of (6) computed by RSC-SI, where they are sensitive to how \( U_j, w_{ij} \) and \( D_j \) are initialized. To this end, we propose the following method for the initialization.

- Choose the first center vector \( b_1 \) uniformly at random from data points \( \{x_1, x_2, \ldots, x_n\} \).
- Compute the first subspace \( U_1 \) by R1-PCA [4] using entire data points subtracted by center \( b_1 \), i.e., \( \{x_1 - b_1, x_2 - b_1, \ldots, x_n - b_1\} \).
- Repeat the following steps \( m - 1 \) times
  - Choose a next center vector \( b_j \) from data points as \( b_j = x_i \) with probability \( \frac{f_D(x_i)}{\sum_{x'} f_D(x')} \), where
    \[
    f_D(x) = \min_{j'} \| (I - U_{j'} U_{j'}^T) (x - b_{j'}) \|_2.
    \]

\(^1\)We denote the trace of \( A \) by \( \text{tr}(A) \).

---

6
Update clustering information as

\[ w_{ij} \left\{ \begin{array}{ll} 1 & \text{if } j = \arg \min_{j'} \| x_i - b_{j'} \|_2 \\ 0 & \text{if otherwise} \end{array} \right. \]

Update \( U_j \) of each cluster by applying R1-PCA using data points (subtracted by its center) in it.

- Update the diagonal matrix \( D_j \) as its \( i \)-th diagonal element is set by

\[
\frac{1}{2 \| (I - U_j U_j^\top) (x_i - b_j^*) \|_2}.
\]

The above initialization is inspired by R1-PCA [4] and K-means++ [1]. R1-PCA provides a globally optimal subspace \( U_j \) given data points and center \( b_j \). In addition, the initialization of K-means++ provides the approximation guarantee of clustering algorithms. Under the initialization, we prove the following approximation guarantee of RSC-SI.

**Theorem 2** Let \( \phi_{\text{global}} \) and \( \phi_{\text{local}} \) be the objective value of (6) at the global minimum and at the local minimum computed by RSC-SI under the above initialization, respectively. Then, it follows that

\[
\mathbb{E}[\phi_{\text{local}}] \leq \left\{ \begin{array}{ll} 2 & \text{if } m = 1 \\ 2\eta (\ln m + 2) & \text{if } m \geq 2 \end{array} \right.
\]

where \( \eta = \max_{k=1,2,...,m} \eta_k \) and

\[
\eta_k = \max_{j \neq k, i \in C_k} \frac{|C_k^*| \| (I - U_j^* (U_j^*)^\top) (x_i - b_j^*) \|_2}{\sum_{\ell \in C_k} \| (I - U_j^* (U_j^*)^\top) (x_\ell - b_j^*) \|_2}.
\]

In the above, \( U_j^* \), \( b_j^* \), \( w_{ij}^* \) be the global optimum of (6) achieving \( \phi_{\text{global}} \) and \( C_j^* = \{ i : w_{ij}^* = 1 \} \) denotes the corresponding \( j \)-th cluster.

The proof of the above theorem is given in the supplementary material due to the space limitation. It first implies that RSC-SI is a 2-approximation scheme for robust PCA optimization (3). On the other hand, for the case of robust subspace clustering, the approximation guarantee depends on some geometric property, captured by \( \eta \), of the global optimum. To convey some intuition for \( \eta \), observe that subspace \( U_j^* \) for the \( j \)-th cluster can be naturally assumed to be statistically independent of data point \( x_i \) in other clusters. Therefore, when \( r \) is much smaller than \( d \), \( \| (I - U_j^* (U_j^*)^\top) (x_i - b_j^*) \|_2 \approx \| x_i - b_j^* \|_2 \), for \( i \in C_k \) with \( k \neq j \). Then, it follows that

\[
\eta_k \approx \max_{j \neq k, i \in C_k} \frac{|C_k^*| \| x_i - b_j^* \|_2}{\sum_{\ell \in C_k} \| x_\ell - b_j^* \|_2}.
\]

One can simply observe that the above quantity can be close to 1 if clusters are well separated, in which case the approximation ratio is close to \( 2(\ln m + 2) \). This intuitive explanation on \( \eta \) helps to understand the performance of RSC-SI.

In addition, we note that the proposed initialization might be expensive since it requires to solve R1-PCA many times. Due to the issue, we also design inexact fast versions of the initialization in our experiments, as reported in Section 5. Through extensive experiments, we found that the careful initializations provide much better starting points for RSC-SI.

**4 Proof of Theorem 1**

Once RSC-SI converges, it is easy to check that \( U_j \) forms singular vectors of \( X H_j \). Under this observation, the convergence to a local optimum of (6) can be shown using same arguments in [18] via...
checking KKT conditions. However, for establishing the monotone convergence of RSC-SI, we need new proof techniques due to our simplified procedures that do not exist in [18]. To this end, we first state the following key lemma.

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

$$
U_A(1) \leftarrow AU(0)
$$

$$
U_A(1) \leftarrow \text{Gram-Schmidt Orthonomalization of } U_A(1),
$$

where $U(0) \in \mathbb{R}^{d \times d}$ is some fixed square matrix. Then, for any $r \leq d$, it follows that

$$
\sum_{i=1}^{r} U(0)_i^\top AU(0)_i \leq \sum_{i=1}^{r} U_A(1)_i^\top AU_A(1)_i,
$$

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

**Proof:** To begin with, one can observe that the subspace-iteration can be summarized as follows:

$$
U_A(1)R_A = AU(0),
$$

where $R_A$ is an upper triangular matrix, i.e., it is a QR decomposition of $AU(0)$. Since $A$ is positive semi-definite, there exists a unique positive semi-definite square-root matrix $B \in \mathbb{R}^{d \times d}$, i.e., $A = B^2$.

Consider the two steps of subspace-iterations for $B$:

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

which implies that

$$
U_B(2)R_B(2)R_B(1) = BU_B(1)R_B(1) = B^2U(0) = AU(0).
$$

Hence, without loss of generality, one can conclude that

$$
U_B(2) = U_A(1). \tag{10}
$$

Furthermore, we let $B(k) := U_B(k)^\top BU_B(k)$ and observe that

$$
B(0) = U_B(0)^\top BU_B(0) = U_B(0)^\top U_B(1)R_B(1) = Q_B(1)R_B(1),
$$

where we define $Q_B(k+1) := U_B(k)^\top U_B(k+1)$. Similarly, it follows that

$$
B(1) = U_B(1)^\top BU_B(1) = Q_B(1)^\top U_B(0)^\top BU_B(0)Q_B(1) = Q_B(1)^\top B(0)Q_B(1) = R_B(1)Q_B(1).
$$

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

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

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

$$
\text{tr} \left( \text{sub} \left[ B(k)^2 \right] \right) = \text{tr} \left( \text{sub} \left[ U_B(k)^\top B^2U_B(k) \right] \right)
$$

$$
= \sum_{i=1}^{r} U_B(k)_i^\top B^2U_B(k)_i = \sum_{i=1}^{r} U_B(k)_i^\top AU_B(k)_i, \tag{11}
$$
where \( U_B(k)_i \) denotes the \( i \)-th column vector of \( U_B(k) \). Since \( R(1) \) is an upper triangular matrix, it follows that

\[
\text{tr}(\text{sub}[B(1)^2]) - \text{tr}(\text{sub}[B(0)^2])
\]

\[
= \text{tr}(\text{sub}[R(1)R(1)^\top]) - \text{tr}(\text{sub}[R(1)^\top R(1)])
\]

\[
= \sum_{i=1}^r \sum_{j=1}^d (R(1))_{ij}^2 - \sum_{i=1}^r \sum_{j=1}^d (R(1))_{ij}^2
\]

\[
= \sum_{i=1}^r (\sum_{j=1}^d (R(1))_{ij}^2) - \sum_{i=1}^r (R(1))_{ij}^2
\]

\[
= \sum_{i=1}^r \sum_{j=r+1}^d (R(1))_{ij}^2 \geq 0,
\] (12)

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

\[
\text{tr}(\text{sub}[B(2)^2]) - \text{tr}(\text{sub}[B(1)^2]) \geq 0.
\] (13)

Combining (10), (11), (12) and (13) leads to the conclusion of Lemma 3.

Let \( W, D, B, U \) and \( \hat{W}, \hat{D}, \hat{B}, \hat{U} \) are current and updated (after one iteration) values of \textbf{RSC-SI}, and we define

\[
J(W, D, U, B) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} d_{ij} \| (I - U_j U_j^\top) (x_i - b_j) \|_2^2.
\]

From Lemma 3 for positive semi-definite \( XH_j H_j^\top X^\top \), it follows that

\[
\sum_j \text{tr}(\hat{U}_j^\top XH_j H_j^\top X^\top \hat{U}_j) \geq \sum_j \text{tr}(U_j^\top XH_j H_j^\top X^\top U_j).
\]

Due to (8), it is equivalent to

\[
J(W, D, \hat{U}, \hat{B}) \leq J(W, D, U, \hat{B}).
\] (14)

On the other hand, the optimality of \( \hat{B} \) implies that

\[
J(W, D, U, \hat{B}) \leq J(W, D, U, B).
\] (15)

Combining (14) and (15) leads to

\[
J(W, D, \hat{U}, \hat{B}) \leq J(W, D, U, B).
\]

This reduces to

\[
\sum_{i=1}^n \sum_{j=1}^m w_{ij} \frac{J_i(\hat{U}_j, \hat{b}_j)^2}{J_i(U_j, b_j)^2} \leq \sum_{i=1}^n \sum_{j=1}^m \frac{w_{ij} J_i(U_j, b_j)^2}{2 J_i(U_j, b_j)}.
\] (16)

where we define \( J_i(U_j, b_j) := \| (I - U_j U_j^\top) (x_i - b_j) \|_2 \) to simply notation and use the definition of \( d_{ij} \).

If we further use Lemma 1 in [17], we have

\[
\sum_{i=1}^n \sum_{j=1}^m \left[ w_{ij} J_i(\hat{U}_j, \hat{b}_j) - \frac{w_{ij} J_i(U_j, b_j)^2}{2 J_i(U_j, b_j)} \right] \leq \sum_{i=1}^n \sum_{j=1}^m \left[ w_{ij} J_i(U_j, b_j) - \frac{w_{ij} J_i(U_j, b_j)^2}{2 J_i(U_j, b_j)} \right].
\] (17)
Finally, from (16) and (17), we obtain
\[ \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} J_i(\hat{U}_j, \hat{b}_j) \leq \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} J_i(U_j, b_j). \]
This establishes the monotone decreasing property of RSC-SI, and completes the proof of Theorem 1.

5 Experiments

In this section, we report experimental performances of our proposed algorithm, RSC-SI under synthetic and real (MNIST) data, compared to other known clustering algorithms. We choose \( k = 1 \) in RSC-SI.

5.1 Synthetic Data

We generate synthetic data of \( n \) points from the mixture of multivariate normal distributions. First, we consider three multivariate normal distributions in dimension \( r = 10 \) generated as follows. Means are selected randomly in the interval \([0, 50]\) and covariance matrices are of form \( AA^T \) where each entry of \( A \) is chosen randomly in \([0, 10]\). The number of data points in each cluster is \( n/3 \). Second, we project each point in \( \mathbb{R}^d \) to \( \mathbb{R}^r \) using a corresponding randomly generated projection matrix to each cluster, where we choose \( d = 500 \). Finally, we randomly select \( p \) fraction of entries in the whole generated dataset, and reset values of each selected entry to be zero.

We compare the number of iterations for convergence of RSC-SI, RPCA-OM and RSC-EM on datasets, under \( n = 15000 \) and \( p = 0.25 \). We also report their running time per iteration (or stage) as data dimension \( d \) increases. Since the initialization in Section 3.2 might be expensive due to R1-PCA, we also test its modified version that runs R1-PCA only at the end of the initialization and chooses \( U_j = 0 \) at intermediate steps. The results are reported in Figure 4. It shows that the convergence rate of RSC-SI is comparable with RPCA-OM and RSC-EM, while its running time per iteration is at an order of magnitude faster than others.

We also compare the clustering performances of RSC-SI to other spectral subspace clustering algorithms including median k-flat algorithm (MKF) [31], local best-fit flats (LBF), spectral LBF (SLBF), their heuristic versions (LBF-MS, SLBF-MS) [32], local subspace affinity (LSA) [28], sparse subspace clustering (SSC) [5], and robust subspace clustering (RSC) [22]. Here, we try three noise levels \( p = 0, 0.25, 0.5 \) and also consider RSC-SI initialized by the K-means algorithm. We use the popular Jaccard index in the interval \([0, 1]\) to measure how much clustering that each algorithm finds is close to the

![Figure 3: Performance comparisons in convergence and running time between RSC-SI and RPCA-OM for \( m = 1 \) (i.e., robust PCA setup) and between RSC-SI and RSC-EM for \( m = 1 \) (i.e., robust SC setup). RSC-SI (E), RSC-SI (I) and RSC-SI (R) uses different initializations, the one in Section 3.2, its modified version and a purely random one, respectively. Each point is obtained from averaging over twenty randomly generated datasets, where we run each algorithm once for each data sample.](image-url)
true clustering, i.e., 1 means perfect clustering. The results are summarized in Table 1. For the synthetic data we generated, we found that spectral methods (SLBF, LSA, SSC, RSC) and RSC-SI perform well in accuracy, while the formers are much slower than the latter in general. However, as we reported in the following section, RSC-SI outperforms in both speed and accuracy for real MNIST data. One can also observe that the K-means initialization for RSC-SI outperforms other initializations in accuracy, where we think this is because K-means already works reasonably well for the synthetic data, i.e., it provide a good starting point and RSC-SI further improves it significantly. Considering the popularity of K-means in many real-world datasets, we recommend it as initialization of RSC-SI for practical purposes.

5.2 MNIST Data

We also test clustering algorithms in the previous section using real MNIST data. We select three or four subsets of ten numbers $\{0, 1, \ldots, 9\}$ to compare their performances, and choose 1000 data points in each cluster. To compare robustness of each clustering algorithm, we select $p = 0, 0.25$ random fractions of entries in data points and set them by 255 (i.e., black). We do not try LBF and SLBF algorithms for MNIST since they are too slow in this case compared to similar methods, LBF-MS and SLBF-MS. We also choose the K-means initialization and subspace dimension $r = 10$ to run RSC-SI since it shows the best performance in our synthetic data experiments. The experimental results are reported in Table 2. We observe that the standard (non-robust) EM algorithm optimizing the sum of squared $\ell_2$-norm objective (2) outperforms K-means algorithm, which implies that our subspace assumption can enhance clustering performance of MNIST data. Furthermore, RSC-SI outperforms the EM algorithm which is due to robustness of our non-squared $\ell_2$-norm objective. Spectral methods (SLBF, LSA, SSC, RSC) is much times slower than RSC-SI, and their accuracies are even worse.

Finally, to justify subspace structures of MNIST dataset and how RSC-SI is effective to find them, we plot the results of PCA and RSC-SI in two-dimensional spaces. This shows an interesting structure of MNIST dataset, and confirms our subspace assumption underlying it.

|            | score |         |         |         |         |
|------------|-------|---------|---------|---------|---------|
|            | $p = 0\%$ | $p = 25\%$ | $p = 50\%$ | time (s)   |
| K-means    | 0.83  | 0.77   | 0.76   | 0.51     |
| SC-EM      | 1.00  | 0.98   | 0.93   | 3.24     |
| RSC-SI (E) | 0.91  | 0.77   | 0.71   | 78.31    |
| RSC-SI (KM)| 1.00  | 1.00   | 1.00   | 1.90     |
| RSC-SI (I) | 0.86  | 0.80   | 0.67   | 8.87     |
| MKF        | 1.00  | 0.85   | 0.49   | 5.61     |
| LBF        | 1.00  | 1.00   | 0.97   | 54.59    |
| LBF-MS     | 1.00  | 0.97   | 0.94   | 14.35    |
| SLBF       | 1.00  | 1.00   | 1.00   | 672.41   |
| SLBF-MS    | 1.00  | 1.00   | 1.00   | 241.98   |
| LSA        | 1.00  | 0.96   | 0.98   | 291.68   |
| SSC        | 1.00  | 1.00   | 1.00   | 66.49    |
| RSC        | 1.00  | 0.99   | 0.99   | 667.32   |

Table 1: Performance comparisons in accuracy (measured by Jaccard index) using synthetic data. RSC-SI (E), RSC-SI (I) and RSC-SI (KM) uses different initializations, the one in Section 3.2, its modified version and K-means, respectively. Their running times differ primarily due to initialization costs running R1-PCA. Each number is obtained from averaging over twenty randomly generated datasets, where we run each algorithm once for each data sample. SC-EM refers the standard (non-robust) EM algorithm optimizing (2) under K-means initialization.
Table 2: Performance comparisons in accuracy (measured by Jaccard index) using MNIST dataset. Each number is obtained from averaging over twenty runs for K-means, SC-EM, RSC-SI, MKF, LBF-MS, and one run for SLBF-MS, LSA, SSC and RSC (since they are quite slow).

| Method  | subsets | [0 1 5] | [2 5 9] | [1 3 6] | [1 2 5] | [1 2 5 6] | [0 1 3 4] | [0 2 4 5] | average |
|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| K-means | score ($p = 0\%$) | 0.64 | 0.61 | 0.75 | 0.64 | 0.53 | 0.76 | 0.61 | 0.65 |
|         | score ($p = 25\%$) | 0.56 | 0.59 | 0.73 | 0.58 | 0.48 | 0.40 | 0.57 | 0.62 |
|         | time (s) | 1.07 | 1.78 | 0.55 | 0.80 | 2.90 | 1.15 | 3.82 | 1.73 |
| SC-EM   | score ($p = 0\%$) | 0.71 | 0.76 | 0.89 | 0.83 | 0.71 | 0.84 | 0.74 | 0.78 |
|         | score ($p = 25\%$) | 0.61 | 0.73 | 0.82 | 0.68 | 0.61 | 0.87 | 0.66 | 0.71 |
|         | time (s) | 5.05 | 6.95 | 3.00 | 4.37 | 9.55 | 7.44 | 6.16 | 6.06 |
| RSC-SI  | score ($p = 0\%$) | 0.85 | 0.78 | 0.92 | 0.96 | 0.84 | 0.93 | 0.91 | 0.88 |
|         | score ($p = 25\%$) | 0.63 | 0.70 | 0.86 | 0.73 | 0.61 | 0.94 | 0.76 | 0.74 |
|         | time (s) | 13.71 | 13.00 | 15.09 | 13.85 | 24.69 | 24.50 | 21.77 | 18.09 |
| MKF     | score ($p = 0\%$) | 0.53 | 0.82 | 0.76 | 0.66 | 0.64 | 0.63 | 0.65 | 0.71 |
|         | score ($p = 25\%$) | 0.21 | 0.22 | 0.21 | 0.21 | 0.16 | 0.17 | 0.17 | 0.19 |
|         | time (s) | 29.51 | 23.42 | 25.77 | 27.35 | 42.11 | 46.14 | 41.51 | 33.69 |
| LBF-MS  | score ($p = 0\%$) | 0.53 | 0.45 | 0.60 | 0.49 | 0.39 | 0.54 | 0.44 | 0.49 |
|         | score ($p = 25\%$) | 0.53 | 0.39 | 0.63 | 0.45 | 0.38 | 0.61 | 0.38 | 0.48 |
|         | time (s) | 36.71 | 35.92 | 36.42 | 36.28 | 52.07 | 52.55 | 51.74 | 43.10 |
| SLBF-MS | score ($p = 0\%$) | 0.54 | 0.89 | 0.49 | 0.57 | 0.57 | 0.59 | 0.87 | 0.65 |
|         | score ($p = 25\%$) | 0.65 | 0.66 | 0.82 | 0.66 | 0.62 | 0.83 | 0.55 | 0.68 |
|         | time (s) | 643.81 | 608.97 | 644.49 | 646.00 | 995.08 | 1004.64 | 932.53 | 782.22 |
| LSA     | score ($p = 0\%$) | 0.52 | 0.65 | 0.52 | 0.52 | 0.35 | 0.33 | 0.67 | 0.51 |
|         | score ($p = 25\%$) | 0.48 | 0.71 | 0.47 | 0.46 | 0.31 | 0.37 | 0.40 | 0.46 |
|         | time (s) | 688.17 | 638.19 | 702.70 | 688.03 | 1186.94 | 1223.17 | 1096.89 | 889.15 |
| SSC     | score ($p = 0\%$) | 0.47 | 0.89 | 0.47 | 0.45 | 0.32 | 0.32 | 0.82 | 0.54 |
|         | score ($p = 25\%$) | 0.47 | 0.82 | 0.90 | 0.75 | 0.63 | 0.79 | 0.65 | 0.72 |
|         | time (s) | 110.80 | 85.78 | 115.43 | 112.45 | 213.71 | 233.58 | 169.61 | 147.34 |
| RSC     | score ($p = 0\%$) | 0.35 | 0.59 | 0.46 | 0.67 | 0.52 | 0.28 | 0.28 | 0.45 |
|         | score ($p = 25\%$) | 0.26 | 0.20 | 0.23 | 0.21 | 0.17 | 0.20 | 0.16 | 0.20 |
|         | time (s) | 1889.49 | 1745.43 | 1965.12 | 1894.18 | 3461.52 | 3951.87 | 3132.42 | 2577.15 |

Figure 4: Comparison between PCA (computed by SVD) and robust SC (computed by RSC-SI) for numbers \{1, 3, 6\} in MNIST dataset. For PCA, we compute the first and second principal components of each data point. For robust SC, we first find a subspace of dimension 10 for each cluster using RSC-SI. Then, we project all data points to each subspace and use PCA to plot them into two-dimensional space $\mathbb{R}^2$. 
6 Conclusion

Various methods have addresses to enhance the robustness of dimensionality reduction techniques including PCA and SC. However, they are very computationally expensive compared to non-robust ones. In this paper, we design and analyze an efficient iterative PCA and SC algorithm via optimizing a sum of non-squared $\ell_2$-norm objective, where it is in particular attractive for high-dimensional and large-scale data. Considering a growing popularity of dimensional reduction techniques in the machine learning problems, we believe that our proposed algorithm will find numerous applications in various domains.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] A. Baccini, P. Besse, and A. De Falguerolles. A 11-norm pea and a heuristic approach. Ordinal and symbolic data analysis, page 359.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] Q. Ke and T. Kanade. Robust subspace computation using 11 norm. 2003.

[11] Q. Ke and T. Kanade. Robust 1 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.

[12] Y. Kotropoulos, Constantine P., and G. R. Arce. 11-graph based music structure analysis. In International Society for Music Information Retrieval Conference, ISMIR, 2011.

[13] 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.
[14] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. \textit{arXiv preprint arXiv:1009.5055}, 2010.

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

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

[17] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In \textit{Advances in Neural Information Processing Systems}, pages 1813–1821, 2010.

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

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

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

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

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

[23] R. Vidal. A tutorial on subspace clustering. \textit{IEEE Signal Processing Magazine}, 28(2):52–68, 2010.

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

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

[26] 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 \textit{Advances in neural information processing systems}, pages 2080–2088, 2009.

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

[28] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In \textit{Computer Vision–ECCV 2006}, pages 94–106. Springer, 2006.

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

[30] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari. Guess who rated this movie: Identifying users through subspace clustering. \textit{arXiv preprint arXiv:1208.1544}, 2012.

[31] T. Zhang, A. Szlam, and G. Lerman. Median k-flats for hybrid linear modeling with many outliers. In \textit{Computer Vision Workshops (ICCV Workshops)}, 2009 \textit{IEEE 12th International Conference on}, pages 234–241. IEEE, 2009.
[32] 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.
A Proof of Theorem 2

To begin with, let $\phi_{\text{init}}$ denotes the objective value of (6) using $b_j, U_j$ and $W_j$ initialized as described in Section 3.2. In addition, we define

$$U_j^{(b)} := \arg\min_U J_{C_j^1}(U, b),$$

where $J_{C_j^1}(U, b) := \sum_{i \in C_j}(I - UU^\top)(x_i - b)_2$, i.e., $U_j^{(b)}$ is an optimal subspace of $C_j^1$ for given $b$, where the authors [4] show that R1-PCA solves the minimization. We first consider the case $m = 1$, and it is easy to check that for any point $b \in \mathbb{R}^d$, $J_{C_j^i}(U_j^{(b)}, b) \leq J_{C_j^i}(U_j^*, b_j^*) + |C_j^i|\|(I - U_j^* (U_j^*)^\top)(b_j^* - b)_2||2$, where it is from the optimality of $U_j^{(b)}$ and triangular inequalities. Using this, we derive that for $m = 1$, i.e., $C_1^1$ is the entire data points,

$$\mathbb{E}[\phi_{\text{init}}] = \frac{1}{|C_1^1|} \sum_{i \in C_1^1} J_{C_j^1}(U_j^{(x_i)}, x_i)\leq \frac{1}{|C_1^1|} \sum_{i \in C_1^1} \left[J_{C_j^i}(U_j^*, b_j^*) + |C_j^i|\|(I - U_j^* (U_j^*)^\top)(x_i - b_j^*)_2\right] = 2 \sum_{i \in C_1^1} \|(I - U_j^* (U_j^*)^\top)(x_i - b_j^*)_2\|2 = 2 \phi_{\text{global}}.

Furthermore, Theorem 1 implies that $\phi_{\text{local}} \leq \phi_{\text{init}}$ with probability 1. This completes the proof of Theorem 2 for $m = 1$.

The proof for the case $m \geq 2$ follows a similar strategy to that of K-means++, i.e., Theorem 3.1 in [1], where the additional technical issue is how to handle subspaces that do not exist in K-means++. To this end, we establish the following key lemma that is the main reason why we introduce parameter $\eta$ that is not necessary in K-means++.

**Lemma 4** If $m = 2$,

$$\mathbb{E}[\phi_{\text{local}}(C_2^*)] \leq 2\eta \phi_{\text{global}}(C_2^*),$$

then under the initialization described in Section 3.2, where $\phi(C_j^*)$ denote the contribution of data points in $C_j^*$ to the objective value $\phi$.

**Proof:** To simplify notation, we define

$$g_j(x, b) := \min \left\{ f_D(x), \|(I - U_j^{(b)} (U_j^{(b)})^\top)(x - b)_2\right\}.$$

Then, under the initialization described in Section 3.2,

$$\mathbb{E}\left[\phi_{\text{init}}(C_2^*)\right] = \sum_{i \in C_2^*} \frac{f_D(x_i)}{\sum_{i' \in C_2^*} f_D(x_{i'})} \sum_{i' \in C_2^*} g_2(x_{i'}, x_i). \quad (18)$$

From the definition of $\eta$, we have that for all $i \in C_2^*$,

$$|C_2^*| f_D(x_i) = \sum_{i' \in C_2^*} f_D(x_i) \leq \sum_{i' \in C_2^*} \eta f_D(x_{i'}). \quad (19)$$
In addition, from the definition of $g_j(x, b)$, it follows that
\[
g_j(x, b) \leq \| (I - U_j^{(b)} (U_j^{(b)})^\top) (x - b) \|_2. \tag{20}
\]

From (18), (19) and (20), we obtain
\[
E[\phi_{\text{init}}(C_j^*)] \leq \frac{\eta}{|C_j^*|} \sum_{i \in C_j^*} J_{C_j^*} (U_j^{(x_i)}, x_i).
\]

Finally, we can use the same arguments for the case $m = 1$ with the above inequality and complete the proof of Lemma 4.

The initialization described in Section 3.2 chooses a new center $b_j$ from data points as
\[
b_j = x_i \quad \text{with probability} \quad \frac{f_D(x_i)}{\sum_{i'} f_D(x_{i'})},
\]
where $f_D(x)$ denote the shortest projected distance from a data point to the closest center we have already chosen:
\[
f_D(x) = \min_{j'} \| (I - U_{j'} U_{j'}^\top) (x - b_{j'}) \|_2.
\]

Hence, a new center is chosen from $C_j^*$, with probability with $\frac{\phi(C_j^*)}{\phi}$ where $\phi$ is an objective value computed by current chosen center vector $b$, subspace $U$ and cluster membership $w$. From these observations and Lemma 4, one can use identical inductive arguments of Lemma 3.4 in [1], and complete the proof of Theorem 2 for $m \geq 2$. We omit further details in this paper.