Journal of Applied Analysis and Computation
Volume 7, Number 2, May 2017, 600–616
Website:http://jaac-online.com/
DOI:10.11948/2017037

A NEW MODEL FOR SPARSE AND LOW-RANK MATRIX DECOMPOSITION*

Zisheng Liu[1], Jicheng Li[1,†], Guo Li[2,†], Jianchao Bai[1] and Xuenian Liu[1]

Abstract The robust principal component analysis (RPCA) model is a popular method for solving problems with the nuclear norm and ℓ_1 norm. However, it is time-consuming since in general one has to use the singular value decomposition in each iteration. In this paper, we introduce a novel model to reformulate the existed model by making use of low-rank matrix factorization to surrogate the nuclear norm for the sparse and low-rank decomposition problem. In such case we apply the Penalty Function Method (PFM) and Augmented Lagrangian Multipliers Method (ALMM) to solve this new nonconvex optimization problem. Theoretically, corresponding to our methods, the convergence analysis is given respectively. Compared with classical RPCA, some practical numerical examples are simulated to show that our methods are much better than RPCA.

Keywords Robust principal component analysis, sparse matrix, low-rank matrix, nuclear norm, matrix decomposition.

MSC(2010) 15A23, 65K05, 90C22.

1. Introduction

1.1. Motivation

Suppose there is a matrix which is formed by the addition of an unknown sparse matrix and a low rank matrix. Our aim is to decompose it into two parts which contain sparse components and low-rank components. Such problem is widely applied in the engineer fields including model selection, system identification, image and computer vision [8, 23], bioinformatics, background modeling and face recognition [26], latent semantic indexing [11, 21], machine learning [1–3] and control [20] etc.. Data matrix generated from these applications may have a high number of dimensions, while the vast majority of the data may have the similar (even same) structure. As we know that the main information of the data matrix lies in low-dimensional subspace or low-dimensional manifolds, and sometimes these key messages will be covered or interfered by sparse components. Therefore, it is necessary and significant to remove

[†] the corresponding author. Email address:jcli@mail.xjtu.edu.cn (J. Li), guoli@szu.edu.cn (G. Li)
[1] School of Mathematics and Statistics, Xi'an Jiaotong University, 710049, Xi'an, China
[2] College of Mathematics and Statistics, Shenzhen University, 518060, Shenzhen, China
[*] The authors were supported by the National Natural Science Foundation of China (11171270), the Natural Science Foundation of Shaanxi Province, China (2014JM1029) and the Fundamental Research Funds for the Central Universities (Xi'an Jiaotong University, Grant No. xkjc2014008).

sparse components and find a low-rank structure matrix.

1.1.1. Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) [12, 16] is a common tool for high-dimensional data processing and analysis, it has a wide range of applications in science and engineering fields [16]. Suppose we have a high-dimensional data which lies near a much lower-dimensional subspace, the main purpose of the PCA is to efficiently and accurately estimate this low-dimensional subspace. We assume that the observed data matrix $D \in \mathbb{R}^{m\times n}$ is decomposed as

$$D = A + E, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ is the low-rank component of D and $E \in \mathbb{R}^{m\times n}$ is the sparse component of D. The classic PCA can seek the best rank-r estimation of A by solving the following constrained optimization

$$\begin{aligned} &\min_{E} \ \|E\|_F \\ &s.t.\ D = A + E, \\ &\qquad Rank(A) \le r, \end{aligned} \tag{1.2}$$

where $r \ll \min(m, n)$ is the target dimension of the subspace and $\|\cdot\|_F$ is the Frobenius norm. The PCA computes singular value decomposition (SVD) of D and then projects the columns of D onto the subspace spanned by the r principal left singular vectors of D. In practical applications, PCA can perform well if the noise magnitude is not large.

1.1.2. Robust Principal Component Analysis (RPCA)

Despite its many advantages, the traditional PCA suffers from the fact that the estimation $\hat{A}$ can be arbitrarily far from the true A, when E is sufficiently sparse (relative to the rank of A). We hope to recover both A and E accurately and efficiently. According to the structure of the low-rank matrix and the property of the sparse component, J. Wright et al. [26] have shown that one can exactly recover the data matrix D from (1) by solving the following convex optimization problem, as long as the matrix E is sufficiently sparse,

$$\begin{aligned} &\min_{A,E} \ \|A\|_* + \lambda\|E\|_1 \\ &s.t.\ D = A + E, \end{aligned} \tag{1.3}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix (i.e., the sum of its singular values), $\|\cdot\|_1$ denotes the ℓ_1 norm of a matrix (i.e., the sum of its absolute values), and λ is a positive weighting parameter that provides a trade-off between the sparse and low-rank components. From the above expression, the goal of (1.3) is to approximate a given matrix by minimizing the addition of two matrices under the nuclear norm and ℓ_1 norm. Due to the ability to exactly recover underlying low-rank structure and sparse component in the data set, this optimization is referred to as the Robust PCA (RPCA) in [26], even the sparse component have arbitrarily large magnitude. Several applications of the RPCA have been demonstrated in [4, 5, 18, 26, 27].

Although the RPCA performs very well, it usually costs much time to calculate the singular value decomposition (SVD) in every iteration when the given data matrix is big. To overcome this shortcoming, in this paper we introduce a novel model named Sparse amd Low-Rank Factorization (SLRF) by using the low-rank factorization of a matrix. Before continuing, we provide here a brief summary of the notations used throughout the paper.

1.2. Notations

For a matrix $A \in \mathbb{R}^{m\times n}$, let $\|A\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|A_{ij}|$ denotes the ℓ_1 norm, $\|A\|_2$ and $\|A\|_*$ denote the spectral norm and the nuclear norm (i.e., the sum of its singular values), respectively. From [6,22] we know that the spectral and nuclear norms are dual from one another. We consider the singular value decomposition (SVD) of a matrix A of rank r

$$A = USV',\ S = diag(\{\sigma_i\}),\ 1 \le i \le r,$$

where U and V are $m\times r$ and $n\times r$ matrices with orthonormal columns, respectively, and the singular values σ_i are positive. We always assume that the SVD of a matrix is given in the reduced form above. Furthermore, $\langle A, B\rangle = \mathrm{trace}(A'B)$ denotes the standard inner product, then the Frobenius norm is

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{tr(A'A)} = \Big(\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2\Big)^{\frac{1}{2}} = \Big(\sum_{i=1}^{r}\sigma_i^2\Big)^{\frac{1}{2}}.$$

1.3. Contributions and Organizations

Lots of iterative algorithms to solve the nuclear minimization problems suffer from high computation cost of singular value decompositions (SVDs) at each iteration. The purpose of this paper is to develop an efficient sparse and low-rank decomposition method, i.e., SLRF. This method offers much enhanced scalability in solving large-scale matrix decomposition problems and speeds up effectively on some difficult applications.

The main contributions and organizations of this work are as follows. In Section 2, we introduce the idea of low-rank matrix factorization into the original RPCA model to drastically reduce the computing time of SVDs of the involved iterations, and propose a new model for the estimation of sparse and low-rank decomposition. Theoretically, we prove that the problem (2.1) is completely equivalent to the problem (1.3). In Section 3, we design two iterative schemes to solve the proposed model. In Section 4, we show some convergence results about the methods presented in Sections 3. In Section 5, some empirical results on synthetic data and real-world data are reported to demonstrate the efficiency and convergence behavior of our new methods. Finally, we conclude the paper and give a short discussion.

2. Problem Formulations

2.1. Sparse and Low-Rank Factorization (SLRF)

We now concentrate on the new model that works with sparse and low-rank factorization (SLRF) of D. The given observation matrix has the form $D = A + E$, where

$A \in \mathbb{R}^{m\times n}$ is a low-rank matrix of rank r and $E \in \mathbb{R}^{m\times n}$ is the sparse component of D. To tackle the low rank property, a popular approach is to utilize the rank factorization of a matrix, that is, $A = LR'$ where L and R are $m \times r$ and $n \times r$ matrices, respectively. Consequently, we can solve the following optimization problem to find its solution

$$\begin{aligned}&\min_{L,R',E}\ \frac{1}{2}(\|L\|_F^2 + \|R'\|_F^2) + \lambda\|E\|_1\\ &s.t.\ D = LR' + E.\end{aligned} \tag{2.1}$$

The main difference between RPCA and SLRF is that in order to characterize the rank constraint, we utilize the method of low-rank matrix factorization to surrogate the nuclear norm for sparse and low-rank decomposition. The biggest advantage of our method is that the model (2.1) avoids the computational burden of the SVD for a large scale matrix, which can greatly save the computing time. Another advantage of this formulation is that it can substantially decrease one of decision variables from mn to $(m + n)r$. Therefore, this model requires less storage space and computation time than the previous methods. This optimization also can be rewritten as a semidefinite program (SDP) [24].

2.2. Semidefinite Program (SDP) Formulation

Let $Z = U\Sigma V'$ be a singular value decomposition of an $m \times n$ matrix Z, where U is an $m \times r$ matrix, V is an $n \times r$ matrix, Σ is an $r \times r$ diagonal matrix, and r is the rank of Z. We recall the fact that the spectral and nuclear norms are dual from one another.

Proposition 2.1 ([22]). *The dual norm of the operator norm $\|\cdot\|_2$ in $R^{m\times n}$ is the nuclear norm $\|\cdot\|_*$.*

The proof can be found in [22]. Therefore, by the Proposition 2.1, for a $m \times n$ matrix Z, we have

$$\|Z\|_* := \max\{tr(Z'Y) \mid \|Y\|_2 \le 1\}. \tag{2.2}$$

From the characterization in (2.2), the optimization problem

$$\begin{aligned}&\max\ tr(Z'Y)\\ &s.t.\ \|Y\|_2 \le 1,\end{aligned}$$

can be formulated as a simple semidefinite program:

$$\begin{aligned}&\max\ tr(Z'Y)\\ &s.t.\ \begin{bmatrix} I_m & Y \\ Y' & I_n \end{bmatrix} \succeq 0.\end{aligned} \tag{2.3}$$

From duality, the nuclear norm has the following SDP characterization:

$$\begin{aligned}&\min_{W_1,W_2}\ \frac{1}{2}(tr(W_1) + tr(W_2))\\ &s.t.\ \begin{bmatrix} W_1 & Z \\ Z' & W_2 \end{bmatrix} \succeq 0.\end{aligned}$$

If we set $W_1 := U\Sigma U'$ and $W_2 := V\Sigma V'$, then, the triple (W_1, W_2, Z) is feasible for (2.3) since

$$\begin{bmatrix} W_1 & Z \\ Z' & W_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Sigma \begin{bmatrix} U \\ V \end{bmatrix}' \succeq 0.$$

Furthermore, we have $tr(W_1) = tr(W_2) = tr(\Sigma)$, and thus the objective function satisfies $\frac{1}{2}[tr(W_1) + tr(W_2)] = tr(\Sigma) = \|Z\|_*$.

Putting these facts together, (1.3) can be reformulated as

$$\begin{aligned} \min_{A,E,W_1,W_2,T} \ & \frac{1}{2}[tr(W_1) + tr(W_2)] + \lambda \mathbf{1}_m^T T \mathbf{1}_n \\ s.t. \ & \begin{bmatrix} W_1 & A \\ A' & W_2 \end{bmatrix} \succeq 0, \\ & -T_{ij} \le E_{ij} \le T_{ij} \ \ \forall\ (i,j), \\ & D = A + E, \end{aligned} \tag{2.4}$$

where T is defined as

$$\begin{cases} T_{ij} = \text{sign}(E_{ij}), & E_{ij} \neq 0 \\ T_{ij} \in [-1, 1], & E_{ij} = 0 \end{cases}.$$

Here, $\mathbf{1}_m$ refers to the vector that has 1 in every entry.

2.3. The Equivalence Between (1.3) and (2.1)

In [22], B. Recht et al. have proved that (2.1) is equivalent to (1.3) without considering sparse matrices E. As the spectral norm and nuclear norm are dual norms, utilizing the Semidefinite Program (SDP), in the following theorem, we prove that the problem (2.1) is completely equivalent to the problem (1.3).

Theorem 2.1. *The minimum nuclear norm relaxation (1.3) is equivalent to the non-convex quadratic optimization problem (2.1).*

Proof. For the sparse matrix E, by the Lagrange function of (1.3) and (2.1), we have

$$\begin{aligned} \mathcal{L}_{RPCA}(A, E, \mu) &= \|A\|_* + \lambda\|E\|_1 + \frac{\mu}{2}\|D - A - E\|_F^2, \\ \mathcal{L}_{SLRF}(L, R', E, \mu) &= \frac{1}{2}(\|L\|_F^2 + \|R'\|_F^2) + \lambda\|E\|_1 + \frac{\mu}{2}\|D - LR' - E\|_F^2, \end{aligned}$$

respectively. If the variables A, L, R' are fixed, then we have that

$$\begin{cases} E_{RPCA} = \arg\min_E \ \mathcal{L}_{RPCA}(A, E, \mu), \\ E_{SLRF} = \arg\min_E \ \mathcal{L}_{SLRF}(L, R', E, \mu), \end{cases}$$

which is equivalent to

$$\begin{cases} E_{RPCA} = \arg\min_E \ \frac{\lambda}{\mu}\|E\|_1 + \frac{1}{2}\|E - (D - A)\|_F^2, \\ E_{SLRF} = \arg\min_E \ \frac{\lambda}{\mu}\|E\|_1 + \frac{1}{2}\|E - (D - LR')\|_F^2. \end{cases}$$

By [5], this implies

$$\begin{cases} E_{RPCA} = \mathcal{S}_{\frac{\lambda}{\mu}}(D - A), \\ E_{SLRF} = \mathcal{S}_{\frac{\lambda}{\mu}}(D - LR'), \end{cases}$$

where the soft thresholding (shrinkage) operator $\mathcal{S}$ is defined as $\mathcal{S}_{\frac{\lambda}{\mu}}(x) =: \max\{|x| - \frac{\lambda}{\mu}, 0\}$, $x \in \mathbb{R}$.

Let the triple (L, R', E) be a feasible solution of (2.1). If we define $W_1 := LL'$, $W_2 := RR'$, and $A := LR'$ be the feasible solution of the primal SDP problem (2.4), then they can reach the same cost of (2.1) and (2.4). Since the SDP formulation is equivalent to the nuclear norm problem, we have that the optimal value of (2.1) is always greater than or equal to the nuclear norm heuristic.

Conversely, from the SVD decomposition $A = U\Sigma V'$ of the optimal solution of the nuclear norm relaxation (1.3), we can explicitly construct matrices $L := U\Sigma^{\frac{1}{2}}$ and $R := V\Sigma^{\frac{1}{2}}$ for (2.1) that yield exactly the same value of the objective. The proof is completed. □

However, the formulation (2.1) is non-convex and is thus potentially subject to local minima that are not globally optimal. This non-convexity does not pose as much of a problem as it could, since we present two methods that are guaranteed to converge to a local minimum for a suitable selected r.

3. Methods for SLRF Problem

Since D is the superposition of a low-rank component A and a sparse component E, some constraint conditions on A and E were proposed in [7] to ensure sufficiently that the unique solution $(\hat{A}, \hat{E})$ of (2.1) is accurate under the trade-off parameter λ, if one wants to reconstruct the original matrix D, or recover each component individually. Therefore, studying and searching efficient algorithms to recover the sparse component and low-rank component of D becomes rather important.

In this section, we introduce two efficient methods to solve the SLRF model proposed in section 2. One is the Penalty Function Method (PFM), and the other is the Augmented Lagrangian Multipliers Method (ALMM). Furthermore, the efficiency and convergence behavior of the proposed methods are validated in the experiment part.

In order to describe the optimality conditions for the norm minimization problem (2.1), we must first characterize the set of all subgradients of the ℓ_1 norm.

Definition 3.1. Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a proper convex function. We say that a vector $d \in \mathbb{R}^n$ is a subgradient of f at a point $x \in \mathbb{R}^n$ if

$$f(z) \geq f(x) + \langle d, z - x \rangle, \ \forall z \in \mathbb{R}^n.$$

The set of all subgradients of f at x is called the subdifferential of f at x and is denoted by $\partial f(x)$.

We recall that any subgradient of ℓ_1 norm is defined as

$$\partial \|A\|_1 = \{sign(A) + W, \ W \text{and } A \text{ have disjoint support and } \|W\|_\infty \leq 1\}.$$

For comparison, the subdifferential of the nuclear norm at X is given by (see [25])

$$\partial \|A\|_* = \{UV' + Q : \ Q \in \mathbb{R}^{m \times n}, \ U'Q = 0, \ QV = 0, \ \|Q\|_2 \leq 1\}.$$

3.1. Penalty Function Method

We first present a Penalty Function Method (PFM) for solving (2.1). The Lagrangian function of (2.1) is defined as

$$\mathcal{L}(L, R', E, \mu) = \frac{1}{2}(\|L\|_F^2 + \|R'\|_F^2) + \lambda\|E\|_1 + \frac{\mu}{2}\|D - LR' - E\|_F^2, \tag{3.1}$$

where $\lambda > 0$ is a constant providing a trade-off between the sparse and low-rank components and $\mu > 0$ is the penalty parameter. By the optimality conditions, we have

$$\begin{cases} \mathbf{0} = L - \mu(D - LR' - E)R, \\ \mathbf{0} = R - \mu(D - LR' - E)'L, \\ \mathbf{0} \in [\lambda\partial(\|E\|_1) - \mu(D - LR' - E)]. \end{cases}$$

As a consequence, we obtain

$$\begin{cases} L = \mu(D - E)R(I_r + \mu R'R)^{-1}, \\ R = \mu(D - E)'L(I_r + \mu L'L)^{-1}, \\ E = \arg\min \ \frac{\lambda}{\mu}\|E\|_1 + \frac{1}{2}\|E - (D - LR')\|_F^2 \\ \quad = \mathcal{S}_{\frac{\lambda}{\mu}}(D - LR'), \end{cases} \tag{3.2}$$

where $\mathcal{S}_{\frac{\lambda}{\mu}}(x) =: \max\{|x| - \frac{\lambda}{\mu}, 0\}$, $x \in \mathbb{R}$. More details of PFM iterative strategy can be found in the following Algorithm 1.

Algorithm 1 Penalty Function Method (PFM)

```
Task: Approximate the solution of (2.1).
Input: Observation matrix $D = A + E$, weights $\lambda$ and penalty parameter $\mu$, rank $r$.
Initialize: $D = U\Sigma V'$, $L_0 := U\Sigma^{\frac{1}{2}}$, $R_0 := V\Sigma^{\frac{1}{2}}$, $E_0 = \mathbf{0}$.
  while the termination criterion is not met, do
    $L_k = \mu(D - E_{k-1})R_{k-1}(I_r + \mu R'_{k-1}R_{k-1})^{-1}$;
    $R_k = \mu(D - E_{k-1})'L_k(I_r + \mu L'_k L_k)^{-1}$;
    $A_k = L_k R'_k$;
    $E_k = \mathcal{S}_{\frac{\lambda}{\mu}}(D - L_k R'_k)$;
  end while
Output: $A \leftarrow A_k$, $E \leftarrow E_k$.
```

3.2. Augmented Lagrangian Multipliers Method

In fact, for the beneficial structure of the well-known Augmented Lagrangian Multipliers method (ALMM), we can utilize it to solve the SLRF model which has a high-level separable structure. As the objective function is non-convex, we next show how to extend the classical analysis of ALMM to such a new objective function.

One may define the Augmented Lagrangian function of (2.1):

$$\begin{aligned}\mathcal{L}(L,R',E,Y,\mu)=&\frac{1}{2}(\|L\|_F^2+\|R'\|_F^2)+\lambda\|E\|_1-\langle Y,D-LR'-E\rangle\\&+\frac{\mu}{2}(\|D-LR'-E\|_F^2,\end{aligned}\tag{3.3}$$

where $\lambda>0$ is a constant providing a trade-off between the sparse and low-rank components and $\mu>0$ is the penalty parameter, and $Y\in\mathbb{R}^{m\times n}$ is the Lagrange multiplier corresponding to the constraint $D-LR'-E=0$. It is well-known that the classic augmented Lagrangian method solves

$$\min_{L,R',E}\ \mathcal{L}(L,R',E,Y,\mu).\tag{3.4}$$

Since it is difficult to obtain the optimal solutions L, R' and E simultaneously from (3.4), based on the idea of the classic alternating direction method for convex optimization problems [14,15], one can minimize the augmented Lagrangian function to solve each block variable at a time by fixing the other two blocks and then update the Lagrange multiplier. Then starting with $Y_0=\mathbf{0}\in\mathbb{R}^{m\times n}$, our method inductively defined as the following framework:

$$\begin{aligned}L_k&=\arg\min_{L\in\mathbb{R}^{m\times r}}\ \mathcal{L}(L,R'_{k-1},E_{k-1},Y_{k-1},\mu),\\R_k&=\arg\min_{R'\in\mathbb{R}^{r\times m}}\ \mathcal{L}(L_k,R',E_{k-1},Y_{k-1},\mu),\\E_k&=\arg\min_{E\in\mathbb{R}^{m\times n}}\ \mathcal{L}(L_k,R'_k,E,Y_{k-1},\mu),\\Y_k&=Y_{k-1}-\delta_k(D-L_kR'_k-E_k),\end{aligned}$$

where $\delta>0$ is a step-length parameter.

Similarly to PFM, by the optimality condition

$$\nabla\mathcal{L}(L,R',E,Y,\mu)=\mathbf{0},\tag{3.5}$$

we have

$$\begin{cases}\mathbf{0}=L-\mu(D-LR'-E)R+YR,\\\mathbf{0}=R-\mu(D-LR'-E)'L+Y'L,\\\mathbf{0}\in[\lambda\partial(\|E\|_1)-\mu(D-LR'-E)+Y].\end{cases}\tag{3.6}$$

As a consequence, we have

$$\begin{cases}L=\mu(D-E-Y)R(I_r+\mu R'R)^{-1},\\R=\mu(D-E-Y)'L(I_r+\mu L'L)^{-1},\\E=\arg\min\ \frac{\lambda}{\mu}\|E\|_1+\frac{1}{2}\|E-(D-LR'-\frac{Y}{\mu})\|_F^2\\\quad=\mathcal{S}_{\frac{\lambda}{\mu}}(D-LR'-\frac{Y}{\mu}).\end{cases}\tag{3.7}$$

Summarizing the above description, for solving (2.1), the ALMM approach is described in Algorithms 2.

Remark 3.1. It is worth noting that although the augmented Lagrangian function (3.3) is non-convex in the pair (L,R',E), it is convex with respect to either L, R' or E while fixing the other. This structure separable property allows the ADM [9,13] scheme to be well defined.

Algorithm 2 Augmented Lagrangian Multipliers Method (ALMM)

```
**Task:** Approximate the solution of (2.1).
**Input:** Observation matrix $D = A + E$, weights $\lambda$ and penalty parameter $\mu$, step size $\delta$, rank $r$.
**Initialize:** $D = U\Sigma V'$, $L_0 := U\Sigma^{\frac{1}{2}}$, $R_0 := V\Sigma^{\frac{1}{2}}$, $E_0 = \mathbf{0}$, $Y_0 = \mathbf{0}$.
  **while** the termination criterion is not met, **do**
    $L_k = \mu(D - E_{k-1} - Y_{k-1})R_{k-1}(I_r + \mu R'_{k-1}R_{k-1})^{-1}$;
    $R_k = \mu(D - E_{k-1} - Y_{k-1})'L_k(I_r + \mu L'_k L_k)^{-1}$;
    $A_k = L_k R'_k$;
    $E_k = \mathcal{S}_{\frac{\lambda}{\mu}}(D - A_k - \frac{Y_{k-1}}{\mu})$;
    $Y_k = Y_{k-1} - \delta_k(D - A_k - E_k)$;
  **end while**
**Output:** $A \leftarrow A_k$, $E \leftarrow E_k$.
```

4. Convergence Analysis

For PFM, we have the following theorem about the convergence of the penalty item $\|D - LR' - E\|_F^2$ in (2.1).

Theorem 4.1. *The penalty item sequence $\|D - LR' - E\|_F^2$ produced by alternative optimizations (3.2) converge to a local minimum.*

Proof. Let the penalty item value $\|D - LR' - E\|_F^2$ after solving the three optimizations (3.2) be P_k^1, P_k^2 and P_k^3, respectively, in the k-th iteration. On the one hand, we have

$$\begin{cases} P_k^1 = \|D - L_k R'_{k-1} - E_{k-1}\|_F^2, \\ P_k^2 = \|D - L_k R'_k - E_{k-1}\|_F^2, \\ P_k^3 = \|D - L_k R'_k - E_k\|_F^2. \end{cases} \tag{4.1}$$

The local optimality of L_k, R_k and E_k yields $P_k^1 \geq P_k^2 \geq P_k^3$. On the other hand,

$$\begin{cases} P_{k+1}^1 = \|D - L_{k+1} R'_k - E_k\|_F^2, \\ P_{k+1}^2 = \|D - L_{k+1} R'_{k+1} - E_k\|_F^2, \\ P_{k+1}^3 = \|D - L_{k+1} R'_{k+1} - E_{k+1}\|_F^2. \end{cases} \tag{4.2}$$

The local optimality of P_{k+1}^1 yields $P_k^3 \geq P_{k+1}^1$. Therefore, the penalty item values $\|D - LR' - E\|_F^2$ keep decreasing throughout PFM (6):

$$P_1^1 \geq P_1^2 \geq P_1^3 \geq P_2^1 \geq P_2^2 \geq P_2^3 \ldots S_k^1 \geq P_k^2 \geq P_k^3 \geq P_{k+1}^1 \ldots. \tag{4.3}$$

Since the objective of (2.1) is monotonically decreasing and the constraints are satisfied all the time, (3.2) produces a sequence of penalty item values that converge to a local minimum. The proof is completed. □

For ALMM, our convergence theorem requires the boundedness of some sequences, which results from the following theorem.

Theorem 4.2 ([18]). *Let $\mathcal{H}$ be a Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle$ and a corresponding norm $\| \cdot \|$, and $y \in \partial\|x\|$, where $\partial f(x)$ is the subgradient of*

$f(x)$. Then $\|y\|^* = 1$ if $x \neq 0$, and $\|y\|^* \leq 1$ if $x = 0$, where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$.

With Theorem 4.2, we can prove the following lemma.

Lemma 4.1 ([18]). *The Lagrange multiplier sequence $\{Y_k\}$ produced by ALMM is bounded, where $Y_k = Y_{k-1} - \delta_k(D - L_k R'_k - E_k)$.*

For convenience, let $\mathcal{A}(Y) = \mu(D - LR' - E) - Y$, we have the following convergence theorem.

Theorem 4.3. *Suppose that (L_k, R_k, E_k, Y_k) is the optimal solution output by ALMM at each iteration. Assume that the sequence $\{Y_k\}$ is bounded. If (L_k, R_k, E_k) converges to $(\hat{L}, \hat{R}, \hat{E})$ and the linear map*

$$\Lambda_k(Y) = \begin{bmatrix} \mathcal{A}(Y)L_k \\ \mathcal{A}'(Y)R_k \\ \mathcal{A}(Y) \end{bmatrix} = \mathbf{0}$$

for all k, then there exists a matrix $\hat{Y}$ such that

$$\nabla\mathcal{L}(\hat{L}, \hat{R}, \hat{E}, \hat{Y}, \mu) = \mathbf{0}.$$

Proof. Since (L_k, R'_k, E_k) minimizes the ALMM at iteration k and $\mathcal{A}(Y_k) = \mu(D - L_k R'_k - E_k) - Y_k$, from (3.5) and (3.6) we have

$$\begin{cases} \nabla_L\mathcal{L} = \mathbf{0} \Rightarrow L_k - \mathcal{A}(Y_k)R_k = \mathbf{0}, \\ \nabla_R\mathcal{L} = \mathbf{0} \Rightarrow R_k - \mathcal{A}'(Y_k)L_k = \mathbf{0}, \\ \nabla_E\mathcal{L} = \mathbf{0} \Rightarrow \mathbf{0} \in [\lambda\partial(\|E_k\|_1) - \mathcal{A}(Y_k)], \end{cases} \tag{4.4}$$

which can be reformed as

$$\Lambda_k(Y_k) = \begin{bmatrix} L_k \\ R_k \\ Z_k \end{bmatrix},$$

where $Z = \frac{\lambda}{\mu}T + (D - LR' - Y)$.

$$\begin{cases} T_{ij} = \text{sign}[D - LR' - Y]_{ij}, & [D - LR' - Y]_{ij} \neq 0, \\ T_{ij} \in [-1, 1], & [D - LR' - Y]_{ij} = 0. \end{cases}$$

Since we have assumed that there is no nonzero Y with $\Lambda_k(Y) = \mathbf{0}$, there exists a left inverse and we can solve for Y_k:

$$Y_k = \Lambda_k^{\dagger} \begin{bmatrix} L_k \\ R_k \\ Z_k \end{bmatrix},$$

where

$$\Lambda_k(Y_k) = \begin{bmatrix} \mathcal{A}(Y_k)L_k \\ \mathcal{A}'(Y_k)R_k \\ \mathcal{A}(Y_k) \end{bmatrix}. \tag{4.5}$$

Based on assumptions that the sequence $\{Y_k\}$ is bounded and (L_k, R_k, E_k) converges to $(\hat{L}, \hat{R}, \hat{E})$, then we deduce that the right-hand side of (4.5) is bounded. Therefore, we must have that Y_k converge to some $\hat{Y}$. Taking the limit of (4.4) completes the proof. □

5. Experiments

In this section, some numerical examples demonstrate the performance and effectiveness of the proposed methods. Three cases will be discussed to illustrate the effectiveness of our methods. One is exact recovery of the sparse and low-rank matrix. The second one is to evaluate the recoverability of our methods for solving SLRF problem. The third one is an application of the sparse and low-rank decomposition problem. All experiments are performed under Windows 7 and MATLAB v7.8 (R2009a) running on a Lenovo desktop with an Intel Core(TM)i5-3470, CPU at 3.2 GHz and 4 GB of memory.

5.1. Exact Recovery of Sparse and Low-rank Matrix

Let $D = A + E$ be the available data, where A and E are, respectively, the original low-rank and sparse matrices that we wish to recover. For the SLRF model, we demonstrate the efficiency of the proposed algorithms on randomly generated matrices. Simply, we restrict our examples to square matrices. We draw A according to the independent random matrices and generate E satisfying the i.i.d.Gaussian distribution. The sparse component of the constructed E can have arbitrarily large magnitude. Specially, the rank of the matrix A and the sparse entries of the matrix E are selected to be $5\%m$ and $5\%m^2$, respectively. We apply our proposed methods PFM and ALMM on the matrix D to recover A and E. The parameters are set as $\lambda = \frac{10}{\sqrt{m}}$, $\mu = 0.5m$ and $\delta = 10^{-2}$. The relative errors are respectively denoted by

$$\begin{cases} \text{rel.err}(D) = \|D - A^k - E^k\|_F / \|D\|_F, \\ \text{rel.err}(A) = \|A - A^k\|_F / \|A\|_F, \\ \text{rel.err}(E) = \|E - E^k\|_F / \|E\|_F. \end{cases} \tag{5.1}$$

In our experiments, we compare RPCA (*`LRSD`: Low Rank and Sparse matrix Decomposition (see [27] for more details)) with SLRF (PFM, ALMM). A brief comparison of the two methods are presented in Tab.1. We set methods are terminated at relatively constant iteration steps. From the aspect of cost time, PFM has a small advantage. For example, PFM recovers a 800 × 800 matrix of rank 40 in less than 26 seconds and recovers a $3,000 \times 3,000$ matrix of rank 250 in just 418.26 seconds, while, ALMM needs a little more seconds than PFM. But compared with

*http://perception.csl.illinois.edu/matrix-rank/

the relative error, the ALMM is superior to the PFM. No matter which method is used to solve the SLRF problem, the relative error of D, A, E can achieve a very low error magnitude. In Tab.2, as the matrix dimension is up to $m = 1,000$, both of our methods for SLRF problem are at least 12 times faster than the `LRSD` for RPCA.

Table 1. The relative error and time cost comparison on synthetic data. Iterations=300.

Algs	m	Rank(A)	$\|E\|_0$	rel.err(D)	rel.err(A)	rel.err(E)	Time(s)
PFM	200	10	1981	5.25×10^{-4}	1.93×10^{-4}	7.95×10^{-3}	0.62
ALMM		10	1981	5.04×10^{-4}	1.70×10^{-4}	7.69×10^{-3}	0.75
PFM	400	20	7983	1.34×10^{-4}	4.91×10^{-5}	2.81×10^{-3}	6.39
ALMM		20	7984	1.29×10^{-4}	3.80×10^{-5}	2.71×10^{-3}	7.20
PFM	800	40	31944	3.31×10^{-5}	1.20×10^{-5}	9.90×10^{-4}	25.77
ALMM		40	31946	3.15×10^{-5}	6.97×10^{-6}	9.07×10^{-4}	26.63
PFM	1000	50	49818	2.12×10^{-5}	7.67×10^{-6}	7.05×10^{-4}	38.21
ALMM		40	49821	2.08×10^{-5}	6.95×10^{-6}	6.15×10^{-4}	40.39
PFM	2000	100	199803	5.28×10^{-5}	1.84×10^{-5}	2.50×10^{-3}	152.56
ALMM		100	199815	5.16×10^{-5}	1.46×10^{-5}	2.43×10^{-3}	161.16
PFM	3000	250	449838	1.88×10^{-5}	8.75×10^{-6}	1.47×10^{-3}	418.26
ALMM		250	449847	1.78×10^{-5}	5.36×10^{-6}	1.33×10^{-3}	425.79

Table 2. The relative error and time cost comparison on synthetic data. Iterations=80.

Algs	m	Rank(A)	$\|E\|_0$	rel.err(D)	rel.err(A)	rel.err(E)	Time(s)
PFM	200	10	1981	5.05×10^{-4}	1.84×10^{-4}	8.22×10^{-3}	0.37
ALMM		10	1981	4.83×10^{-4}	1.08×10^{-4}	7.69×10^{-3}	0.75
RPCA		10	1973	7.19×10^{-5}	2.61×10^{-5}	1.18×10^{-3}	0.57
PFM	400	20	7983	1.30×10^{-4}	4.72×10^{-5}	2.81×10^{-3}	1.77
ALMM		20	7980	1.37×10^{-4}	6.95×10^{-5}	2.35×10^{-3}	2.19
RPCA		10	7825	1.73×10^{-4}	6.61×10^{-5}	2.76×10^{-3}	11.60
PFM	800	40	31744	3.28×10^{-4}	1.15×10^{-4}	9.86×10^{-3}	5.54
ALMM		40	31746	3.18×10^{-4}	7.58×10^{-5}	9.39×10^{-3}	6.33
RPCA		40	30555	1.88×10^{-4}	3.28×10^{-4}	1.07×10^{-2}	59.89
PFM	1000	50	49718	2.10×10^{-4}	7.33×10^{-5}	7.04×10^{-3}	9.02
ALMM		50	49716	2.02×10^{-4}	4.45×10^{-5}	6.58×10^{-3}	9.29
RPCA		50	44255	3.02×10^{-4}	3.45×10^{-3}	1.09×10^{-1}	120.48

Fig.1 shows the comparison between SLRF and RPCA. There are two reasons for our acceleration:

a) Since the RPCA needs SVDs in each step, which is a high time-consuming operation, then the improvement of speed is due to that the SLRF is searching a low-rank matrix A by using rank factorization.

b) In the process of solving the Lagrangian function (3.1) and (3.3), we split them into three sub-problems and obtain (3.2) and (3.7), respectively. Using an alternate method to solve sub-problems can speed up effectively.

Therefore, in terms of running time, our methods for solving SLRF model outperform the method for RPCA model. In particular, existing subroutines for efficient SVD (e.g. [17, 19]) guarantees the efficiency of RPCA model for sparse and low-rank recovery problem.

Fig.2 and Fig.3 reflect the relative errors intuitively with different matrix size. To be specific, Fig.2 shows the result with the matrix of size 200×200. At the beginning of the iterations, RPCA has a poor performance, after about 30 iterations, we find that it achieves a better results than the SLRF, however, it is very clear that our

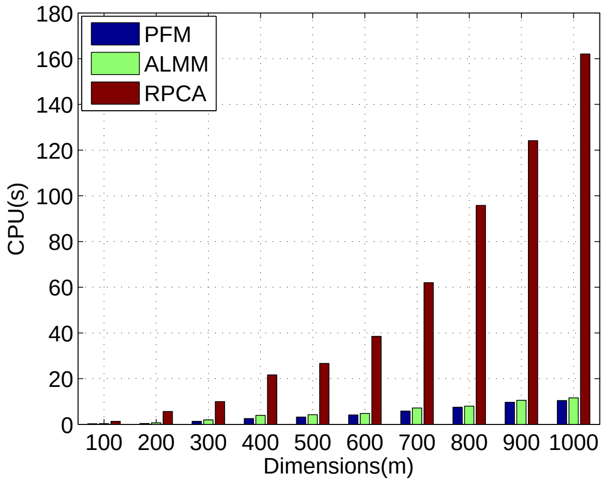


Figure 1. The recovery time for matrices of different size and different rank.

approach has a faster convergence rate than the RPCA. When the matrices size is up to $1,000 \times 1,000$, Fig.3 demonstrates that our methods perform better than RPCA, regardless of the relative error D or the relative errors of matrices A and E. As a summary, our approach in dealing with big data problems performs much better than the traditional RPCA method.

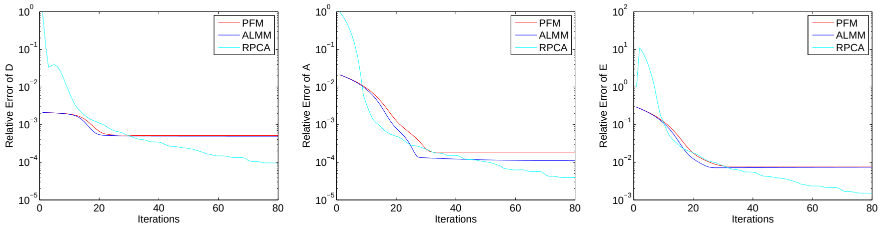


Figure 2. Relative error results of the sparse and low-rank decomposition tasks where the size of matrix is 200×200.

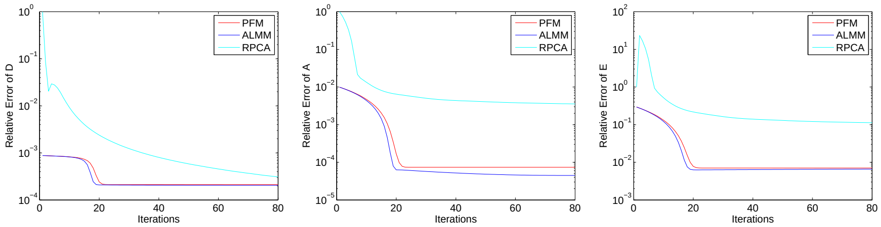


Figure 3. Relative error results of the sparse and low-rank decomposition tasks where the size of matrix is $1,000 \times 1,000$.

5.2. Evaluation of the Recoverability

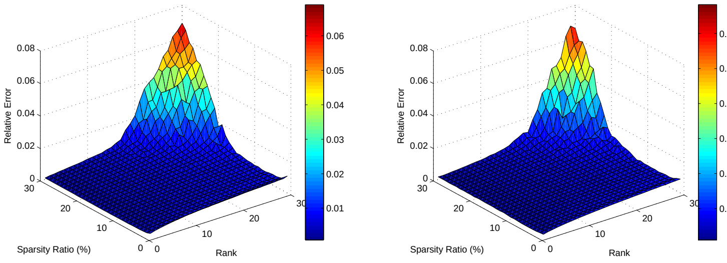
We list numerical results to illustrate the accuracy of our methods for SLRF. Specifically, we set $m = n = 100$ and test (r, spr) for which the decomposition problem roughly changes. Here the sparsity ratio is defined as

$$spr = \frac{\text{number-of-non-zero-entries}}{m^2} \times 100\%.$$

For each pair (r, spr), the maximal iteration is set to 300. The relative error is defined as

$$\text{Rel.Err} = \frac{\|(A^k, E^k) - (A, E)\|_F}{\|(A, E)\|_F + 1}.$$

The following results show the recoverability of the PFM and ALMM methods when r varies from 1 to 30 and spr varies from 1% to 30%.



(a) PFM

(b) ALMM

Figure 4. Recoverability results from varying rank and sparsity ratioby PFM(a) and ALMM(b).

For each choice of r and spr, by applying the proposed methods for SLRF model, we give a recoverability test. Fig.4 shows exact recoverability when either the sparsity ratio of E or the rank of A are properly small. Specifically, for Gaussian sparse matrices, the relative errors are as small as 10^{-3} for a pair $(10, 10\%)$. When $r \leq 5$, the PFM (Fig.4(a)) and ALMM (Fig.4(b)) methods result to faithful recoveries with spr as high as 20%, meanwhile, high accuracy recovery is attainable for r as high as 20 when $spr \leq 5\%$.

5.3. Background Modeling from Surveillance Video

Since background modeling [5, 10, 26] can reveal the correlation between video frames, then, an important application of sparse and low-rank decomposition is proposed to separate model background variations and foreground moving objects of the supervision video. If the individual frames are stacked as columns of a matrix D, then D can be expressed as the sum of a low-rank background matrix and a sparse error matrix representing the activity in the scene.

We apply ALMM to surveillance videos which consist of 200 frames of a scene. Then, the matrix D is composed of these frames with the resolution 144×176. We convert each frame as a vector and thus the matrix D is of size 25344×200. The decomposition results of one frame in each video sequence are shown in Fig.5. As is observed, the background and moving objects are precisely separated (the person in A of Fig.5 does not move throughout the video). Therefore, we find that our methods are very effective in separating the background from the activity.

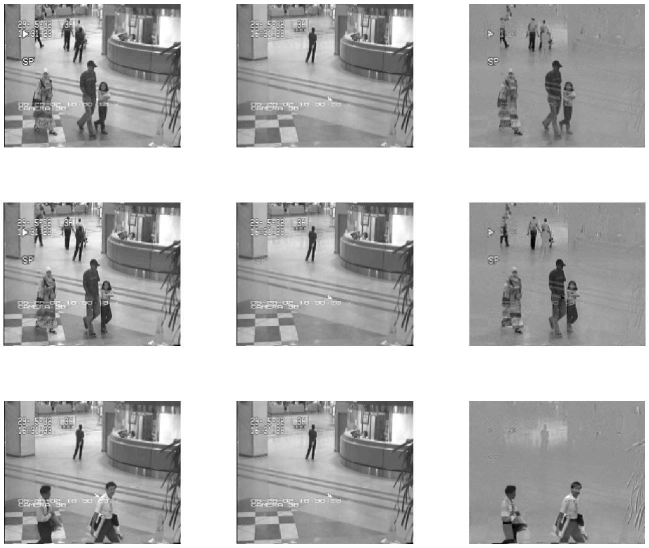


Figure 5. Background modeling results of the 200-frame surveillance video sequences in $D = A + E$ mode. (Left Column) Video sequence of a scene in an airport. (Middle Column) Static background recovered by ALMM. (Right Column) Sparse error recovered by ALMM represents activity in the frame.

6. Conclusion and Future Development

RPCA can be viewed as the combination of the convex relaxation of a sparse problem and a rank minimization problem. The main disadvantages of the optimization problem (1.3) is to cost much time on the calculation of SVDs for large matrices. In order to overcome this shortcoming, we propose a new strategy by introducing rank factorization into RPCA framework. Then in this paper, a novel model for matrices decomposition problem named as Sparse and Low-Rank Factorization (SLRF) is presented, which is fast to be solved and surprisingly effective in terms of computation cost and storage requirement. Furthermore, we also develop two efficient iterative schemes to solve the problem (2.1). The efficiency and convergence behavior of the proposed methods are validated through the comprehensive numerical experimental results on synthetic data and real-world data.

Our results in this paper address only the case of exact (noiseless) measurements or observations. In some real applications, most data matrices are corrupted by some dense noise (i.e., Gaussian). Alternatively to (1.1), we can consider that the matrix D is generated by adding both sparse errors and small but dense noise to a perfectly low-rank matrix A:

$$D = A + E + N,$$

where N is a Gaussian matrix whose entries have small variance. For instance, one could start with the following relaxed version of the non-convex program

$$\begin{aligned} &\min_{L,R',E} \ \frac{1}{2}(\|L\|_F + \|R'\|_F) + \lambda \|E\|_1 \\ &s.t. \ \ \|D - LR' - E\|_F^2 \leq \epsilon^2, \end{aligned}$$

where ϵ is a upper bound on the noise level $\|N\|_F$. For the noise case, we may consider the stability analysis of the methods. In particular, we need to estimate the error bounds for the sparse and low-rank decomposition problem.

Acknowledgements

The authors would like to express their special thanks to the editors and the reviewers for their carefully reading of this manuscript and for their valuable suggestions.

References

[1] A. Argyriou, T. Evgeniou and M. Pontil, *Multi-task feature learning*, Adv. Neural Inform. Process. Syst., 2007, 19, 41–48.

[2] J. Abernethy, F. Bach, T. Evgeniou and J. P. Vert, *Low-rank Matrix Factorization with Attributes*, Arxiv preprint cs/0611124, 2006.

[3] Y. Amit, M. Fink, N. Srebro and S. Ullman, *Uncovering shared structures in multiclass classification*, in Proceedings of the 24th International Conference on Machine Learning, ACM, Providence, RI, 2007, 17–24.

[4] J-F. Cai, E. J. Candès and Z. Shen, *A singular value thresholding algorithm for matrxi completion*, SIAM J. Optim., 2010, 20(4), 1956–1982.

[5] E. J. Candès, X. Li, Y. Ma and J. Wright, *Robust principal component analysis*? Journal of the ACM, 2011, 58(3), 1–37.

[6] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 2009, 9(6), 717–772.

[7] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo and A. S. Willskyc, *Rank-sparsity incoherence for matrix decomposition*, SIAM J. Optim., 2011, 21(2), 572–596.

[8] P. Chen and D. Suter, *Recovering the missing components in a large noisy low-rank matrix: Application to SFM*, IEEE Trans. Pattern Anal. Machine Intelligence, 2004, 26(8), 1051–1063.

[9] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Math. Program., 1994, 64, 81–101.

[10] L. Cheng, M. Gong, D. Schuurmans and T. Caelli, *Real-time discriminative background subtraction*, IEEE Trans. Image Process., 2011, 20(5), 1401–1414.

[11] S. Deerwester, S. T. Dumains, T. Landauer, G. Furnas and R. Harshman, *Indexing by latent semantic analysis*, J. Soc. Inf. Sci., 1990, 41(6), 391–407.

[12] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1936, 1(3), 211–218.

[13] E. Esser, *Applications of Lagrangian-based Alternating Direction Methods and Connections to Split Bregman*, CAM report, 2009.

[14] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[15] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, volume 9 of SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1989.

[16] I. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[17] R. M. Larsen, *PROPACK-Software for large and sparse SVD calculations*, Available from http://sun.stanford.edu/~rmunk/PROPACK/.

[18] Z. Lin, M. Chen and Y. Ma, *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*, version 3, 2013 http://arxiv.org/abs/1009.5055.

[19] S. Ma, D. Goldfarb and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., 2011, 128(1), 321–353.

[20] M. Mesbahi and G. P. Papavassilopoulos, *On the rank minimization problem over a positive semidefinitelinear matrix inequality*, IEEE Trans. Automat. Control, 1997, 42, 239–243.

[21] C. Papadimitriou, P. Raghavan, H. Tamaki and S. Vempala, *Latent semantic indexing, a probabilistic analysis*, J. Comput. Syst. Sci., 2000, 61(2), 217–235.

[22] B. Recht, M. Fazel and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 2010, 52(3), 471–501.

[23] C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: A factorization method*, Int. J. Comput. Vision, 1992, 9, 137–154.

[24] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 1996, 38, 49–95.

[25] G. A. Watson, *Characterization of the subdifferential of some matrix norms*, Linear Algebra Appl., 1992, 170, 1039–1053.

[26] J. Wright, A. Ganesh, S. Rao, Y. Peng and Y. Ma, *Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization*, Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS 2009), 12/2009.

[27] X. Yuan and J. Yang, *Sparse and low-rank matrix decomposition via alternating direction methods*, Pacific Journal of Optimization, 2013, 9(1), 167–180.