Applied Mathematical Modelling 39 (2015) 4859–4870

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A penalty decomposition method for rank minimization problem with affine constraints

Zheng-Fen Jin[a], Zhongping Wan[a,*], Xiaoke Zhao[a], Yunhai Xiao[b]

[a] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, PR China
[b] College of Mathematics and Information Science, Henan University, Kaifeng 475004, PR China

A R T I C L E   I N F O

Article history:
Received 21 May 2014
Accepted 23 March 2015
Available online 24 April 2015

Keywords:
Rank minimization
Penalty decomposition method
Block coordinate descent method

A B S T R A C T

The rank minimization problem with affine constraints is widely applied in the fields of control, system identification, and machine learning, and attracted much attention and well studied in the past few years. Unlike most of the existing methods where a nuclear norm is used to approximate the rank term, in this paper, we apply the penalty decomposition method to solve the rank minimization problem directly. One subproblem can be solved effectively by using linear conjugate gradient method, and the other one has closed-form solutions by taking full use of the problem's favorable structure. Under some suitable assumptions, the convergence results for the proposed method are given. Finally, we do numerical experiments on randomly generated and real data, the results show that the proposed method is effective and promising.

1. Introduction

In this paper, we consider the affine constrained rank minimization problem

$$\min_{X\in\mathbb{R}^{m\times n}} \operatorname{rank}(X), \quad \text{s.t. } \mathcal{A}(X) = b, \tag{1}$$

where $X \in \mathbb{R}^{m\times n}$ is a decision variable, $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ is a linear map and $b \in \mathbb{R}^p$ is a given measurement vector. This problem appears in many applications arising in various areas, for instance, the low-order realization of linear control system[1], system identification in engineering[2] and machine learning[3], etc. A particular case of problem (1) is the matrix completion problem

$$\min_{X\in\mathbb{R}^{m\times n}} \operatorname{rank}(X), \quad \text{s.t. } X_{i,j} = M_{i,j}, \quad \forall (i,j) \in \Omega, \tag{2}$$

where M is an unknown matrix with some available sampled entries, and Ω is a set of index pairs (i,j). Given a subset of entries of a matrix, problem (2) is to recover the missing entries to obtain a complete low-rank matrix. The problem (1) is generally viewed as NP-hard because of the combinational nature of the function "rank$(\cdot)$" [4]. Instead of solving the rank minimization problem, many researchers have used the nuclear norm as a surrogate. The nuclear norm of X is defined as the sum of its positive singular values, and it is the best convex approximation of the rank function over the unit ball of matrices with norm less than one. Assume that the matrix X has r positive singular values of $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$, then its nuclear norm is formulated as $\|X\|_* = \sum_{i=1}^r \sigma_i(X)$.

* Corresponding author.
E-mail addresses: jinzhengfen@whu.edu.cn (Z.-F. Jin), mathwanzhp@whu.edu.cn (Z. Wan), zhaoxiaokehenan@126.com (X. Zhao), yhxiaomath@gmail.com (Y. Xiao).

http://dx.doi.org/10.1016/j.apm.2015.03.054
0307-904X/© 2015 Elsevier Inc. All rights reserved.

Using the nuclear norm, the problem (1) can be transformed into

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_*, \quad \text{s.t. } \mathcal{A}(X) = b. \tag{3}$$

In some applications, the measurement b may be contained with a small amount of noise. Hence, the constraint should be relaxed in the noisy case. There are two typical relaxation of models for describing the situation with noise. One is the inequality constrained nuclear norm minimization

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_*, \quad \text{s.t. } \|\mathcal{A}(X) - b\|_2 \leqslant \delta, \tag{4}$$

where $\delta \geqslant 0$ measures the noisy level. The other one is the nuclear norm regularized least square problem

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_* + \frac{\gamma}{2}\|\mathcal{A}(X) - b\|_2^2, \tag{5}$$

where $\gamma \geqslant 0$ balances both terms for minimization. Clearly, when δ is equal to zero, the model (4) reduces to (3). Particularly, if the parameter γ and δ are chosen appropriately, both of the solutions coincide. The nuclear norm minimization problem has attracted considerable attention and has taken good progress. Problem (3) is convex, which can be rewritten as an equivalent semidefinte programming problem and solved subsequently by some SDP solvers. However, these solvers can usually solve a problem with relatively lower dimension. To overcome this disadvantage, Cai et al. [5] proposed a singular value thresholding (SVT) algorithm for solving

$$\min_{X\in\mathbb{R}^{m\times n}} \tau\|X\|_* + \frac{1}{2}\|X\|_F^2, \quad \text{s.t. } \mathcal{A}(X) = b, \tag{6}$$

where $\tau \geqslant 0$ is a given parameter and $\|\cdot\|_F$ is the Frobenius norm. When $\tau \to \infty$, the solution of (6) converges to the one of (3). Nevertheless, a large value of τ may make the thresholding step to eliminate most of the small singular values and produce a low rank output [5]. However, when the matrix is not very low rank, the efficiency of SVT may not be remarkable. Ma et al. [4] proposed a fixed point continuation with approximate SVD (FPCA) method to solve a Lagrangian form of (3)

$$\min_{X\in\mathbb{R}^{m\times n}} \mu\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2, \tag{7}$$

where $\mu > 0$ is a given parameter. The FPCA is an extension of the well-known fixed point continuation algorithm of Hale et al. [6] for ℓ_1-regularized minimization. It can handle large scale problems robustly, but its efficiency for the noiseless affine constraint case is not clear. Subsequently, Toh and Yun [7] proposed an accelerated proximal gradient (APG) method to accelerate the rate of convergence. Yang and Yuan [8] presented an alternating direction method (ADM), where a linearized technique is used to ensure that both subproblems admit closed-form solutions. Another ADM type algorithm due to Xiao and Jin [9], in which one of the subproblems is solved iteratively by the Barzilai–Borwein gradient method [10]. Other ADM type of algorithms can refer to the references [23,24].

The low rank matrix has been recovered successfully via the nuclear norm minimization, but the matrix produced from its least measurements may be with much higher rank than the real one unless some strict condition satisfied [11–13]. Hence, the efficient algorithms to solve the rank minimization problem directly are highly needed. For instance, the greedy algorithms ADMiRA [14] and SVP [15] were proposed to solve the following rank approximation minimization problem

$$\min_{X\in\mathbb{R}^{m\times n}} \|\mathcal{A}(X) - b\|_2, \quad \text{s.t. } \operatorname{rank}(X) \leqslant r.$$

Based on a smooth approximation of rank function, another algorithm was proposed, analyzed, and tested in [16]. As far as we know, the algorithms to directly solve the affine rank minimization problem (1) are relatively fewer.

Recently, Lu and Zhang [18] proposed a penalty decomposition (PD) method to solve the general rank minimization problem and tested it by the matrix completion problem (2). Due to the well performance of their method, in this paper, we extend it to solve the rank minimization problem with affine constraints, and test its numerical performance by comparing with other well-known solvers. The main idea of PD method is to solve a quadratic penalty function of the problem (1) by block coordinate descent (BCD) method. Since the BCD method works well if each subproblem minimization is performed quite efficiently [18]. In our proposed method, one of the subproblems admits closed-form solution, and the other one is solved efficiently by linear conjugate gradient method. We establish the convergence result under some mild conditions. Finally, we do numerical experiments and compare with other algorithms in the literature.

The rest of this paper is organized as follows. In Section 2, we will briefly review some results proposed in the [18], discuss the construction of our method and the convergence analysis. The numerical results are presented in Section 3. Finally, we conclude this paper with some remarks in Section 4.

2. Algorithm

In this section, we construct our PD method to solve the affine rank minimization problem (1). We begin with reviewing some results proposed in [18], which take a class of special rank minimization problems having closed form solutions and play an important role in solving the resulting subproblems.

2.1. Preliminaries

In the first place, we list a couple of definitions as follows.

Definition 2.1. A norm $\|\cdot\|$ is a unitarily invariant norm on $\mathbb{R}^{m\times n}$ if $\|UXV\| = \|X\|$ for all $U \in \mathbb{U}^m,\ V \in \mathbb{U}^n$, where the $\mathbb{U}^n$ is the set of all unitary matrices in $\mathbb{R}^{m\times n}$.

Definition 2.2. A function $F : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a unitary invariant function if $F(UXV) = F(X)$ for all $U \in \mathbb{U}^m,\ V \in \mathbb{U}^n,\ X \in \mathbb{R}^{m\times n}$. Moreover, a set $\mathcal{X} \subseteq \mathbb{R}^{m\times n}$ is unitary invariant set if

$$\{UXV : U \in \mathbb{U}^m, V \in \mathbb{U}^n, X \in \mathcal{X}\} = \mathcal{X}.$$

Proposition 2.1 [18]. *Let $\|\cdot\|$ be a unitarily invariant norm on $\mathbb{R}^{m\times n}$, and let $F : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a unitarily invariant function. Suppose that $\mathcal{X} \subseteq \mathbb{R}^{m\times n}$ is a unitarily invariant set. Let matrix $A \in \mathbb{R}^{m\times n}$ be given, $q = \min(m,n)$, and let ϕ be a non-decreasing function on $[0,\infty]$. Suppose that $U\Sigma(A)V^\top$ is the singular value decomposition of A. Then $X^* = U\mathcal{D}(x^*)V^T$ is an optimal solution of the problem*

$$\min F(X) + \phi(\|X - A\|) \quad \text{s.t. } X \in \mathcal{X}, \tag{8}$$

where $x^* \in \mathbb{R}^q$ is an optimal solution of the problem

$$\min F(\mathcal{D}(x)) + \phi(\|\mathcal{D}(x) - \Sigma(A)\|) \quad \text{s.t. } \mathcal{D}(x) \in \mathcal{X}. \tag{9}$$

where $\mathcal{D}(x)$ denotes a $m \times n$ matrix with $\mathcal{D}_{ij}(x) = x_i$ if $i = j$, and $\mathcal{D}_{ij}(x) = 0$ if not.

This result shows that some matrix optimization problems over a subset of $\mathbb{R}^{m\times n}$ can be solved by means of solving their corresponding lower dimensional vector optimization problems [18]. The special case of Proposition 2.1 is listed as follows.

Corollary 2.1 [18]. *Let $v \geqslant 0,\ A \in \mathbb{R}^{m\times n}$, and $q = \min(m,n)$. Suppose that $\mathcal{X} \subseteq \mathbb{R}^{m\times n}$ is a unitarily invariant set, and $U\Sigma(A)V^T$ is the singular value decomposition of A. Then $X^* = U\mathcal{D}(x^*)V^T$ is an optimal solution of the problem*

$$\min\left\{v\ rank(X) + \frac{1}{2}\|X - A\|_F^2 : X \in \mathcal{X}\right\}, \tag{10}$$

where $x^* \in \mathbb{R}^q$ is an optimal solution of the problem

$$\min\left\{v\|x\|_0 + \frac{1}{2}\|x - \sigma(A)\|_2^2 :\ \mathcal{D}(x) \in \mathcal{X}\right\}, \tag{11}$$

where $\sigma(A)$ denotes the vector consisting of all singular values of A arranged in nondecreasing order.

From [19], it is easy to see that the closed-form solution of (11) satisfies

$$x^* = \mathcal{H}_{(2v)^{0.5}}(\sigma(A)),$$

where $\mathcal{H}_{(.)}$ is an element wise hard thresholding operator proposed in [20], which is defined by

$$\mathcal{H}_{(2v)^{0.5}}(\sigma(A)) = \begin{cases} \sigma(A), & \text{if } \sigma(A) > (2v)^{0.5}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

2.2. Penalty decomposition method for problem (1)

In this subsection, we mainly focus on solving the affine rank minimization problem (1) via PD method. Introducing an auxiliary variable Y, problem (1) is equivalently transformed into

$$\min_{X,Y} \text{rank}(Y) \quad \text{s.t. } \mathcal{A}(X) = b, \quad X = Y \quad X \in \mathbb{R}^{m\times n}, \quad Y \in \mathbb{R}^{m\times n}. \tag{13}$$

Given a parameter $\rho > 0$, the corresponding quadratic penalty function of (13) is

$$P_\rho(X,Y) = \text{rank}(Y) + \frac{\rho}{2}\left(\|\mathcal{A}(X) - b\|_2^2 + \|X - Y\|_F^2\right), \tag{14}$$

where ρ is a penalty parameter. Clearly, as $\rho \to \infty$, the solution of problem (14) converges to the one of (13). Now we apply the PD method to minimize the penalty problem, i.e.,

$$\min_{X,Y}\{P_\rho(X,Y) : X, Y \in \mathbb{R}^{m\times n}\}. \tag{15}$$

For simplicity, we set the outer iteration index as k, and use l to represent the inner iteration index.

Given $\{(X_l^k, Y_l^k)\}$, the main steps are to apply the BCD method for solving the both penalty subproblems, which can be described as follows:

$$X_{l+1}^k \in \text{Argmin}_{X\in\mathbb{R}^{m\times n}} P_\rho(X, Y_l^k), \tag{16}$$

$$Y_{l+1}^k \in \text{Argmin}_{Y\in\mathbb{R}^{m\times n}} P_\rho(X_{l+1}^k, Y). \tag{17}$$

Firstly, it is not hard to see that problem (16) can be reformulated as

$$X_{l+1}^k \in \text{Argmin}_{X\in\mathbb{R}^{m\times n}} P_\rho(X, Y_l^k) = \text{Argmin}_{X\in\mathbb{R}^{m\times n}} \frac{\rho}{2}(\|\mathcal{A}(X) - b\|_2^2 + \|X - Y_l^k\|_F^2). \tag{18}$$

Let $Q(X) = \rho(\|\mathcal{A}(X) - b\|_2^2 + \|X - Y_l^k\|_F^2)/2$, it is clear to see that $Q(X)$ is a quadratic function. Then let $G(X) = \rho(\mathcal{A}^*(\mathcal{A}(X) - b) + X - Y_l^k)$ be the gradient of $Q(X)$. Forcing $G(X) = 0$ reduces to the following linear system

$$(\mathcal{A}^*\mathcal{A} + I)X = \mathcal{A}^*(b) + Y_l^k, \tag{19}$$

where I is an identity matrix. Subsequently, we get

$$X_{l+1}^k = (\mathcal{A}^*\mathcal{A} + I)^{-1}(\mathcal{A}^*(b) + Y_l^k),$$

where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$. Although we can directly solve the linear equations to get an exact solution, the computing on the inverse of $(\mathcal{A}^*\mathcal{A} + I)$ is more expensive especially when the matrix is large. Hence, it is natural to use a linear conjugate gradient method instead.

For the sake of simplicity, we denote $C = I + \mathcal{A}^*\mathcal{A}$, and $D_l^k = \mathcal{A}^*(b) + Y_l^k$. Choosing $\hat{X}_0 = X_l^k$, $\hat{R}_0 = C\hat{X}_0 - D_l^k$ and $\hat{P}_0 = -\hat{R}_0$, the sequence $\{\hat{X}_i\}$ is computed by the following iterative process while $\|C\hat{X}_i - D_l^k\|_F \neq 0$,

$$\begin{cases} \alpha_i = -\frac{\langle \hat{R}_i, \hat{P}_i\rangle}{\langle \hat{P}_i, C\hat{P}_i\rangle}, \\ \hat{X}_{i+1} = \hat{X}_i + \alpha_i \hat{P}_i, \\ \hat{R}_{i+1} = C\hat{X}_{i+1} - D_l^k, \\ \beta_{i+1} = \frac{\langle \hat{R}_{i+1}, C\hat{P}_i\rangle}{\langle \hat{P}_i, C\hat{P}_i\rangle}, \\ \hat{P}_{i+1} = -\hat{R}_{i+1} + \beta_{i+1}\hat{P}_i \end{cases} \tag{20}$$

and set $X_{l+1}^k = \hat{X}_i$. In (20), the symbol $\langle\cdot,\cdot\rangle$ denotes the standard inner trace product of matrices. In practical computation, it is not necessary to compute the X_{l+1}^k exactly. For sufficiently small $\epsilon > 0$, it is shown that $\|C\hat{X}_i - D_l^k\|_F \leqslant \epsilon$ may produce acceptable solutions. Note that the matrix–vector multiplication is only required at each iteration, thus it is not necessary to store the matrices C and D_l^k explicitly. The linear conjugate gradient method is very efficient to solve linear system, and its convergence properties are well-studied. We refer to [21] for more details.

Secondly, the subproblem (17) is equivalent to

$$Y_{l+1}^k \in \text{Arg}\min_{Y\in\mathbb{R}^{m\times n}} \text{rank}(Y) + \frac{\rho}{2}\|Y - X_{l+1}^k\|_F^2. \tag{21}$$

Let $X_{l+1}^k = U\Sigma(X_{l+1}^k)V^T$. According to the Corollary 2.1, it is easy to show that the closed-form solutions of (21) can be described as

$$Y_{l+1}^k = U\mathcal{D}(y^*)V^T, \tag{22}$$

where

$$y^* = \mathcal{H}_{(2/\rho)^{0.5}}(\sigma(X_{l+1}^k)),$$

where $\mathcal{H}(\cdot)$ is defined in (12).

If (X_{l+1}^k, Y_{l+1}^k) satisfies the following condition

$$\|\nabla_X Q_{\rho_k}(X_{l+1}^k, Y_{l+1}^k)\|_F \leqslant \epsilon_k, \tag{23}$$

where the ϵ_k is a given positive sequence, then let $(X^k, Y^k) = (X_{l+1}^k, Y_{l+1}^k)$, and starting the outer loops.

In light of all the analysis above, we state the basic framework of the penalty decomposition method joining with conjugate gradient algorithm (abbreviated as PD-CG) as follows:

Algorithm 1. (PD-CG)

```
Initialization: Given $Y_0^0 \in \mathbb{R}^{m\times n}$, constants $\rho_0 > 0,\ \sigma > 1,\ \epsilon > 0,\ \Upsilon \geqslant \max\{\mathrm{rank}(X^{feas}),\ \min_X P_{\rho_0}(X, Y_0^0)\},\ \mu > 0$, integer $\bar{i} > 0$, and positive decreasing sequence $\{\epsilon_k\}$. Set $k = 0$.
Step 1. Set $l = 0$ and find an approximate solution $(X^k, Y^k)$ via the following steps (1)–(4):
    (1) Compute $X_{l+1}^k$ via (16) for fixed $Y_l^k$. Let $C = I + \mathcal{A}^*\mathcal{A}$ and $D_l^k = \mathcal{A}^*(b) + Y_l^k$.
        (1a). Let $\hat{X}_0 = X_l^k, \hat{R}_0 = C\hat{X}_0 - D_l^k$ and $\hat{P}_0 = -\hat{R}_0$. Set $i = 0$;
        (1b). While $\hat{R}_i > \epsilon$ and $i < \bar{i}$ do,
            Compute $\hat{X}_i$ via (20);
            Let $i = i + 1$;
        (1c). Set $X_{l+1}^k = \hat{X}_i$.
    (2) Compute $Y_{l+1}^k$ via (17) with fixed $X_{l+1}^k$.
    (3) Set $(X^k, Y^k) := (X_{l+1}^k, Y_{l+1}^k)$. If $(X^k, Y^k)$ satisfies (23), then go to Step 2.
    (4) Set $l \leftarrow l + 1$ and go to step (1).
Step 2. Set $\rho_{k+1} := \sigma\rho_k$.
Step 3. If $\min_X P_{\rho_{k+1}}(X, Y^k) \geqslant \Upsilon$, set $Y_0^{k+1} := X^{feas}$. Otherwise, set $Y_0^{k+1} := Y^k$.
Step 4. Set $k \leftarrow k + 1$ and go to Step 1.
```

Remark 2.1. In the **Step 1**, (1a)–(1c) are the iterative process of linear conjugate gradient method, where $\bar{i} = 5$. Due to the simple structure of the coefficient matrix C, it can get the optimal solution of the subproblem within 5 steps by a series of experiments.

Remark 2.2. As stated in [18], the terminate condition (23) of the Algorithm 1 is used to establish the global convergence of PD method. In practical applications, we also choose the following terminate condition

$$\frac{\left|Q_{\rho_k}(X_l^k, Y_l^k) - Q_{\rho_k}(X_{l-1}^k, Y_{l-1}^k)\right|}{max(|Q_{\rho_k}(X_l^k, Y_l^k)|, 1)} \leqslant \epsilon_I, \tag{24}$$

for some $\epsilon_I > 0$. Unlike the [18], we take the terminate condition of the outer iteration in this paper as follows

$$\|X^k - Y^k\|_F \leqslant \epsilon_0 \ and\ \|\mathcal{A}(X^k) - b\|_2 \leqslant \epsilon_0, \tag{25}$$

where $\epsilon_0 > 0$ is a given constant.

Remark 2.3. Due to the nonconvex property of function P_ρ, the method may only obtain a local stationary point. Therefore in order to enhance the efficiency, we apply the same techniques proposed in the [18], which is to execute it multiple times by restarting from a suitable perturbation of the current best approximate solution.

2.3. Convergence analysis

In this subsection, we firstly give the convergence of the inner iterations of Algorithm 1. The following convergence result shows that the inner method is effectively for solving the problem (15).

Theorem 2.1. *Let $\{(X_l, Y_l)\}$ be the sequence generated by the above BCD method, and let $\epsilon > 0$ be given, suppose that $\{(X_l, Y_l)\}$ has at least an accumulation point, then there exists a sufficiently large $l > 0$ such that*

$$\|\nabla_X Q_\rho(X_l, Y_l)\|_F \leqslant \epsilon, \tag{26}$$

where $Q_\rho(X_l, Y_l) = \frac{\rho}{2}(\|\mathcal{A}(X_l) - b\|_2^2 + \|X_l - Y_l\|_F^2)$.

Proof. The inner iterations show that

$$P_\rho(X_{l+1}, Y_l) \leqslant P_\rho(X, Y_l), \quad \forall X \in \mathbb{R}^{m\times n}, \tag{27}$$

$$P_\rho(X_{l+1}, Y_{l+1}) \leqslant P_\rho(X_{l+1}, Y), \quad \forall Y \in \mathbb{R}^{m\times n}. \tag{28}$$

Subsequently,

$$P_\rho(X_{l+1},Y_{l+1}) \leqslant P_\rho(X_{l+1},Y_l) \leqslant P_\rho(X_l,Y_l), \tag{29}$$

which indicates that the sequence $P_\rho(X_l,Y_l)$ is non-increasing. By the assumption, (X_l,Y_l) has at least an accumulation point, denoted by (X^*,Y^*), then there exists a subsequence $\mathcal{L}$ such that

$$\lim_{l\in\mathcal{L},\,l\to\infty}(X_l,Y_l) = (X^*,Y^*). \tag{30}$$

Because of the monotonicity of $P_\rho(X_l,Y_l)$ and the boundary of $P_\rho(X_l,Y_l)_{l\in L}$, it is clearly that $P_\rho(X_l,Y_l)$ is bounded below and hence $\lim_{l\to\infty}P_\rho(X_l,Y_l)$ exists. Combining with the (29), we obtain

$$\lim_{l\to\infty}P_\rho(X_l,Y_l) = \lim_{l\to\infty}P_\rho(X_{l+1},Y_l). \tag{31}$$

Because the $\{\text{rank}(Y_l)\}_{l\in L}$ is bounded, there exists a subsequence $l\in\bar{\mathcal{L}}, l\to\infty$ such that $\lim_{l\in\bar{\mathcal{L}},l\to\infty}\text{rank}(Y_l)$ exists. Moreover, together with (31), we have

$$\begin{aligned}\lim_{l\in\bar{\mathcal{L}},l\to\infty}P_\rho(X_{l+1},Y_l) &= \lim_{l\in\bar{\mathcal{L}},l\to\infty}\text{rank}(Y_l)+\frac{\rho}{2}(\|\mathcal{A}(X_{l+1})-b\|_2^2+\|X_{l+1}-Y_l\|_F^2)\\ &= \lim_{l\in\bar{\mathcal{L}},l\to\infty}\text{rank}(Y_l)+\frac{\rho}{2}(\|\mathcal{A}(X_l)-b\|_2^2+\|X_l-Y_l\|_F^2)\\ &= \lim_{l\in\bar{\mathcal{L}},l\to\infty}P_\rho(X_l,Y_l)\\ &= P_\rho(X^*,Y^*).\end{aligned} \tag{32}$$

Taking limits on both sides of (27) as $l\in\bar{\mathcal{L}}, l\to\infty$ and using the above results (32), we further obtain

$$P_\rho(X,Y^*) \geqslant P_\rho(X^*,Y^*).$$

Because the rank(Y^*) is a constant, it holds that

$$Q_\rho(X,Y^*) \geqslant Q_\rho(X^*,Y^*), \quad \forall X\in\mathbb{R}^{m\times n}.$$

By the first order optimal condition, we have

$$\|\nabla_X Q_\rho(X^*,Y^*)\|_F = 0.$$

Note that $\nabla_X Q_\rho(\cdot,\cdot)$ is continuous and together with (30), it is easy to see that

$$\lim_{l\in\bar{\mathcal{L}},l\to\infty}\|\nabla_X Q_\rho(X_l,Y_l)\|_F = \|\nabla_X Q_\rho(X^*,Y^*)\|_F = 0, \tag{33}$$

which shows that desirable result (26). □

Next, we assume that an approximate solution (X^k,Y^k) produced by the BCD method satisfies $(X^k,Y^k)\in\mathcal{X}\times\mathcal{Y}$, then the problem (1) can be viewed as a special case of the general rank minimization problem [18] with $f(X)\equiv 0,\ v=1,\ h(X)=\mathcal{A}(X)-b=0$, where $\mathcal{X}$ and $\mathcal{Y}$ are compact convex set. Both of the convergence results and its proof for the PD method can be seen in [18]. Here we just give the convergence result as follows.

Theorem 2.2. *Let $\{(X^k,Y^k)\}$ be the sequence generated by the above PD method satisfying the condition (26), and let $U^k\in\mathbb{R}^{m\times r}$ and $V^k\in\mathbb{R}^{r\times n}$ be such that $(U^k)^T U^k = I,\ Y^k = U^kV^k$. Let $\{(u^k,Z_X^k)\}$ be defined by the following form: $u^k=\rho_k(\mathcal{A}(X^k)-b),\quad Z_X^k=\rho_k(X^k-Y^k)$. Assume that $\epsilon_k\to 0$. Suppose that the level set $\mathcal{X}_\Upsilon=\{X\in\mathcal{X}|rank(X)\leqslant\Upsilon\}$ is closed, then the following statements hold:*

(1) The sequence $\{(X^k,Y^k)\}$ is bounded.
(2) Suppose that a subsequence $\{(X^k,Y^k)\}_{k\in\bar{K}}$ converges to (X^*,Y^*) and $rank(Y^k)=r$ for all $k\in\bar{K}$, where $r=rank(Y^*)$, then$\{(X^k,Y^k,U^k,V^k)\}_{k\in\bar{K}}$ is bounded.

Let $K\subseteq\bar{K}$ be a subsequence such that $\{(X^k,Y^k,U^k,V^k)\}_{k\in\bar{K}}$ converges to (X^*,Y^*,U^*,V^*). Assume that the Robinson condition holds at (X^*,Y^*,U^*,V^*) and $\{d_Y-d_UV^k-U^kd_V : d_U\in\mathbb{R}^{m\times r}, d_V\in\mathbb{R}^{r\times n},\ d_Y\in\mathcal{T}(Y^k)\}=\mathbb{R}^{m\times n}$ holds for sufficiently large $k\in K$. Then$\{u^k,Z_X^k\}_{k\in K}$ is bounded,and each accumulation point $\{u^*,Z_X^*\}$ of $\{u^k,Z_X^k\}_{k\in K}$ together with (X^*,Y^*,U^*,V^*) and some $Z_X^*\in\mathbb{R}^{m\times n}$ satisfies the first order optimality condition.

3. Numerical experiments

In this section, we report some numerical results to evaluate the performance of the PD-CG method (Algorithm 1). Firstly, we apply the PD-CG method to solve the problem (1) with different experimental settings. Then we compare it with some competitive algorithms for solving randomly generated matrix completion problems. Finally, we test the proposed algorithm's practical performance on recovering three gray images from their partial pixels.

All experiments are performed under Window 7 premium and MATLAB v7.8(2009a) running on a Lenovo laptop with an Intel core CPU at 2.4 GHz and 2 GB memory.

In all tests, we firstly generated matrices $M_L \in \mathbb{R}^{m\times r}$ and $M_R \in \mathbb{R}^{n\times r}$ with independent identically distributed Gaussian entries, and then set $M = M_L M_R^T$. It is clear that the rank of M is r. We use the partial discrete cosine transform (DCT) matrix as an encoder. Since the DCT matrix is implicity storied as fast transforms, this enables us to test problem more efficiently. In addition, we choose $X^{fea} = \mathcal{A}^*(b)$ and $\sigma = \sqrt{10}$, and set $\rho_k = \sigma * \rho_{k-1}$ with the initial penalty parameter $\rho_0 = 0.1$. We use r and p to denote the rank of original matrix M and the number of measurement respectively. Additionally, $sr = p/(mn)$ denotes the sampling ratio, and $dr = r(m+n-r)$ denotes the number of degree of freedom in a real-valued rank r matrix. As mentioned in [11,16], when the ratio p/dr is greater than 3, the problem can be viewed as an easy problem. On the contrary, it is considered to be a hard problem. Another radio is $FR = r(m+n-r)/p$, it is also important for successfully recovering the matrix M. If $FR > 1$, it is impossible to recover matrix with the given entries because there is an infinite number of matrices X with rank r [4]. Therefore the FR varies in $(0, 1)$ generally. In all tests, we use X^* to represent the optimal solution produced by the PD-CG method, and use relative error (RelErr) to measure the quality of X^* for original matrix M, i.e.

$$RelErr = \frac{\|X^* - M\|_F}{\|M\|_F}. \tag{34}$$

We say that M is recovered successfully by X^* if the corresponding $RelErr$ is less than 10^{-3}, which has been used in [4,5].

3.1. Test on affine rank minimization problems

In this subsection, we mainly test our PD-CG method for solving the problem (1). Firstly, we test it as r is increasing from 1 to 10. We run the code 50 times at each case. The test results are shown in Table 1, in which "NS" represents the number of matrices which are successfully recovered, and "ATime" shows the averaged CPU time in seconds. Observing the Table 1, it is clear that the PD-CG method can successfully solve these problems at each case. Moreover, it is worth noting that the PD-CG method requires more time as r increases.

Secondly, we test the performance of PD-CG method with different sr for problem (1). In this test, we fixed $r = 5$ and set $n = 100$ and $n = 200$. The test results are given in Table 2. It can be seen from the table that as the higher the sampling ratio is, the easier to be solved the problem becomes. At the same time, less time is required to attain a higher accuracy.

In the third test, we test the PD-CG method at the case of $m \neq n$. We choose m and n from 50 to 500, and set $sr = 0.5$. The test results are given in Table 3. From this table, we can see that all experimental examples are solved successfully and very

Table 1
Numerical results for problem (1) with different r ($m = n,\ sr = 0.5$).

(n, r)	dr	p/dr	FR	NS	ATime	RelErr
(50, 1)	99	12.6263	0.0792	50	0.6994	8.255e−007
(50, 2)	196	6.3776	0.1568	50	1.0061	6.895e−007
(50, 3)	291	4.2955	0.2328	50	1.4532	6.908e−007
(50, 4)	384	3.2552	0.3072	50	1.9956	7.072e−007
(50, 5)	475	2.6316	0.3800	50	2.4750	7.233e−007
(50, 6)	564	2.2163	0.4512	50	3.3209	7.937e−007
(50, 7)	651	1.9201	0.5208	50	4.5643	8.671e−007
(50, 8)	736	1.6984	0.5888	49	7.4500	1.034e−006
(50, 9)	819	1.5263	0.6552	42	12.2541	1.256e−006
(50, 10)	900	1.3889	0.7200	42	16.4418	1.337e−006

Table 2
Numerical results for problem (1) with different sr ($m = n,\ r = 5$).

(n, r)	sr	dr	p/dr	FR	Time	RelErr
(100, 5)	0.3	975	3.0769	0.3250	13.91	4.527e−007
(100, 5)	0.5	975	5.1282	0.1950	4.27	2.122e−007
(100, 5)	0.7	975	7.1795	0.1393	2.22	1.263e−007
(100, 5)	0.9	975	9.2308	0.1083	1.41	7.117e−008
(200, 5)	0.3	1975	6.0759	0.1646	36.93	1.549e−007
(200, 5)	0.5	1975	10.1266	0.0988	14.57	8.834e−008
(200, 5)	0.7	1975	14.1772	0.0705	8.89	4.737e−008
(200, 5)	0.9	1975	18.2278	0.0549	5.51	3.568e−008

Table 3
Numerical results for problem (1) with $m \neq n,\ sr = 0.5$.

(m, n)	r	p/dr	dr	FR	Time	RelErr
(50, 60)	5	2.8571	525	0.3500	2.22	6.038e–007
(60, 50)	5	2.8571	525	0.3500	2.21	5.465e–007
(100, 150)	10	3.1250	2400	0.3200	8.36	1.573e–007
(150, 100)	10	3.1250	2400	0.3200	9.68	1.857e–007
(200, 300)	15	4.1237	7275	0.2425	36.36	5.169e–008
(300, 200)	15	4.1237	7275	0.2425	33.31	5.355e–008
(400, 500)	20	5.6818	17600	0.1760	195.18	2.028e–008
(500, 400)	20	5.6818	17600	0.1760	171.56	1.881e–008

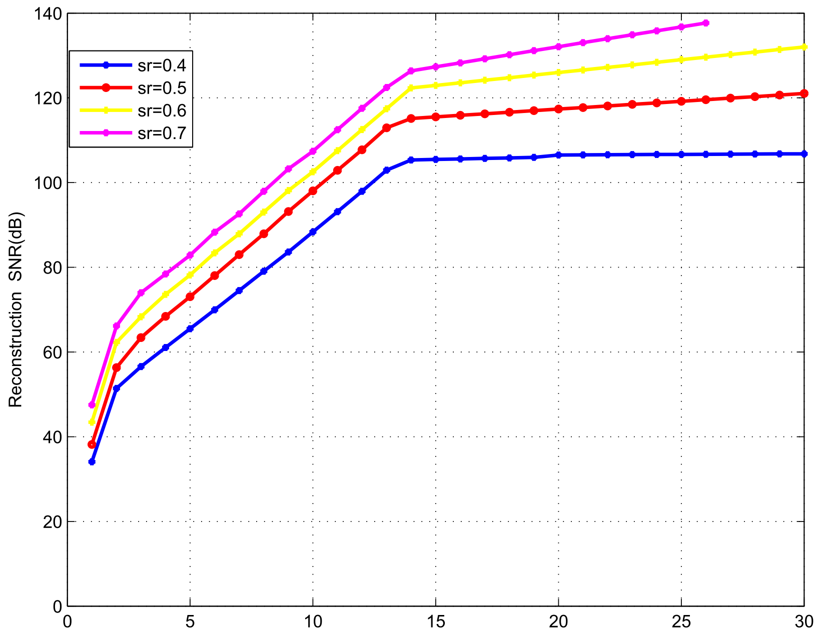

Fig. 1. $m = n = 100,\ r = 10,\ sr = 0.4$ and $sr = 0.5$ are hard problems, and the others are easy problems.

high accuracy of the optimal solutions are obtained. Therefore according to the above analysis, we can conclude that the PD-CG method is efficient and promising for solving the problem (1).

In order to further reveal the efficiency of the PD-CG method, we use the *SNR* (signal-to-noise ratio) to measure the accuracy of the optimal solution X^*. The *SNR* is defined as $SNR_{rec} = 20 * log_{10}(\|M\|_F / \|M - X^*\|_F)$ [16]. In this test, we use the PD-CG method to solve problem (1) with different r and sr. The statistics data are listed in Figs. 1 and 2 respectively, where the horizontal axis represents the index of ρ_k, and the vertical axis denotes SNR_{rec}. In Fig. 1, we set $m = n = 100$ and $r = 10$ with different sr. It can be seen that all SNR_{rec}s surpass 80 dB, which indicates that the propose algorithm performs well. In Fig. 2, we set $sr = 0.5$, and increase r monotonely from 5 to 20 with gap 5. We can clearly observe that the higher accuracy, i.e. more than 100 dB, can be obtained.

To end this subsection, we apply the PD-CG method to solve some hard affine rank minimization problems and compare it with IADM_NNLS[1] [8], which is proposed to solve the nuclear norm regularized least square problem

$$\min_{X \in \mathbb{R}^{m\times n}} \|X\|_* + \frac{1}{2\mu}\|\mathcal{A}(X) + b\|_2^2. \tag{35}$$

In running both codes, we set the parameters $maxit = 1000$ and $tol\_relchg = 1e-5$. Moreover, default values are used for other parameters in IADM_NNLS. As indicated in Table 4 that, the PD-CG method performs little better than IADM_NNLS in these problems. Moreover, we note that IADM_NNLS failed to get a higher accuracy within 1000 iterations and failed to attain a correct rank in some cases. In this table, the symbol "- - -" means that the method solved the corresponding problem unsuccessfully, and "rX" is the recovered rank by IADM_NNLS rather than the true rank. In addition, we can see that the algorithm for nuclear norm minimization is not very efficient when the rank of the matrix is not very low. Nevertheless, the

[1] The code is downloaded from http://math.nju.edu.cn/jfyang/IADM_NNLS/index.html.

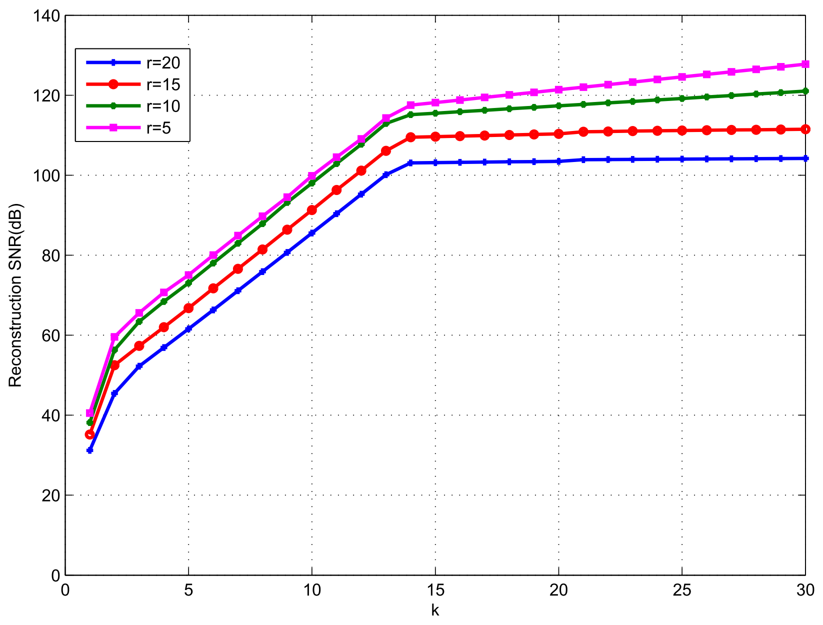

Fig. 2. $m = n = 100,\ sr = 0.5,\ r = 5$ is easy problem, and the others are hard problems.

Table 4
Comparison of PD-CG and IADM_NNLS for affine rank minimization, $m = n,\ sr = 0.5$.

(m, r)	dr	FR	p/dr	PD-CG		IADM_NNLS		
				Time	RelErr	Time	RelErr	rX
(50, 5)	475	0.38	2.6316	1.90	5.728e–006	18.88	1.68e–001	6
(100, 10)	1900	0.38	2.6316	5.53	8.619e–006	15.88	3.80e–002	
(100, 15)	2775	0.56	1.8018	10.17	8.554e–006	- - -	- - -	
(200, 20)	7600	0.38	2.6316	25.99	7.184e–006	40.84	5.01e–002	
(200, 25)	9375	0.47	2.133	33.05	8.640e–006	121.27	3.13e–001	35
(300, 30)	17100	0.38	2.6316	80.23	6.601e–006	100.04	3.84e–004	
(300, 35)	19775	0.44	2.2756	88.34	9.069e–006	162.22	1.59e–001	38
(400, 40)	30400	0.38	2.6316	172.48	7.985e–006	195.58	7.05e–004	
(400, 45)	33975	0.42	2.3547	185.19	9.221e–006	209.74	1.04e–001	
(500, 50)	47500	0.38	2.6316	307.35	6.547e–006	307.87	2.17e–004	
(500, 55)	51975	0.41	2.4050	356.38	7.463e–006	337.24	6.14e–002	

Table 5
Numerical results of PD-CG, FPCA, PD and FPC for matrix completion, $m = n, sr = 0.5$.

(m, r)	p/dr	PD-CG		FPCA		PD		FPC		
		Time	RelErr	Time	RelErr	Time	RelErr	Time	RelErr	rX
(50, 5)	2.632	2.78	5.98e–6	2.19	1.53e–5	0.60	2.43e–5	1.96	1.09e–3	8
(100,10)	2.632	4.27	9.59e–6	6.88	1.04e–5	1.74	1.76e–5	12.32	3.51e–4	11
(100, 15)	1.802	10.06	7.21e–6	8.03	1.90e–5	3.51	3.51e–5	105.4	1.05e–2	51
(200, 20)	2.632	11.95	5.77e–6	34.39	9.12e–6	8.19	1.53e–5	27.22	1.68e–4	
(200, 25)	2.133	19.37	9.50e–6	38.19	1.42e–5	12.46	2.22e–5	69.81	2.20e–4	
(300, 30)	2.632	31.79	6.66e–6	98.16	6.98e–6	27.20	2.01e–5	57.08	1.08e–4	
(300, 35)	2.276	44.85	8.18e–6	117.2	1.03e–5	58.98	2.00e–5	104.0	1.24e–4	
(400, 40)	2.632	64.01	7.98e–6	210.5	6.28e–6	82.21	1.89e–5	109.5	7.87e–5	
(400, 45)	2.355	85.11	9.61e–6	213.9	9.09e–6	104.1	2.16e–5	170.2	8.74e–5	46
(500, 50)	2.632	122.5	6.11e–6	390.3	6.13e–6	160.9	2.05e–5	211.7	6.40e–5	
(500, 55)	2.405	161.1	7.46e–6	442.7	8.22e–6	240.4	2.18e–5	335.4	6.82e–5	
(600, 60)	2.632	239.0	8.89e–6	626.6	5.98e–6	610.6	1.40e–5	430.8	5.26e–5	
(600, 65)	2.439	200.2	9.73e–6	627.7	7.73e–6	538.7	2.17e–5	546.7	5.53e–5	
(700, 70)	2.632	321.8	7.26e–6	997.5	6.19e–6	991.7	1.84e–5	700.3	4.47e–5	
(700, 75)	2.465	384.8	8.30e–6	984.9	7.52e–6	1293.9	2.02e–5	905.8	4.71e–5	
(800, 80)	2.632	487.2	5.98e–6	1502.9	6.50e–6	1588.8	1.83e–5	1102.1	3.91e–5	
(800, 85)	2.485	497.8	6.95e–6	1507.8	7.10e–6	2942.3	1.99e–5	1273.8	4.03e–5	
(900, 90)	2.632	632.5	9.80e–6	1948.3	5.10e–6	6134.2	1.94e–5	1468.5	3.45e–5	
(900, 95)	2.500	873.8	6.55e–6	2300.4	6.06e–6	3991.4	2.09e–5	1913.5	3.58e–5	

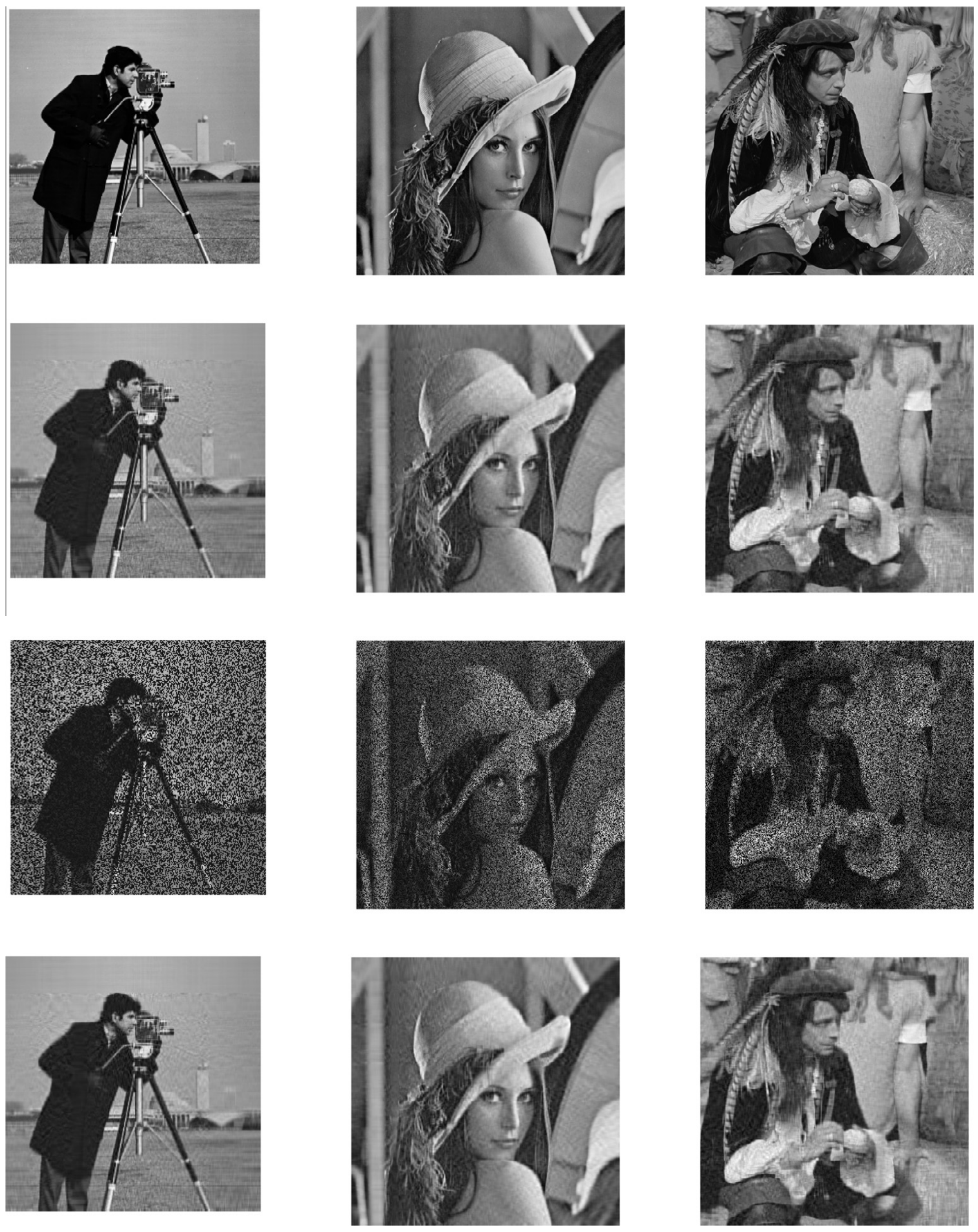

Fig. 3. Images with full rank (first line); Corresponding low rank images with $r = 40$ (second line); Randomly masked images from rank 40 with $sr = 50\%$ (third line); Recovered images by PD-CG method (last line).

IADM_NNLS method is truly very efficient for easy large scale low rank problems. Therefore we conclude from these numerical experiments that the PD-CG method is efficient and promising for solving the affine rank minimization problem.

3.2. Test on matrix completion problems

In this subsection, we report the efficiency of the PD-CG method for solving the matrix completion (MC) problem (2). Our first test is to compare the recovery ability of PD-CG with FPCA[2] [4], FPC[3] [4] and PD[4] [18]. Both of FPCA and FPC solve the following nuclear norm matrix completion problem

[2] The FPCA code is downloaded form http://www1.se.cuhk.edu.hk/sqma/FPCA.html.
[3] The FPC code is downloaded from http://svt.stanford.edu/code/.
[4] The PD code is downloaded from http://www.sfu.ca/yza30/homepage/PD_Rank/downloads.html.

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_*, \quad \text{s.t. } X_{i,j} = M_{i,j}, \quad \forall (i,j) \in \Omega. \tag{36}$$

Here we give the MATLAB codes $'\Omega = randsample(m*n,p)'$ and $'p = round(sr*m*n)'$ to generate the randan matrix $M_{i,j}$. In this test, we use PD and PD-CG methods to solve the problem (2). Comparing PD-CG with PD, we know that the main difference between both algorithms lies in solving the resulting subproblem on X per-iteration. More preciously, PD solves it by project method, and PD-CG uses a linear conjugate gradient method instead. In the first test, we choose some hard problems solved by the four algorithms with different m and r. Particularly, due to the increasing scale of problems and the expensive computing of singular value decomposition at each iteration, a welcome software PROPACK [22] is used to compute the singular values which is bigger than a threshold and corresponding vectors for the PD-CG method. As shown in [7] that, initializing $sv_0 = \min(m,n)/20$, we predict the number of singular values and vectors at each iteration as follows:

$$sv_{k+1} = \begin{cases} svp_k + 1, & \text{if } svp_k < sv_k, \\ svp_k + 5, & \text{if } svp_k = sv_k, \end{cases}$$

where svp_k denotes the number of positive singular values of X^k_{l+1}. In particular, we set the parameters $tol = 10^{-5}$, $mu\_final = 0.01$ and $maxiter = 500$ for FPC. Additionally, for FPCA and PD, default values are used for all parameters. From Table 5, we can see that PD-CG requires less time to attain higher level of accuracy when comparing with FPCA and FPC. It thus conclude that PD-CG performs better than FPCA and FPC in these tested problems. In the meantime, comparing with PD, PD-CG performs better at the case of $m > 300$. In addition, we can see that the FPC method is not so effective for solving hard problems because it can not produce the true rank of original matrix such as shown in the last column of Table 5. Overall, PD-CG method is very comparable with the other three method for solving the MC problems on these limited experiments.

To further show the superiority of the PD-CG method, we test it for recovering three real gray images. These images are widely used to test algorithms' performance in many simulations. In this test, the size of 'cameraman' is $256*256$, and the sizes of others are $512*512$. We firstly get the low rank-40 images by using the singular value decomposition (SVD) to each original images, which are shown in the second line of Fig. 3. Then we randomly select 50% samples from each resulting low rank images and recover the missed pixels via solving the matrix completion model. In addition, the relative error form left to right in the Fig. 3 is $RelErr = 4.856e-004$, $RelErr = 8.763e-005$, $RelErr = 3.420e-005$ respectively. From the last two lines in this figure, we visibly see that all the images are successfully recovered.

4. Conclusions

In this paper, we proposed a PD-CG method for solving the affine rank minimization problem directly. The proposed method mainly minimized the quadratic penalty function of the problem (1) by alternatively solving the resulting penalty subproblems on X and Y. The subproblem on X can be effectively solved by using the linear conjugate gradient method, and the other subproblem has a closed-form solution. Numerical experiments on the affine rank minimization problem with different sizes, ranks, and sampling ratios have illustrated that the proposed PD-CG method has well performance. The accuracy measure SNR_{rec} has further demonstrated that the proposed method is efficient. Moreover, numerical comparison with IADM_NNLS shows that solving the affine rank minimization may produce more truthful rank than solving the nuclear norm minimization. Finally, the comparison with three state-of-the-art methods and the recovery of some corrupted images have further illustrated that the PD-CG method is promising and practical.

Acknowledgment

This work is supported by the Natural Science Foundation of China Grant NSFC-71171150 and NSFC-11471101.

References

[1] M. Fazel, H. Hindi, S.P. Boyd, A rank minimization heuristic with application to minimum order system approximation, Proceedings of the American Control Conference, 6, IEEE, 2001, pp. 4734–4739.
[2] Z. Liu, L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM J. Matrix Anal. Appl. 31 (2009) 1235–1256.
[3] A. Rakotomamonjy, R. Flamary, G. Gasso, S. Canu, Lp-Lq penalty for sparse linear and sparse multiple kernel multitask learning, IEEE Trans. Neural Netw. 22 (2011) 1307–1320.
[4] S. Ma, D. Goldfarb, L. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, Math. Programming 128 (2011) 321–353.
[5] J.F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (2010) 1956–1982.
[6] E.T. Hale, W. Yin, Y. Zhang, Fixed-point continuation for ℓ_1-minimization: methodology and convergence, SIAM J. Optim. 19 (2008) 1107–1130.
[7] K.C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific J. Optim. 6 (2010) 615–640.
[8] J. Yang, X.M. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, Math. Comput. 82 (2013) 301–329.
[9] Y.H. Xiao, Z.-F. Jin, An alternating direction method for linear-constrained matrix nuclear norm minimization, Numer. Linear Algebra Appl. 19 (2012) 541–554.
[10] J. Barzilai, J.M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141–148.
[11] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (2009) 717–772.

[12] B. Recht, M. Fazel, P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (2010) 471–501.
[13] S.L. Zhou, N.H. Xiu, Z.Y. Luo, et al, Sparse and low-rank covariance matrix estimation, J. Oper. Res. Soc. China (2014) 1–20.
[14] K. Lee, Y. Bresler, Admira: atomic decomposition for minimum rank approximation, IEEE Trans. Inf. Theory 56 (2010) 4402–4416.
[15] P. Jain, R. Meka, I.S. Dhillon, Guaranteed rank minimization via singular value projection, NIPS 23 (2010) 937–945.
[16] M. Malek-Mohammadi, M. Babaie-Zadeh, A. Amini, C. Jutten, Recovery of low rank matrices under affine constrain via a smoothed rank function, IEEE Trans. Signal Processing 62 (2014) 981–992.
[17] Z. Lu, Y. Zhang, Penalty decomposition methods for rank minimization, Optim. Methods Softw. (2014) 1–28 [Epub ahead of print].
[18] Z. Wen, D. Goldfarb, K. Scheinberg, Block coordinate descent methods for semidefinite programming, Int. Ser. Oper. Res. Manage. Sci. 166 (2012) 533–564.
[19] Z. Lu, Y. Zhang, Sparse approximation via penalty decomposition methods, SIAM J. Optim. 23 (2013) 2448–2478.
[20] T. Blumensath, M.E. Davies, Iterative thresholding for sparse approximations, J. Fourier Anal. Appl. 14 (2008) 629–654.
[21] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.
[22] R.M. Larsen, PROPACK-Software for large and sparse SVD calculations, 2004. http://sun.stanford.edu/rmunk/PROPACK/
[23] C. Chen, B. He, X. Yuan, Matrix completion via an alternating direction method, IMA. J. Numer. Anal. 32 (2012) 227–245.
[24] Z.-F. Jin, Q . Wang, Z. Wan, Recovering low-rank matrices from corrupted observations via the linear conjugate gradient algorithm, J. Comput. Appl. Math. 256 (2014) 114–120.