

# LOW-RANK AND SPARSE MATRIX RECOVERY FROM NOISY OBSERVATIONS VIA 3-BLOCK ADMM ALGORITHM

Peng Wang<sup>1,2,†</sup>, Chengde Lin<sup>1</sup>, Xiaobo Yang<sup>3</sup>  
and Shengwu Xiong<sup>1,†</sup>

**Abstract** Recovering low-rank and sparse matrix from a given matrix arises in many applications, such as image processing, video background subtraction, and so on. The 3-block alternating direction method of multipliers (ADMM) has been applied successfully to solve convex problems with 3-block variables. However, the existing sufficient conditions to guarantee the convergence of the 3-block ADMM usually require the penalty parameter  $\gamma$  to satisfy a certain bound, which may affect the performance of solving the large scale problem in practice. In this paper, we propose the 3-block ADMM to recover low-rank and sparse matrix from noisy observations. In theory, we prove that the 3-block ADMM is convergent when the penalty parameters satisfy a certain condition and the objective function value sequences generated by 3-block ADMM converge to the optimal value. Numerical experiments verify that proposed method can achieve higher performance than existing methods in terms of both efficiency and accuracy.

**Keywords** Low-rank, sparse, nuclear norm minimization,  $\ell_1$ -norm minimization, 3-block alternating direction method.

**MSC(2010)** 49, 04.

## 1. Introduction

The low-rank and sparse matrix recovery problem arises widely in statistical model selection, system identification, and machine learning, such as image recovery [11], face recognition [20, 21] and background modeling [3, 12]. A central goal of the problem is to decompose a given high-dimensional matrix  $C$  into a sum of a low rank matrix  $A$  and a sparse matrix  $B$ . This can be mathematically formulated as

$$\min_{A,B} \text{rank}(A) + \lambda \|B\|_0, \quad \text{s.t.} \quad A + B = C, \quad (1.1)$$

where  $\|\cdot\|_0$  is  $l_0$  norm counting the number of nonzero entries,  $\lambda > 0$  is the hyperparameter, and  $C \in \mathbb{R}^{m \times n}$  is the given matrix.

<sup>†</sup>the corresponding author. Email address: [swxiong@whut.edu.cn](mailto:swxiong@whut.edu.cn) (S. Xiong), [p\\_wong@126.com](mailto:p_wong@126.com) (P. Wang)

<sup>1</sup>School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China

<sup>2</sup>School of Mathematics and Computational Science, Wuyi University, Jiangmen 529020, China

<sup>3</sup>Henan Branch of China Development Bank, Zhengzhou 450008, China

However, the problem (1.1) is generally NP-hard due to the non-convexity and non-smoothness of  $\text{rank}(\cdot)$  and  $\|\cdot\|_0$ . To overcome this difficulty, based on the fact that the  $\ell_1$  norm and the nuclear norm have been shown to be effective surrogate for  $\|\cdot\|_0$  and  $\text{rank}(\cdot)$  respectively [8], the problem (1.1) was often translated into the convex relaxation formulation [4, 5, 18]:

$$\min_{A,B} \|A\|_* + \lambda \|B\|_1, \quad \text{s.t.} \quad A + B = C, \quad (1.2)$$

where  $\|A\|_* = \sum_{i=1}^n \sigma_i(A)$  denotes the nuclear norm of matrix  $A$ .  $\sigma_i(A)$  is the  $i$ th singular value of  $A$ , and  $\|B\|_1 = \sum_{ij} |B_{ij}|$  denotes the  $\ell_1$  norm of  $B$  vectorized as a long vector in  $\mathbb{R}^{m \times n}$ . It has been shown that the above convex model can exactly recover a low-rank matrix  $A$  and a sparse matrix  $B$  under some conditions [3].

When the observation matrix  $C$  is corrupted by Gaussian noise, problem (1.2) is then formulated as the following nonsmooth convex optimization problem [13, 23]:

$$\min_{A,B} \|A\|_* + \lambda \|B\|_1, \quad \text{s.t.} \quad \|C - A - B\|_F \leq \delta, \quad (1.3)$$

where  $\delta > 0$  is a constant to describe noise intensity.

For the convenience of optimization, problem (1.3) can be usually transformed into its penalized form

$$\min_{A,B} \|A\|_* + \lambda \|B\|_1 + \frac{\rho}{2} \|A + B - C\|_F^2, \quad (1.4)$$

where  $\rho > 0$  is a penalty parameter.

In practice, a more common formulation for the corrupted observations is often expressed as

$$A + B + D = C,$$

where  $D$  denotes the random noise and  $\|D\|_F \leq \delta$  for some  $\delta > 0$ . Therefore, the original low-rank and sparse matrix decomposition can be formulated as

$$\min_{A,B,D} \|A\|_* + \lambda \|B\|_1 + \frac{\rho}{2} \|D\|_F^2, \quad \text{s.t.} \quad A + B + D = C. \quad (1.5)$$

In recent years, the two problems (1.2) and (1.3) have been proposed to extract low-dimensional structure from a possibly noisy data matrix [3, 23]. The work [13] has shown that accelerated proximal gradient (APG) method has impressive performance for solving (1.4) as well as (1.2). A twisted version of proximal alternating direction method of multipliers (TADMM) [19] was presented for recovering low-rank and sparse matrix from noisy observation, and the convergence and effectiveness of TADMM was also proved in the work. Alternating direction method of multipliers (ADMM) [1, 5, 15, 18, 22] is another widely used approach to efficiently solve such problems.

Optimizing the problem of recovering the low-rank and sparse matrix from noisy observations (1.5) is a challenge task. In this work, we employ the 3-block ADMM algorithm to solve (1.5). More importantly, we prove that the iterative sequence generated by proposed algorithm converges to an optimal solution of (1.5) under some mild conditions. Numerical experiments are executed to demonstrate the superiority of the proposed method to some existing methods.

The rest of the paper is organized as follows. In section 2, we introduce some notations and preliminaries which are necessary for our main results. Section 3

describes the details of recovering low rank and sparse matrix from noisy observation via 3-block ADMM. Section 4 analyzes the convergence properties of 3-block ADMM for (1.5) and shows that the objective function value sequence generated by 3-block ADMM algorithm converges to the optimal value. In Section 5, we compare 3-block ADMM with some existing ADMM based algorithms to demonstrate the performance of it. Finally, Section 6 concludes the paper.

## 2. Preliminaries and notations

In this section, we briefly introduce some notations and results on the shrinkage operator, which are key to the forthcoming sections.

For matrices  $X$  and  $Y$  with the same size, the inner product in  $\mathbb{R}^{m \times n}$  is defined as  $\langle X, Y \rangle = \text{tr}(X^T Y) = \sum_{i=1}^m \sum_{j=1}^n X_{ij} Y_{ij}$ . The norm associated with this inner product is called the Frobenius norm  $\|\cdot\|_F$ . For symmetric and positive semi-definite matrix  $P \succeq 0$ , let  $\|x\|_P^2 = x^T P x$  where  $x \in \mathbb{R}^n$ . When  $P$  is the  $n \times n$  identity matrix,  $\|\cdot\|_P$  denotes the  $\ell_2$ -norm  $\|\cdot\|_2$ .

**Lemma 2.1** ([18]). *For  $\mu > 0$  and  $Y \in \mathbb{R}^{m \times n}$ , the solution of the problem*

$$\min_{X \in \mathbb{R}^{m \times n}} \mu \|X\|_1 + \frac{1}{2} \|X - Y\|_F^2$$

is given by  $S_\mu(Y) \in \mathbb{R}^{m \times n}$ , which is defined component-wisely by

$$(S_\mu(Y))_{ij} = \max\{\text{abs}(Y_{ij}) - \mu, 0\} \cdot \text{sign}(Y_{ij}),$$

where  $\text{abs}(\cdot)$  and  $\text{sign}(\cdot)$  are absolute value and sign functions, respectively.

**Lemma 2.2** ([2]). *Given a matrix  $Y \in \mathbb{R}^{m \times n}$  with  $\text{rank}(Y) = r$ , let its Singular Value Decomposition be  $Y = U_Y \text{Diag}(\{\sigma_i\}_{1 \leq i \leq r}) V_Y^T$ , where  $U_Y \in \mathbb{R}^{m \times r}$ ,  $V_Y \in \mathbb{R}^{n \times r}$ , and a scalar  $\tau > 0$ . Then*

$$X = \mathcal{D}_\tau(Y) = U_Y \text{Diag}(\{\sigma_i - \tau\}_+) V_Y^T$$

is an optimal solution of the following problem

$$\min_{X \in \mathbb{R}^{m \times n}} \tau \|X\|_* + \frac{1}{2} \|X - Y\|_F^2,$$

where plus function  $\{\cdot\}_+$  is defined by  $a_+ = \max\{a, 0\}$  for  $a \in \mathbb{R}$ .

## 3. The 3-block ADMM algorithm

The classical alternating direction method of multipliers (ADMM) is also called 2-block ADMM with two variables, which has been studied in the literature (see [6, 7, 10]). The 3-block ADMM is extended from the classical ADMM for solving the following convex problem:

$$\begin{aligned} & \min f_1(x_1) + f_2(x_2) + f_3(x_3), \\ \text{s.t. } & A_1 x_1 + A_2 x_2 + A_3 x_3 = y, \\ & x_1 \in \chi_1, x_2 \in \chi_2, x_3 \in \chi_3, \end{aligned} \tag{3.1}$$

where  $x_i \in \mathbb{R}^{n_i}$ ,  $A_i \in \mathbb{R}^{m \times n_i}$ ,  $y \in \mathbb{R}^m$ ,  $f_i : \mathbb{R}^{n_i} \rightarrow (-\infty, +\infty]$  are closed proper convex functions, and  $\chi_i$  are closed convex sets for  $i = 1, 2, 3$ . For given  $(x_2^k, x_3^k; \lambda^k)$ , 3-block ADMM for (3.1) can be summarized as

$$\begin{cases} x_1^{k+1} = \arg \min_{x_1} \mathcal{L}_\beta(x_1, x_2^k, x_3^k; \lambda^k), \\ x_2^{k+1} = \arg \min_{x_2} \mathcal{L}_\beta(x_1^{k+1}, x_2, x_3^k; \lambda^k), \\ x_3^{k+1} = \arg \min_{x_3} \mathcal{L}_\beta(x_1^{k+1}, x_2^{k+1}, x_3; \lambda^k), \\ \lambda_{k+1} = \lambda_k - \beta(\sum_{i=1}^3 A_i x_i^{k+1} - y), \end{cases} \quad (3.2)$$

where

$$\mathcal{L}_\beta(x_1, x_2, x_3; \lambda) = \sum_{i=1}^3 f_i(x_i) - \langle \lambda, \sum_{i=1}^3 A_i x_i - y \rangle + \frac{\beta}{2} \left\| \sum_{i=1}^3 A_i x_i - y \right\|_2^2$$

denotes the augmented Lagrangian function of (3.1),  $\lambda$  is the Lagrange multiplier and  $\beta > 0$  represents a penalty parameter.

Though the convergence properties of the 2-block ADMM have been well-established [10, 16, 17], the convergence of 3-block ADMM (3.2) has remained unclear for a very long time. In recent work [6], a counterexample was given which showed that without further conditions the 3-block ADMM may actually fail to converge. If all the functions  $f_1$ ,  $f_2$  and  $f_3$  are strongly convex, Han et al. [9] proved the global convergence of the 3-block ADMM (3.2). When the second block in the objective function is a strongly convex function and constrained by one coupled linear equation, the work [14] presented a semi-proximal alternating direction of multiplier (sP-ADMM) for the problem (3.1). Specifically, the iterative scheme was described as follows:

$$\begin{cases} x_1^{k+1} = \arg \min_{x_1} \mathcal{L}_\beta(x_1, x_2^k, x_3^k; \lambda^k) + \frac{1}{2} \|x_1 - x_1^k\|_{P_1}^2, \\ x_2^{k+1} = \arg \min_{x_2} \mathcal{L}_\beta(x_1^{k+1}, x_2, x_3^k; \lambda^k) + \frac{1}{2} \|x_2 - x_2^k\|_{P_2}^2, \\ x_3^{k+1} = \arg \min_{x_3} \mathcal{L}_\beta(x_1^{k+1}, x_2^{k+1}, x_3; \lambda^k) + \frac{1}{2} \|x_3 - x_3^k\|_{P_3}^2, \\ \lambda_{k+1} = \lambda_k - \gamma\beta(\sum_{i=1}^3 A_i x_i^{k+1} - y), \end{cases} \quad (3.3)$$

where  $P_1$ ,  $P_2$ , and  $P_3$  are positive semidefinite matrices, and  $\gamma > 0$  is a step size for the dual update. Moreover, for  $\gamma \in (0, (1 + \sqrt{5})/2)$  and  $\beta \in (0, +\infty)$ , the global convergence of sP-ADMM was proved [14]. However, the above results usually require some restrictive conditions on  $P_i$ ,  $\gamma$  and  $\beta$  which may affect the performance of solving large-scale problem in practice. The 3-block ADMM for (3.1) was researched under the conditions that  $P_i = 0$ ,  $\beta > 0$  and  $\gamma = 1$  [15]. Moreover, they showed the algorithm was convergent and effective with any penalty parameter  $\beta > 0$  under no additional assumptions.

This paper mainly considers the 3-block ADMM for recovering low rank and sparse matrices from a given corrupted observation matrix expressed as (1.5). In order to solve (1.5) by 3-block ADMM, we first consider its augmented Lagrangian function

$$\begin{aligned} \mathcal{L}_\beta(A, B, D, \Lambda) &= \|A\|_* + \lambda \|B\|_1 + \frac{\rho}{2} \|D\|_F^2 - \langle \Lambda, A + B + D - C \rangle \\ &\quad + \frac{\beta}{2} \|A + B + D - C\|_F^2, \end{aligned} \quad (3.4)$$

where  $\Lambda \in \mathbb{R}^{m \times n}$  is a Lagrange multiplier matrix, and  $\beta > 0$  is the penalty parameter. The 3-block ADMM algorithm for recovering low-rank and sparse matrix is then formulated as follows:

$$\begin{cases} A_{k+1} = \arg \min_A \mathcal{L}_\beta(A, B_k, D_k, \Lambda_k), \\ B_{k+1} = \arg \min_B \mathcal{L}_\beta(A_{k+1}, B, D_k, \Lambda_k), \\ D_{k+1} = \arg \min_D \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D, \Lambda_k), \\ \Lambda_{k+1} = \Lambda_k - \beta(A_{k+1} + B_{k+1} + D_{k+1} - C). \end{cases}$$

This can be equivalently transformed into

$$\begin{cases} 0 \in \partial \|A_{k+1}\|_* - \Lambda_k + \beta(A_{k+1} + B_k + D_k - C), \\ 0 \in \lambda \partial \|B_{k+1}\|_1 - \Lambda_k + \beta(A_{k+1} + B_{k+1} + D_k - C), \\ \rho D_{k+1} - \Lambda_k + \beta(A_{k+1} + B_{k+1} + D_{k+1} - C) = 0, \\ \Lambda_{k+1} = \Lambda_k - \beta(A_{k+1} + B_{k+1} + D_{k+1} - C), \end{cases} \quad (3.5)$$

where  $\partial(\cdot)$  denotes the subgradient of a convex function. Eq. (3.5) implies that the equation  $\Lambda_{k+1} = \rho D_{k+1}$  holds, for any  $k \geq 0$ .

The first-order optimal conditions for (3.5) are given by  $A_{k+1}, B_{k+1} \in \mathbb{R}^{m \times n}$  and

$$\langle \hat{A} - A_{k+1}, G_1 - \Lambda_k + \beta(A_{k+1} + B_k + D_k - C) \rangle \geq 0, \quad \forall \hat{A} \in \mathbb{R}^{m \times n}, \quad (3.6)$$

$$\langle \hat{B} - B_{k+1}, S_1 - \Lambda_k + \beta(A_{k+1} + B_{k+1} + D_k - C) \rangle \geq 0, \quad \forall \hat{B} \in \mathbb{R}^{m \times n}, \quad (3.7)$$

where  $G_1 \in \partial \|A_{k+1}\|_*$  and  $S_1 \in \partial \lambda \|B_{k+1}\|_1$ .

By Lemmas 2.1 and 2.2, (3.5) can be solved by

$$\begin{cases} A_{k+1} = \mathcal{D}_\tau(\tau \Lambda_k - (B_k + D_k - C)), \\ B_{k+1} = \mathcal{S}_{\lambda \tau}(\tau \Lambda_k - (A_{k+1} + D_k - C)), \\ D_{k+1} = \frac{1}{\rho + \beta} [\Lambda_k - \beta(A_{k+1} + B_{k+1} - C)], \end{cases} \quad (3.8)$$

where  $\tau = \frac{1}{\beta}$ . Consequently, for the problem (1.5), there exists a saddle point  $(A^*, B^*, D^*, \Lambda^*)$  satisfying the KKT conditions:

$$\begin{cases} 0 \in \partial \|A^*\|_* - \Lambda^*, \\ 0 \in \partial \lambda \|B^*\|_1 - \Lambda^*, \\ \rho D^* - \Lambda^* = 0, \\ A^* + B^* + D^* - C = 0. \end{cases} \quad (3.9)$$

Based on the above analysis, we propose the algorithm for solving (1.5) which is described in Algorithm 1.

---

**Algorithm 1** Low-rank and sparse matrix recovery via 3-block ADMM

---

**Require:**  $B_1 \in \mathbb{R}^{m \times n}$ ;  $D_1 \in \mathbb{R}^{m \times n}$ ;  $\Lambda_1 \in \mathbb{R}^{m \times n}$ ;  $\beta, \lambda, \rho \in \mathbb{R}$ .

**Ensure:** low-rank matrix  $A_k$  and sparse matrix  $B_k$ .

- 1: **repeat**
  - 2:    $A_{k+1} = \mathcal{D}_\tau(\tau\Lambda_k - (B_k + D_k - C))$ , where  $\tau = \frac{1}{\beta}$ ;
  - 3:    $B_{k+1} = \mathcal{S}_{\lambda\tau}(\tau\Lambda_k - (A_{k+1} + D_k - C))$ ;
  - 4:    $D_{k+1} = \frac{1}{\rho+\beta}[\Lambda_k - \beta(A_{k+1} + B_{k+1} - C)]$ ;
  - 5:    $\Lambda_{k+1} = \Lambda_k - \beta(A_{k+1} + B_{k+1} + D_{k+1} - C)$ ;
  - 6: **until** the convergence conditions are not satisfied.
- 

## 4. Convergence of the proposed 3-block ADMM

In this section, we will show the convergence of Algorithm 1 for the problem (1.5). The following identity will be used frequently,

$$\langle V_1 - V_2, V_3 - V_1 \rangle = \frac{1}{2}(\|V_2 - V_3\|_F^2 - \|V_1 - V_2\|_F^2 - \|V_1 - V_3\|_F^2). \quad (4.1)$$

As a main result of proposed work, we first present the following theorem.

**Theorem 4.1.** *Let sequence  $\{(A_k, B_k, D_k, \Lambda_k)\}$  be generated by Algorithm 1,  $\beta > \rho$  where  $\rho$  and  $\beta$  are penalty parameters of the problem (1.4) and (1.5), respectively. Then the sequence  $\{\mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k)\}$  is monotonically decreasing.*

**Proof.** From (3.4), it follows that

$$\begin{aligned} & \mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_k, D_k, \Lambda_k) \\ &= \|A_k\|_* - \|A_{k+1}\|_* - \langle \Lambda_k, A_k - A_{k+1} \rangle + \frac{\beta}{2}\|A_k + B_k + D_k - C\|_F^2 \\ & \quad - \frac{\beta}{2}\|A_{k+1} + B_k + D_k - C\|_F^2 \\ & \geq \langle G_1, A_k - A_{k+1} \rangle - \langle \Lambda_k, A_k - A_{k+1} \rangle + \beta\langle A_k - A_{k+1}, A_{k+1} + B_k + D_k - C \rangle \\ & \quad + \frac{\beta}{2}\|A_k - A_{k+1}\|_F^2 \\ &= \langle A_k - A_{k+1}, G_1 - \Lambda_k + \beta(A_{k+1} + B_k + D_k - C) \rangle + \frac{\beta}{2}\|A_k - A_{k+1}\|_F^2 \\ & \geq \frac{\beta}{2}\|A_k - A_{k+1}\|_F^2, \end{aligned} \quad (4.2)$$

where  $G_1 \in \partial\|A_{k+1}\|_*$ , the first inequality is based on the property of convex function and identity (4.1), and the second inequality is obtained by setting  $\hat{A} = A_k$  from (3.6). Similarly,

$$\begin{aligned} & \mathcal{L}_\beta(A_{k+1}, B_k, D_k, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_k, \Lambda_k) \\ &= \lambda\|B_k\|_1 - \lambda\|B_{k+1}\|_1 - \langle \Lambda_k, B_k - B_{k+1} \rangle + \frac{\beta}{2}\|A_{k+1} + B_k + D_k - C\|_F^2 \\ & \quad - \frac{\beta}{2}\|A_{k+1} + B_{k+1} + D_k - C\|_F^2 \\ & \geq \langle S_1, B_k - B_{k+1} \rangle - \langle \Lambda_k, B_k - B_{k+1} \rangle + \beta\langle B_k - B_{k+1}, A_{k+1} + B_{k+1} + D_k - C \rangle \end{aligned}$$

$$\begin{aligned}
& + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 \\
& = \langle B_k - B_{k+1}, S_1 - \Lambda_k + \beta(A_{k+1} + B_{k+1} + D_k - C) \rangle + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 \\
& \geq \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2, \tag{4.3}
\end{aligned}$$

where  $S_1 \in \partial\lambda\|B_{k+1}\|_1$ . The first inequality is from the property of convex function and identity (4.1), and the second inequality is obtained by setting  $\tilde{B} = B_k$  from (3.7). By (3.5) and (4.1), we obtain

$$\begin{aligned}
& \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_k, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_k) \\
& = \frac{\rho}{2} \|D_k\|_F^2 - \frac{\rho}{2} \|D_{k+1}\|_F^2 + \langle D_{k+1} - D_k, \Lambda_k \rangle + \frac{\beta}{2} \|A_{k+1} + B_{k+1} + D_k - C\|_F^2 \\
& \quad - \frac{\beta}{2} \|A_{k+1} + B_{k+1} + D_{k+1} - C\|_F^2 \\
& = \frac{\rho}{2} \|D_k\|_F^2 - \frac{\rho}{2} \|D_{k+1}\|_F^2 + \langle D_{k+1} - D_k, \Lambda_k - \beta(A_{k+1} + B_{k+1} + D_{k+1} - C) \rangle \\
& \quad + \frac{\beta}{2} \|D_k - D_{k+1}\|_F^2 \\
& = \frac{\beta + \rho}{2} \|D_k - D_{k+1}\|_F^2. \tag{4.4}
\end{aligned}$$

Combining (4.2), (4.3) and (4.4), we have

$$\begin{aligned}
& \mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_k) \\
& \geq \frac{\beta}{2} \|A_k - A_{k+1}\|_F^2 + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 + \frac{\beta + \rho}{2} \|D_k - D_{k+1}\|_F^2. \tag{4.5}
\end{aligned}$$

From (3.5) and  $\Lambda^{k+1} = \rho D_{k+1}$ , we get

$$\begin{aligned}
& \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_{k+1}) \\
& = \langle \Lambda_{k+1} - \Lambda_k, A_{k+1} + B_{k+1} + D_{k+1} - C \rangle \\
& = \langle \Lambda_{k+1} - \Lambda_k, \frac{1}{\beta}(\Lambda_k - \rho D_{k+1}) \rangle \\
& = -\frac{1}{\beta} \|\Lambda_{k+1} - \Lambda_k\|_F^2 = -\frac{\rho^2}{\beta} \|D_k - D_{k+1}\|_F^2. \tag{4.6}
\end{aligned}$$

By (4.5) and (4.6), we obtain

$$\begin{aligned}
& \mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_{k+1}) \\
& \geq \frac{\beta}{2} \|A_k - A_{k+1}\|_F^2 + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 + \frac{\beta + \rho}{2} \|D_k - D_{k+1}\|_F^2 \\
& \quad - \frac{\rho^2}{\beta} \|D_k - D_{k+1}\|_F^2 \\
& = \frac{\beta}{2} \|A_k - A_{k+1}\|_F^2 + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 + \left(\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta}\right) \|D_k - D_{k+1}\|_F^2. \tag{4.7}
\end{aligned}$$

Since  $\beta \in (\rho, +\infty)$ , we have  $\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta} > 0$ . From the above inequality, it follows that the sequence  $\{\mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k)\}$  is monotonically decreasing. The proof is completed.  $\square$

Based on Theorem 4.1, the convergence of the sequence  $\{(A_k, B_k, D_k, \Lambda_k)\}$  generated by Algorithm 1 can be achieved by the following theorem.

**Theorem 4.2.** *Let the sequence  $\{(A_k, B_k, D_k, \Lambda_k)\}$  be generated by Algorithm 1, and  $\beta > \rho$ , then  $\{(A_k, B_k, D_k, \Lambda_k)\}$  is bounded, and any of its accumulation point  $(A^*, B^*, D^*, \Lambda^*)$  is an optimal solution of (1.5).*

**Proof.** First, we will show that the augmented Lagrangian function has a lower bound.

$$\begin{aligned} & \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_{k+1}) \\ &= \|A_{k+1}\|_* + \lambda \|B_{k+1}\|_1 + \frac{\rho}{2} \|D_{k+1}\|_F^2 - \langle \Lambda_{k+1}, A_{k+1} + B_{k+1} + D_{k+1} - C \rangle \\ & \quad + \frac{\beta}{2} \|A_{k+1} + B_{k+1} + D_{k+1} - C\|_F^2 \\ &= \|A_{k+1}\|_* + \lambda \|B_{k+1}\|_1 + \frac{1}{2\rho} \|\Lambda_{k+1} - \rho(A_{k+1} + B_{k+1} + D_{k+1} - C)\|_F^2 \\ & \quad + \frac{\beta - \rho}{2} \|A_{k+1} + B_{k+1} + D_{k+1} - C\|_F^2 \\ & \geq 0, \end{aligned} \tag{4.8}$$

where the second equality holds from the fact that  $\Lambda_{k+1} = \rho D_{k+1}$  and the inequality is obtained by  $\beta > \rho$ . By Theorem 4.1 and (4.8), we conclude the convergence of  $\{\mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k)\}$ .

Combining (4.7) and (4.8), for any  $k > 0$ , it follows that

$$\begin{aligned} & \sum_{i=1}^k \left( \frac{\beta}{2} \|A_i - A_{i+1}\|_F^2 + \frac{\beta}{2} \|B_i - B_{i+1}\|_F^2 + \left(\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta}\right) \|D_i - D_{i+1}\|_F^2 \right) \\ & \leq \sum_{i=1}^k \left( \mathcal{L}_\beta(A_i, B_i, D_i, \Lambda_i) - \mathcal{L}_\beta(A_{i+1}, B_{i+1}, D_{i+1}, \Lambda_{i+1}) \right) \\ & = \mathcal{L}_\beta(A_1, B_1, D_1, \Lambda_1) - \mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_{k+1}) \\ & \leq \mathcal{L}_\beta(A_1, B_1, D_1, \Lambda_1), \end{aligned} \tag{4.9}$$

where  $\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta} > 0$ .

Taking limit over the sequence in (4.9) and letting  $k \rightarrow +\infty$ , we have

$$\sum_{i=1}^{\infty} \left( \frac{\beta}{2} \|A_i - A_{i+1}\|_F^2 + \frac{\beta}{2} \|B_i - B_{i+1}\|_F^2 + \left(\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta}\right) \|D_i - D_{i+1}\|_F^2 \right) < \infty.$$

The above inequality implies that

$$\lim_{k \rightarrow \infty} \left( \frac{\beta}{2} \|A_k - A_{k+1}\|_F^2 + \frac{\beta}{2} \|B_k - B_{k+1}\|_F^2 + \left(\frac{\beta + \rho}{2} - \frac{\rho^2}{\beta}\right) \|D_k - D_{k+1}\|_F^2 \right) = 0. \tag{4.10}$$

From (4.8), we obtain

$$\mathcal{L}_\beta(A_{k+1}, B_{k+1}, D_{k+1}, \Lambda_{k+1}) \geq \|A_{k+1}\|_* + \lambda \|B_{k+1}\|_1.$$

Since sequence  $\{\mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k)\}$  is convergent and bounded, the sequence  $\{(A_{k+1}, B_{k+1})\}$  is also bounded. By (3.8) and  $\Lambda_{k+1} = \rho D_{k+1}$ , we can conclude that  $\{(D_{k+1}, \Lambda_{k+1})\}$  is a bounded sequence. Therefore,  $\{(A_k, B_k, D_k, \Lambda_k)\}$  is bounded.

Furthermore, there exists a convergent subsequence  $\{(A_{k_j}, B_{k_j}, D_{k_j}, \Lambda_{k_j})\}$  such that

$$\lim_{j \rightarrow \infty} A_{k_j} = \bar{A}, \quad \lim_{j \rightarrow \infty} B_{k_j} = \bar{B}, \quad \lim_{k \rightarrow \infty} D_{k_j} = \bar{D}.$$

By (4.10), we have

$$\lim_{j \rightarrow \infty} A_{k_j+1} = \lim_{j \rightarrow \infty} (A_{k_j+1} - (A_{k_j} - \bar{A})) = \bar{A}. \quad (4.11)$$

Similarly,

$$\lim_{j \rightarrow \infty} B_{k_j+1} = \bar{B}, \quad \lim_{j \rightarrow \infty} D_{k_j+1} = \bar{D}, \quad \lim_{j \rightarrow \infty} \Lambda_{k_j+1} = \bar{\Lambda}. \quad (4.12)$$

Consequently, we can concluded that

$$\lim_{k \rightarrow \infty} \mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k) = \mathcal{L}_\beta(\bar{A}, \bar{B}, \bar{D}, \bar{\Lambda}). \quad (4.13)$$

From (3.5), (4.10), (4.11), and (4.12), we obtain

$$\begin{aligned} \lim_{j \rightarrow \infty} \|\Lambda_{k_j} - \Lambda_{k_j+1}\|_F &= \lim_{j \rightarrow \infty} \rho \|D_{k_j} - D_{k_j+1}\|_F \\ &= \lim_{k \rightarrow \infty} \beta \|A_{k_j+1} + B_{k_j+1} + D_{k_j+1} - C\|_F \\ &= \|\bar{A} + \bar{B} + \bar{D} - C\|_F = 0, \end{aligned} \quad (4.14)$$

and

$$\begin{aligned} 0 \in \partial \|A_{k_j+1}\|_* - \Lambda_{k_j} + \beta(A_{k_j+1} + B_{k_j+1} + D_{k_j+1} - C) + \beta(B_{k_j} - B_{k_j+1}) \\ + \beta(D_{k_j} - D_{k_j+1}). \end{aligned} \quad (4.15)$$

Taking over the limit of (4.15) for  $j \rightarrow \infty$ , we have

$$0 \in \partial \|\bar{A}\|_* - \bar{\Lambda}. \quad (4.16)$$

Similarly,

$$0 \in \partial \|\bar{B}\|_1 - \bar{\Lambda}, \quad \text{and} \quad 2\rho\bar{D} - \bar{\Lambda} = 0. \quad (4.17)$$

Combining (4.14), (4.16) and (4.17),  $(\bar{A}, \bar{B}, \bar{D}, \bar{\Lambda})$  satisfies the optimal conditions for (1.5), and is an optimal solution of (1.5). Letting  $(\bar{A}, \bar{B}, \bar{D}, \bar{\Lambda}) = (A^*, B^*, D^*, \Lambda^*)$ , thus we complete the proof.  $\square$

From Theorem 4.2, if  $(A^*, B^*, D^*, \Lambda^*)$  is an optimal solution of (1.5), the optimal s of the objective function on the problem (1.5) can be denoted as  $f^*$ , and  $f^* = \|A^*\|_* + \lambda \|B^*\|_1 + \frac{\rho}{2} \|D^*\|_F^2$ . Therefore, the sequence of objective function value converges to the optimal value.

**Corollary 4.1.** *Let the sequence  $\{(A_k, B_k, D_k, \Lambda_k)\}$  be generated by Algorithm 1. Then the sequence of objective function value converges to the optimal value, i.e.*

$$\lim_{k \rightarrow \infty} (\|A_k\|_* + \lambda \|B_k\|_1 + \frac{\rho}{2} \|D_k\|_F^2) = f^*,$$

and

$$\lim_{k \rightarrow \infty} (A_k + B_k + D_k) = C.$$

**Proof.** From (3.5) and Theorem 4.2, it follows that

$$\|A_k + B_k + D_k - C\|_F = \frac{1}{\beta} \|\Lambda_{k-1} - \Lambda_k\|_F \rightarrow 0, \quad (4.18)$$

when  $k \rightarrow \infty$ . By (4.13) and (4.18), we have

$$\begin{aligned} & \left| \|A_k\|_* + \lambda \|B_k\|_1 + \frac{\rho}{2} \|D_k\|_F^2 - f^* \right| \\ & \leq \left| \mathcal{L}_\beta(A_k, B_k, D_k, \Lambda_k) - \mathcal{L}_\beta(A^*, B^*, D^*, \Lambda^*) \right| + \|\Lambda_k\|_F \cdot \|A_k + B_k + D_k - C\|_F \\ & \quad + \frac{\beta}{2} \|A_k + B_k + D_k - C\|_F^2 \rightarrow 0, \quad \text{for } k \rightarrow \infty. \end{aligned}$$

Thus the proof is concluded.  $\square$

## 5. Experiments

In this section, we present several numerical experiments to demonstrate the effectiveness of Algorithm 1. All the algorithms are implemented by Matlab R2015a, and are tested on a PC with 4 GB RAM and Intel Core i5-3550 CPU.

### 5.1. Background subtraction from video

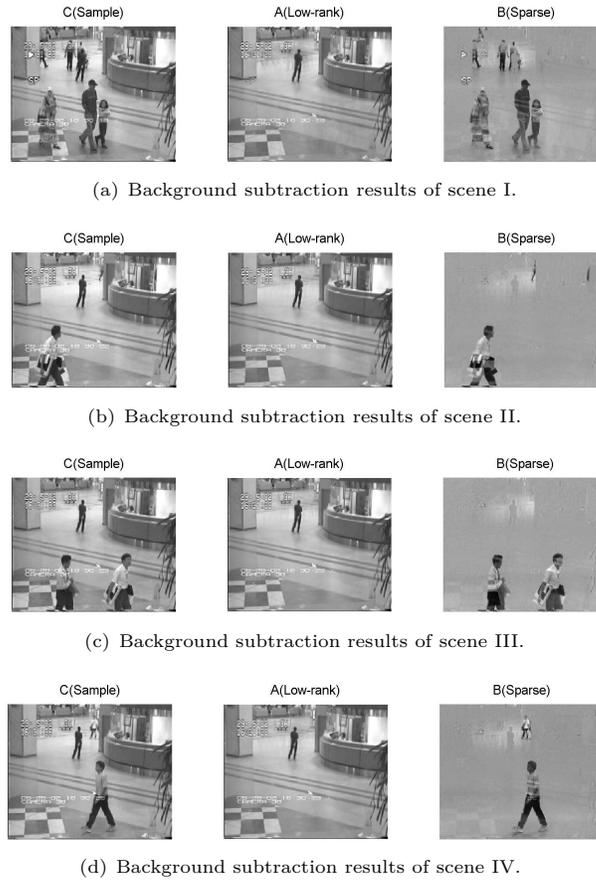
Background subtraction which aims to detect the moving objects from the background of a video stream is an important research field of computer vision. The background subtraction is usually modeled as a matrix decomposition problem. In this case, the background of each frame is relatively invariant while the foreground moving objects are sparse. Therefore, the background subtraction problem can be modeled as the low-rank and sparse structure as problem (1.5). As a result, 3-block ADMM can be employed to solve the problem (1.5) and eventually separate the background and foreground objects.

In this experiment, a surveillance video consisting of a static background and a number of people moving in the foreground is used to test the algorithm. The video is captured from an airport, which is composed of 200 frames with the resolution  $144 \times 176$ . Each frame is then stacked into a vector and the video is converted as a matrix  $A$  with the size of  $25344 \times 200$ .

To execute the experiment, some experimental settings are listed as follows. The parameters  $\lambda$  and  $\rho$  are selected by trail and error method and set as  $\lambda = 8$ ,  $\rho = 100$ .  $C$  (Sample) is original frame or observation, and  $D(\text{noise}) = C - A - B$ , where  $A$  (Low-rank) and  $B$  (Sparse) are, respectively, the background and the foreground that will be separated. We only take four frames of the video as an example and show the decomposition results in Figure 1. Figure 1 illustrates that the proposed method can successfully separate the background and the moving objects of all the four scenes.

### 5.2. Matrix recovery

In this section, we provide numerical tests for matrix recovery task and compare the 3-block alternating direction method (3b-ADMM) with some other algorithms, including 3-block semi-proximal ADMM (3b-sPADMM) [14] and twisted version of the proximal ADMM (TADMM) [19], for solving the problem (1.5).



**Figure 1.** Background subtraction from surveillance videos of four different scenes via 3-block ADMM.

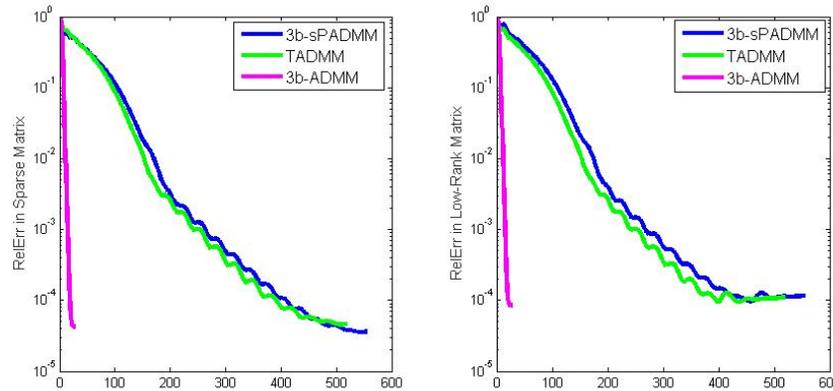
In this experiment, we generate the data in the same way as [22]. Let  $A^*$  and  $B^*$  are, respectively, the ground-truth of the low-rank and sparse matrices which will be modeled and recovered by (1.5),  $r$  is the rank of matrix  $A^*$ . More specifically, the low-rank matrix  $A^*$  is generated by  $A^* = UV$ , where  $U = rand(m, r)$  and  $V = rand(r, n)$ .  $B^*$  is generated as  $B^* = zeros(m, n)$ ,  $p = randperm(m \times n)$ ,  $L = round(spr \times m \times n)$  and  $B^*(p(1 : L)) = randn(L, 1)$ , where  $spr$  is the sparsity ratio,  $rand$ ,  $randn$ ,  $zeros$ ,  $randperm$  and  $round$  are the corresponding MATLAB functions. Let the noise  $D = 0.001 \cdot rand(m, n)$  and let  $C = A^* + B^* + D$ , then we obtain the observation  $C$  with noise.

In the following, we will recover the true matrices  $A^*$  and  $B^*$  from the observation  $C$ . Let  $\frac{\|A - A^*\|_F}{\|A^*\|_F}$  and  $\frac{\|B - B^*\|_F}{\|B^*\|_F}$  denote the relative error of low-rank matrix and in sparse matrix, respectively. In our experiments, we set the parameter  $\rho = 1e + 3$ ,  $\beta = 1e + 2$ , and the initial iteration  $(A_0, B_0, D_0, \Lambda_0) = (0, 0, 0, 0)$ . The stopping criterion of algorithms is achieved by using

$$\frac{\|(A_k, B_k) - (A_{k+1}, B_{k+1})\|}{\|(A_k, B_k)\|_F + 1} \leq \epsilon,$$

where  $\epsilon$  is tolerance.

Figure 2 shows the convergent curves of the three algorithms with the parameters setting as  $m = n = 100$ ,  $\text{rank}=10$ , and a given noise  $D = 0.001 \cdot \text{rand}(100, 100)$ . The sparsity ratio is set as 5%. It illustrates that the performance of 3b-sPADMM and TADMM is roughly equal, while the 3b-ADMM converges much faster than them. 3b-ADMM can achieve satisfying result at most 50 iterations.



**Figure 2.** The relative error of recovering sparse matrix and low-rank matrix via TADMM, 3b-ADMM and 3b-sPADMM algorithm respectively.

**Table 1.** Numerical comparison of algorithms

Size	Rank	Spr	Algorithm	$\frac{\ A-A^*\ _F}{\ A^*\ _F}$	$\frac{\ B-B^*\ _F}{\ B^*\ _F}$	Iter	Time(s)
(200,200)	15	10%	3b-ADMM	6.11e-05	3.17e-05	45	3.2
			3b-sPADMM	1.10e-04	2.88e-05	1000	31.4
			TADMM	1.30e-04	2.64e-05	770	25.2
(300,300)	20	5%	3b-ADMM	5.14e-05	2.60e-05	36	3.7
			3b-sPADMM	1.03e-04	2.10e-05	789	62.2
			TADMM	1.14e-04	1.66e-05	679	55.6
(400,400)	25	5%	3b-ADMM	4.11e-05	2.30e-05	46	7.8
			3b-sPADMM	9.94e-05	2.21e-05	1000	162.1
			TADMM	1.04e-04	1.66e-05	932	149.6

According to the data in Table 1, we can also observe that 3b-ADMM performs the better among the tested algorithms in terms of both the accuracy and the speed. Therefore, the proposed 3b-ADMM for solving the low-rank and sparse matrices recovery problem has favorable numerical performance and converges faster than some state-of-the-art algorithms.

## 6. Conclusions

In this paper, we mainly emphasize the applicability of the 3-block alternating direction method (ADMM) for solving the low-rank and sparse matrix recovery problem from noisy observations, and show the convergence of 3-block ADMM with a strongly convex block when the penalty parameters of the problem (1.5) satisfy  $\beta > \rho$ . Our algorithm is successful to separate background and foreground in video and recover the low-rank and sparse matrix. Moreover, numerical experiments demonstrate the efficiency of the proposed method in comparison with several existing algorithms, and our algorithm can spend less iteration and achieve higher precision.

## Acknowledgements

This work was in part supported by the National Key Research and Development Program of China(2017YFB1402203), in part by the National Natural Science Foundation of China(11601393, 11871379), in part by the Defense Industrial Technology Development Program(JCKY2018110C165), in part by Major Technological Innovation Projects in Hubei Province(2019AAA024), in part by the Fundamental Research Funds for the Central Universities(2017-YB-027), in part by the Natural Science Foundation of Guangdong Province (2018A030313063), the Innovation Project of Department of Education of Guangdong Province(2018KTSCX231).

## References

- [1] S. Boyd, N. Parikh, E. Chu et al., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends<sup>®</sup> in Machine learning, 2011, 3(1), 1–122.
- [2] J.-F. Cai, E. J. Candès and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM Journal on Optimization, 2010, 20(4), 1956–1982.
- [3] E. J. Candès, X. Li, Y. Ma and J. Wright, *Robust principal component analysis?*, Journal of the ACM (JACM), 2011, 58(3), 11.
- [4] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo and A. S. Willsky, *Rank-sparsity incoherence for matrix decomposition*, SIAM Journal on Optimization, 2011, 21(2), 572–596.
- [5] R. Chartrand, *Nonconvex splitting for regularized low-rank+ sparse decomposition*, IEEE Transactions on Signal Processing, 2012, 60(11), 5810–5819.
- [6] C. Chen, B. He, Y. Ye and X. Yuan, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, Mathematical Programming, 2016, 155(1-2), 57–79.
- [7] J. Eckstein and D. P. Bertsekas, *On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 1992, 55(1-3), 293–318.
- [8] M. Fazel, *Matrix rank minimization with applications*, Ph.D. thesis, Doctoral dissertation, Stanford University, 2002.
- [9] D. Han and X. Yuan, *A note on the alternating direction method of multipliers*, Journal of Optimization Theory and Applications, 2012, 155(1), 227–238.

- 
- [10] B. He and X. Yuan, *On the  $O(1/n)$  convergence rate of the douglas–rachford alternating direction method*, SIAM Journal on Numerical Analysis, 2012, 50(2), 700–709.
  - [11] Y. Hu, X. Liu and M. Jacob, *A generalized structured low-rank matrix completion algorithm for MR image recovery*, IEEE Transactions on Medical Imaging, 2019, 38(8), 1841–1851.
  - [12] S. Javed, A. Mahmood, T. Bouwmans and S. K. Jung, *Spatiotemporal low-rank modeling for complex scene background initialization*, IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(6), 1315–1329.
  - [13] H. Ji, S. Huang, Z. Shen and Y. Xu, *Robust video restoration by joint sparse and low rank matrix approximation*, SIAM Journal on Imaging Sciences, 2011, 4(4), 1122–1142.
  - [14] M. Li, D. Sun and K.-C. Toh, *A convergent 3-block semi-proximal admm for convex minimization problems with one strongly convex block*, Asia-Pacific Journal of Operational Research, 2015, 32(04), 1550024.
  - [15] T. Lin, S. Ma and S. Zhang, *Global convergence of unmodified 3-block admm for a class of convex minimization problems*, Journal of Scientific Computing, 2018, 76(1), 69–88.
  - [16] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 1979, 16(6), 964–979.
  - [17] R. D. Monteiro and B. F. Svaiter, *Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers*, SIAM Journal on Optimization, 2013, 23(1), 475–507.
  - [18] M. Tao and X. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM Journal on Optimization, 2011, 21(1), 57–81.
  - [19] J. J. Wang and W. Song, *An algorithm twisted from generalized admm for multi-block separable convex minimization models*, Journal of Computational and Applied Mathematics, 2017, 309, 342–358.
  - [20] J. Wright, A. Y. Yang, A. Ganesh et al., *Robust face recognition via sparse representation*, IEEE transactions on pattern analysis and machine intelligence, 2009, 31(2), 210–227.
  - [21] S. Yang, L. Zhang, L. He and Y. Wen, *Sparse low-rank component-based representation for face recognition with low-quality images*, IEEE Transactions on Information Forensics and Security, 2019, 14(1), 251–261.
  - [22] X. Yuan and J. Yang, *Sparse and low rank matrix decomposition via alternating direction method*, Pacific Journal of Optimization, 2013, 9(1), 167.
  - [23] Z. Zhou, X. Li, J. Wright et al., *Stable principal component pursuit*, in *2010 IEEE international symposium on information theory*, IEEE, 2010, 1518–1522.