HOMOTOPY METHOD FOR MATRIX RANK MINIMIZATION BASED ON THE MATRIX HARD THRESHOLDING METHOD

ZHENGSHAN DONG
Ruijie Networks Co., Ltd, Fuzhou 350108, China
Center for Discrete Mathematics and Theoretical Computer Science
Fuzhou University, Fuzhou 350108, China

JIANLI CHEN AND WENXING ZHU
Center for Discrete Mathematics and Theoretical Computer Science
Fuzhou University, Fuzhou 350108, China

(Communicated by Wanquan Liu)

Abstract. Based on the matrix hard thresholding method, a homotopy method is proposed for solving the matrix rank minimization problem. This method iteratively solves a series of regularization subproblems, whose solutions are given in closed form by the matrix hard thresholding operator. Under some mild assumptions, convergence of the proposed method is proved. The proposed method does not depend on a prior knowledge of exact rank value. Numerical experiments demonstrate that the proposed homotopy method weakens the affection of the choice of the regularization parameter, and is more efficient and effective than the existing state-of-the-art methods.

1. Introduction. In this paper, we consider recovering a low rank matrix under a part of linear measurements. Generally, undersampling makes it impossible to find an accurate matrix. However, the information that the matrix has a low rank structure makes it be recovered. So the interesting problem could be formed as the matrix rank minimization problem

\[
\min_X \text{rank}(X) \quad s.t. \quad A(X) = y,
\]

where \( X \in \mathbb{R}^{m \times n} \) is an unknown variable, \( A \) is a linear map: \( \mathbb{R}^{m \times n} \to \mathbb{R}^q \) \((q \ll mn \text{ and rank}(X) \ll \min\{m, n\})\), and \( y \in \mathbb{R}^q \) is the problem data. Since this problem extracts low dimensional structures from high dimensional data, it has many applications in different fields, such as system identification [7, 14, 19], optimal control [6], and signal processing [21, 24]. Moreover, the problem becomes especially interesting when it has increasingly massive data.

A special case of problem (1) is the matrix completion problem,

\[
\min_X \text{rank}(X) \quad s.t. \quad X_{ij} = M_{ij}, \forall (i, j) \in \Omega.
\]
where $X$ and $M$ are in $\mathbb{R}^{m \times n}$, and $\Omega$ is the subset of indices for the known entries. This problem has applications in online recommendation systems, collaborative filtering, image recovery, data mining, etc. One famous problem is the Netflix problem, which is about movie ratings. In the problem, it is impossible for all users to rate all movies, but we must assess the unknown ratings basing on the known ratings of users.

Although the general matrix rank minimization problems (1) and (2) are NP-hard [10, 21], a large number of methods have been proposed. One popular strategy is relaxing the objective function of the problem, by approximating the rank function $\text{rank}(X)$ using some surrogate functions, such as the nuclear norm $\|X\|_*$ and the $L_p$ norm ($0 < p < 1$). The nuclear norm is defined as the sum of the nonzero singular values of $X$, which is the tightest convex relaxation of the rank function. By the nuclear norm, problem (1) is relaxed as

$$\min_X \|X\|_* \quad \text{s.t.} \quad A(X) = y, \quad (3)$$

and its regularization version is

$$\min_X \frac{1}{2}\|A(X) - y\|_2^2 + \lambda\|X\|_* \quad (4)$$

In problem (4), $\lambda > 0$ is a regularization parameter which balances the sampling error and the nuclear norm, and $\|\cdot\|_2$ denotes the standard Euclidean norm.

Several algorithms have been proposed for solving problems (3) and (4). Liu et al. [14] proposed interior-point methods to solve a semidefinite programming version of problem (3). However, their methods could not be applied to large scale problems. Then some first-order methods were proposed, including the Fixed Point Continuation (FPC) method [8, 18], and the iteratively reweighted nuclear norm algorithms [15, 19, 20].

Amounts of work from the theoretical aspect shows that, under some assumptions, the nuclear norm minimization (3) could solve the matrix rank minimization problem (1) with high probability [4, 5, 21]. However, the corresponding methods solve the relaxation problem and are restricted to those assumptions to guarantee effective. Hence, there have been lots of methods [1, 3, 9, 11, 12, 13, 17, 22, 23] solving the low rank problem without the nuclear norm relaxation. In fact, most of them solve the rank-constrained problem

$$\min_{X \in \mathbb{R}^{m \times n}} \frac{1}{2}\|A(X) - y\|_2^2, \quad (5)$$

where $C_r = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) \leq r\}$. That is to say, the exact rank $r$ must be known prior. However, in some cases, we just know that the matrix is low rank, but do not know the exact rank value. Hence we will consider in this paper how to solve the low rank minimization problem without knowing the rank value.

In [17], the Penalty Decomposition methods (PD) were proposed for the general matrix rank minimization problems with the rank function appearing either in the constraints or in the objective function. In the latter case, the PD method solves the low rank minimization problem (1) without knowing the rank value.

To the best of our knowledge, except the PD method, there is no other method which does not need to know the rank value prior and solves the low rank minimization problem using the rank function directly. In this paper, we propose a homotopy method (denote it as HIHT) for the low rank minimization problem (1)
by solving a series of regularization problem, which is
\[
\min_X \frac{1}{2} \|A(X) - y\|_2^2 + \lambda \text{rank}(X),
\]
where \(\lambda > 0\) is a regularization parameter. This method does not require a rank value prior. Though problem (6) is not easy to solve, we could linearize the first term and turn to iteratively solve some subproblems, which have closed form solutions based on the matrix hard thresholding operator (for more details refer to Section 2). For the regularization parameter \(\lambda\), we adopt the homotopy technique to avoid the difficulty of choosing a suitable value of \(\lambda\). Under some mild assumptions, convergence of the proposed method is proved. Experimental results show that our homotopy method is more efficient and effective than the existing state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 is the preliminary, in which some notations and the IHT method for the low rank matrix minimization are described. Section 3 presents the homotopy method for the matrix rank minimization problem (1), which combines the homotopy technique with the IHT method. Moreover, convergence analyses are given. In Section 4, some practical issues are considered. Numerical experiments are shown in Section 5 and conclusions are made in Section 6, respectively.

2. Preliminaries.

2.1. Notations. In this subsection, some notations are presented to simplify presentation. Let \(X \in \mathbb{R}^{m \times n}\). \(r = \text{rank}(X)\) denotes the rank of matrix \(X\). \(\sigma(X)\) denotes the singular values of \(X\) with decreasing order, where \(\sigma_i(X)\) is the \(i\)-th largest singular value. The spectral norm \(\|X\|_2\) is the largest singular value of matrix \(X\), i.e., \(\|X\|_2 = \max_i \sigma_i(X) = \sigma_1(X)\), and the Frobenious norm of matrix \(X\) is \(\|X\|_F = (\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}^2)^{1/2}\). \(A^*(\cdot)\) denotes the adjoint operator of \(A(\cdot)\). The transpose of matrix \(X\) is denoted by \(X^T\). For a square matrix \(X\), its trace is defined by the sum of its diagonal entries. The inner product of two matrices \(X\) and \(Y\) is denoted by \((X, Y) = Tr(X^T Y) = Tr(Y^T X)\).

2.2. Iterative hard thresholding method. In this subsection, the iterative hard thresholding (IHT) method with singular value decomposition is presented, and the convergence of the method is proved.

We consider the problem
\[
\min_X \phi_\lambda(X) = f(X) + \lambda \text{rank}(X),
\]
where \(f(X) = \frac{1}{2} \|A(X) - y\|_2^2\) and its gradient is Lipschitz continuous (denote the Lipschitz constant as \(L_f\)). To solve this problem, we approximate the first term by linearizing \(f(X)\) at the current solution \(X^k\), and replace the objective function by
\[
P_{L,\lambda}(X^k, X) = f(X^k) + \langle \nabla f(X^k), X - X^k \rangle + \frac{L}{2} \|X - X^k\|_F^2 + \lambda \text{rank}(X),
\]
where the quadratic term is the proximal term, \(L > 0\) is a constant. Then problem (7) can be approximated by
\[
\min_X \frac{1}{2} \|X - (X^k - \frac{1}{L} \nabla f(X^k))\|_F^2 + \frac{\lambda}{L} \text{rank}(X).
\]
The above problem has a closed form solution based on the following lemmas (refer to Corollary 2.2 in [17]).

**Lemma 2.1.** [17] Let $\alpha > 0$, and $Z \in \mathbb{R}^{m \times n}$ be given, and let $t = \min(m, n)$. Suppose $U \Sigma(Z) V^T$ is the singular value decomposition of $Z$. Then, $X^* = U D(x^*) V^T$ is an optimal solution of the problem

$$
\min_X \frac{1}{2} \|X - Z\|_F^2 + \alpha \text{rank}(X),
$$

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

$$
\min_x \frac{1}{2} \|x - \sigma(Z)\|_2^2 + \alpha \|x\|_0,
$$

and $D(\cdot)$ transforms a vector to a diagonal matrix with the vector as its diagonal entries.

**Lemma 2.2.** [2, 16] The solution of problem (11) is given by the hard thresholding operator

$$
x^*_i = [H_\alpha(\sigma(Z))]_i = \begin{cases} 
\sigma(Z)_i, & \text{if } [\sigma(Z)_i]^2 > 2\alpha; \\
0, & \text{otherwise.}
\end{cases}
$$

where $[\cdot]_i$ denotes the $i$-th component of the vector.

In fact, the above two lemmas present the matrix hard thresholding operator for problem (10):

$$
X^* = H_\alpha(Z) = U D(H_\alpha(\sigma(Z))) V^T.
$$

For simplification, we denote $H_\alpha(\cdot)$ as the hard thresholding operator on a vector or a matrix, which is defined by (12) and (13). Then we adopt the iterative hard thresholding method to solve problem (7). The frame is presented as Algorithm 1.

---

**Algorithm 1:** $X^* \leftarrow \text{IHT}(L, \lambda, X^0)$

**Input:** $L > L_f$, $\lambda$, $X^0$;

**Output:** $X^*$;

1: initialization $k \leftarrow 0$;
2: repeat
3: $Z^{k+1} \leftarrow X^k - \frac{1}{\lambda} \nabla f(X^k)$;
4: $X^{k+1} \leftarrow H_{2\lambda/L}(Z^{k+1})$;
5: until some termination condition reaches
6: $X^* \leftarrow X^k$.

---

**Remark 1.** The iterative hard thresholding method proposed in Algorithm 1 is different from the one proposed in [8]. In [8], the iterative hard operator is a rank-$r$ projection on the set $C_r$. More precisely, it just keeps the $r$ largest singular values and sets others as zeros. However, in Algorithm 1, the corresponding operator depends on a thresholding value, which controls the rank value. Hence we need no information about the exact rank value.

Next, we show the convergence and some properties of Algorithm 1, which will be used later.
Theorem 2.3. Let \( \{X^k\} \) be the sequence generated by Algorithm 1. If \( L > L_f \), then the following statements hold:

(i) \( \{P_{L, \lambda}(X^k, X^{k+1})\} \) is nonincreasing, and if \( X^k \neq X^{k-1} \), then \( P_{L, \lambda}(X^k, X^{k+1}) < P_{L, \lambda}(X^{k-1}, X^k) \);

(ii) \( \{\phi_\lambda(X^k)\} \) converges and \( \{||X^k - X^{k+1}||_F^2\} \) converges to 0. Indeed,

\[
\phi_\lambda(X^k) - \phi_\lambda(X^{k+1}) \geq \frac{L - L_f}{2} ||X^{k+1} - X^k||_F^2;
\]

(14)

(iii) \( \text{rank}(X^k) \) changes only in a finite number of steps.

Proof. (i) By the iteration process of Algorithm 1, we have

\[
P_{L, \lambda}(X^k, X^{k+1}) = f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L}{2} ||X^{k+1} - X^k||_F^2 + \lambda \text{rank}(X^{k+1})
\]

\[
\leq f(X^k) + \lambda \text{rank}(X^k)
\]

\[
\leq f(X^{k-1}) + \langle \nabla f(X^{k-1}), X^k - X^{k-1} \rangle + \frac{L_f}{2} ||X^k - X^{k-1}||_F^2 + \lambda \text{rank}(X^k)
\]

\[
\leq f(X^{k-1}) + \langle \nabla f(X^{k-1}), X^k - X^{k-1} \rangle + \frac{L_f}{2} ||X^{k-1} - X^{k-1}||_F^2 + \lambda \text{rank}(X^k)
\]

\[
= P_{L, \lambda}(X^{k-1}, X^k).
\]

In the above equations, the second inequality follows from that \( \nabla f \) is Lipschitz continuous and the Lipschitz constant is \( L_f \); the third inequality follows from the assumption that \( L > L_f \). Thus, \( P_{L, \lambda}(X^k, X^{k+1}) \) is nonincreasing. Moreover, if \( X^k \neq X^{k-1} \), then the above third inequality is strict and \( P_{L, \lambda}(X^k, X^{k+1}) < P_{L, \lambda}(X^{k-1}, X^k) \), since \( L > L_f \).

(ii) By the iteration process of Algorithm 1 and \( L > L_f \), we also have

\[
\phi_\lambda(X^k) = f(X^k) + \lambda \text{rank}(X^k)
\]

\[
\geq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L}{2} ||X^{k+1} - X^k||_F^2 + \lambda \text{rank}(X^{k+1})
\]

\[
\geq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L_f}{2} ||X^{k+1} - X^k||_F^2 + \lambda \text{rank}(X^{k+1})
\]

\[
\geq f(X^{k+1}) + \lambda \text{rank}(X^{k+1})
\]

\[
= \phi_\lambda(X^{k+1}),
\]

(15)

where the last inequality follows from that \( \nabla f \) is Lipschitz continuous. Thus \( \{\phi_\lambda(X^k)\} \) is nonincreasing and if \( X^k \neq X^{k+1} \) then it is strictly decreasing. Furthermore, \( \phi_\lambda(X) \) is bounded below for \( \lambda \geq 0 \). Therefore, \( \{\phi_\lambda(X^k)\} \) converges for \( k \to \infty \).

Inequalities (15) also imply that

\[
\phi_\lambda(X^k) - \phi_\lambda(X^{k+1}) \geq \frac{L - L_f}{2} ||X^{k+1} - X^k||_F^2,
\]

(16)

which together with the convergence of \( \{\phi_\lambda(X^k)\} \) and \( L > L_f \) imply that \( \{||X^k - X^{k+1}||_F^2\} \) converges to 0.

(iii) Without loss of generality, suppose that \( \lambda > 0 \). If \( \text{rank}(X^k) \) still changes when \( k \) is large enough, then \( |\text{rank}(X^k) - \text{rank}(X^{k+1})| \geq 1 \). However, the convergence of \( \{X^k\} \) and the continuity of \( f(X^k) \) imply that, for any \( \varepsilon > 0 \) such that
$\varepsilon < \lambda$, then $|f(X^k) - f(X^{k+1})| < \varepsilon$ when $k$ is large enough. Furthermore,

$$
|\phi_\lambda(X^k) - \phi_\lambda(X^{k+1})|
\geq \lambda|\text{rank}(X^k) - \text{rank}(X^{k+1})| - |f(X^k) - f(X^{k+1})|
> \lambda - \varepsilon,
$$

which contradicts the convergence of $\{\phi_\lambda(X^k)\}$. Hence, $\text{rank}(X^k)$ changes only in a finite number of steps.

From Theorem 2.3.(ii), we can see that if $X^{k+1} = X^k$, then all the following solutions generated by Algorithm 1 are equal to $X^k$, and Algorithm 1 will terminate. Hence, by (14) the sequence $\{\phi_\lambda(X^k)\}$ is strictly decreasing.

3. **Homotopy method.** In the previous section, we convert problem (1) to the regularization problem (7), which is tackled by iteratively solving problem (9). Since the regularization parameter $\lambda$ balances the rank of $X$ and the reconstruction error, a suitable value of $\lambda$ could produce a better solution. However, finding a good value of $\lambda$ is still open. In this section, we adopt the homotopy technique to weaken the affection of the regularization parameter, and efficiently calculate and trace the solutions of the regularized matrix rank problem along a homotopy path.

Based on the hard thresholding operator with the singular value decomposition, the core idea of our homotopy method is as follows. Set a large initial value of the regularization parameter $\lambda$ and gradually decrease it with some strategy. More precisely, the decreasing speed of $\lambda$ could be geometrical: for an initial value $\lambda_0$ and a parameter $\rho \in (0, 1)$, set $\lambda_{k+1} = \rho \lambda_k$ for $k = 0, 1, 2, \cdots$. For every fixed value of the regularization parameter $\lambda$, Algorithm 1 is used to find an approximate optimal solution, and then use it as the initial solution for the next iteration.

The framework of the proposed Homotopy Iterative Hard Thresholding (HIHT) algorithm is described as Algorithm 2.

**Algorithm 2:** $X^* \leftarrow HIHT(L, \lambda_0, X^0)$

**Input:** $L$, $\lambda_0$, $X^0$;

**Output:** $X^*$;

1. initialization $k \leftarrow 0$, $\rho \in (0, 1)$;

2. repeat

3. $i \leftarrow 0$;

4. $X^{k,0} = X^k$;

5. repeat

6. $Z_{k,i+1} \leftarrow X^{k,i} - \frac{1}{L} \nabla f(X^{k,i})$;

7. $X_{k,i+1} \leftarrow H_{2\lambda_k/L}(Z_{k,i+1})$;

8. until rank$(X^{k,i})$ does not change when increasing $i$

9. $X^{k+1} \leftarrow X^{k,i}$;

10. $\lambda_{k+1} \leftarrow \rho \lambda_k$;

11. $k \leftarrow k + 1$;

12. until some termination condition reaches

13. $X^* \leftarrow X^k$.

**Remark 2.** According to (iii) of Theorem 2.3, the iteration between lines 5-8 of Algorithm 2 terminates in a finite number of steps. Let $n_k$ be the number of
iterations between lines 5-8 at the $k$-th outer iteration. Then for each $\lambda_k$, when $i \geq n_k$, $\text{rank}(X^{k,i}) = \text{rank}(X^{k,n_k})$ and
\[
\text{rank}(X^{k,n_k}) \neq \text{rank}(X^{k,n_k-1}).
\]
However, the condition used in line 8 of Algorithm 2 is not easy to verify, since at some $i$ $\text{rank}(X^{k,i}) = \text{rank}(X^{k,i+1})$ does not indicate that $\text{rank}(X^{k,i})$ will not change forever. We will consider some practical termination conditions in Section 4.

The convergence analysis of Algorithm 2 is presented as the following theorem. The proof is based on some properties of IHT in Section 2.

**Theorem 3.1.** Let $\{X^k\}$ be the sequence generated by Algorithm 2. If $L > L_f$, then the following statements hold:

(i) $\{\phi_{\lambda_k}(X^k)\}$ is strictly decreasing and convergent. Indeed,
\[
\phi_{\lambda_k}(X^k) - \phi_{\lambda_{k+1}}(X^{k+1}) \geq \lambda_k (1 - \rho) \text{rank}(X^{k+1}) > 0;
\]
(ii) $\lim_{k \to \infty} \|X^{k,i} - X^{k,i+1}\|_F^2 = 0$;
(iii) if $f(X)$ is strongly convex with modulus $\beta$, then $\{X^k\}$ converges.

**Proof.** (i) Firstly, for each fixed $\lambda_k$, by (iii) of Theorem 2.3, lines 5-8 in Algorithm 2 will terminate in a finite number of iterations. Furthermore, by (i) of Theorem 2.3 and the choice of $X^{k,i}$ from Algorithm 2, we can obtain that
\[
\phi_{\lambda_k}(X^k) = f(X^k) + \lambda_k \text{rank}(X^k)
\geq f(X^k) + \langle \nabla f(X^k), X_{\mu}^1 - x^k \rangle + \frac{L}{2} \|X^k_{\mu} - X^k\|_F^2 + \lambda_k \text{rank}(X^{k,1})
= f(X^{k,0}) + \langle \nabla f(X^{k,0}), X_{\mu}^1 - X^{k,0} \rangle + \frac{L}{2} \|X^k_{\mu} - X^{k,0}\|_F^2 + \lambda_k \text{rank}(X^{k,1})
\geq \cdots
\geq f(X^{k,n_k-1}) + \langle \nabla f(X^{k,n_k-1}), X^{k+1} - X^{k,n_k-1} \rangle + \frac{L}{2} \|X^{k+1} - X^{k,n_k-1}\|_F^2 + \lambda_k \text{rank}(X^{k+1}),
\]
where $n_k$ is the number of iterations between lines 5-8 at the $k$-th outer iteration.

On the other hand, since $\nabla f$ is Lipschitz continuous, one can observe that
\[
\phi_{\lambda_{k+1}}(X^{k+1}) = f(X^{k+1}) + \lambda_{k+1} \text{rank}(X^{k+1})
\leq f(X^{k,n_k-1}) + \langle \nabla f(X^{k,n_k-1}), X^{k+1} - X^{k,n_k-1} \rangle + \frac{L_f}{2} \|X^{k+1} - X^{k,n_k-1}\|_F^2 + \lambda_{k+1} \text{rank}(X^{k+1}),
\]
which together with (17), $X^{k+1} = X^{k,n_k}$, and $L > L_f$ imply that
\[
\phi_{\lambda_k}(X^{k}) - \phi_{\lambda_{k+1}}(X^{k+1})
\geq \frac{L - L_f}{2} \|X^{k+1} - X^{k,n_k-1}\|_F^2 + (\lambda_k - \lambda_{k+1}) \text{rank}(X^{k+1})
\geq \frac{L - L_f}{2} \|X^{k+1} - X^{k,n_k-1}\|_F^2 + \lambda_k (1 - \rho) \text{rank}(X^{k+1})
\geq \lambda_k (1 - \rho) \text{rank}(X^{k+1}) > 0.
\]
Hence $\{\phi_{\lambda_k}(X^k)\}$ is strictly decreasing. Furthermore, $\phi_{\lambda}(X)$ is lower bounded for $\lambda > 0$. Therefore $\{\phi_{\lambda_k}(X^k)\}$ converges for $k \to \infty$.\]
(ii) By the proof of Theorem 2.3, i.e., for every fixed $\lambda_k$, $\{\phi_{\lambda_k}(X^{k,i})\}$ is nonincreasing, the convergence of $\{\phi_{\lambda_k}(X^k)\}$ and nonincreasing $\lambda_k$, one can observe that the sequence

$$
\phi_{\lambda_0}(X^0) \to \phi_{\lambda_0}(X^{0,0}), \phi_{\lambda_0}(X^{0,1}), \ldots, \phi_{\lambda_0}(X^{0,n_0}) \to \phi_{\lambda_0}(X^1) \\
\phi_{\lambda_1}(X^1) \to \phi_{\lambda_1}(X^{1,0}), \phi_{\lambda_1}(X^{1,1}), \ldots, \phi_{\lambda_1}(X^{1,n_1}) \to \phi_{\lambda_1}(X^2) \\
\ldots \\
\phi_{\lambda_k}(X^k) \to \phi_{\lambda_k}(X^{k,0}), \phi_{\lambda_k}(X^{k,1}), \ldots, \phi_{\lambda_k}(X^{k,n_k}) \to \phi_{\lambda_k}(X^{k+1}) \\
\ldots
$$

is nonincreasing and converges.

Next, by $\phi_{\lambda_k}(X^k) - \phi_{\lambda_{k+1}}(X^{k+1}) \to 0$ when $k \to \infty$, and the second inequality of (18), one can get that

$$
\|X^{k+1} - X^{k,n_k-1}\|_F = \|X^{k,n_k} - X^{k,n_k-1}\|_F \to 0, \quad \text{when } k \to \infty.
$$

Similarly, $\|X^{k,i+1} - X^{k,i}\|_F \to 0$, when $k \to \infty$, $i = 0, 1, 2, \ldots, n_k - 1$.

(iii) Denote

$$
F := \{X : \mathcal{A}(X) = y, \|X\|_F^2 < \infty\}.
$$

Then $F$ is bounded, and for $\tilde{X} \in F$,

$$
f(\tilde{X}) = 0, \nabla f(\tilde{X}) = \mathcal{A}^*(\mathcal{A}(\tilde{X}) - y) = 0. \tag{19}
$$

From Theorem 3.1.(i), we know that the sequence $\{\phi_{\lambda_k}(X^k)\}$ is strictly decreasing, which together with the strong convex of $f(X)$ implies that

$$
\phi_{\lambda_0}(X^0) \geq \phi_{\lambda_k}(X^k) = f(X^k) + \lambda_k \text{rank}(X^k) \\
\geq f(X) + \langle \nabla f(X), X^k - X \rangle + \frac{\beta}{2} \|X^k - X\|_F^2 + \lambda_k \text{rank}(X^k)
$$

for all $X \in \mathbb{R}^{m \times n}$ and $k > 0$. Substituting $X = \tilde{X}$ into the above inequalities and using (19), we obtain

$$
\phi_{\lambda_0}(X^0) \geq \frac{\beta}{2} \|X^k - \tilde{X}\|_F^2 + \lambda_k \text{rank}(X^k)
$$

for all $k > 0$. Hence the sequence $\{X^k\}$ is bounded, which together with Theorem 3.1.(ii) and the finiteness of $n_k$ implies that the sequence $\{X^k\}$ is convergent. \qed

**Remark 3.** (i) From the proof of Theorem 3.1, it is not difficult to see that the entire sequences $\{X^{k,i}\}$ and $\{\phi_{\lambda_k}(X^{k,i})\}$ converge.

(ii) Note that we could reformulate problem (7) equivalently such that $\frac{1}{\lambda}$ is a penalty parameter on $f(X)$. Then by the theory of the penalty function method, any limit point of $\{X^{k,i}\}$ is a feasible solution of problem (1), i.e., if $X^* = \lim_{k \to \infty} X^{k,i}$, then $\mathcal{A}(X^*) = y$.

4. **Practical issues.** Since the matrix rank minimization problem is always large scale in practice, finding an upper bound on $L_f$ is difficult. In this section, some issues will be considered, such as the value of $L$ and the termination conditions.
4.1. Upper bound on $L_f$. One simple choice is to set an upper bound on $L_f$ as a large constant value. But this strategy may cause the algorithm converging slow or converging to a bad solution. An alternative method is setting $L$ by a linear search: initializing $L$ as a suitable value and then increasing it until it satisfies

$$f(X^{k+1}) \leq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L}{2} ||X^{k+1} - X^k||_F^2,$$

or

$$\phi_\lambda(X^k) - \phi_\lambda(X^{k+1}) \geq \frac{\varrho}{2} ||X^{k+1} - X^k||_F^2.$$

The first condition is based on the definition of $L_f$ and $L > L_f$. The second one is based on (16) with $\varrho \geq L - L_f$ (similar strategy can refer to [16]).

For solving the matrix completion problem, $L$ could be $1 + \delta$ ($\delta \geq 0$). This is by the reason that the corresponding sampling matrix $A$ of $A$ is a projection matrix and satisfies $||AT||_2 = 1$.

4.2. Termination conditions. In the previous sections, we have proved that the sequence $\{X^k\}$ generated by our method is convergent. Hence, we could terminate our method when

$$\frac{||X^{k+1} - X^k||_F}{\max\{1, ||X^k||_F\}} < \epsilon$$

reaches for a small number $\epsilon > 0$.

For Algorithm 2, we stop the algorithm when $\lambda$ reaches a small given value $\lambda_t$. For the inner termination condition (line 8) of Algorithm 2, it may not be easily and exactly detected. Since Algorithm 1 converges for each fixed $\lambda$, we may use

$$\frac{||X^{k,i+1} - X^{k,i}||_F}{\max\{1, ||X^{k,i}||_F\}} < \epsilon_{in}$$

as the inner stopping criterion for a small number $\epsilon_{in} > 0$. In addition, from the proof of Theorem 3.1, we can find that the number of inner iterations can be fixed as some constant and Theorem 3.1 still holds. More precisely, Algorithm 2 is still convergent. Hence, we will stop the inner loop between lines 5-8 of Algorithm 2 when the number of inner iterations reaches $step$ or the criterion (21) is satisfied in our later computational experiments.

5. Computational experiments. Since most of algorithms for matrix rank minimization are tested on the matrix completion problem, in this section, we also test the performances of our algorithms on the matrix completion problem. In subsection 5.1, we show the influence of some parameters on our methods. In subsection 5.2, we compare our methods with two state-of-the-art algorithms, i.e., the FPCA method [8, 18] and the PD method [17]. All experiments are performed on a personal computer with an Intel(R) Core(TM)2 Duo CPU T6600 (2.20GHz) and 3GB memory, using a MATLAB toolbox (version R2012b).

We generate test data as follows. The matrix $M \in \mathbb{R}^{m \times n}$ with rank $r$ is created by the product of two random Gaussian or uniform matrices, i.e., $M = M_L M_R^T$, where $M_L \in \mathbb{R}^{m \times r}$ and $M_R \in \mathbb{R}^{r \times n}$, and their entries follow independently the standard Gaussian or uniform distributions. The subset $\Omega$ of $q$ entries are sampled independently and uniformly at random.

If without special statement, some parameters are set as follows. We set $m = n = 500$ with sampling ratio $\tau$. Then the sampling number $q = \tau mn$. Also, the initial solution of our algorithms is set as $X^0 = 0$, the initial value of the upper bound
on \( \nabla f(X) \) is set as \( L = 1 \), and the tolerances as \( \epsilon = \epsilon_{in} = 10^{-5} \). The initial value \( \lambda_0 \) of the regularization parameter \( \lambda \) is set as \( \eta \| A^*(y) \|_\infty \) and \( \rho = 0.3 \) controls its decreasing speed, i.e., \( \lambda_0 = \eta \| A^*(y) \|_2 \), \( \lambda_{k+1} = 0.3 \lambda_k \) for \( \eta > 0 \), \( k = 0, 1, 2, \cdots, \lambda_t \), and \( \lambda_t = 0.001 \). All parameters in the compared methods are set as their default values if without special statement.

As \([4, 18]\), for a given approximate recovery \( \hat{X} \) for \( M \), the relative error, which is defined as

\[
\text{rel.err.} := \frac{\| \hat{X} - M \|_F}{\| M \|_F},
\]

is used to estimate the reconstruction error. We call that \( \hat{X} \) recovers a matrix \( M \), if the relative error is less than \( 10^{-3} \) and the recovered rank is exactly \( r \). This definition is different from that in \([4, 18]\), in which only the relative error less than \( 10^{-3} \) is required. We redefine this concept due to that, though some methods can reconstruct low rank matrix with small relative error, they cannot recover a matrix with the exact rank in some cases (refer to Table 4). We denote \( NS \) as the number of matrices that are successfully recovered.

5.1. Influence of parameters. First of all, we test the influence of some parameters on our method, such as the regularization parameter \( \lambda \), and the maximum number of inner iterations for each fixed \( \lambda \) in Algorithm 2. In this subsection, the entries of \( M_L \in \mathbb{R}^{m \times r} \) and \( M_R \in \mathbb{R}^{r \times n} \) follow independently the standard Gaussian distribution.

In the first experiment, we investigate the influence of the regularization parameter \( \lambda \) on Algorithm 1 and the influence of the initial value \( \lambda_0 \) on Algorithm 2. All parameters are set as above except that the sampling ratio is set as \( \tau = 0.5 \), rank as \( r = 40 \), the number of inner iterations \( \text{step} \) as 50, and For each value of \( M \), sampling number \( q \), rank \( r \), and the iteration \( \text{step} \), 50 random trials are made. The averaged experimental results are put in Table 1, including \( NS \), the average recovered rank \( r \), the average \( \text{rel.err.} \) and the average running time (in seconds).

| \( \eta \) | \( \eta \) | \( \text{HIHT} \) | \( \text{HIHT} \) |
|---|---|---|---|
| 50 | 4 | 122.43 | 38.56 | 1.29E-01 | 50 | 35.33 | 40 | 1.17E-05 |
| 20 | 50 | 18.83 | 40.00 | 5.82E-05 | 50 | 19.86 | 40 | 1.17E-05 |
| 10 | 47 | 23.69 | 40.06 | 2.52E-03 | 50 | 23.25 | 40 | 1.38E-05 |
| 5 | 0 | 52.92 | 227.10 | 6.33E-01 | 0 | 45.41 | 227.72 | 6.34E-01 |

From Table 1, we can find that, Algorithm 1 can recover the low rank matrix only when choosing a suitable value of the regularization parameter \( \lambda \). However, without knowing the rank value \( r \), the HIHT method can solve the problem when \( \lambda_0 \) is not too small. Moreover, Table 1 also shows that the proposed HIHT method can exactly recover the low rank matrix without prior knowledge of the exact matrix rank. Furthermore, considering performance of our homotopy method, we set \( \eta = 20 \) for the HIHT method if without special statement in the following experiments.

In the second experiment, we investigate the influence of the maximum number of inner iterations \( \text{step} \) (the loop between lines 5-8) on Algorithm 2. Values of all parameters are set as in the first experiment except the number of inner iterations. The maximum number of inner iterations \( \text{step} \) ranges from 5 to 70. For each value of
Table 2. Influence of maximum number of inner iterations on Algorithm 2

| step | NS | time | rank | rel.err. |
|------|----|------|------|----------|
| 5    | 50 | 16.72| 40   | 4.35E-05 |
| 10   | 50 | 18.13| 40   | 2.09E-05 |
| 20   | 50 | 18.97| 40   | 1.35E-05 |
| 30   | 50 | 19.29| 40   | 1.16E-05 |
| 40   | 50 | 19.27| 40   | 1.16E-05 |
| 50   | 50 | 19.24| 40   | 1.17E-05 |
| 60   | 50 | 19.72| 40   | 1.01E-05 |
| 70   | 50 | 19.71| 40   | 1.02E-05 |

$M$, sampling number $q$, rank $r$, and the number of inner iterations $step$, 50 random trials are made. The averaged experimental results are put in Table 2.

From Table 2, we can find out that the running time just changes in several seconds and the relative error keeps on the same order of magnitude when the number of inner iterations ranges from 10 to 70. Hence, in the following experiments, considering robustness of our homotopy method, we set $step = 50$.

5.2. Comparing with other methods. In this subsection, we compare the performances of our HIHT method with the state-of-the-art methods: FPCA [8, 18], PD [17] for solving the matrix completion problem on random instances. The two methods can solve the problem without prior knowledge of the rank value. More precisely, the FPCA method solves the nuclear norm minimization problem (3), while the PD method solves the low rank minimization problem (1) directly. We mainly test the compared methods on instances with different ranks.

All the methods are tested on random Gaussian and uniform matrices with problem size $m = n = 500$ and fixed sampling ratio $\tau = 0.5$, but with different rank. All parameters of our methods are set as stated before, except with different rank. We also set parameters of all other compared methods as default values except that $tol = 10^{-4}, \mu = 10^{-4}$ for FPCA. 50 instances are randomly generated for each rank value. The averaged experimental results are put in Table 3 and Table 4, respectively.

Table 3 shows that all compared methods can reconstruct the low rank Gaussian matrices well, but with obviously different running times. Among the compared methods, The FPCA method runs fast when the rank is lower than 50 but much slow when the rank becomes larger. The PD method is almost the slowest on instances with different ranks. This is due to that there are many subproblems to be solved and each iteration uses the exact SVD. Our HIHT method also uses the exact SVD, but it is much faster than the PD method.

From Table 4, we can find out that our method is still efficient and effective. Specifically, when the rank is larger, our method outperforms all the other methods both in running time and solution quality. The FPCA method could reconstruct low rank matrix only when the rank is smaller than 30. The PD method performs better than the FPCA method when the rank is not less than 30, and presents almost the same performance as our HIHT method but with higher CPU time.
Table 3. Numerical results on random Gaussian matrices with size $m = n = 500$ and sampling ratio $\tau = 0.5$.

| Alg. | $r=5$ | $r=10$ | $r=30$ | $r=50$ | $r=70$ | $r=90$ |
|------|-------|-------|-------|-------|-------|-------|
|      | NS    | time  | rank  | rel.err. | NS    | time  | rank  | rel.err. | NS    | time  | rank  | rel.err. |
| FPCA | 50    | 4.69  | 5     | 1.24E-06 | 50    | 5.19  | 10    | 2.90E-06 | 50    | 255.32 | 50    | 1.61E-04 |
| PD   | 50    | 95.92 | 5     | 5.92E-05 | 50    | 102.83 | 10    | 8.54E-05 | 50    | 106.19 | 50    | 1.50E-04 |
| HIHT | 50    | 10.91 | 5     | 7.49E-05 | 50    | 12.54 | 10    | 7.02E-05 | 50    | 24.13  | 50    | 2.67E-05 |

Table 4. Numerical results on random uniform matrices with size $m = n = 500$ and sampling ratio $\tau = 0.5$.

| Alg. | $r=5$ | $r=10$ | $r=30$ | $r=50$ | $r=70$ | $r=90$ |
|------|-------|-------|-------|-------|-------|-------|
|      | NS    | time  | rank  | rel.err. | NS    | time  | rank  | rel.err. | NS    | time  | rank  | rel.err. |
| FPCA | 50    | 5.78  | 30    | 1.44E-05 | 50    | 255.32 | 50    | 1.61E-04 | 50    | 329.90 | 90    | 2.71E-04 |
| PD   | 50    | 102.38 | 30    | 9.40E-05 | 50    | 106.19 | 50    | 1.50E-04 | 50    | 347.48 | 90    | 2.51E-04 |
| HIHT | 50    | 18.57 | 30    | 6.63E-05 | 50    | 24.13  | 50    | 2.67E-05 | 50    | 31.32  | 90    | 2.08E-04 |

In the last experiment, we compare the three methods on the matrix rank minimization problem with $m = n = 4000$, $r = 200$, and sampling ratio $\tau = 0.3$. We set parameters as before except that $\eta = 200$. We generate 50 instances randomly for each case. The average experimental results are put in Table 5.

Table 5 shows that, though the FPCA method is the best on the Gaussian matrices, it cannot recover the uniform matrices. However, the HIHT method works well in both cases. It also shows that our HIHT method performs better than the PD method both in running time and solution quality on the Gaussian matrices. For uniform matrices, our method is comparable with the PD method.

6. Conclusions. A homotopy method, which combines the advantage of the homotopy technique with the matrix hard thresholding method, is proposed for solving the matrix rank minimization problem. The method is strictly decreasing and convergent, and can overcome the difficulty of the iterative hard thresholding method on the choice of the regularization parameter, by tracing solutions of the sparse problem along a homotopy path. In addition, the homotopy method with exact
Table 5. Computational results on random Gaussian (or uniform) matrices with size $m = n = 4000$, $r = 200$, and sampling ratio $\tau = 0.3$.

| Alg. | NS | time | rank | rel.err. | NS | time | rank | rel.err. |
|------|----|------|------|----------|----|------|------|----------|
| FPFA | 50 | 284.36 | 200 | 8.28E-05 | 0  | 1887.44 | 1.06 | 2.35E-02 |
| PD   | 50 | 13232.85 | 200 | 1.02E-04 | 50 | 5976.50 | 200 | 9.89E-05 |
| HIHT | 50 | 4427.63 | 200 | 1.63E-05 | 50 | 7637.74  | 200 | 1.70E-05 |

SVD does not require a prior knowledge of the rank value. Numerical experiments demonstrate effectiveness of the proposed algorithm in accurately and efficiently recovering a low rank matrix. In particular, the low computational complexity per iteration makes the proposed method be efficient for large scale problems.

**Acknowledgments.** The authors would like to thank the anonymous reviewers, whose insightful comments and suggestions helped improving the quality of the manuscript greatly.

**REFERENCES**

[1] J. D. Blanchard, J. Tanner and K. Wei, CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion, *Information and Inference: A Journal of the IMA*, 4 (2014), 289–327.

[2] T. Blumensath and M. E. Davies, Iterative thresholding for sparse approximations, *Journal of Constructive Approximation*, 14 (2008), 629–654.

[3] T. Blumensath and M. E. Davies Normalised iterative hard thresholding: guaranteed stability and performance, *IEEE Journal of Selected Topics in Signal Processing*, 4 (2010), 298–309.

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

[5] E. J. Candès and T. Tao, The power of convex relaxation: Near-optimal matrix completion, *IEEE Transactions on Information Theory*, 56 (2009), 2053–1080.

[6] M. Fazel, H. Hindi and S. Boyd, Rank minimization and applications in system theory, *Proceedings of the American Control Conference*, 4 (2004), 3273–3278.

[7] M. Fazel, T. K. Pong, D. Sun and P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, *SIAM Journal on Matrix Analysis and Applications*, 34 (2013), 946–977.

[8] D. Goldfarb and S. Ma, Convergence of fixed-point continuation algorithms for matrix rank minimization, *Foundations of Computational Mathematics*, 11 (2011), 183–210.

[9] J. P. Haldar and D. Hernando, Rank-constrained solutions to linear matrix equations using powerfactorization, *IEEE Signal Processing Letters*, 16 (2009), 584–587.

[10] N. J. A. Harvey, D. R. Karger and S. Yekhanin, The complexity of matrix completion, *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, (2006), 1103–1111.

[11] A. Kyrillidis and V. Cevher, Martix ALPS: Accelerated low rank and sparse matrix reconstruction, Technical Report, 2012.

[12] A. Kyrillidis and V. Cevher, Martix recips for hard thresholding methods, *Journal of Mathematical Imaging and Vision*, 48 (2014), 235–265.

[13] Y. Liu, J. Tao, H. Zhang, X. Xiu and L. Kong, Fused LASSO penalized least absolute deviation estimator for high dimensional linear regression, *Numerical Algebra, Control & Optimization*, 8 (2018), 97–117.

[14] Z. Liu and L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, *SIAM Journal on Matrix Analysis and Applications*, 31 (2009), 1235–1256.

[15] C. Lu, J. Tang, S. Yan and Z. Lin, Generalized nonconvex nonsmooth low-rank minimization, *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[16] Z. Lu, Iterative hard thresholding methods for $l_1$ regularized convex cone programming, *Mathematical Programming*, 147 (2014), 125–154.
[17] Z. Lu and Y. Zhang, Penalty decomposition methods for rank minimization, *Optimization Methods and Software*, 30 (2015), 531–558.
[18] S. Ma, D. Goldfarb and L. Chen, Fixed point and bregman iterative methods for matrix rank minimization, *Mathematical Programming*, 128 (2011), 321–353.
[19] K. Mohan and M. Fazel, Reweighted nuclear norm minimization with application to system identification, *Proceedings of the American Control Conference*, 2010.
[20] K. Mohan and M. Fazel, Iterative reweighted algorithms for matrix rank minimization, *Journal of Machine Learning Research*, 13 (2012), 3441–3473.
[21] B. Recht, M. Fazel and P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Review*, 52 (2010), 471–501.
[22] J. Tanner and K. Wei, Normalized iterative hard thresholding for matrix completion, *SIAM Journal on Scientific Computing*, 35 (2013), S104–S125.
[23] Z. Wen, W. Yin and Y. Zhang, Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm, *Mathematical Programming Computation*, 4 (2012), 333–361.
[24] Z. Weng and X. Wang, Low-rank matrix completion for array signal processing, *IEEE International Conference on Speech and Signal Processing*, (2012), 2697–2700.

Received May 2018; 1st revision July 2018; final revision August 2018.

E-mail address: 1030656689@qq.com
E-mail address: jilchen@fzu.edu.cn
E-mail address: wxzhu@fzu.edu.cn