Subspace Clustering Using Log-determinant Rank Approximation

Chong Peng, Zhao Kang, Huiqing Li, and Qiang Cheng*
Southern Illinois University
Carbondale, IL, 62901, USA
{pchong, zhao.kang, huiqing.li, qcheng}@siu.edu

ABSTRACT

A number of machine learning and computer vision problems, such as matrix completion and subspace clustering, require a matrix to be of low-rank. To meet this requirement, most existing methods use the nuclear norm as a convex proxy of the rank function and minimize it. However, the nuclear norm simply adds all nonzero singular values together instead of treating them equally as the rank function does, which may not be a good rank approximation when some singular values are very large. To reduce this undesirable weighting effect, we use a log-determinant function as a non-convex rank approximation which reduces the contributions of large singular values while keeping those of small singular values close to zero. We apply the method of augmented Lagrangian multipliers to optimize this non-convex rank approximation-based objective function and obtain closed-form solutions for all subproblems of minimizing different variables alternatively. The log-determinant low-rank optimization method is used to solve subspace clustering problem, for which we construct an affinity matrix based on the angular information of the low-rank representation to enhance its separability property. Extensive experimental results on face clustering and motion segmentation data demonstrate the effectiveness of the proposed method.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Clustering; H.2.8 [**Database Applications**]: Metrics—*Data mining*; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Subspace clustering, rank approximation, low-rank representation, nuclear norm

*to whom all correspondence should be sent.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783303.

1. INTRODUCTION

In many areas of machine learning, signal and image processing, and data mining, high-dimensional data are commonly used. Apart from being uniformly distributed, high-dimensional data often lie on low-dimensional structures. Recovering the low-dimensional subspace can well preserve and reveal the latent structure of the data in many problems. For example, face images of an individual under different lighting conditions span a low-dimensional subspace in an ambient high-dimensional space [2]. For low-dimensional subspaces, there is often a need for subspace clustering in many applications, such as face clustering [26] and motion segmentation [9, 22]. This requires partitioning data points into multiple groups based on the underlying subspaces.

Recent subspace clustering methods, such as LRR [25–27] and LRSC [14, 36], usually depend on the nuclear norm as a convex proxy of the rank function in seeking low-rank subspaces. However, unlike the rank function's treating them equally, the nuclear norm simply adds all nonzero singular values together, where the large ones may contribute exclusively to the approximation, rendering it much deviated from the true rank. Nuclear norm is essentially the l_1 norm of singular values, which is known for being biased in estimation and detrimental to large singular values. To resolve this problem, we propose to use a special log-determinant function to approximate the rank function. It attenuates the contributions from large singular values significantly compared to the nuclear norm while keeping the contributions from the small singular values close to zero. By assigning much reduced weights to large singular values, this function approximates the real rank of a matrix more closely than the nuclear norm. Because the log-determinant proxy is non-convex, for potential applications on large-scale data, we propose to optimize the associated objective function by using the method of augmented Lagrangian multipliers (ALM). For optimizing our rank approximation, we derive closed-form solutions to the subproblems of minimizing different variables alternatively within each iteration. As an application, we use our new rank approximation method for subspace clustering and experimentally validate its effectiveness.

The main contributions of this paper are summarized as follows: We use a non-convex log-determinant function for more accurate rank approximation than the nuclear norm for applications to the subspace clustering problem and we achieve promising results in a new level that are superior to several state-of-the-art methods; the optimization of a general non-convex function is generally difficult, but we build

an iterative optimization algorithm of which the core idea could be applied to a set of non-convex optimization problems; our optimization is efficient and the theoretical proof of convergence to a stationary point is provided, which is generally difficult for non-convex optimization problems.

The rest of this paper is organized as follows: We briefly review related work on subspace clustering in Section 2; Section 3 presents the proposed formulation, optimization, theoretical proof of convergence and complexity analysis; we show how to construct the affinity matrix and do the clustering in Section 4; we report experimental results on face clustering and motion segmentation benchmark data in Section 5; and Section 6 concludes our paper.

2. RELATED WORK

In the last decade a number of subspace clustering algorithms have been developed such as algebraic methods [5, 16, 29, 37], statistical methods [19, 28, 30, 32], iterative methods [20, 34, 40], and spectral clustering based methods [8, 11, 13, 14, 18, 25, 31, 36, 38]; see [35] for a review. Local spectral clustering based methods such as the local subspace affinity (LSA) [38] and spectral local best-fit flats (SLBF) [41] are based on the observation that data points in the nearest neighbor (NN) often belong to the same subspace. LSA and SLBF first project data points onto a subspace and then compute the K-NNs to fit a local affine space between each data point and its neighbor. By using the K-NNs, the affinity matrix is constructed. The locally linear manifold clustering (LLMC) [18] also fits a local subspace to each point and its K-NNs. Different from LSA and SLBF, LLMC is applicable to nonlinear subspaces. These methods are robust to outliers because an outlier is unlikely to be chosen as a neighbor of any data point. However, these methods may fail to deal well with those data points in intersections because their neighbors may belong to different subspaces.

Global spectral clustering based approaches try to resolve this problem by using global information to build similarities. Spectral curvature clustering (SCC) [8] uses data points from the entire data set to define the affinity between two points and leads to promising construction. Sparse subspace clustering (SSC) [11, 12], low-rank representation (LRR) [25–27] and low-rank subspace clustering (LRSC) [14, 36] seek a low-rank representation of the data. The similarity is then built based on this representation and used for the data segmentation. These methods are usually able to handle noise and outliers well.

In the following, we give a brief review of LRR [26], LRSC [36] and SSC [11, 13] that are closely related to our work.

2.1 Low Rank Representation (LRR)

Given the data matrix $A = [a_1, a_2, ..., a_n] \in \mathcal{R}^{d\times n}$ that contains the data points in $\mathcal{R}^d$, in the case that A is clean, LRR aims to find a low-rank representation matrix X showing mutual similarity of the points, such that $A = AX$. LRR algorithm finds X by solving the following problem:

$$\min_X \|X\|_* \qquad s.t. \quad A = AX, \tag{1}$$

where $\|\cdot\|_*$ denotes the nuclear norm. In this case, the optimal solution to Eq. (1) is given by $X = V_r V_r^T$, where $A = V_r \Sigma_r V_r^T$ is the rank r SVD of A [25] which coincides with the affinity matrix proposed by [9]. It is shown in [22] that this matrix ensures that $X_{ij} = 0$ if the i-th and j-th nodes are from different subspaces and hence X can be used to build an affinity matrix. In the case of corrupted data, LRR solves the following convex optimization problem:

$$\min_{X,E} \|X\|_* + \gamma\|E\|_{2,1} \qquad s.t. \quad A = AX + E, \tag{2}$$

where E contains corrupted entries, $\|E\|_{2,1} := \sum_j \sqrt{\sum_i E_{ij}^2}$. Subsequently, the segmentation of data A is obtained by applying a spectral clustering algorithm to the affinity matrix defined as $|X| + |X^T|$, here $|X|$ denotes the absolute value matrix of X, i.e., its ij-th element is $|X_{ij}|$.

2.2 Low Rank Subspace Clustering (LRSC)

LRSC aims to solve the non-convex optimization problem given in the following:

$$\begin{aligned} \min_{X,S,E,B} &\|X\|_* + \frac{\tau}{2}\|B - BX\|_F^2 + \frac{\alpha}{2}\|E\|_F^2 + \gamma\|S\|_1 \\ s.t. \quad & A = B + S + E, \quad X = X^T, \end{aligned} \tag{3}$$

where B is the underlying clean data matrix, S is a sparse matrix containing the gross errors, E is a matrix of fitting residuals, $\|\cdot\|_F$ denotes the Frobenius norm and $||S||_1 := \sum_i \sum_j |S_{ij}|$. Several special cases of LRSC have been considered: (1) If A is clean, letting $\alpha \to \infty$ and $\gamma \to \infty$, then $E = 0$ and $S = 0$; (2) if there is only noise, letting $\gamma \to \infty$, then $S = 0$; (3) if there is only gross error, letting $\alpha \to \infty$, then $E = 0$. After obtaining the optimal X, the same spectral clustering procedure as LRR is used, with the affinity matrix defined to be $|X|$.

2.3 Sparse Subspace Clustering (SSC)

Similar to LRR and LRSC, SSC tries to find a sparse representation of A by solving the following convex optimization problem:

$$\min_X \|X\|_1 \qquad s.t. \quad A = AX + S + E, \quad diag(X) = 0. \tag{4}$$

As shown in [11, 12] that under some conditions on the data and subspaces, the optimal solution to Eq. (4) is such that $X_{ij} = 0$ if i-th and j-th nodes are from different subspaces and hence an affinity matrix can be defined as $|X| + |X^T|$. In the case that the data is contaminated by noise and gross errors, the SSC tries to recover the sparse representation by solving the convex optimization problem:

$$\begin{aligned} \min_{X,S,E} &\|X\|_1 + \frac{\alpha}{2}\|E\|_F^2 + \gamma\|S\|_1 \\ s.t. \quad & A = AX + S + E, \quad diag(X) = 0. \end{aligned} \tag{5}$$

The subsequent clustering is essentially the same as LRR.

3. FORMULATION AND OPTIMIZATION

3.1 A Log-Determinant Rank Approximation

The nuclear norm has been theoretically proven to be the tightest convex approximation to the rank function [15]. However, it may not be a good approximation to the rank function in practical problems, because the rank function regards all nonzero singular values to have equal contributions while the nuclear norm treats the nonzero singular values differently; i.e., the larger the singular value is, the more contribution it makes to the approximation. In addition, it is usually difficult to check whether the nuclear norm satisfies

the theoretical requirements for near-optimally approximating the rank function, e.g., the incoherence property [6, 7]. To approximate the rank function more closely, we define a non-convex approximation as

$$F(X) = \log\det\left(I + X^T X\right) = \sum_{i=1}^{n} \log\left(1+\sigma_i^2\right), \quad (6)$$

where σ_i is the singular values of $X \in \mathcal{R}^{n\times n}$ for $i = 1, 2, ..., n$. When $\sigma_i = 0$, it contributes nothing to the rank approximation $F(X)$, because the corresponding term $\log(1+\sigma_i^2) = 0$. Thus the behavior of zero singular values in $F(X)$ is the same as the true rank function and the nuclear norm. However, for a large nonzero singular value, its behavior in $F(X)$ is much better than in the nuclear norm for approximating the rank, because $\log(1+\sigma_i^2) \ll \sigma_i$ when $\sigma_i > 1$. Hence, $F(X)$ approximates the rank function better by reducing the contributing weights of large singular values significantly. In addition, those small nonzero singular values can be reduced further in $F(X)$, because $\log(1+\sigma_i^2) < \sigma_i$ when $0 < \sigma_i < 1$. This implies that $F(X)$ has better noise-attenuation property than the nuclear norm, as those very small singular values are often regarded as coming from noise.

3.2 Formulation for Subspace Clustering

As an application of the above-defined rank approximation, we consider using $F(X)$ to recover the low-rank representation for subspace clustering and deriving its optimization method. We model the data matrix A by assuming it is corrupted by sparse error entries and Gaussian noise. With this modeling, we may write the data matrix as $A = B + S + E$, where B is the unknown underlying clean data matrix which is self-expressive by satisfying $B = BX$, and S and E represent the sparse error matrix and Gaussian noise respectively. Usually, those data points in the same subspace have large similarity to each other, which is called within-cluster cohesiveness; while those in different subspaces have small or ideally vanishing similarity, which is called between-cluster separability. With an ideal similarity measure, these phenomenons may render X to have low-rank. By minimizing the rank of X, we aim at recovering the underlying cluster structure of X. Using $F(X)$ as a proxy of the rank of X, we optimize the following objective function:

$$\begin{aligned} &\min_{X,S,B,E} F(X) + \alpha\|S\|_l + \beta\|E\|_F^2 \\ &s.t. \quad B = BX, A = B + S + E, \end{aligned} \quad (7)$$

where $\|S\|_l$ denotes a proper norm that captures sparse within-sample or inter-sample outliers, and α and β are positive balancing parameters. Here, we consider $\|S\|_l$ to be $\|S\|_1$ and $\|S\|_{2,1}$ for capturing within-sample and inter-sample outliers, respectively. The way of minimizing $\|S\|_l$ and $\|E\|_F^2$ to capture the outliers and data fitting residues in (7) is similar to LRSC; however, (7) is different from (3) used in LRSC in the following: 1) $F(X)$ is used to approximate the rank function rather than the nuclear norm, 2) X is not required to be symmetric as in LRSC, and 3) $\|S\|_{2,1}$ norm is considered for capturing the inter-sample outliers. Consequently, there are also considerable differences in optimization and solutions.

By relaxing the equality constraint, $B = BX$, and adding an additional term, $\|B - BX\|_F^2$, into the objective function, the constrained optimization problem (7) becomes an unconstrained optimization:

$$\min_{X,S,B} F(X) + \alpha\|S\|_l + \beta\|A - B - S\|_F^2 + \gamma\|B - BX\|_F^2, \quad (8)$$

where γ is a positive trade-off parameter. Because $F(X)$ and $B - BX$ are not convex, the optimization of (8) is not straightforward, especially when the scale of the data becomes large. We discuss the optimization of (8) in the following section.

3.3 Optimization Algorithm

To facilitate optimization on potentially large-scale data, we apply the method of ALM [4] to solve (8). By doing so, we are able to obtain closed-form solutions to all subproblems in optimizing different variables alternatively within each iteration. Because it is difficult to directly optimize X for (8), we make a change of variable, $Y = I - X$, and rewrite (8) into an augmented Lagrangian optimization:

$$\begin{aligned} L(X,S,B,Y,\Lambda,\rho) = F(X) + \alpha\|S\|_l + \beta\|A-B-S\|_F^2 \\ +\gamma\|BY\|_F^2 + \frac{\rho}{2}\|Y-(I-X)+\frac{1}{\rho}\Lambda\|_F^2, \end{aligned} \quad (9)$$

where Λ is the Lagrangian multiplier, and ρ is the penalty parameter. In each iteration, we optimize one variable by fixing all the others. Specifically, the procedure is given in the following.

3.3.1 Computing B

We optimize B while fixing the other variables. The subproblem is

$$B \leftarrow \arg\min_B \beta\|A - B - S\|_F^2 + \gamma\|BY\|_F^2. \quad (10)$$

The objective function in (10) is quadratic and strongly convex in B. By taking the derivative with respect to B and setting it to zero, we obtain a closed-form solution to (10):

$$B = \beta(A - S)(\gamma YY^T + \beta I)^{-1}. \quad (11)$$

3.3.2 Computing S

For S-minimization, we have the following subproblem:

$$S \leftarrow \arg\min_S \alpha\|S\|_l + \beta\|A - B - S\|_F^2, \quad (12)$$

which admits closed-form solutions for both $\|S\|_1$ and $\|S\|_{2,1}$ norms. Specifically, for $\|S\|_1$ norm, we apply the shrinkage-thresholding operator defined in [3, 10] element-wisely and obtain the optimal S given by

$$[S]_{ij} = \left(\left|[A-B]_{ij}\right| - \alpha/2\beta\right)_+ sign\left([A-B]_{ij}\right), \quad (13)$$

where $(z)_+ = max(0, z)$ is a nonnegative mapping. For $\|S\|_{2,1}$ norm, according to Proposition 1 of [39] which is proved by [21] as Lemma 1, we have the solution to the S-minimization problem by columns of S:

$$[S^*]_i = \begin{cases} \frac{\|[A-B]_i\|_2 - \frac{\alpha}{2\beta}}{\|[A-B]_i\|_2}[A-B]_i, & \text{if } \|[A-B]_i\|_2 > \frac{\alpha}{2\beta} \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

3.3.3 Computing X

Denoting $D = I - Y - \frac{1}{\rho}\Lambda$, the X-minimization reduces to the following:

$$X \leftarrow \arg\min_X \log\det\left(I + X^T X\right) + \frac{\rho}{2}\|X - D\|_F^2. \quad (15)$$

To solve (15), we give some definitions and theorems.

DEFINITION 3.1. *A function $f : \mathcal{R}^n \to (-\infty, \infty)$ is* absolutely symmetric *if $f(x)$ is invariant under arbitrary permutations and sign changes of the elements of x.*

THEOREM 3.1. *([24]) For an absolutely symmetric function f, function $F : \mathcal{R}^{n_1 \times n_2} \to \mathcal{R}$ is unitarily invariant if and only if $F(X) = f(\sigma(X)) = f \circ \sigma(X)$, where $X \in \mathcal{R}^{n_1 \times n_2}$ has the singular value decomposition (SVD) $X = Udiag(\sigma(X))V^T$, $\sigma(X) \in \mathcal{R}^n$ are singular values of X, and $n = \min(n_1, n_2)$. And provided f is differentiable, the gradient of $F(X)$ at X is $\frac{\partial F(X)}{\partial X} = Udiag(\theta)V^T$ where $\theta = \frac{\partial f(y)}{\partial y}|_{y=\sigma(X)}$.*

Hence, $F(X) = \log\det\left(I + X^T X\right)$ is unitarily invariant and differentiable, because $F(X) = f \circ \sigma(X)$, and $f(\sigma) = \sum_{i=1}^n \log\left(1 + \sigma_i^2\right)$ is absolutely symmetric and differentiable.

DEFINITION 3.2. *The Moreau-Yosida proximity operator, denoted as $prox_{f,\rho}(v)$, is defined as a function of v given by $prox_{f,\rho}(v) = \arg\min_x (f(x) + \frac{\rho}{2}||x - v||_2^2)$.*

We obtain a result that reduces the problem of finding the Moreau-Yosida proximity operator of a matrix to that of finding the corresponding proximity operator of a vector.

THEOREM 3.2. *If $F(X)$ is unitarily invariant, and the SVD of $D \in \mathcal{R}^{m \times n}$ is $D = U\Sigma_D V^T$, with $\Sigma_D = diag(\sigma(D))$, then the minimizer of the optimization problem,*

$$\min_X F(X) + \frac{\rho}{2}\|X - D\|_F^2, \quad (16)$$

is $X^* = U\Sigma_X^* V^T$, and $\Sigma_X^* = diag(\sigma^*)$ with $\sigma^* = prox_{f,\rho}(\sigma(D))$, for some absolutely symmetric function f such that $F(X) = f \circ \sigma(X)$.

PROOF. Let $D = U\Sigma_D V^T$ be the skinny SVD of D, then then $\Sigma_D = U^T D V$. Denoting $Z = U^T X V$ which has exactly the same singular values as X, we have

$$\begin{aligned}
&F(X) + \frac{\rho}{2}\|X - D\|_F^2 && (17)\\
=&F(Z) + \frac{\rho}{2}\|Z - \Sigma_D\|_F^2 && (18)\\
=&F(\Sigma_Z) + \frac{\rho}{2}\|Z - \Sigma_D\|_F^2 && (19)\\
=&F(\Sigma_Z) + \frac{\rho}{2}\left(\|Z\|_F^2 + \|\Sigma_D\|_F^2 - 2\left\langle Z, \Sigma_D\right\rangle\right) && (20)\\
\geq&F(\Sigma_Z) + \frac{\rho}{2}\left(\|\Sigma_Z\|_F^2 + \|\Sigma_D\|_F^2 - 2\left\langle \Sigma_Z, \Sigma_D\right\rangle\right) && (21)\\
=&F(\Sigma_Z) + \frac{\rho}{2}\|\Sigma_Z - \Sigma_D\|_F^2 && (22)\\
=&F(\Sigma_X) + \frac{\rho}{2}\|\Sigma_X - \Sigma_D\|_F^2 && (23)\\
=&f \circ \sigma(X) + \frac{\rho}{2}\|\sigma(X) - \sigma(D)\|_2^2 && (24)\\
\geq&f(\sigma^*) + \frac{\rho}{2}||\sigma^* - \sigma(D)||_2^2. && (25)
\end{aligned}$$

In the above, (18) holds because the Frobenius norm is unitarily invariant; (19) holds because $F(Z)$ is unitarily invariant; (20) is true by the von Neumann's trace inequality; and (23) holds as $\Sigma_Z = \Sigma_X$. Therefore, (23) is a lower bound of (17), where Σ_X^* is obtained by minimizing (23). Note that the equality in (21) is attained if $Z = \Sigma_Z$. Because $\Sigma_X = \Sigma_Z = Z = U^T X V$, the SVD of X is $X = U\Sigma_X V^T$, which is the minimizer of problem (16). Furthermore, when $F(X) = f \circ \sigma(X)$, (24) is true; and (25) holds because σ^* is the Moreau-Yosida proximity operator of f with penalty ρ. Hence the proof is completed. □

Based on Theorem 3.2, the function f for our F is

$$f(\sigma) = \sum_i \log(1 + \sigma_i^2), \quad (26)$$

which is separable for each σ_i. Hence, the X-minimization problem can be reduced to scalar minimization problems

$$\arg\min_{\sigma_i} f_{X,D}(\sigma_i), \quad s.t. \quad \sigma_i \geq 0,$$

where $f_{X,D}(\sigma_i) = \log\left(1 + \sigma_i^2\right) + \frac{\rho}{2}\left(\sigma_i - \sigma_i^D\right)^2$ and σ_i^D are singular values of D, $i = 1, 2, \cdots, n$. By the first-order optimality condition, we take the derivative of $f_{X,D}(\sigma_i)$ and set it to be zero, giving rise to

$$\rho\sigma_i^3 - \rho\sigma_i^D\sigma_i^2 + (\rho + 2)\sigma_i - \rho\sigma_i^D = 0 \quad s.t. \quad \sigma_i \geq 0. \quad (27)$$

In general, the equation in (27) has three roots. But we need to enforce the nonnegativity constraint. It can be easily shown that there is at least one nonnegative root of (27) located in $(0, \sigma_i^D)$ when $\sigma_i^D > 0$. We have a closed-form analytical solution to the Moreau-Yosida proximity operator of f in the following.

PROPOSITION 3.1. *Consider $g(x) = \log(1 + x^2)$. When $\sigma_i^D = 0$, $prox_{g,\rho}(\sigma_i^D) = 0$. Under the condition that $\sigma_i^D > 0$ and $\rho > 1/4$, $prox_{g,\rho}(\sigma_i^D)$ is located in $(0, \sigma_i^D)$ and is the unique positive root of the cubic equation (27).*

PROOF. In order to get $prox_{g,\rho}(\sigma_i^D)$, we need to minimize $f_{X,D}(\sigma_i)$ under the constraint of $\sigma_i \geq 0$. The derivative of $f_{X,D}(\sigma_i)$ is

$$f'_{X,D}(\sigma_i) = \frac{2\sigma_i}{1 + \sigma_i^2} + \rho(\sigma_i - \sigma_i^D),$$

and the second derivative is

$$f''_{X,D}(\sigma_i) = \frac{\rho\sigma_i^4 + (2\rho - 2)\sigma_i^2 + (2 + \rho)}{(1 + \sigma_i^2)^2}.$$

Case 1: If $\sigma_i^D = 0$, because $\sigma_i \geq 0$, we always have $f'_{X,D}(\sigma_i) \geq 0$. That is, $f_{X,D}(\sigma_i)$ is nondecreasing for any $\sigma_i \geq 0$ and strictly increasing for any $\sigma_i > 0$. Hence the minimizer is σ_i^*=0.

Case 2: If $\sigma_i^D > 0$, then there exists at least one root in $(0, \sigma_i^D)$ $f'_{X,D}(\sigma_i) = 0$, because $f'_{X,D}(0) = -\rho\sigma_i^D < 0$, and $f'_{X,D}(\sigma_i^D) > 0$. If $\rho > \frac{1}{4}$, it is easily shown by using the discriminant of $\rho\sigma_i^4 + (2\rho - 2)\sigma_i^2 + (2 + \rho)$ that $f''_{X,D}(\sigma_i) > 0$. Therefore, $f_{X,D}(\sigma_i)$ is a strictly convex function with a unique global minimizer, which is the unique root of $f'_{X,D}(\sigma_i) = 0$. Note that $f'_{X,D}(\sigma_i) = 0$ is equivalent to (27), hence the proposition is proven. □

In the case that $0 < \rho \leq \frac{1}{4}$, we need to determine the minimizer in the following way: Denote the set of nonnegative root(s) of the cubic equation in (27) by Ω_+. By

the first-order necessary optimality condition, the minimizer $prox_{g,\rho}(\sigma_i^D)$ needs to be chosen from $\{0\} \cup \Omega_+$. That is, $\sigma_i^* = \arg\min_{\sigma \in \{0\} \cup \Omega_+} f_{X,D}(\sigma)$. In our experiments, we choose to initialize $\rho = 1$ and increase its value in each iteration. Therefore, as long as $\sigma_i^D > 0$, we know the minimizer $\sigma_i^* \in (0, \sigma_i^D)$, and it is the unique positive root of (27); otherwise, $\sigma_i^* = 0$.

3.3.4 Computing Y

Similarly to B-minimization, we compute Y by solving the following optimization problem

$$Y \leftarrow \arg\min_{Y} \gamma \|BY\|_F^2 + \frac{\rho}{2}\|Y - (I - X) + \frac{1}{\rho}\Lambda\|_F^2. \quad (28)$$

The objective function of (28) is also quadratic and strongly convex in Y. We take the derivative with respect to Y and let it equal to zero, giving rise to a closed-form solution

$$Y = \left(2\gamma B^T B + \rho I\right)^{-1} \left(\rho I - \rho X - \Lambda\right). \quad (29)$$

3.3.5 Update Λ and ρ

For Λ and ρ, we update them in the standard way:

$$\Lambda \leftarrow \Lambda + \rho\left(Y - I + X\right), \quad (30)$$

$$\rho \leftarrow \mu\rho, \quad (31)$$

where $\mu > 1$ is a parameter that controls the convergence speed. The larger μ is, the fewer iterations are needed for the algorithm to converge. But for large μ, we may lose some precision of the final objective function value. Moreover, our theoretical convergence analysis requires that the rate of increase of ρ should be slower than the vanishing rate of $Y_{k+1} - Y_k$ to guarantee the convergence of our algorithm. In our experiments, we set $\mu = 1.1$ and $\rho_0 = 1$.

3.3.6 Initialization

Since the objective function in (8) is not convex, different initializations may lead to different solutions. As numerically shown in our experiments, it is effective to initialize the variables as $S_0 = 0$, $Y_0 = 0$, $\Lambda_0 = 0$.
In summary, we outline the optimization procedure of solving (8) in Algorithm 1.

Algorithm 1: Solving (8) by ALM

Input: α, β, γ, μ, A, k_{max}
Initialize: S_0, Y_0, Λ_0, ρ_0, and $k = 0$.
repeat
  $D_{k+1} = I - Y_k - \rho_k\Lambda_k$.
  Compute SVD of D_{k+1}, $D_{k+1} = U_{D_{k+1}}\Sigma_{D_{k+1}}V_{D_{k+1}}^T$.
  Solve problems of $\sigma_i^* = prox_{\log(1+x^2),\rho_k}(\sigma_i^{D_{k+1}})$, $i = 1, 2, \cdots, n$, using Proposition 3.1.
  $X_{k+1} = U_{D_{k+1}} diag\{\sigma_1^*, \sigma_2^*, ..., \sigma_n^*\}V_{D_{k+1}}^T$.
  $B_{k+1} = \beta\left(A - S_k\right)\left(\gamma Y_k Y_k^T + \beta I\right)^{-1}$.
  Get S_{k+1} by (12) or (14) depending on l norm we use.
  $Y_{k+1} = \left(2\gamma B_{k+1}^T B_{k+1} + \rho_k I\right)^{-1}\left(\rho_k I - \rho_k X_{k+1} - \Lambda_k\right)$.
  Update $\Lambda_{k+1} = \Lambda_k + \rho_k\left(Y_{k+1} - I + X_{k+1}\right)$; $\rho_{k+1} = \mu\rho_k$; $k = k + 1$.
until $k \geq k_{max}$ or $\{X_k, B_k, S_k\}$ converges
Output: $X^* = X_k$

3.4 Theoretical Convergence Analysis

It is known to be hard to prove an algorithm for nonconvex optimization to converge to a stationary point. In this section, we manage to theoretically prove that our Algorithm 1 converges to a stationary point of the objective function. Empirically, we obtain stronger results than this convergence guarantee, because in our experiments we always observe that this stationary point is a local minimum.

In the following, for a simpler notation of (8) and its augmented Lagrangian, we write them as

$$\begin{aligned} G(X, S, B) =& F(X) + \alpha\|S\|_l \\ &+ \beta\|A - B - S\|_F^2 + \gamma\|B - BX\|_F^2, \end{aligned} \quad (32)$$

and

$$\begin{aligned} L(X, S, B, Y, \Lambda, \rho) =& G(X, S, B) + \langle Y - (I - X), \Lambda\rangle \\ &+ \frac{\rho}{2}\|Y - (I - X)\|_F^2. \end{aligned} \quad (33)$$

THEOREM 3.3. *The sequences $\{\Lambda_k\}$, $\{X_k\}$, $\{S_k\}$, $\{B_k\}$ and $\{Y_k\}$ are bounded as long as $\rho_k(Y_k - Y_{k+1})$ is bounded, $\sum \frac{\rho_{k+1}}{\rho_k^2} < \infty$ and $\sum \frac{1}{\rho_k} < \infty$.*

PROOF. To minimize X at iteration $k+1$, the optimal X_{k+1} needs to satisfy the first-order optimality condition, that is,

$$\begin{aligned} &\nabla_X L\left(X, S_k, B_k, Y_k, \Lambda_k, \rho_k\right)|_{X_{k+1}} \\ =&\nabla_X F\left(X\right)|_{X_{k+1}} + \rho_k\left(Y_k - (I - X_{k+1}) + \frac{1}{\rho}\Lambda_k\right) \\ =&\nabla_X F\left(X\right)|_{X_{k+1}} + \rho_k\left(Y_k - Y_{k+1} + Y_{k+1} - I + X_{k+1} + \frac{1}{\rho}\Lambda_k\right) \\ =&0. \end{aligned} \quad (34)$$

Note that the updating rule for Λ is

$$\Lambda_{k+1} = \Lambda_k + \rho_k\left(Y_{k+1} - I + X_{k+1}\right), \quad (35)$$

hence $\nabla_X F\left(X\right)|_{X_{k+1}} + \Lambda_{k+1} + \rho_k\left(Y_k - Y_{k+1}\right) = 0$. By Theorem 3.1, we know

$$\begin{aligned} &\nabla_X F\left(X\right)|_{X_{k+1}} \\ =&U diag\left(\frac{df(\sigma_1)}{d\sigma_1}, \cdots, \frac{df(\sigma_n)}{d\sigma_n}\right)V^T \\ =&U diag\left(\frac{2\sigma_1}{1+\sigma_1^2}, \cdots, \frac{2\sigma_n}{1+\sigma_n^2}\right)V^T. \end{aligned} \quad (36)$$

Because $0 \leq \frac{2\sigma_i}{1+\sigma_i^2} \leq 1$, we have $\nabla_X F\left(X\right)|_{X_{k+1}}$ is bounded. Under the condition that $\rho_k(Y_k - Y_{k+1})$ bounded, it is seen that Λ_{k+1} is bounded. Then

$$\begin{aligned} &L\left(X_k, S_k, B_k, Y_k, \Lambda_k, \rho_k\right) \\ =&L\left(X_k, S_k, B_k, Y_k, \Lambda_{k-1}, \rho_{k-1}\right) \\ &+ \langle\Lambda_k, Y_k - I + X_k\rangle + \frac{\rho_k}{2}\|Y_k - I + X_k\|_F^2 \\ &- \langle\Lambda_{k-1}, Y_k - I + X_k\rangle - \frac{\rho_{k-1}}{2}\|Y_k - I + X_k\|_F^2 \\ =&L\left(X_k, S_k, B_k, Y_k, \Lambda_{k-1}, \rho_{k-1}\right) \\ &+ \langle\Lambda_k - \Lambda_{k-1}, Y_k - I + X_k\rangle + \frac{\rho_k - \rho_{k-1}}{2}\|Y_k - I + X_k\|_F^2 \\ =&L\left(X_k, S_k, B_k, Y_k, \Lambda_{k-1}, \rho_{k-1}\right) + \frac{\rho_k + \rho_{k-1}}{2\rho_{k-1}^2}\|\Lambda_k - \Lambda_{k-1}\|_F^2. \end{aligned} \quad (37)$$

Here, the last equation is obtained by using the Λ-updating rule (35). So, by the alternative optimization of our algorithm by repeating the optimization w.r.t Y, S, B and X, we have

$$\begin{aligned} &L\left(X_{k+1}, S_{k+1}, B_{k+1}, Y_{k+1}, \Lambda_k, \rho_k\right) \\ \leq &L\left(X_k, S_k, B_k, Y_k, \Lambda_{k-1}, \rho_{k-1}\right)+\frac{\rho_k+\rho_{k-1}}{2 \rho_{k-1}^2}\left\|\Lambda_k-\Lambda_{k-1}\right\|_F^2 . \end{aligned} \tag{38}$$

Iterating the inequality in (38) for k times, we arrive at:

$$\begin{aligned} &L\left(X_{k+1}, S_{k+1}, B_{k+1}, Y_{k+1}, \Lambda_k, \rho_k\right) \\ \leq &L\left(X_1, S_1, B_1, Y_1, \Lambda_0, \rho_0\right)+\sum_{i=1}^k \frac{\rho_i+\rho_{i-1}}{2 \rho_{i-1}^2}\left\|\Lambda_i-\Lambda_{i-1}\right\|_F^2 \\ \leq &L\left(X_1, S_1, B_1, Y_1, \Lambda_0, \rho_0\right)+C \sum_{i=1}^k \frac{\rho_i+\rho_{i-1}}{2 \rho_{i-1}^2}, \end{aligned} \tag{39}$$

where C is a an upper bound for $\{\|\Lambda_i - \Lambda_{i-1}\|_F^2\}$. The existence of such a C is guaranteed because, under the condition that $\rho_k(Y_k - Y_{k+1})$ is bounded, and that $\{\Lambda_i\}$ is bounded, and thus $\|\Lambda_i - \Lambda_{i-1}\|_F^2 \leq 2\|\Lambda_i\|_F^2 + 2\|\Lambda_{i-1}\|_F^2$ is also bounded. Under the conditions of Theorem 3.3, it is clear that $L\left(X_{k+1}, S_{k+1}, B_{k+1}, Y_{k+1}, \Lambda_k, \rho_k\right)$ is bounded. Adding $\frac{1}{2\rho_k}||\Lambda_k||_F^2$ to the augmented Lagrangian function (33), we can rewrite (33) as

$$\begin{aligned} &L\left(X_{k+1}, S_{k+1}, B_{k+1}, Y_{k+1}, \Lambda_k, \rho_k\right)+\frac{1}{2 \rho_k}||\Lambda_k||_F^2 \\ =&F(X_{k+1})+\alpha\|S_{k+1}\|_l+\beta\|A-B_{k+1}-S_{k+1}\|_F^2 \\ &+\gamma\|B_{k+1}Y_{k+1}\|_F^2+\frac{\rho_k}{2}\|Y_{k+1}-I+X_{k+1}+\frac{1}{\rho_k}\Lambda_k\|_F^2 . \end{aligned} \tag{40}$$

Because $\{\Lambda_k\}$ is bounded, the left-hand side of (40) is bounded. As each term on the right-hand side of (40) is nonnegative, each term must be bounded. $F(X_{k+1}) = \sum_i \log(1 + \sigma_i^2(X_{k+1}))$ being bounded implies that all singular values of X_{k+1} are bounded and thus X_{k+1} is bounded. From the second term it is obvious that S_{k+1} is bounded for both $\|S\|_1$ and $\|S\|_{2,1}$ cases. Then B_{k+1} is bounded because $\beta\|A-B_{k+1}-S_{k+1}\|_F^2$ is bounded. Since $Y_{k+1} = \frac{1}{\rho_k}(\Lambda_{k+1} - \Lambda_k) + I - X_{k+1}$according to (35) and $\{\Lambda_{k+1}\}$ is bounded, clearly we have Y_{k+1} is bounded. □

THEOREM 3.4. *Let $\{X_k, S_k, B_k, Y_k, \Lambda_k\}$ be the sequence generated by Algorithm 1. Under the assumptions that $\sum \frac{\rho_{k+1}}{\rho_k^2} < \infty$, $\sum \frac{1}{\rho_k} < \infty$ and $\rho_k\left(Y_{k+1} - Y_k\right) \to 0$, this sequence has at least one accumulation point. For any accumulation point $\{X^*, S^*, B^*, Y^*, \Lambda^*\}$, $\{X^*, S^*, B^*\}$ is a stationary point of optimization problem (8).*

PROOF. Because $\rho_k(Y_{k+1}-Y_k) \to 0$, it is clear that $\rho_k(Y_{k+1} - Y_k)$ is bounded. Under the conditions on ρ_k, we know that $\{X_k, S_k, B_k, Y_k, \Lambda_k\}$ is a bounded sequence by Theorem 3.4. By the Bolzano-Weierstrass theorem, the sequence must have at least one accumulation point, which is denoted by $\{X^*, S^*, B^*, Y^*, \Lambda^*\}$. Without loss of generality, we assume that $\{X_k, S_k, B_k, Y_k, \Lambda_k\}$ itself converges to $\{X^*, S^*, B^*, Y^*, \Lambda^*\}$. Next, we prove that $\{X^*, S^*, B^*\}$ is a stationary point of problem (8). As $\Lambda_{k+1} = \Lambda_k + \rho_k(Y_{k+1} - I + X_{k+1})$, we have $Y_{k+1} - I + X_{k+1} = \frac{1}{\rho_k}(\Lambda_{k+1} - \Lambda_k)$. Because $\rho_k \to \infty$ and $\{\Lambda_k\}$ is bounded, we get $Y_{k+1} - I + X_{k+1} \to 0$, i.e., $Y^* = I - X^*$, thus the primal feasibility condition is satisfied by Y^* and X^*. By first-order optimality condition and the definition of X_{k+1}, we have $\nabla_X F(X)|_{X_{k+1}} + \Lambda_k + \rho_k(Y_k - I + X_{k+1}) = \nabla_X F\left(X\right)|_{X_{k+1}} + \Lambda_{k+1} + \rho_k(Y_k - Y_{k+1}) = 0$. Let $k \to \infty$, with the assumption that $\rho_k(Y_{k+1} - Y_k) \to 0$, we get $\nabla_X F\left(X\right)|_{X^*} + \Lambda^* = 0$. For the Y-update at the $(k+1)$th iteration, Y_{k+1} satisfies the following equation:

$$2\gamma B_{k+1}^T B_{k+1} Y_{k+1} + \rho_k\left[Y_{k+1} - (I - X_{k+1}) + \frac{1}{\rho_k}\Lambda_k\right] = 0;$$

i.e., $2\gamma B_{k+1}^T B_{k+1} Y_{k+1} = -\Lambda_k + \rho_k\left(I - X_{k+1}\right) - \rho_k Y_{k+1} = -\Lambda_k - \Lambda_{k+1}$. Let $k \to \infty$, we get $\gamma {B^*}^T B^* Y^* + \Lambda^* = 0$. The other KKT conditions that $\nabla_S L(X, S, B, Y, \Lambda)|_{S^*} = 0$ and $\nabla_B L(X, S, B, Y, \Lambda)|_{B^*} = 0$ are easy to verify.

Now we can see that $\{X^*, S^*, B^*, Y^*, \Lambda^*\}$ satisfies the KKT conditions of $L(X, S, B, Y, \Lambda)$; therefore $\{X^*, S^*, B^*\}$ is a stationary point of problem (8). □

3.5 Computational Cost and Empirical Convergence

The computational cost of updating each variable in each iteration is of Algorithm 1 as follows: Solving B and Y, the complexity is dominated by matrix inversion and multiplication, which cost $O(n^3 + dn^2)$ for both of them. Updating X takes $O(n^3)$ operations by SVD. To update Λ, there are $O(n^2)$ multiplications and to update S, it takes $O(dn)$ for both $\|S\|_1$ and $\|S\|_{2,1}$. In all, the total cost is $O(n^3+dn^2)$ per iteration. Let k denote the number of iterations to converge, the computational cost of our algorithm is $O(kn^3 + kdn^2)$.

Because the first and last terms in the objective function, $\log\det(I + X^T X)$ and $\|B - BX\|_F^2$, are not convex, in general, it is difficult to prove the convergence of the optimization algorithm to an optimal point. We have managed to obtain a theoretical proof of the convergence of Algorithm 1 to a stationary point in Section 3.4. Empirically we always observe the convergence of our optimization algorithm and the converged point gives rise to promising numerical results. We also illustrate examples of X constructed from the data with different numbers of subspaces in Fig. 1. They clearly demonstrate dominant block diagonal structures which may facilitate the construction of salient affinity matrix for subspace clustering.

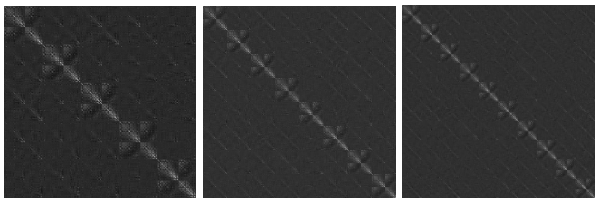

Figure 1: Sample images of X obtained from Extended Yale B data set. There are 5, 8 and 10 objects in the data from left to right respectively. The parameters are $\alpha = 0.1$, $\beta = 0.03$ and $\gamma = 0.08$.

4. SUBSPACE CLUSTERING WITH LOG-DETERMINANT APPROXIMATION

After obtaining the optimal X^* from minimization of (8), we define an affinity matrix using an idea inspired by [25]. Assuming the skinny SVD of X^* is $U\Sigma V^T$, the matrix UU^T is useful for subspace segmentation as U consists of the left eigenvectors of X^* and UU^T identifies the column space of X^* [25]. We use the weighted column space for constructing

an affinity matrix. For this purpose, let $M = U\Sigma^{1/2}$ which represents weighted left eigenvectors. Then we normalize the rows of M to get $\bar{U}$. Finally, the affinity matrix W is defined as

$$[W]_{ij} = \left(|[\bar{U}\bar{U}^T]_{ij}|\right)^{\phi}, \tag{41}$$

where $\phi \geq 1$ is a parameter that controls the sharpness of the affinity between two data points. A large ϕ may help separate the clusters, but as ϕ becomes large, the intra-cluster cohesiveness would be degraded. Therefore, we need to achieve a balance between within-cluster cohesiveness and between-cluster separability. In our experiments, we usually set ϕ to be 4 or 6. Having defined the affinity matrix W, we use it to perform spectral clustering in a way similar to [1]. We outline this spectral clustering step in Algorithm 2. The overall procedure for subspace clustering with log-determinant approximation, called as SCLA, is given in Algorithm 3. As special cases, SCLA_1 and $\text{SCLA}_{2,1}$ denote the proposed algorithm with $\|S\|_l$ being $\|S\|_1$ and $\|S\|_{2,1}$ norms, respectively.

Algorithm 2: Spectral Clustering

1: **Input**: Affinity matrix $W \in \mathcal{R}^{n\times n}$ and the number of subspaces K
2: Define $L = Z^{-1/2}WZ^{-1/2}$, where Z is diagonal with $Z_{ii} = \sum_{j=1}^{n} W_{ij}$.
3: Find the eigenvectors u_k, $k = 1, 2, ..., n$, corresponding to n eigenvalues of L and construct the matrix $U = [u_1, ..., u_n] \in \mathcal{R}^{n\times n}$.
4: Normalize U by rows by $U \leftarrow PU$, where P is diagonal and $P_{ii} = 1/\|U_{i,:}\|_2 = 1/\sqrt{\sum_{j=1}^{n} U_{ij}^2}$.
5: Apply k-means algorithm to cluster U into K groups by treating the rows of U as examples.
6: **Output**: Cluster labels for all data points

5. EXPERIMENTS

In this section, we evaluate the proposed subspace clustering algorithm on two applications: (1) face image clustering; and (2) motion segmentation in video data. We use the subspace clustering error rate, which is the number of misclassified points divided by the number of total points, to evaluate the performance. We compare our method with several state-of-the-art subspace clustering algorithms: LSA [38], SCC [8], LRR [26], SSC [13], and LRSC [14]. For a fair comparison, we specify the number of clusters in the clustering step for all algorithms.

5.1 Face Clustering

Face clustering groups a set of face images into different clusters that correspond to different individuals. In Fig. 2, we show some face image examples from the Extended Yale B data set [23]. We use this data set to evaluate the performance of the proposed method. In this data, there are

Algorithm 3: Subspace Clustering with Log-determinant Approximation (SCLA)

1: Obtain X^* using Algorithm 1.
2: Construct W by (41).
3: Apply Algorithm 2 to W.

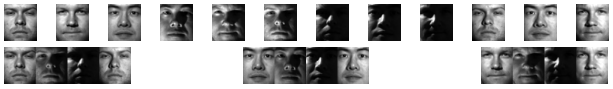

Figure 2: Face clustering example: A set of 12 images belonging to three persons (shown in the top row) are clustered into different classes as shown in the bottom row.

38 individuals, each of which has 64 frontal face images under different lighting conditions. The images are cropped to 192×168 pixels. Following [13, 36], we downsample all the images to 48×42 pixels to reduce the computational cost, and vectorize each image as a data point. Also following [13], we divide those 38 objects into four groups corresponding to objects 1-10, 11-20, 21-30 and 31-38, respectively. For the first three groups, we consider all the choices of $n \in \{2, 3, 5, 8, 10\}$ objects; and for the last group, we consider all the choices of $n \in \{2, 3, 5, 8\}$ objects. We conduct experiments for all these sets. Table 1 reports the clustering error rates of SCLA along with the other algorithms whose results are obtained by following [26, 36] on Extended Yale B data set. We show the mean and median error rates of each method with different number of objects. From Table 1, we observe that SCLA has significantly higher mean accuracy than the other methods in all cases; and in terms of median accuracy, SCLA has inferior or comparable performance than SSC on cases with 2 or 3 objects, but superior performance on other cases. It is seen that SCLA works well on the task of segmenting face images with a large number of subjects and performs quite resiliently to the change of object numbers. The parameter values of α, β, γ and ϕ for SCLA_1 are 0.1, 0.03, 0.08 and 4; and 0.1, 0.03, 0.05 and 4 for $\text{SCLA}_{2,1}$, respectively. These parameters are chosen such that we may achieve around the highest accuracy that is commonly used in the literature, which will be analyzed later. Using this above given set of parameters, the sparse term S in SCLA_1 is zero. This may imply that in this data set the images have no outstanding within-sample outliers throughout the data set. Under varying illumination conditions, shadows may be present in the images, but it appears that SCLA_1 is quite insensitive to shadows or illumination conditions, possibly owing to the condensation effect of the log function used in the rank approximation. On the other hand, in another experiment on Yale B[1] data set [17], SCLA_1 produces nonzero sparse matrices for the best clustering performance. The error rates on Yale B data are shown in Table 2. $\text{SCLA}_{2,1}$ produces nonzero sparse term S using the above mentioned parameters. Fig. 3 shows that S and E capture certain "outlying" and noise information, respectively, by SCLA_1 and $\text{SCLA}_{2,1}$ on Yale B data set. We observe that $\text{SCLA}_{2,1}$ has better performance than SCLA_1. This implies that capturing sparse entries by examples is more suitable than by the whole data set. To demonstrate the effectiveness of SCLA, we also list the Wilcoxon signed rank test results in Table 5 and it reveals that SCLA outperforms the other methods under the significant level 0.05.

[1]Yale B contains the first 10 classes of Extended Yale B data set half of which are corrupted by gross error, i.e., the shadow. The other methods do not work well on this data and we show the robustness of our method on the heavily corrupted data with significant improvement in accuracy.

Table 1: Clustering error rates (%) on EYaleB data set

Method	LSA	SCC	LRR	LRR-H	LRSC	SSC	$SCLA_1$	$SCLA_{2,1}$
2 Objects								
Mean	32.80	16.62	9.52	2.54	5.32	1.86	1.31	**1.17**
Median	47.66	7.82	5.47	0.78	4.69	**0.00**	0.78	0.78
3 Objects								
Mean	52.29	38.16	19.52	4.21	8.47	3.10	1.98	**1.89**
Median	50.00	39.06	14.58	2.60	7.81	**1.04**	1.56	**1.04**
5 Objects								
Mean	58.02	58.90	34.16	6.90	12.24	4.31	2.76	**2.57**
Median	56.87	59.38	35.00	5.63	11.25	2.50	2.50	**2.19**
8 Objects								
Mean	59.19	66.11	41.19	14.34	23.72	5.85	3.48	**3.21**
Median	58.59	64.65	43.75	10.06	28.03	4.49	3.12	**2.73**
10 Objects								
Mean	60.42	73.02	38.85	22.92	30.36	10.94	3.85	**3.70**
Median	57.50	75.78	41.09	23.59	28.75	5.63	**3.12**	3.28

The best performance is boldfaced.

Table 2: Clustering error rates (%) on Yale B data set

Algorithms	LSA	LRR-H	LRSC	SSC	$SCLA_1$	$SCLA_{2,1}$
Error rate (%)	59.52	20.94	35.78	35	**3.28**	**3.28**

5.2 Motion Segmentation

Motion segmentation clusters a set of points, each of which consists of x- and y-coordinates, into multiple groups corresponding to different rigid-body motions. For such a problem, the data matrix $A \in \mathcal{R}^{2F \times N}$, where F is the number of frames in a video sequence and N is the number of 2-dimensional points on the motion trajectory on each frame. We use the Hopkins-155 data[2] [33] to evaluate motion segmentation performance of SCLA. This data consists of 155 video sequences belonging to three categories: traffic, articulated, and checkerboard. Among these sequences, 120 have two motions and 35 have three motions. Several examples from this data are shown in Fig. 4 and some statistics are listed in Table 3. Table 4 shows the clustering error rates of SCLA on Hopkins-155 along with the other methods whose results are obtained by following [26, 36]. The parameter values of α, β, γ and ϕ for $SCLA_1$ are 0.2, 150, 50, and 6; and 1, 150, 50 and 6 for $SCLA_{2,1}$, respectively and will be analyzed later. From Table 4, it is evident that SCLA achieves the best performance among all seven methods. Furthermore, $SCLA_{2,1}$ almost equally with $SCLA_1$ since their performance are very close. This may be due to the small noise in Hopkins-155 data and hence the difference between the effects of $\|S\|_1$ and $\|S\|_{2,1}$ is not so significant in this case. Also, we have the statistical test to show the effectiveness of SCLA compared with other methods. Since the data in this scenario is small, we combine the results of Tables 1 and 4 and give the p-values in the bottom line of Table 5. The statistical results demonstrates that SCLA outperforms the other methods under the significant level of 0.01.

[2] `http://www.vision.jhu.edu/data/hopkins155`

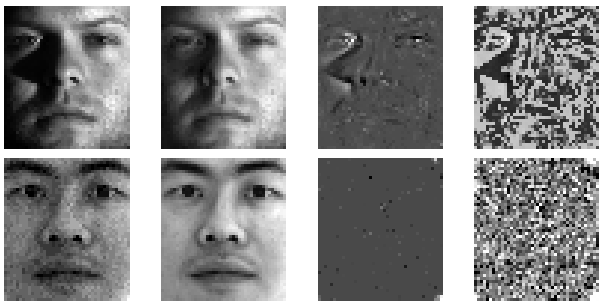

Figure 3: Sample images using Yale B data set showing different data matrices captured by $SCLA_1$ (top) and $SCLA_{2,1}$ (bottom). From left to right on each row are the original image, the underlying clean image, the sparse error image, and Gaussian noise image contained (as columns) in A, BX, S, and E, respectively.

Figure 4: Examples from Hopkins-155 data with ground truth segmentation consisting of 3, 2, and 3 objects, respectively. From left to right are video sequences of traffic, articulated and checkerboard.

Table 3: Statistics of Hopkins-155 data set

Categories	2 Motions			3 Motions		
	# of Seq.	Points	Frames	# of Seq.	Points	Frames
Traffic	31	241	30	7	332	31
Articulated	11	155	40	2	122	3
Checkerboard	78	291	28	26	437	28
All	120	266	30	35	398	29

5.3 Parameter Sensitivity

In Fig. 5, we show how the error rates change with various parameter values. Each figure shows the effects of one parameter in x-coordinate while the others are fixed to the values given in Sections 5.1 and 5.2. It is seen that the error rates can be kept at a low level with a range of values for α, β and γ. Also, from Fig. 5, we can see that $SCLA_1$ and $SCLA_{2,1}$ can achieve good performance using similar parameter values.

5.4 Computational Time Comparison

The average computational time as a function of the number of objects for some algorithms are shown in Fig. 6. We compare the computational time of SCLA with SSC and LRR, because SSC has lower error rates than the other methods (worse than SCLA) as shown above, and LRR (also LRSC) is the fastest algorithm as shown in [13, 36]. For a fair of comparison, the time recorded is the computational time to run the complete procedure for all these methods. Experiments in this section are conducted on a 4-core Intel Xeon E3-1240 V2 3.40 GHz Linux Server with 8 GB memory. It is seen from Fig. 6 that both $SCLA_1$ and $SCLA_{2,1}$

Table 4: Clustering error rates (%) on Hopkins-155 data set

Method	LSA	SCC	LRR	LRR-H	LRSR	SSC	$SCLA_1$	$SCLA_{2,1}$
2 Motions								
Mean	4.23	2.89	4.10	2.13	3.69	1.52	**1.29**	1.30
Median	0.56	**0.00**	0.22	**0.00**	0.29	**0.00**	**0.00**	**0.00**
3 Motions								
Mean	7.02	8.25	9.89	4.03	7.69	4.40	2.69	**2.67**
Median	1.45	0.24	6.22	1.43	3.80	0.56	0.21	**0.19**
All								
Mean	4.86	4.10	5.41	2.56	4.59	2.18	**1.61**	**1.61**
Median	0.89	**0.00**	0.53	**0.00**	0.60	**0.00**	**0.00**	**0.00**

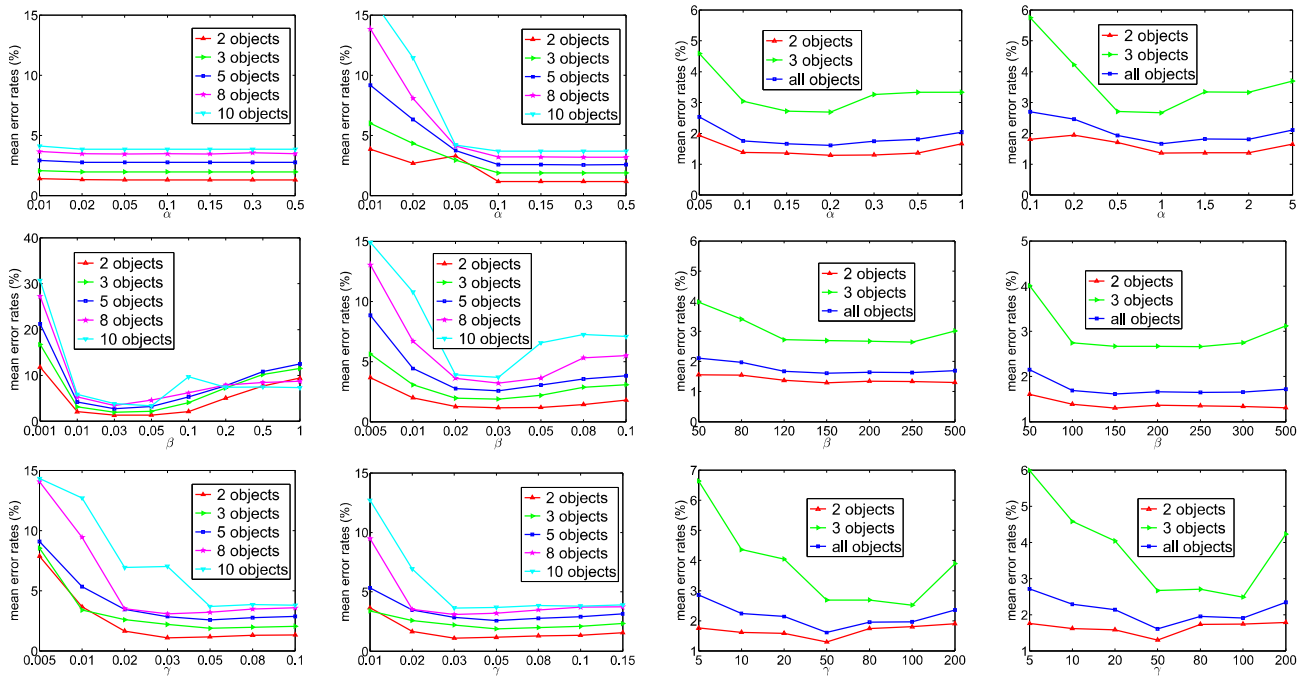

Figure 5: Plots of average error rates v.s. the change of parameters for face clustering and motion segmentation. From the left to the right column panels are SCLA_1 on face clustering, $\text{SCLA}_{2,1}$ on face clustering, SCLA_1 on motion segmentation and $\text{SCLA}_{2,1}$ on motion segmentation, respectively. From top to bottom rows are the figures of error rates v.s. α, β and γ, respectively. The parameter in x-axis changes while the other parameters are fixed.

Table 5: Wilcoxon Signed Rank Test of SCLA with Other Methods

Method	(LSA,SCLA)	(SCC,SCLA)	(LRR,SCLA)	(LRR-H,SCLA)	(LRSR,SCLA)	(SSC,SCLA)
Face	0.0020	0.0020	0.0020	0.0039	0.0020	0.0195
All	4.4E-4	1.2E-4	4.4E-4	2.4E-4	4.4E-4	0.0034

For SCLA, the best performance is used among SCLA_1 and $\text{SCLA}_{2,1}$.

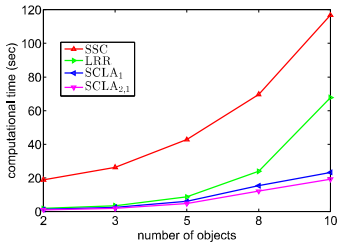

Figure 6: Average computational time (second) of SCLA_1, $\text{SCLA}_{2,1}$, SSC and LRR as a function of the number of objects.

are faster than SSC and LRR. Since LRR (also LRSC) and SSC are the fastest three algorithms among all the methods that we compare SCLA with[13, 36], it is evident to believe SCLA have satisfying speed.

6. CONCLUSION

We propose a subspace clustering method based on a non-convex log-determinant approximation to the rank function. We apply the method of augmented Lagrangian multipliers to the optimization of associated non-convex objective function, which admits closed-form solutions to all subproblems in each iteration. The convergence of the proposed optimization algorithm to a stationary point is mathematically proved. We conduct experiments on standard benchmark data sets and achieve promising results. Empirical results demonstrate the effectiveness and efficiency in both accuracy and speed. Though empirically the proposed algorithm converges fast, theoretical rate of convergence needs to be derived, which is a future line of our research.

Acknowledgment

This work is supported by National Science Foundation under grant IIS-1218712. The authors would like to thank all the reviewers for their comments.

References

[1] P. K. Agarwal and N. H. Mustafa. k-means projective clustering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 155–165. ACM, 2004.

[2] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] D. P. Bertsekas. Nonlinear programming. 1999.

[5] T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 179–186. IEEE, 1991.

[6] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[7] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[8] G. Chen and G. Lerman. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330, 2009.

[9] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[10] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[11] 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.

[12] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1926–1929. IEEE, 2010.

[13] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2765–2781, 2013.

[14] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1801–1807. IEEE, 2011.

[15] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, 2002.

[16] C. W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.

[17] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):643–660, 2001.

[18] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[19] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–707. IEEE, 2004.

[20] J. Ho, M.-H. Yang, J. Lim, K.-C. 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.

[21] J. Huang, F. Nie, H. Huang, and C. Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):11, 2014.

[22] K. Kanatani. Motion segmentation by subspace separation and model selection. *image*, 1:1, 2001.

[23] K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):684–698, 2005.

[24] A. S. Lewis and H. S. Sendov. Nonsmooth analysis of singular values. part i: Theory. *Set-Valued Analysis*, 13(3):213–241, 2005.

[25] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, 2013.

[26] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 663–670, 2010.

[27] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1615–1622. IEEE, 2011.

[28] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1546–1562, 2007.

[29] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM review*, 50(3):413–458, 2008.

[30] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1832–1845, 2010.

[31] M. Soltanolkotabi, E. J. Candes, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

[32] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[33] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[34] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

[35] R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, March 2011.

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

[37] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–621. IEEE, 2003.

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

[39] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[40] 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.

[41] 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.