Research Article

LogDet Rank Minimization with Application to Subspace Clustering

Zhao Kang,[1] Chong Peng,[1] Jie Cheng,[2] and Qiang Cheng[1]

[1]*Computer Science Department, Southern Illinois University, Carbondale, IL 62901, USA*
[2]*Department of Computer Science and Engineering, University of Hawaii at Hilo, Hilo, HI 96720, USA*

Correspondence should be addressed to Qiang Cheng; qcheng@siu.edu

Low-rank matrix is desired in many machine learning and computer vision problems. Most of the recent studies use the nuclear norm as a convex surrogate of the rank operator. However, all singular values are simply added together by the nuclear norm, and thus the rank may not be well approximated in practical problems. In this paper, we propose using a log-determinant (LogDet) function as a smooth and closer, though nonconvex, approximation to rank for obtaining a low-rank representation in subspace clustering. Augmented Lagrange multipliers strategy is applied to iteratively optimize the LogDet-based nonconvex objective function on potentially large-scale data. By making use of the angular information of principal directions of the resultant low-rank representation, an affinity graph matrix is constructed for spectral clustering. Experimental results on motion segmentation and face clustering data demonstrate that the proposed method often outperforms state-of-the-art subspace clustering algorithms.

1. Introduction

Matrix rank minimizing [1] is ubiquitous in machine learning, computer vision, control, signal processing, and system identification. For instance, low-rank representation based subspace clustering [2–4] and matrix completion [5, 6] methods have achieved great success recently. Subspace clustering [7] is one of the fundamental topics with numerous applications, for example, image representation [8, 9], face clustering [3, 10], and motion segmentation [11, 12]. It is assumed that high-dimensional data is more likely a union of low-dimensional subspaces rather than one individual subspace. For example, different subspaces are needed to describe trajectories of different moving objects in a video sequence. Subspace clustering is an intrinsically difficult problem, since we need to simultaneously cluster all data points into multiple groups and find a low-dimensional subspace fitting each group of points.

Subspace clustering has been an active research topic over the past decades. Four main categories of methods are proposed [10]: iterative, algebraic, statistical, and spectral clustering-based methods. The first three kinds of approaches are sensitive to initialization, noise, and outliers; in addition,

they are difficult to optimize [10]. Spectral clustering-based methods have achieved promising performance, where the key is to learn a good affinity matrix of data points. For instance, the algorithms of local subspace affinity (LSA) [13], locally linear manifold clustering (LLMC) [14], and spectral local best-fit flats (SLBF) [15] use local information around each point to construct the affinity matrix, while spectral curvature clustering (SCC) [16] method preserves the global structures of the whole dataset in deriving the affinity matrix. Subsequently, K-means [17] or Normalized Cuts (NCuts) [18, 19] are applied to the affinity matrix to obtain clustering results.

Recently, some spectral clustering-based methods, such as sparse representation (SSC) [10] and low-rank representation (LRR) [3], have been proposed to obtain state-of-the-art results in subspace clustering. SSC represents each data point as a sparse linear combination of the other points and solves an l_1-norm regularized minimization problem for sparsity. SSC shows promising results if the subspaces are either independent or disjoint [20].

The basic idea of LRR is to learn a low-rank representation of data by capturing the global Euclidean structure of the whole data. In this scheme, each data point is represented

as a linear combination of the examples in the data matrix itself, and a convex nuclear norm minimization is used as a surrogate of the rank function to obtain the desired low-rank representation. Though its optimization is well studied and has a global optimum, its performance may be far from optimal in real applications because the nuclear norm might not be a good approximation to the rank function. Compared to the rank function to which all nonzero singular values have equal contributions, the nuclear norm treats those values differently by simply adding them together. As a result, the nuclear norm may be dominated by a few very large singular values and significantly deviated from the true value of the rank. Several papers have considered this problem of using the nuclear norm and designed methods to alleviate it by either thresholding or removing some of the singular values; for instance, singular value thresholding [21] and truncated nuclear norm [6] both considerably enhance the performance of matrix completion.

In this paper, we propose using a log-determinant (LogDet) function for rank approximation and study its minimization in subspace clustering. Different from the nuclear norm-based approaches which minimize the summation of all singular values, our approach aims to minimize the rank by making the contribution to be much closer to one from a big singular value, while being zero from a small singular value. In this way, we can get closer and more robust approximation to the rank function than the nuclear norm. Since the LogDet function is nonconvex, we apply the method of augmented Lagrange multipliers (ALM) to solve the associated optimization for potentially large-scale applications, in which the subproblem for minimizing the LogDet function in each iteration has a closed-form solution. To demonstrate the effectiveness of our LogDet minimization method, we apply it to subspace clustering. By employing a rather simple formulation based on the LogDet function, we obtain a low-rank representation for subspace clustering. Subsequently, we exploit the angular information of principal directions of such a representation to further enhance the separation ability of the affinity matrix. In summary, our main contributions of this work include the following.

(i) More accurate and robust rank approximation is used to obtain the low-rank representation, which is able to capture the global structure of the dataset.

(ii) An iterative optimization algorithm is designed for minimizing this rank approximation-based objective function. Theoretical analysis shows that our algorithm converges to a stationary point. Specifically, the proposed optimization method is applied to subspace clustering.

(iii) Angular information of principal directions of the low-rank representation is employed to further exploit the intrinsic local geometrical structure relevant to the membership of data points.

(iv) Extensive experiments demonstrate the effectiveness of the proposed LogDet minimization method for

rank approximation. Particularly, when used for subspace clustering, our simple formulation shows favorable performance compared to other state-of-the-art methods, although we do not explicitly account for outliers in our model. This demonstrates the robustness of our approach.

The remainder of the paper is organized as follows: Section 2 provides a brief review of LRR and SSC. In Section 3, we present the proposed approximation and design an efficient optimization scheme. We give convergence analysis in Section 4. Experimental results are shown in Section 5. Finally, conclusions are drawn in Section 6.

2. Review of LRR and SCC

In this section, we give a brief review of SSC and LRR.

Let $X = [x_1, x_2, \ldots, x_n] \in \mathcal{R}^{d \times n}$ be a set of d-dimensional data points drawn from an unknown union of k linear subspaces $S_1, S_2, \ldots, S_k$. The task of subspace clustering is to segment data points into k subspaces.

LRR tries to seek the lowest rank representation among many possible linear combinations of the bases in a given dictionary, which typically is the data matrix itself. The problem can be formulated as

$$\min_{Z} \quad \text{rank}(Z) \tag{1}$$
$$\text{s.t.} \quad X = XZ,$$

where $Z = [z_1, z_2, \ldots, z_n]$ is the coefficient matrix with each z_i being the representation of x_i. The above problem is NP-hard due to the combinatorial nature of the rank function.

The tightest convex relaxation of the rank function [22] is the nuclear norm. For a matrix $D \in \mathcal{R}^{m \times n}$, its nuclear norm is defined as $\|D\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(D)$, where $\sigma_i(D)$ means the ith singular value of D. Using this relaxation, LRR solves the following problem:

$$\min_{Z} \quad \|Z\|_* \tag{2}$$
$$\text{s.t.} \quad X = XZ.$$

After obtaining Z, the affinity matrix W is defined as

$$W = |Z| + \left|Z^T\right|. \tag{3}$$

Then the spectral clustering algorithm, Normalized Cuts [18], is used to produce the final segmentation.

SSC aims to find a sparse representation of X by solving the following convex optimization problem:

$$\min_{Z,E,S} \quad \|Z\|_1 + \frac{\alpha}{2} \|E\|_F^2 + \gamma \|S\|_1, \tag{4}$$
$$\text{s.t.} \quad X = XZ + E + S, \quad \text{diag}(Z) = 0,$$

where $\|S\|_1 = \sum_{ij} |S_{ij}|$, S is a sparse matrix containing the gross error, $\|E\|_F^2 = \sum_i \sum_j E_{ij}^2$, and E is a matrix of fitting residuals. After obtaining Z, subsequent procedures are similar to LRR.

3. LogDet Rank Approximation and Its Minimization Algorithm

A function $f : \mathcal{R}^n \rightarrow [-\infty, \infty]$ is absolutely symmetric if $f(x)$ is invariant under arbitrary permutations and sign changes of the elements of x. Based on this function $f(x)$, we have the following theorem [23].

Theorem 1. *Function $F : \mathcal{R}^{n_1 \times n_2} \rightarrow \mathcal{R}$ is unitarily invariant if $F(X) = f(\sigma(X)) = f \circ \sigma(X)$, where $X \in \mathcal{R}^{n_1 \times n_2}$, whose singular value decomposition is $X = U \operatorname{diag}(\{\sigma_i\}_{1 \le i \le n})V^T$, $\sigma(X) : \mathcal{R}^{n_1 \times n_2} \rightarrow \mathcal{R}^n$ are singular values of X, and $n = \min(n_1, n_2)$. And the gradient of $F(X)$ at X is*

$$\frac{\partial F(X)}{\partial X} = U \operatorname{diag}(\theta) V^T, \tag{5}$$

where $\theta = \partial f(y)/\partial y|_{y=\sigma(X)}$.

Equation (5) can be obtained directly from Theorem 3.1 of [23].

In this work, we utilize unitarily invariant function LogDet to achieve a closer, though not convex, rank relaxation than the nuclear norm. We apply the method of ALM for LogDet rank approximation associated minimization. To explain our method, we specifically consider using LogDet as a rank surrogate in subspace clustering. We first obtain a low-rank representation of high-dimensional data based on the LogDet optimization. Then we construct an affinity graph matrix for spectral clustering by using the angular information of principal directions of the low-rank representation.

3.1. LogDet Rank Minimization. We use $\operatorname{LogDet}(I + Z^T Z)$ as a surrogate of the rank function of Z. It is obvious that $\operatorname{LogDet}(I + Z^T Z) = \sum_{i=1}^n \log(1 + \sigma_i^2(Z))$. Because it can be easily verified that $\log(1 + \sigma_i^2(Z)) \le \sigma_i(Z)$ for any $\sigma_i(Z) \ge 0$, we always have $\operatorname{LogDet}(I + Z^T Z) \le \|Z\|_*$; particularly, if there are large nonzero singular values, the LogDet function will be much smaller than the nuclear norm since $\log(1 + \sigma_i^2(Z)) \ll \sigma_i(Z)$ for a large $\sigma_i(Z) > 1$. It is noted that, for small nonzero singular values, their contribution to the LogDet function will be significantly reduced compared to the nuclear norm. Because small nonzero singular values are often regarded as being from noise in the data, the LogDet function reduces noise effect more compared to the nuclear norm.

It is worthwhile to note that a similar function $\operatorname{LogDet}(X + \delta I)$ was proposed in [24] to approximate rank and iterative linearization was used to find a local minimum. However, δ is a very small constant (e.g., 10^{-6}), which leads to biased approximation for small singular values.

This LogDet function is differentiable with respect to the singular values by Theorem 1, and even though it is nonconvex, its minimization is rather simple by using our optimization method. To explain its minimization, we consider its specific application to subspace clustering. By employing the above LogDet function, we simply formulate the subspace clustering into the following unconstrained nonconvex minimization problem:

$$\min_Z \operatorname{LogDet}\left(I + Z^T Z\right) + \rho \|X - XZ\|_F^2, \tag{6}$$

where $I \in \mathcal{R}^{n \times n}$ is the identity matrix. The first term of (6) is to minimize the rank of Z, while the second is a relaxation of $X = XZ$, which is referred to as the self-expressiveness of X with Z representing the similarity between data points. Because the LogDet function is not convex in Z, we resort to ALM technique to solve (6), by rewriting (6) as follows:

$$\min_Z \quad \operatorname{LogDet}\left(I + Z^T Z\right) + \rho \|X - XW\|_F^2$$
$$\text{s.t.} \quad Z = W. \tag{7}$$

We turn to the minimizing of the following augmented Lagrangian function:

$$L(Y, Z, W, \beta) = \operatorname{LogDet}\left(I + Z^T Z\right) + \rho \|X - XW\|_F^2$$
$$+ \frac{\beta}{2} \|Z - W\|_F^2 + \operatorname{Tr}\left(Y^T (Z - W)\right), \tag{8}$$

where $\beta > 0$ is a penalty parameter and Y is the Lagrangian dual variable. With a sufficiently large β, the objective function converges to objective function in (6). This can be solved by updating Z, W, and Y alternatively while fixing the other variables. Specifically, assume that at the kth iteration we have obtained Z^k, W^k, and Y^k; then, for the $(k + 1)$th iteration, optimization problem (8) can be updated via the following four steps.

Step 1. Compute W^{k+1}. Fix Z^k and Y^k and then calculate W^{k+1}:

$$W^{k+1} = \arg \min_W \rho \|X - XW\|_F^2$$
$$+ \frac{\beta_k}{2} \left\| Z^k - \left(W - \frac{1}{\beta_k} Y^k\right) \right\|_F^2, \tag{9}$$

which has a closed-form solution:

$$W^{k+1} = \left(\beta_k I + 2\rho X^T X\right)^{-1} \left(2\rho X^T X + Y^k + \beta_k Z^k\right). \tag{10}$$

Step 2. Compute Z^{k+1}. Fix W^{k+1} and Y^k and minimize $L(Y^k, Z, W^{k+1}, \beta_k)$ as follows:

$$Z^{k+1} = \arg \min_Z L\left(Y^k, Z, W^{k+1}, \beta_k\right)$$
$$= \arg \min_Z \operatorname{LogDet}\left(I + Z^T Z\right) \tag{11}$$
$$+ \frac{\beta_k}{2} \left\| Z - \left(W^{k+1} - \frac{1}{\beta_k} Y^k\right) \right\|_F^2.$$

This can be converted to a scalar minimization problem due to the following theorem. As we notice, this can also be rewritten as a special case of the problem in a recent work [25].

Theorem 2. *For unitarily invariant function* $F(Z) = f \circ \sigma(Z)$, *assuming SVD of* $A \in \mathcal{R}^{m \times n}$ *is* $A = U \Sigma_A V^T$, $\Sigma_A = \text{diag}(\{\sigma_{i,A}\}_{i=1}^{\min(m,n)})$, *the optimal solution to the problem*

$$\min_Z F(Z) + \frac{\beta}{2} \|Z - A\|_F^2 \tag{12}$$

is $Z^* = U \Sigma_Z^* V^T$, *with* $\Sigma_Z^* = \text{diag}(\{\sigma_i^*\}_{i=1}^{\min(m,n)})$ *obtained by solving scalar minimization problems*

$$\sigma_i^* = \arg\min_{\sigma_i} f(\sigma_i) + \frac{\beta}{2}(\sigma_i - \sigma_{i,A})^2, \tag{13}$$

$$i = 1, \ldots, \min(m, n).$$

Proof. Let $A = U \Sigma_A V^T$ be SVD of A; then $\Sigma_A = U^T A V$. Denoting $X = U^T Z V$ which has exactly the same singular values as Z, that is, $\Sigma_X = \Sigma_Z$, we have

$$F(Z) + \frac{\beta}{2} \|Z - A\|_F^2 \tag{14}$$

$$= F(X) + \frac{\beta}{2} \|X - \Sigma_A\|_F^2 \tag{15}$$

$$= F(\Sigma_X) + \frac{\beta}{2} \|X - \Sigma_A\|_F^2 \tag{16}$$

$$= F(\Sigma_X) + \frac{\beta}{2} \left(\|X\|_F^2 + \|\Sigma_A\|_F^2 - 2 \langle X, \Sigma_A \rangle \right) \tag{17}$$

$$\geq F(\Sigma_X) + \frac{\beta}{2} \left(\|\Sigma_X\|_F^2 + \|\Sigma_A\|_F^2 - 2 \langle \Sigma_X, \Sigma_A \rangle \right) \tag{18}$$

$$= F(\Sigma_X) + \frac{\beta}{2} \|\Sigma_X - \Sigma_A\|_F^2 \tag{19}$$

$$= F(\Sigma_Z) + \frac{\beta}{2} \|\Sigma_Z - \Sigma_A\|_F^2 \tag{20}$$

$$= \sum_i \left[f(\sigma_i) + \frac{\beta}{2}(\sigma_i - \sigma_{i,A})^2 \right] \tag{21}$$

$$\geq \sum_i f(\sigma_i^*) + \frac{\beta}{2}(\sigma_i^* - \sigma_{i,A})^2. \tag{22}$$

In the above, (15) holds because the Frobenius norm is unitary invariant; (16) holds because $F(Z)$ is unitary invariant; (17) is true by von Neumann's inequality; and (20) holds as $\Sigma_X = \Sigma_Z$. The inequality between (15) and (19) can also be obtained by the Hoffman-Wielandt inequality. Therefore, (20) is a lower bound of (14), where Σ_Z^* is obtained by minimizing (20). Note that the equality in (18) is attained if $X = \Sigma_X$. Because $\Sigma_Z = \Sigma_X = X = U^T Z V$, the SVD of Z is $Z = U \Sigma_Z V^T$, which is the minimizer of problem (12). Hence the proof is completed. $\square$

The first-order optimality condition is that the gradient of (13) with respect to each singular value should vanish. Thus, for subproblem (11), we have

$$\frac{2\sigma_i}{1 + \sigma_i^2} + \beta_k \left(\sigma_i - \Sigma_i^k \right) = 0, \tag{23}$$

$$\text{s.t. } \sigma_i \geq 0, \text{ for } i = 1, \ldots, n,$$

where SVD of $W^{k+1} - (1/\beta_k) Y^k$ is $U \text{diag}(\{\Sigma_i^k\}_{i=1}^n) V^T$. The above equation is cubic and gives three roots. In addition, we need to enforce the nonnegativity of σ_i. It is easily seen that there exists at least one nonnegative root. And there is a unique minimizer $\sigma_i^* \in [0, \Sigma_i^k)$ if $\beta_k > 1/4$. Finally, we obtain the update of Z variable with $Z^{k+1} = U \text{diag}(\sigma_1^{k*}, \ldots, \sigma_n^*) V^T$.

Step 3. Compute Y^{k+1}. Fix Z^{k+1} and W^{k+1}, and then we calculate Y^{k+1} as follows:

$$Y^{k+1} = Y^k + \beta_k \left(Z^{k+1} - W^{k+1} \right). \tag{24}$$

Step 4. Update β_{k+1} as $\beta_{k+1} = \gamma \beta_k$. The complete procedure is summarized in Algorithm 1.

Problem (6) is nonconvex. It is difficult to give a rigorous mathematical argument for convergence to (local) optimum. We will provide a theoretical proof that our algorithm converges to an accumulation point and this accumulation point is a stationary point. Our empirical experiments confirm the convergence of the proposed method on the benchmark datasets. The experimental results are promising, despite the fact that the solution obtained by the proposed optimization method may be a local optimum.

3.2. Affinity Graph Matrix Construction. Now we will construct an affinity matrix W for subspace clustering. Optimal Z^* may not accurately describe the relationship between samples if the data is severely corrupted. Therefore, in general, it is not a good idea to construct W by directly using Z^*. In the spirit of [3, 12], we construct an affinity matrix in the following way.

Assuming the skinny SVD of Z^* is $U^* \Sigma^* (V^*)^T$, we define $M = U^* (\Sigma^*)^{1/2}$ and $N = (\Sigma^*)^{1/2} (V^*)^T$. Based on the weighted eigenvector matrix M or N, we construct an affinity matrix W as follows:

$$W_{ij} = \left(\frac{m_i^T m_j}{\|m_i\|_2 \|m_j\|_2} \right)^{2\alpha}$$

$$\text{or } W_{ij} = \left(\frac{n_i^T n_j}{\|n_i\|_2 \|n_j\|_2} \right)^{2\alpha}, \tag{25}$$

where m_i (n_i) and m_j (n_j) represent the ith and jth columns (rows) of M (N), respectively, and parameter $\alpha \in \mathcal{N}$ tunes the sharpness of the affinity between two points, with $\alpha > 1$ helping in separating the clusters. When α increases, while the between-cluster separability can be increased, the intra-cluster cohesiveness would nevertheless be degraded. Thus,

Input: data matrix X, parameters $\rho > 0$, $\gamma > 1$, and $\beta_0 > 0$.
Initialize: $Z = I \in \mathscr{R}^{n \times n}$, $Y = 0$.
Repeat
 (1) Update W as:
   $W^{k+1} = \left(\beta_k I + 2\rho X^T X \right)^{-1} \left(2\rho X^T X + Y^k + \beta_k Z^{k+1} \right)$.
 (2) Solve Z using (11) and (23).
 (3) Update the augmented multiplier Y and the augmented Lagrange multiplier β:
   $Y^{k+1} = Y^k + \beta_k \left(Z^{k+1} - W^{k+1} \right)$,
   $\beta_{k+1} = \gamma \beta_k$.
Until stopping criterion is satisfied.
Return $Z^* = Z^{k+1}$.

ALGORITHM 1: LogDet Rank Minimization.

Input: data matrix X, number of subspaces k, parameters $\rho > 0$, $\gamma > 1$, and $\beta_0 > 0$.
(1) Obtain Z^* from Algorithm 1.
(2) Compute the skinny SVD $Z^* = U^* \Sigma^* (V^*)^T$.
(3) Calculate $M = U^* (\Sigma^*)^{1/2}$ or $N = (\Sigma^*)^{1/2} (V^*)^T$.
(4) Construct the affinity graph matrix W by (25).
(5) Apply W to perform NCuts.

ALGORITHM 2: The SCLD Algorithm.

a suitable α needs to balance within-cluster cohesiveness and between-cluster separability. In this paper, we set α to be 2. Then we have the same postprocessing as LRR. (For LRR, we use (12) in [3] rather than (3) to construct W. We also confirmed with an author of [3] that the power 2 of (12) is a typo and it should be 4.) As U^* or V^* spans the principal directions of Z^*, we employ the angle information or powered correlation coefficients of the examples, because their lengths may be affected significantly by the noise or outliers in the data.

Now using the resultant affinity matrix, we can apply spectral clustering algorithm to do segmentation. In this paper, we simply perform NCuts [18] on W. The proposed subspace clustering procedure is summarized in Algorithm 2.

4. Convergence Analysis

In this section, we give the convergence analysis for Algorithm 1. We will show that our optimization algorithm attains at least one stationary point of problem (7). We first rewrite the objective function of (7) as

$$G(Z, W) = F(Z) + \rho \|X - XW\|_F^2 \quad \text{s.t. } Z = W, \quad (26)$$

$$H(Z, W, Y) = G(Z, W) + \langle Z - W, Y \rangle, \quad (27)$$

$$L(Z, W, Y, \beta) = H(Z, W, Y) + \frac{\beta}{2} \|Z - W\|_F^2. \quad (28)$$

Lemma 3. *The sequence $\{Y_k\}$ is bounded.*

Proof. To minimize Z at step $k + 1$, the optimal Z_{k+1} needs to satisfy the first-order optimality condition

$$\begin{aligned} & \nabla_Z L\left(Z, W_{k+1}, Y_k, \beta_k\right)\big|_{Z_{k+1}} \\ & = \nabla_Z F(Z)\big|_{Z_{k+1}} + \beta_k \left(Z_{k+1} + \frac{1}{\beta_k} Y_k - W_{k+1} \right) = 0. \end{aligned} \quad (29)$$

Note that the updating rule for Y is

$$Y_{k+1} = Y_k + \beta_k \left(Z_{k+1} - W_{k+1} \right); \quad (30)$$

thus $\nabla_Z F(Z)\big|_{Z_{k+1}} + Y_{k+1} = 0$. We know from (5) that

$$\nabla_Z F(Z)\big|_{Z_{k+1}} = U \, \mathrm{diag}\left(\frac{2\sigma_1}{1 + \sigma_1^2}, \dots, \frac{2\sigma_n}{1 + \sigma_n^2} \right) V^T, \quad (31)$$

and $0 \le 2\sigma_i/(1 + \sigma_i^2) \le 1$, so $\nabla_Z F(Z)\big|_{Z_{k+1}}$ is bounded. Then it is seen that Y_{k+1}; that is, $\{Y_k\}$ is bounded. $\square$

Lemma 4. *$\{W_k\}$ and $\{Z_k\}$ are bounded if $\sum(\beta_{k+1}/\beta_k^2) < \infty$ and $\sum(1/\beta_k) < \infty$.*

Proof. Consider

$$\begin{aligned} L\left(Z_k, W_k, Y_k, \beta_k\right) &= L\left(Z_k, W_k, Y_{k-1}, \beta_{k-1}\right) \\ & \quad + \frac{\beta_k - \beta_{k-1}}{2} \|Z_k - W_k\|_F^2 \\ & \quad + \mathrm{Tr}\left(\left(Y_k - Y_{k-1}\right) \left(Z_k - W_k\right) \right) \\ &= L\left(Z_k, W_k, Y_{k-1}, \beta_{k-1}\right) \\ & \quad + \frac{\beta_k + \beta_{k-1}}{2\beta_{k-1}^2} \|Y_k - Y_{k-1}\|_F^2. \end{aligned} \quad (32)$$

Thus,

$$L\left(Z_{k+1}, W_{k+1}, Y_k, \beta_k\right)$$

$$\leq L\left(Z_k, W_{k+1}, Y_k, \beta_k\right)$$

$$\leq L\left(Z_k, W_k, Y_k, \beta_k\right)$$

$$\leq L\left(Z_k, W_k, Y_{k-1}, \beta_{k-1}\right) + \frac{\beta_k + \beta_{k-1}}{2\beta_{k-1}^2} \left\| Y_k - Y_{k-1} \right\|_F^2 \quad (33)$$

$$\leq \cdots$$

$$\leq L\left(Z_1, W_1, Y_0, \beta_0\right) + \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\beta_{i-1}^2} \left\| Y_i - Y_{i-1} \right\|_F^2 .$$

Since the second term in the above inequality is finite, $L(Z_{k+1}, W_{k+1}, Y_k, \beta_k)$ is bounded. We can rewrite $L(Z_{k+1}, W_{k+1}, Y_k, \beta_k)$ as

$$L\left(Z_{k+1}, W_{k+1}, Y_k, \beta_k\right) + \frac{1}{2\beta_k} \left\| Y_k \right\|_F^2$$

$$= F\left(Z_{k+1}\right) + \rho \left\| X - XW_{k+1} \right\|_F^2 \quad (34)$$

$$+ \frac{\beta_k}{2} \left\| Z_{k+1} - W_{k+1} + \frac{1}{\beta_k} Y_k \right\|_F^2 .$$

Because $L(Z_{k+1}, W_{k+1}, Y_k, \beta_k)$ and $(1/\beta_k)\|Y_k\|_F^2$ are bounded and each term on the right hand side of (34) is nonnegative, each term will be bounded. $F(Z_{k+1}) = \sum_i \log(1 + \sigma_i^2(Z_{k+1}))$ being bounded implies that all singular values of Z_{k+1} are bounded and Z_{k+1} is bounded. Since $(1/\beta_k)(Y_{k+1} - Y_k) = Z_{k+1} - W_{k+1}$, clearly we have bounded W_k. Therefore $\{W_k\}$ and $\{Z_k\}$ are bounded. □

Theorem 5. $\{Y_k, W_k, Z_k\}$ *has at least one accumulation point* $\{Y^*, W^*, Z^*\}$, *and* $\{W^*, Z^*\}$ *is a stationary point of optimization problem (7) with the assumption that* $\lim_{k \to \infty} \beta_{k-1}(Z_k - Z_{k-1}) \to 0$.

Proof. $\{Y_k, W_k, Z_k\}$ is a bounded sequence; hence, by the Bolzano-Weierstrass theorem, there must be at least one accumulation point, which is denoted by $\{Y^*, W^*, Z^*\}$. Without loss of generality, we assume that $\{Y_k, W_k, Z_k\}$ itself converges to $\{Y^*, W^*, Z^*\}$. Next, we prove that this accumulation point is a stationary point of problem (26). As $Y_k = Y_{k-1} + \beta_{k-1}(Z_k - W_k)$, we have $Z_k - W_k = (1/\beta_{k-1})(Y_k - Y_{k-1})$. Because $\beta_{k-1} \to \infty$ and $\{Y_k\}$ is bounded, we get $Z_k - W_k \to 0$; that is, $Z^* = W^*$. By first-order optimality condition and the definition of Z_k, we have $\nabla_Z F(Z)|_{Z_k} + Y_{k-1} + \beta_{k-1}(Z_k - W_k) = \nabla_Z F(Z)|_{Z_k} + Y_k = 0$. Letting $k \to \infty$, we get $\nabla_Z F(Z)|_{Z^*} + Y^* = 0$. At the kth step, W_k satisfies $(2\rho X^T X + \beta_{k-1} I)W_k = 2\rho X^T X + \beta_{k-1} Z_{k-1} + Y_{k-1}$; that is, $2\rho X^T X(W_k - I) = \beta_{k-1} Z_{k-1} - \beta_{k-1} W_k + Y_{k-1} = \beta_{k-1}(Z_k - W_k) + \beta_{k-1}(Z_{k-1} - Z_k) + Y_{k-1} = \beta_{k-1}(Z_{k-1} - Z_k) + Y_k$. With the assumption that $\beta_{k-1}(Z_k - Z_{k-1}) \to 0$ [26], we get $2\rho X^T X(W^* - I) = Y^*$.

Now we can see that $\{Y^*, W^*, Z^*\}$ satisfies the KKT conditions of $L(W, Z, Y)$ and thus $\{W^*, Z^*\}$ is a stationary point of (7). □
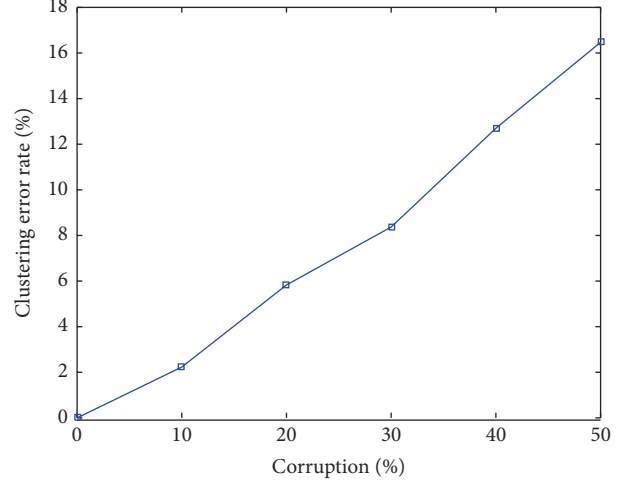

FIGURE 1: The clustering error rate with different percentage of corruption on synthetic data. The parameter ρ is tuned to obtain the best performance.

5. Experiments and Analysis

In this section, we conduct experiments on the subspace clustering task with both synthetic and real data.

5.1. Experiments with Synthetic Data. We construct 5 independent subspaces whose bases $\{U_i\}_{i=1}^5$ are generated by a random rotation matrix R through $U_{i+1} = RU_i$, $1 \leq i \leq 4$, where $U_1 \in \mathcal{R}^{100 \times 4}$ is a random orthogonal matrix [2]. We sample 20 data vectors from each subspace by $X_j = U_j T_j$, $1 \leq j \leq 5$, where T_j is a 4×20 i.i.d. $\mathcal{N}(0, 1)$ matrix. Some data vectors are randomly chosen to corrupt; for example, for a data vector x, it is corrupted by adding Gaussian noise with zero mean and variance $0.2\|x\|$. We then use SCLD to segment the data into 5 clusters. Subspace clustering error rate defined as # of misclassified points/total # of points is used to assess the performance. We report the clustering error rate (averaged from 30 trials) with different corruption levels in Figure 1. Without any corruption, SCLD can cluster all data points correctly.

5.2. Experiments with Real Data. In this section, we evaluate the effectiveness and robustness of SCLD on benchmark datasets, Extended Yale B (EYaleB) [27, 28] and Hopkins 155 [29]. We compare the proposed method SCLD with several state-of-the-art subspace clustering algorithms: LRR [3], SSC [10], LRSC [4, 30], and local subspace affinity (LSA) [13]. For these methods, we use the parameters given by the respective authors. For our method, we also tune ρ to obtain the best performance. Generally, ρ should be relatively large if the data are slightly corrupted. β and γ have little influence on the clustering results, so we just set $\beta_0 = 0.3$ to ensure the uniqueness of minimizer and use $\gamma = 1.1$ empirically. Other parameters are shown in Table 1. The experiments are conducted on Windows 7 with 16 GM memory and Intel Core i5-2300 CPU.

TABLE 1: Parameter settings of different algorithms.

| Method | Face clustering | | Motion segmentation |
	Scenario 1	Scenario 2	
LRR	$\lambda = 0.18$		$\lambda = 4$
LSA	$K = 3, d = 5$		$K = 8, d = 5$
SSC	$\lambda_e = \dfrac{8}{\mu_e}$	$\lambda_e = \dfrac{20}{\mu_e}$	$\lambda_z = \dfrac{800}{\mu_z}$
LRSC	$\tau = 0.4, \alpha = 0.045$	$\tau = 0.045, \alpha = 0.045$	$\tau = 420, \alpha = 3000,$ or $\alpha = 5000$
SCLD	$\rho = 0.08$	$\rho = 0.03$	$\rho = 55$

FIGURE 2: Sample images from the Extended Yale B Database.

TABLE 2: Clustering error rate on the first 10 classes of EYaleB.

Method	LRR	SSC	LSA	LRSC	SCLD
Error rate (%)	20.94	35	59.52	35.78	**3.59**

5.2.1. Face Clustering. Face clustering is to cluster a set of face images from multiple individuals in a hope to reveal the identity of these individuals. EYaleB Database includes 2414 frontal images of 38 individuals. For each individual, the images are taken under 64 lighting conditions and can be described by a low-dimensional subspace [31]. The images are resized to 48×42 pixels and each vectorized image is regarded as a data point. Figure 2 shows some example images from the database.

(1) First Experiment Scenario. As done in [2], we test the algorithms on the first 10 classes of EYaleB, which consists of 640 frontal face images. More than half of the images are corrupted by shadow and noise. We use this heavily corrupted data to test the effectiveness of our method. As shown in Table 2, SCLD significantly enhances the performance. Specifically, it improves the clustering accuracy by at least 17% when compared to the other algorithms. Since the only difference between our approach and LRR is rank approximation, this improvement is due to LogDet.

(2) Second Experiment Scenario. For a fair comparison, we have followed the experimental setup of [10]. We divide the 38 subjects into four groups: subjects 1 to 10, 11 to 20, 21 to 30, and 31 to 38. We consider all choices of $n \in \{2, 3, 5, 8, 10\}$ subjects for the first three groups. For the last group, we consider all choices of $n \in \{2, 3, 5, 8\}$. We implement our subspace clustering algorithm on each set of n subjects. For all experiments, the stopping criterion for Z is triggered by a relative difference of 10^{-5} between two successive iterations or by a maximum of 100 iterations.

The results are presented in Table 3. For other methods, we cited the results from Table 5 of [10]. SCLD consistently has low clustering error rates and is more stable than the other methods whose error rates increase drastically as the number

TABLE 3: Clustering error rates (%) on EYaleB.

Method	LRR	SSC	LSA	LRSC	SCLD
2 subjects					
Mean	2.54	**1.86**	32.80	5.32	2.79
Median	0.78	**0.00**	47.66	4.69	0.78
3 subjects					
Mean	4.21	**3.10**	52.29	8.47	3.72
Median	2.60	**1.04**	50.00	7.81	1.56
5 subjects					
Mean	6.90	**4.31**	58.02	12.24	4.83
Median	5.63	**2.50**	56.87	11.25	**2.50**
8 subjects					
Mean	14.34	5.85	59.19	23.72	**5.45**
Median	10.06	4.49	58.59	28.03	**3.52**
10 subjects					
Mean	22.92	10.94	60.42	30.36	**6.25**
Median	23.59	5.63	57.50	28.75	**4.84**

of subjects increases to 8 and 10. As shown in Figure 2, there are many sparse within-sample outliers in the face images, for example, shadows. Although LRR uses a regularization term to count for corruptions, the regularization term does not appear to be well suited to EYaleB. LSA has inferior performance possibly because it does not explicitly exploit the low-rank structure of the data.

(3) Third Experiment Scenario. In this section, we compare SCLD with other algorithms with RPCA [32] as a preprocessing step. In practice, we do not know the clustering of the data beforehand and hence we apply RPCA to the collection of all data points for each trial prior to clustering. As shown in Table 4, SCLD is still superior to other methods though they apply RPCA to deal with sparse outlying entries. Compared to Table 3, only the clustering error rates of LRSC reduced in some cases. We can conclude that applying RPCA to all data points simultaneously is not effective in improving clustering performance. This is due to the fact that RPCA seeks a common low-rank subspace, which will decrease

FIGURE 3: Example frames from four video sequences of the Hopkins 155 Dataset with traced feature points.

TABLE 4: Clustering error rates (%) on EYaleB after applying RPCA simultaneously to all the data in each trial.

Method	LRR	SSC	LSA	LRSC	SCLD
2 subjects					
Mean	5.72	**2.09**	32.53	5.67	2.79
Median	3.91	**0.78**	47.66	4.69	**0.78**
3 subjects					
Mean	10.01	3.77	53.02	8.72	**3.72**
Median	9.38	2.60	51.04	8.33	**1.56**
5 subjects					
Mean	15.33	6.79	58.76	10.99	**4.83**
Median	15.94	5.31	56.87	10.94	**2.50**
8 subjects					
Mean	28.67	10.28	62.32	16.14	**5.45**
Median	31.05	9.57	62.50	14.65	**3.52**
10 subjects					
Mean	32.55	11.46	62.40	21.82	**6.25**
Median	30.00	11.09	62.50	25.00	**4.84**

TABLE 5: Segmentation error rate (%) on the Hopkins 155 Dataset (155 sequences).

Method	LRR	SSC	LSA	LRSC	SCLD
2 motions					
Mean	2.13	1.52	4.23	3.69	**1.31**
Median	**0.00**	**0.00**	0.56	0.29	**0.00**
3 motions					
Mean	4.03	4.40	7.02	7.69	**3.43**
Median	1.43	**0.56**	1.45	3.80	**0.56**
All					
Mean	2.56	2.18	4.86	4.59	**1.79**
Median	**0.00**	**0.00**	0.89	0.60	**0.00**
Time (sec)	1.30	1.04	3.40	**0.16**	1.49

the principal angles between subspaces and decrease the distance between data points in different subjects [10].

5.2.2. Motion Segmentation. Motion segmentation is to segment the trajectories associated with n different moving objects into different groups according to their motions in a video sequence. Because different motions can be treated as different subspaces, we use the Hopkins 155 Dataset to validate SCLD. This dataset is slightly corrupted as shown in Figure 3. It consists of 155 sequences of two or three motions and 1 sequence of 5 motions; the latter is regarded as outlier. Each sequence is regarded as a separate clustering problem.

The experimental results are reported in Table 5. We also used the results in Table 1 of [10]. It can be seen that SCLD produces superior results compared to the other methods. For all 155 sequences, the error rate is as low as 1.79%. If we use all 156 sequences, the overall error rate of our proposed algorithm will be 1.87%. We report the average computation time for every sequence at the bottom of Table 5. The computational cost of LRSC is much lower than the other methods, while LRR, SSC, and SCLD are comparable.

To testify the influence of parameter ρ in our algorithm, we show the clustering error rates of SCLD for different ρ over all 155 sequences in Figure 4. As we can see, when ρ was between 1 and 200, the clustering error varied between 1.79%

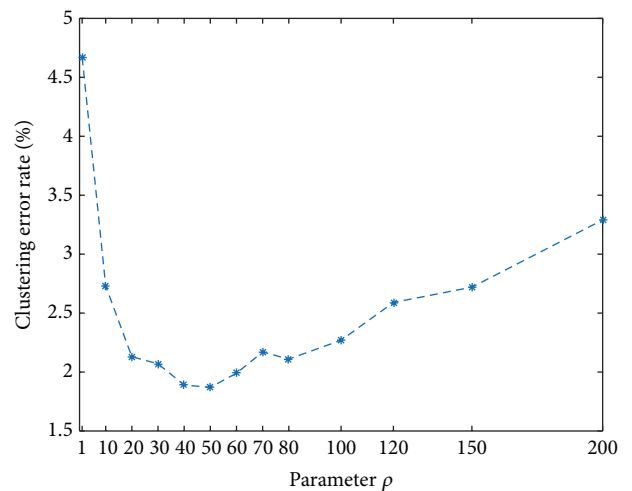
FIGURE 4: Changes in clustering error rate when varying ρ.

and 4.67%. This implies that SCLD performs well under a wide range of values of ρ.

To test the dependence of SCLD on initialization, we apply another two different initializations. First, we use the solutions from LRR as initial guess for SCLD. Second, we just generate some random numbers. We find that we can still get the same results. Actually, it is recommended to use convex relaxation solutions as initialization for nonconvex formulations [33, 34].

6. Conclusion

In this paper we propose using a log-determinant function (LogDet) as a rank approximation to recover the low-rank representation of high-dimensional data. When applied to subspace clustering, the proposed algorithm, called SCLD, exploits both global and local structures of the data through the LogDet rank approximation and angle-based affinity matrix. Consequently, it captures more intrinsic information of the data that benefits subspace clustering. Our extensive experimental results show that it outperforms other low-rank representation algorithms based on the nuclear norm. Therefore LogDet appears to be an effective rank approximation function well suited to subspace clustering applications. Although our model is simple and with no explicit modeling of outliers, it is resilient to various corruptions. Our future research will consider modeling corruptions explicitly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
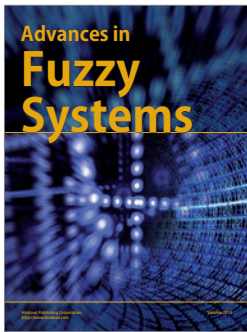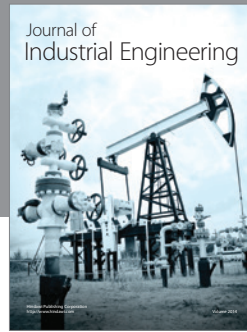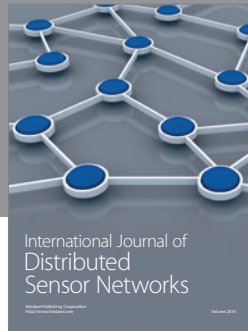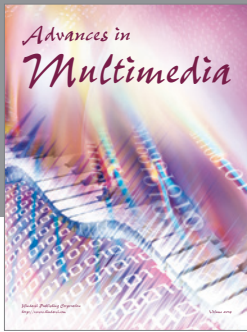
Acknowledgment

References

[1] M. Fazel, *Matrix rank minimization with applications [Ph.D. thesis]*, Stanford University, Stanford, Calif, USA, 2002.

[2] 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)*, pp. 663–670, 2010.

[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[4] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, no. 1, pp. 47–61, 2014.

[5] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[6] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2117–2130, 2013.

[7] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[8] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[9] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[10] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[11] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1832–1845, 2010.

[12] F. Lauer and C. Schnörr, "Spectral clustering of linear subspaces for motion segmentation," in *Proceedings of the 12th International Conference on Computer Vision, (ICCV '09)*, pp. 678–685, IEEE, October 2009.

[13] 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*, vol. 3954 of *Lecture Notes in Computer Science*, pp. 94–106, Springer, Berlin, Germany, 2006.

[14] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–6, IEEE, Minneapolis, Minn, USA, June 2007.

[15] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International Journal of Computer Vision*, vol. 100, no. 3, pp. 217–240, 2012.

[16] G. Chen and G. Lerman, "Spectral Curvature Clustering (SCC)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[17] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026–1041, 2007.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[19] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[20] E. Elhamifar and R. Vidal, "Clustering disjoint subspaces via sparse representation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (CASSP '10)*, pp. 1926–1929, March 2010.

[21] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[22] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[23] A. S. Lewis, "The convex analysis of unitarily invariant matrix functions," *Journal of Convex Analysis*, vol. 2, no. 1-2, pp. 173–183, 1995.

[24] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proceedings of the American Control Conference*, vol. 3, pp. 2156–2162, IEEE, Denver, Colo, USA, June 2003.

[25] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *Proceedings of the 29th Conference on Artificial Intelligence (AAAI '15)*, Austin, Tex, USA, January 2015.

[26] F. Nie, Y. Huang, X. Wang, and H. Huang, "New primal SVM solver with linear computational cost for big data classifications," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, Beijing, China, June 2014.

[27] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[28] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[29] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[30] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1801–1807, June 2011.

[31] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[32] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, article 11, 2011.

[33] J. Fan, L. Xue, and H. Zou, "Strong oracle optimality of folded concave penalized estimation," *The Annals of Statistics*, vol. 42, no. 3, pp. 819–849, 2014.

[34] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *Journal of Machine Learning Research*, vol. 11, pp. 1081–1107, 2010.