Singular value half thresholding algorithm for ℓ_p regularized matrix optimization problems

Dingtao Peng, Xian Zhang, Shouyu Yi

Abstract. In this paper, we study the low-rank matrix optimization problem, where the penalty term is the ℓ_p $(0 < p < 1)$ regularization. Inspired by the good performance of half thresholding function in sparse/low-rank recovery problems, we propose a singular value half thresholding (SVHT) algorithm to solve the ℓ_p regularized matrix optimization problem. The main iteration in SVHT algorithm makes use of the closed-form solution of the subproblem but is different from the existing algorithm, which is in essence to make local 1/2 approximation to the objective function at the current point, instead of local linear or local quadratic approximation. By constructing Lipschitz and non-Lipschitz approximate functions of the objective function, we prove that any accumulation point of the sequence generated by SVHT algorithm is a first-order stationary point of the problem. In numerical experiments, we test SVHT algorithm by low-rank matrix completion problem on both simulated data and real image data. Extensive numerical results show that SVHT algorithm is very efficient for low-rank matrix optimization problems in terms of speed, accuracy, robustness and so on.

Keywords Low-rank matrix optimization problem; ℓ_p regularization; closed-form solution; singular value half thresholding algorithm; first-order stationary point

MSC(2010) 90C26 · 90C46

1 Introduction

In this paper, we consider the following ℓ_p regularized matrix optimization problem:

$$\min_{X \in \mathbb{R}^{m\times n}} F(X) = f(X) + \lambda \|X\|_p^p, \tag{1.1}$$

where $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a smooth function with L_f-Lipschitz continuous gradient in $\mathbb{R}^{m\times n}$, i.e.,

$$\|\nabla f(X) - \nabla f(Y)\|_F \leq L_f \|X - Y\|_F, \quad \forall X, Y \in \mathbb{R}^{m\times n},$$

and f is bounded from below. In (1.1), $\lambda > 0, p \in (0, 1)$ and $\|X\|_p^p = \|\sigma(X)\|_p^p = \sum_{i=1}^{\min\{m,n\}} \sigma_i^p(X)$ is the ℓ_p quasi-norm of X, and $\sigma_i(X)$ is the i-th largest singular value of X. Problem (1.1)

July. 28, 2022

Dingtao Peng ✉
School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China. E-mail: dingtaopeng@126.com
Xian Zhang
School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China.
Shouyu Yi
School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China.

arises in many important contemporary applications in signal and image processing, control [9,10,37], statistics [8,35] and compressed sensing [3,46]. In particular, problem (1.1) extends the following ℓ_p regularized least squares problem:

$$\lim_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathcal{A}(X)-b\|_2^2+\lambda\|X\|_p^p, \tag{1.2}$$

where $b\in\mathbb{R}^l$, $\mathcal{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^l$ is a linear transformation.

When $p\downarrow 0$, problem (1.2) tends to the ℓ_0 regularized least squares problem:

$$\lim_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathcal{A}(X)-b\|_2^2+\lambda\|\sigma(X)\|_0, \tag{1.3}$$

where $\|\sigma(X)\|_0 = \mathrm{rank}(X) := \sum\limits_{i=1}^{\min\{m,n\}} I(\sigma_i(X))$ with $I(t)=0$ if $t=0$ and $I(t)=1$ otherwise.

In addition, when $p\uparrow 1$, problem (1.2) tends to the ℓ_1 regularized least squares problem:

$$\lim_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathcal{A}(X)-b\|_2^2+\lambda\|\sigma(X)\|_1, \tag{1.4}$$

where $\|\sigma(X)\|_1 = \|X\|_* := \sum\limits_{i=1}^{\min\{m,n\}} \sigma_i(X)$ is the nuclear norm of X.

Since theoretical studies show that $\|\sigma(X)\|_1$ is the tightest convex lower bound of the rank function [34], it is a popular strategy to relax the $\|\sigma(X)\|_0$ minimization problem to the $\|\sigma(X)\|_1$ minimization problem. There are a few well-known algorithms for solving problem (1.4). For examples, singular value thresholding (SVT) algorithm [1], singular value projection (SVP) algorithm [14], fixed point continuation (FPC) and Bregman iterative algorithm [29], accelerated proximal gradient with linesearch (APGL) algorithm [38], prime and dual proximal point algorithm [23], LMaFit [44], linearized augmented Lagrangian methods [47], alternating direction methods [5,47] and some other types of algorithms [15,18,22,36,43].

Although $\|\sigma(X)\|_1$ is a good substitution for $\|\sigma(X)\|_0$ and can obtain promising results, there still exists a large gap between $\|\sigma(X)\|_1$ and $\|\sigma(X)\|_0$. It is obvious that the $\|\sigma(X)\|_p^p$ $(0<p<1)$ can reduce this gap. But problem (1.1) is hard to solve since it is nonconvex, non-Lipshcitz and even NP-hard. By virtue of some special optimality conditions such as subspace optimality [26,20,39] and fixed point equations [33,41,42], some researchers have proposed a series of algorithms to obtain the approximate solution of problem (1.1). These algorithms include majorization minimization method [27,28], thresholding algorithms [20, 41,42], fixed point method [32,33] and et al..

Recall that in the case of sparse optimization, some iterative reweighted methods are very efficient in solving ℓ_p regularized sparse optimization problems. The first type of iterative reweighted methods is iterative reweighted ℓ_1 method [2,6,16,24,45], which generates x^{k+1} via substituting $\|x\|_p^p$ by $\sum_i \omega_i^k|x_i|$ with $\omega_i^k=(|x_i^k|+\epsilon_i)^{p-1}$ in the kth iteration. The second type of iterative reweighted methods is iterative reweighted ℓ_2 method [4,7,24,45], which generates x^{k+1} via substituting $\|x\|_p^p$ by $\sum_i \omega_i^k x_i^2$ with $\omega_i^k=(|x_i^k|+\epsilon_i)^{p-2}$ in the kth iteration.

Inspired by iterative reweighted methods for ℓ_p regularized sparse optimization problems, some iterative reweighted methods for ℓ_p regularized matrix optimization problems were proposed. Iterative reweighted ℓ_1 (IRL_1) [12,21,26,30] was such a method for ℓ_p regularized matrix optimization problems, in the kth iteration of which $\|\sigma(X)\|_p^p$ is approximated by $\sum_i \omega_i^k\sigma_i(X)$ with $\omega_i^k=(\sigma_i(X^k)+\epsilon_i)^{p-1}$. Iterative reweighted ℓ_2 (IRL_2) [11,17,31] was another type of method for ℓ_p regularized matrix optimization problems, in the kth iteration of which $\|\sigma(X)\|_p^p$ is approximated by $\sum_i \omega_i^k \sigma_i^2(X)$ with $\omega_i^k = (\sigma_i(X^k) + \epsilon_i)^{p-2}$. The convergence analysis of IRL_1 and IRL_2 methods both take advantage of the closed-form solutions and the convexity of the objective functions of the subproblems. In fact, the essence of IRL_1 and IRL_2 method is to approximate $\|\sigma(X)\|_p^p$ by locally linear functions and locally quadratic functions respectively at each iteration. However, as pointed out in [26], the IRL_2 methods usually do not produce a low-rank solution. Although the IRL_1 methods tend to produce a low-rank solution under some conditions, one can expect that the locally folded concave approximation of $\|\sigma(X)\|_p^p$ can produce a lower rank solution under some weaker conditions, which has been confirmed in sparse optimization.

Therefore, different from previous works, in this paper we consider a new method to solve (1.1), which generates a new iteration X^{k+1} by

$$X^{k+1} \in \arg\min_{X \in \mathbb{R}m\times n} \left\{ \frac{L}{2} \left\| X - \left(X^k - \frac{1}{L} \nabla f(X^k) \right) \right\|_F^2 + 2\lambda p \sum_{i=1}^{n} s_i^k \sigma_i^{1/2}(X) \right\}, \quad (1.5)$$

where $s_i^k = (\sigma_i^{1/2}(X^k) + \epsilon_i^k)^{2p-1}$. This idea is inspired by the $\ell_{1/2}$ regularization method in sparse optimization. Since results in [41,46] revealed that the $\ell_{1/2}$ regularization not only has a closed-form solution which is described by half thresholding function, but also has very powerful recovering ability of sparse solution. Thus one can expect that the above locally $\ell_{1/2}$ approximation of $\|\sigma(X)\|_p^p$ can produce a lower rank solution under some weaker conditions.

Our algorithm based on (1.5) for problem (1.1) is called singular value half thresholding (SVHT) algorithm. It is worthy of mentioning that subproblem (1.5) is nonconvex and non-Lipschitz, which is different from the subproblems in IRL_1 and IRL_2 methods. This will bring some difficulties in convergence analysis. In our analysis, we prove the convergence by constructing Lipschitz and non-Lipschitz approximate functions of the objective function.

This paper is organized as follows. In section 2, we give some preliminaries and technical results used in this paper. In section 3, we propose the singular value half thresholding algorithm and establish the convergence results. In section 4, we test the proposed algorithms by recovering simulated low-rank matrices and real images, and present the numerical results. In section 5, we give the conclusion of this paper.

2 Preliminaries

Through out this paper, without loss of generality, we always suppose $n \le m$. For any $X \in \mathbb{R}^{m\times n}$, let $\sigma(X) = (\sigma_1(X), \cdots, \sigma_n(X))^\top$ denote the vector of singular values of X arranged in nonincreasing order, $\mathrm{Diag}(\sigma(X))$ denote a diagonal matrix whose diagonal vector is $\sigma(X)$, $\|X\|_F$ denote the Frobenius norm of X, namely, $\|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^{1/2} = \left(\sum_{i,j} \sigma_i^2(X)\right)^{1/2}$, and

$$\mathcal{O}(X) = \left\{ (U, V) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r} : U^\top U = V^\top V = I, X = U\mathrm{Diag}(\sigma(X))V^\top \right\}, \quad (2.1)$$

with $r = \mathrm{rank}(X)$. For any $X, Y \in \mathbb{R}^{m\times n}$, let $\langle X, Y\rangle = \mathrm{tr}(Y^\top X)$ denote the inner product of these two matrices.

Next, we introduce the half thresholding function which is the fundament of our algorithm. The following lemma follows from [3,32,46].

Lemma 2.1 *Let $\omega > 0$. Define*

$$h_\omega(t) := \arg\min_{x\in\mathbb{R}} \left\{ (x-t)^2 + \omega|x|^{1/2} \right\}, \quad \forall t \in \mathbb{R},$$

then $h_\omega(t)$ can be analytically expressed by

$$h_\omega(t) = \begin{cases} \phi_\omega(t), & |t| > \frac{\sqrt[3]{54}}{4}\omega^{2/3}, \\ \{\phi_\omega(t), 0\}, & |t| = \frac{\sqrt[3]{54}}{4}\omega^{2/3}, \\ 0, & |t| < \frac{\sqrt[3]{54}}{4}\omega^{2/3}. \end{cases} \tag{2.2}$$

where

$$\phi_\omega(t) = \frac{2}{3}t\left(1 + \cos\left(\frac{2\pi}{3} - \frac{2}{3}\arccos\left(\frac{\omega}{8}\left(\frac{|t|}{3}\right)^{-3/2}\right)\right)\right).$$

The function $h_\omega : \mathbb{R} \rightrightarrows \mathbb{R}$ given by (2.2) is called a half thresholding function. Note that this half thresholding function may be multi-valued (set-valued) since the objective function has two minimizers when $|t| = \frac{\sqrt[3]{54}}{4}\omega^{2/3}$. Hence "argmin" means the set of all global minimizers of the problem.

Similar to [3,32,33,46], we give the following definitions.

Definition 2.2 *(Vector-valued half thresholding operator) For any $w = (w_1, \cdots, w_n)^\top \in \mathbb{R}^n$ with $w_i > 0$, the vector-valued half thresholding operator $H_w : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is defined as*

$$H_w(x) := (h_{w_1}(x_1), \cdots, h_{w_n}(x_n))^\top \in \mathbb{R}^n, \quad \forall x = (x_1, \cdots, x_n)^\top \in \mathbb{R}^n.$$

Definition 2.3 *(Matrix-valued half thresholding operator) For any $w = (w_1, \cdots, w_n)^\top \in \mathbb{R}^n$ with all $w_i > 0$, the matrix-valued half thresholding operator $\mathcal{H}_w : \mathbb{R}^{m\times n} \rightrightarrows \mathbb{R}^{m\times n}$ is defined as*

$$\mathcal{H}_w(Y) := U\text{Diag}(H_w(\sigma(Y)))V^\top \in \mathbb{R}^{m\times n}, \quad \forall\ Y = U\text{Diag}(\sigma(Y))V^\top \in \mathbb{R}^{m\times n}.$$

For any given $w = (w_1, \cdots, w_n)^\top \in \mathbb{R}^n$ with $w_n \geq \cdots \geq w_1 > 0$, let $\Phi(X) = \sum_{i=1}^{n} w_i \cdot \sigma_i^{1/2}(X)$. Then $\Phi(X)$ can be reexpressed as the composition of an absolutely symmetric function ϕ [19] and the vector $\sigma(X)$ of singular values of X, i.e.,

$$\Phi(X) = (\phi \circ \sigma)(X) \text{ with } \phi(x) = \sum_{i=1}^{n} w_i \phi_i^{1/2}(x),$$

where $\phi_i(x)$ is the i-th largest element of $\{|x_1|, |x_2|, \cdots, |x_n|\}$.

According to Lemma 2.1 and [25, Proposition 2.1], we immediately obtain the following Lemma 2.4, which shows that $\mathcal{H}_w(Y)$ is a proximal operator associated with the nonconvex and non-Lispschitz function $\sum_{i=1}^n w_i\sigma_i^{1/2}(X)$.

Lemma 2.4 *For any given $w = (w_1, \cdots, w_n)^\top \in \mathbb{R}^n$ with $w_n \geq \cdots \geq w_1 > 0$ and any $Y \in \mathbb{R}^{m\times n}$, it holds that*

$$\mathcal{H}_w(Y) = \arg\min_{X\in\mathbb{R}^{m\times n}} \|X - Y\|_F^2 + \sum_{i=1}^{n} w_i\sigma_i^{1/2}(X).$$

It follows from [[19], Theorem 2.1] that the limiting subdifferential of a singular value function $(\phi \circ \sigma)$ at X is given by

$$\partial(\phi \circ \sigma)(X) = \Big\{U\text{Diag}(\xi)V^\top : \xi \in \partial\phi(\sigma(X)), (U, V) \in \mathcal{O}(X)\Big\}, \tag{2.3}$$

where $\partial\phi(\cdot)$ is the limiting subdifferent of ϕ.

Following [26, Definition 1], we give the first-order stationary points for problem (1.1).

Definition 2.5 *$X^* \in \mathbb{R}^{m\times n}$ of rank r is a first-order stationary point of problem (1.1) if*

$$0 \in \Big\{U^\top \nabla f(X^*)V + \lambda p \mathrm{Diag}(\sigma^{p-1}(X^*)) : (U,V) \in \mathcal{O}(X^*)\Big\}, \tag{2.4}$$

where $\mathcal{O}(X^*)$ defined in (2.1) and $\sigma^{p-1}(X^*) := (\sigma_1^{p-1}(X^*), \cdots, \sigma_r^{p-1}(X^*), 0, \cdots, 0)^\top$.

3 Singular value half thresholding algorithm for problem (1.1)

In this section, we propose a singular value half thresholding (SVHT) algorithm for solving problem (1.1) and provide its convergence analysis.

3.1 Closed-form solution of the subproblem

For any $Z \in \mathbb{R}^{m\times n}$, denote

$$B_L(Z) = Z - \frac{1}{L}\nabla f(Z).$$

In our SVHT algorithm, the key subproblem at each iteration is to solve a weighted singular value minimization problem in the following form

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{L}{2}\|X - B_L(Z)\|_F^2 + 2\lambda p \sum_{i=1}^{n} s_i \sigma_i^{\frac{1}{2}}(X), \tag{3.1}$$

where $s_i = (\sigma_i^{1/2}(Z) + \epsilon_i)^{2p-1}$ and $\epsilon_i > 0$. From Lemma 2.4, we obtain that problem (3.1) has a closed-form solution, which is given by the following lemma.

Lemma 3.1 *For any $\lambda > 0$, $L \geq L_f$ and $\epsilon_i > 0, i = 1, \cdots, n$, let $B_L(Z) = Z - \frac{1}{L}\nabla f(Z)$ admit the following SVD:*

$$B_L(Z) = U\mathrm{Diag}(\sigma(B_L(Z)))V^\top.$$

Then the closed-form solution $\tilde{X}$ of problem (3.1) can be analytically given by

$$\tilde{X} = \mathcal{H}_{\frac{4\lambda p s}{L}}(B_L(Z)), \tag{3.2}$$

where $s = (s_1, \cdots, s_n)^\top$ with $s_i = (\sigma_i^{1/2}(Z) + \epsilon_i)^{2p-1}$ for $i = 1, \cdots, n$.

3.2 Scheme of SVHT algorithm

The scheme of SVHT algorithm is given as Algorithm 1.

Algorithm 1 singular value half thresholding (SVHT) algorithm

– **Initialize:** Choose $\{\epsilon^k\}$ being a sequence of component-wise non-increasing positive vectors in $\mathbb{R}^n$. Set $X^0 \in \mathbb{R}^{m\times n}$ and $k = 0$.
– **Step1.** Compute

$$X^{k+1} \in \arg\min_{X\in\mathbb{R}^{m\times n}} \Bigg\{ f(X^k) + \langle \nabla f(X^k), X - X^k\rangle + \frac{L}{2}\|X - X^k\|_F^2 + 2\lambda p \sum_{i=1}^{n} s_i^k \sigma_i^{1/2}(X) \Bigg\}. \tag{3.3}$$

where $s^k = (s_1^k, \cdots, s_n^k)^\top$ with $s_i^k = \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right)^{2p-1}$ for $i = 1, \cdots, n$.
– **Step2.** Let $k := k + 1$, return to **Step 1**.
– **Output:** X^k

Let

$$\begin{aligned} Q_L(X, X^k) :&= f(X^k) + \langle \nabla f(X^k), X - X^k\rangle + \frac{L}{2}\|X - X^k\|_F^2 + 2\lambda p \sum_{i=1}^{n} s_i^k \sigma_i^{1/2}(X), \\ &= \frac{L}{2}\|X - (X^k - \frac{1}{L}\nabla f(X^k))\|_F^2 + 2\lambda p \sum_{i=1}^{n} s_i^k \sigma_i^{1/2}(X) + C_k, \end{aligned} \tag{3.4}$$

where C_k is a constant with respect to X, then (3.3) can be rewritten as

$$X^{k+1} \in \arg\min_{X\in\mathbb{R}^{m\times n}} Q_L(X, X^k),$$

that is,

$$Q_L(X^{k+1}, X^k) \le Q_L(X, X^k) \tag{3.5}$$

for any $X \in \mathbb{R}^{m\times n}$. From Lemma 3.1, we know that X^{k+1} can be selected from $\mathcal{H}_{\frac{4\lambda p s^k}{L}}(B_L(X^k))$, i.e., $X^{k+1} \in \mathcal{H}_{\frac{4\lambda p s^k}{L}}(B_L(X^k))$, where $s^k = (s_1^k, \cdots, s_n^k)^\top$ with $s_i^k = \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right)^{2p-1}$ for $i = 1, \cdots, n$.

3.3 Convergence analysis

Next, we provide the convergence analysis for SVHT algorithm.

We first define

$$F_\epsilon(X) := f(X) + \lambda \sum_{i=1}^{n} (\sigma_i^{1/2}(X) + \epsilon_i)^{2p}. \tag{3.6}$$

For any $p \in (0, 1/2]$, let q be such that

$$\frac{1}{2p} + \frac{1}{q} = 1, \tag{3.7}$$

then $q < 0$, and it is not hard to check that

$$t^{2p-1} = \arg\min_{\tau \geq 0} 2p \left\{ t\tau - \frac{\tau^q}{q} \right\} \text{ and } t^{2p} = \min_{\tau \geq 0} 2p \left\{ t\tau - \frac{\tau^q}{q} \right\} \tag{3.8}$$

for any $t > 0$. Using (3.8) and the definition of s^k, we obtain that

$$s_i^k = \arg\min_{s_i \geq 0} 2p \left\{ \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right) s_i - \frac{s_i^q}{q} \right\}, \tag{3.9}$$

$$\left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right)^{2p} = \min_{s_i \geq 0} 2p \left\{ \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right) s_i - \frac{s_i^q}{q} \right\}, \text{ and} \tag{3.10}$$

$$s^k = \arg\min_{s \in \mathbb{R}_+^n} \tilde{F}_{\epsilon^k}(X^k, s), \tag{3.11}$$

where $s^k = (s_1^k, \cdots, s_n^k)$, $\mathbb{R}_+^n = \{(a_1, \cdots, a_n)^\top \in \mathbb{R}^n : a_i \geq 0\}$ and

$$\tilde{F}_\epsilon(X, s) := f(X) + 2\lambda p \sum_{i=1}^n \left[\left(\sigma_i^{1/2}(X) + \epsilon_i\right) s_i - \frac{s_i^q}{q} \right]. \tag{3.12}$$

Particularly, by (3.6),(3.10) and (3.12), we have

$$F_{\epsilon^k}(X^k) = \tilde{F}_{\epsilon^k}(X^k, s^k) \tag{3.13}$$

Remark 3.2 *The functions $F_\epsilon(X)$ and $\tilde{F}_\epsilon(X, s)$ are Lipschitz and non-Lipschitz approximation functions of $F(X)$ respectively. They will play very important roles in the analysis of convergence of SVHT algorithm.*

Lemma 3.3 *Let $p \in (0, 1/2]$, $\lambda > 0$, $L \geq L_f$, $\{\epsilon^k\}$ be a sequence of component-wise non-increasing positive vectors in $\mathbb{R}^n$ with $\epsilon^k \to 0$ as $k \to \infty$, and $\{X^k\}$ be the sequence generated by Algorithm 1. Then the sequence $\{F_{\epsilon^k}(X^k)\}$ is non-increasing and satisfies*

$$F_{\epsilon^{k+1}}(X^{k+1}) - F_{\epsilon^k}(X^k) \leq -\frac{1}{2}(L - L_f)\|X^{k+1} - X^k\|_F^2. \tag{3.14}$$

Proof From (3.4) and (3.5), we have $Q_L(X^{k+1}, X^k) \leq Q_L(X^k, X^k)$, i.e.,

$$\begin{aligned} &f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L}{2}\|X^{k+1} - X^k\|_F^2 + 2\lambda p \sum_{i=1}^n s_i^k \sigma_i^{1/2}(X^{k+1}) \\ &\leq f(X^k) + 2\lambda p \sum_{i=1}^n s_i^k \sigma_i^{1/2}(X^k). \end{aligned} \tag{3.15}$$

By the fact that ∇f is L_f-Lipschitz continuous, we have

$$f(X^{k+1}) \leq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{L_f}{2}\|X^{k+1} - X^k\|_F^2. \tag{3.16}$$

Combining (3.15) and (3.16), we obtain that

$$\begin{aligned} &f(X^{k+1}) + 2\lambda p \sum_{i=1}^n s_i^k \sigma_i^{1/2}(X^{k+1}) + \frac{1}{2}(L - L_f)\|X^{k+1} - X^k\|_F^2 \\ &\leq f(X^k) + 2\lambda p \sum_{i=1}^n s_i^k \sigma_i^{1/2}(X^k). \end{aligned} \tag{3.17}$$

From (3.12), it follows that

$$\tilde{F}_{\epsilon^k}(X^{k+1}, s^k) = f(X^{k+1}) + 2\lambda p \sum_{i=1}^{n} \left[\left(\sigma_i^{1/2}(X^{k+1}) + \epsilon_i^k\right) s_i^k - \frac{(s_i^k)^q}{q} \right],$$
$$\tilde{F}_{\epsilon^k}(X^k, s^k) = f(X^k) + 2\lambda p \sum_{i=1}^{n} \left[\left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right) s_i^k - \frac{(s_i^k)^q}{q} \right].$$

Using these two equalities with (3.17), we have

$$\tilde{F}_{\epsilon^k}(X^{k+1}, s^k) + \frac{1}{2}(L - L_f)\|X^{k+1} - X^k\|_F^2 \le \tilde{F}_{\epsilon^k}(X^k, s^k). \tag{3.18}$$

In addition, we see from (3.8) that $F_\epsilon(X) = \min_{s \in \mathbb{R}_+^n} \tilde{F}_\epsilon(X, s)$, which together with (3.11), (3.13), (3.18) and $\{\epsilon^k\}$ being a sequence of component-wise non-increasing positive vectors, we have

$$\begin{aligned} F_{\epsilon^{k+1}}(X^{k+1}) &= \tilde{F}_{\epsilon^{k+1}}(X^{k+1}, s^{k+1}) \le \tilde{F}_{\epsilon^k}(X^{k+1}, s^{k+1}) \le \tilde{F}_{\epsilon^k}(X^{k+1}, s^k) \\ &\le \tilde{F}_{\epsilon^k}(X^k, s^k) - \frac{1}{2}(L - L_f)\|X^{k+1} - X^k\|_F^2 \\ &= F_{\epsilon^k}(X^k) - \frac{1}{2}(L - L_f)\|X^{k+1} - X^k\|_F^2. \end{aligned}$$

Since $L \ge L_f$, the sequence $\{F_{\epsilon^k}(X^k)\}$ is non-increasing and (3.14) holds. □

Theorem 3.4 *Let $p \in (0, 1/2]$, $\lambda > 0$, $L \ge L_f$ and $\{\epsilon^k\}$ be a sequence of component-wise non-increasing positive vectors in $\mathbb{R}^n$ with $\epsilon^k \to 0$ as $k \to \infty$, and $\{X^k\}$ be the sequence generated by Algorithm 1. Then the following two statements hold.*

(i) *The sequence $\{X^k\}$ is bounded and $F_{\epsilon^k}(X^k)$ converges to $F(X^*)$, where X^* is any accumulation point of $\{X^k\}$.*

(ii) *The sequence $\{X^k\}$ is asymptotically regular, i.e., $\lim_{k\to\infty} \|X^{k+1} - X^k\| = 0$;*

Proof Since $\{F_{\epsilon^k}(X^k)\}$ is non-increasing, $F_{\epsilon^k}(X^k) \le F_{\epsilon^0}(X^0)$ for every $k \ge 0$. In addition, $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is bounded from below, then we have

$$\begin{aligned} \inf_{X \in \mathbb{R}^{m\times n}} f(X) + \lambda \sum_{i=1}^{n} \sigma_i^p(X) &\le \inf_{X \in \mathbb{R}^{m\times n}} f(X) + \lambda \sum_{i=1}^{n} \sigma_i^p(X^k) \\ &\le \inf_{X \in \mathbb{R}^{m\times n}} f(X) + \lambda \sum_{i=1}^{n} \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right)^{2p} \\ &\le f(X^k) + \lambda \sum_{i=1}^{n} \left(\sigma_i^{1/2}(X^k) + \epsilon_i^k\right)^{2p} \\ &= F_{\epsilon^k}(X^k) \le F_{\epsilon^0}(X^0). \end{aligned} \tag{3.19}$$

This means that $\lambda\|X^k\|_p^p \le \left(F_{\epsilon^0}(X^0) - \inf_{X \in \mathbb{R}^{m\times n}} f(X)\right)$, and then $\{X^k\}$ is bounded. Thus, the sequence $\{X^k\}$ has at least one accumulation point.

Inequality (3.19) also means that $F_{\epsilon^k}(X^k)$ is bounded from below, which together with the non-increasing property of $\{F_{\epsilon^k}(X^k)\}$, we know that $F_{\epsilon^k}(X^k)$ converges to a constant F^*. Let X^* be an accumulation point of $\{X^k\}$. By the continuity of $F(\cdot)$ as well as $\epsilon^k \to 0$ as $k \to \infty$ and convergence of $F_{\epsilon^k}(X^k)$, we get $F^* = F(X^*)$.

Next, we proved (ii). From Lemma 3.3, we have

$$\|X^{k+1} - X^k\|_F^2 \le \frac{1}{2}(L - L_f)(F_{\epsilon^k}(X^k) - F_{\epsilon^{k+1}}(X^{k+1})).$$

Due to $L \ge L_f$ and (i), we obtain that $\|X^{k+1} - X^k\| \to 0$ as $k \to \infty$. □

Theorem 3.5 *Let $p \in (0, 1/2]$, $\lambda > 0$, $L \ge L_f$ and $\{\epsilon^k\}$ be a sequence of component-wise non-increasing positive vectors in $\mathbb{R}^n$ with $\epsilon^k \to 0$ as $k \to \infty$, and $\{X^k\}$ be the sequence generated by Algorithm 1. Then any accumulation point of $\{X^k\}$ is a stationary point of problem (1.1).*

Proof Let $B_L(X^k) = X^k - \frac{1}{L}\nabla f(X^k)$ and $U^k \mathrm{Diag}(\sigma(B_{\bar{L}}(X^k)))(V^k)^\top$ be an SVD of $B_L(X^k)$, where $U^k \in \mathbb{R}^{m\times n}$, $V^k \in \mathbb{R}^{n\times r}$ with orthonormal columns and satisfy $(U^k)^\top U^k = (V^k)^\top V^k = I$, and $\sigma(B_L(X^k)) \in \mathbb{R}^n_+$ is the vector of singular values of $B_L(X^k)$ arranged in non-increasing order. From (3.3) and Lemma 3.1, we have

$$X^{k+1} \in \mathcal{H}_{\frac{4\lambda p s^k}{L}}(B_L(X^k)) = U^k \mathrm{Diag}\left(H_{\frac{4\lambda p s^k}{L}}\left(\sigma(B_L(X^k))\right)\right)(V^k)^\top \tag{3.20}$$

and

$$\sigma_i(X^{k+1}) \in h_{\frac{4\lambda p s_i^k}{L}}\left(\sigma_i(B_L(X^k))\right), \forall i = 1, \cdots, n, \tag{3.21}$$

where $s_i^k = (\sigma_i^{1/2}(X^k) + \epsilon_i^k)^{2p-1}$.

Let $\{X^{k_j}\}$ be a convergent subsequence of $\{X^k\}$ and the limit point be X^*, i.e.,

$$X^{k_j} \to X^*, \quad as \quad k_j \to \infty. \tag{3.22}$$

From (3.22), the continuous of $\sigma_i(\cdot)$, $s_i^k = (\sigma_i^{\frac{1}{2}}(X^k) + \epsilon_i^k)^{2p-1}$ and $\epsilon_i^{k_j} \to 0$ (as $k_j \to \infty$), we obtain that

$$s_i^{k_j} \to [\sigma_i(X^*)]^{p-\frac{1}{2}}, \quad \frac{4\lambda p s_i^k}{L} \to \frac{4\lambda p[\sigma_i(X^*)]^{p-\frac{1}{2}}}{L} \quad as \quad k_j \to \infty. \tag{3.23}$$

Due to (3.22) and the asymptotically regular of sequence $\{X^k\}$ by Theorem 3.4-(ii), we have

$$X^{k_j+1} = X^{k_j} + (X^{k_j+1} - X^{k_j}) \to X^*, \quad as \quad k_j \to \infty. \tag{3.24}$$

Due to (3.4), there holds

$$X^{k_j+1} \in \arg\min_{X\in\mathbb{R}^{m\times n}} \left\{ \frac{L}{2}\left\| X - \left[X^{k_j} - \frac{1}{L}\nabla f(X^{k_j})\right]\right\|_F^2 + 2\lambda p \sum_{i=1}^n \left(\sigma_i^{\frac{1}{2}}(X^{k_j}) + \epsilon_i^{k_j}\right)^{2p-1} \sigma_i^{\frac{1}{2}}(X) \right\},$$

then it holds

$$\begin{aligned} &\frac{L}{2}\left\| X^{k_j+1} - \left[X^{k_j} - \frac{1}{L}\nabla f(X^{k_j})\right]\right\|_F^2 + 2\lambda p \sum_{i=1}^n \left(\sigma_i^{\frac{1}{2}}(X^{k_j}) + \epsilon_i^{k_j}\right)^{2p-1} \sigma_i^{\frac{1}{2}}(X^{k_j+1}) \\ \le\ &\frac{L}{2}\left\| X - \left[X^{k_j} - \frac{1}{L}\nabla f(X^{k_j})\right]\right\|_F^2 + 2\lambda p \sum_{i=1}^n \left(\sigma_i^{\frac{1}{2}}(X^{k_j}) + \epsilon_i^{k_j}\right)^{2p-1} \sigma_i^{\frac{1}{2}}(X) \end{aligned}$$

for any $X \in \mathbb{R}^{m\times n}$. Taking limitation on both sides of the above inequality and by virtue of the continuity of $\sigma_i(\cdot)$ and $\nabla f(\cdot)$, as well as (3.22) and (3.24), we obtain that

$$\begin{aligned}&\frac{L}{2}\left\|X^* - \left[X^* - \frac{1}{L}\nabla f(X^*)\right]\right\|_F^2 + 2\lambda p\sum_{i=1}^{n}[\sigma_i(X^*)]^{p-\frac{1}{2}}\sigma_i^{\frac{1}{2}}(X^*)\\ \leq &\frac{L}{2}\left\|X - \left[X^* - \frac{1}{L}\nabla f(X^*)\right]\right\|_F^2 + 2\lambda p\sum_{i=1}^{n}[\sigma_i(X^*)]^{p-\frac{1}{2}}\sigma_i^{\frac{1}{2}}(X)\end{aligned}$$

holds for any $X \in \mathbb{R}^{m\times n}$. This implies that

$$X^* \in \arg\min_{X\in\mathbb{R}^{m\times n}} \frac{L}{2}\left\|X - \left[X^* - \frac{1}{L}\nabla f(X^*)\right]\right\|_F^2 + 2\lambda p\sum_{i=1}^{n}[\sigma_i(X^*)]^{p-\frac{1}{2}}\sigma_i^{\frac{1}{2}}(X). \quad (3.25)$$

Denote $r := \text{rank}(X^*)$ and suppose an SVD of X^* is $X^* = U^*\text{Diag}(\sigma(X^*))(V^*)^\top$ with $(U^*)^\top V^* = (V^*)^\top V^* = I$ and $\sigma(X^*)$ being arranged in non-increasing order. By the optimality condition of problem (3.25) at X^*, it holds

$$\nabla f(X^*) + \lambda p\sum_{i=1}^{r}(\sigma_i(X^*))^{p-1}\, U_i^*\,(V_i^*)^\top = 0. \quad (3.26)$$

Through pre- and port-multiplying (3.26) by $(U^*)^\top$ and V^*, we get

$$(U^*)^\top\nabla f(X^*)V^* + \lambda p\text{Diag}(\sigma^{p-1}(X^*)) = 0, \quad (3.27)$$

where $\sigma^{p-1}(X^*) = (\sigma_1^{p-1}(X^*), \cdots, \sigma_r^{p-1}(X^*), 0, \cdots, 0)^\top$. By considering Definition 2.5, equation (3.27) implies that X^* is a stationary point of problem (1.1). The proof is thus completed. □

4 Numerical experiments

In this section, we conduct numerical experiments on low-rank matrix completion problems to test the performance of the proposed SVHT algorithm. Particularly, low-rank matrix completion problem [1] is described by the following model

$$\begin{aligned}&\min_{X\in\mathbb{R}^{m\times n}} \text{rank(X)}\\ &\text{s.t. } X_{i,j} = M_{i,j}, (i,j) \in \Omega,\end{aligned} \quad (4.1)$$

which aims to fill in the missing entries of a partially observed low-rank matrix $M \in \mathbb{R}^{m\times n}$ with the known entries $\{M_{i,j} : (i,j) \in \Omega\}$.

We compared our SVHT algorithm with three competitive low-rank matrix optimization solvers: SVT [1], augmented Lagrange method for weighted nuclear norm minimization (WNNM) algorithm [12] and low-rank matrix fitting algorithm (LMaFit) [44] for simulation data and real 2D/3D images recovery. The platform is Matlab R2018b under Windows 10 on a desktop of a 3.00GHz CPU and 8.00GB memory.

We emphasize that although we prove the convergence of SVHT algorithm only for $p \in (0, 1/2]$ in theory (Theorem 3.5), SVHT algorithm is also convergent for $p \in (1/2, 1)$ in numerical practice, which can be seen from the following experiments.

4.1 Simulated Data

In this subsection, we use the same way as [1,17,29,32,33] to generate $m\times n$ matrices of rank r. Firstly, two matrices $M_L \in \mathbb{R}^{m\times r}$ and $M_R \in \mathbb{R}^{n\times r}$ with i.i.d. in Gaussian distribution are generated randomly by the MATLAB procedure "randn(m,r)" and "randn(n,r)", then M is obtained by $M = M_L M_R^\top$. The goal is to recover the target matrix $M \in \mathbb{R}^{m\times n}$ of rank r based on some observed entries $\{M_{ij}\}_{(i,j)\in\Omega}$. The observed entries of M is obtained by sampling from M with sampling ratio SR uniformly at random, where SR$= |\Omega|/(mn)$.

4.1.1 Choice of p

First, we set $m = n = 100$, SR$= 0.6$ and let the real rank r increase from 4 to 30 per 2 increases. To choose a good value of p for recovering low-rank matrices, we let p vary among $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each test simulation, we run 10 instances and report the following results: the frequency of success, the average of relative error, the average number of iteration and the average CPU time. The recovery was regarded successful if $\frac{\|M-X^*\|_F}{\|M\|_F} < 10^{-3}$, where X^* stands for the output matrix.

Numerical results of this experiment are displayed in Figure 1.

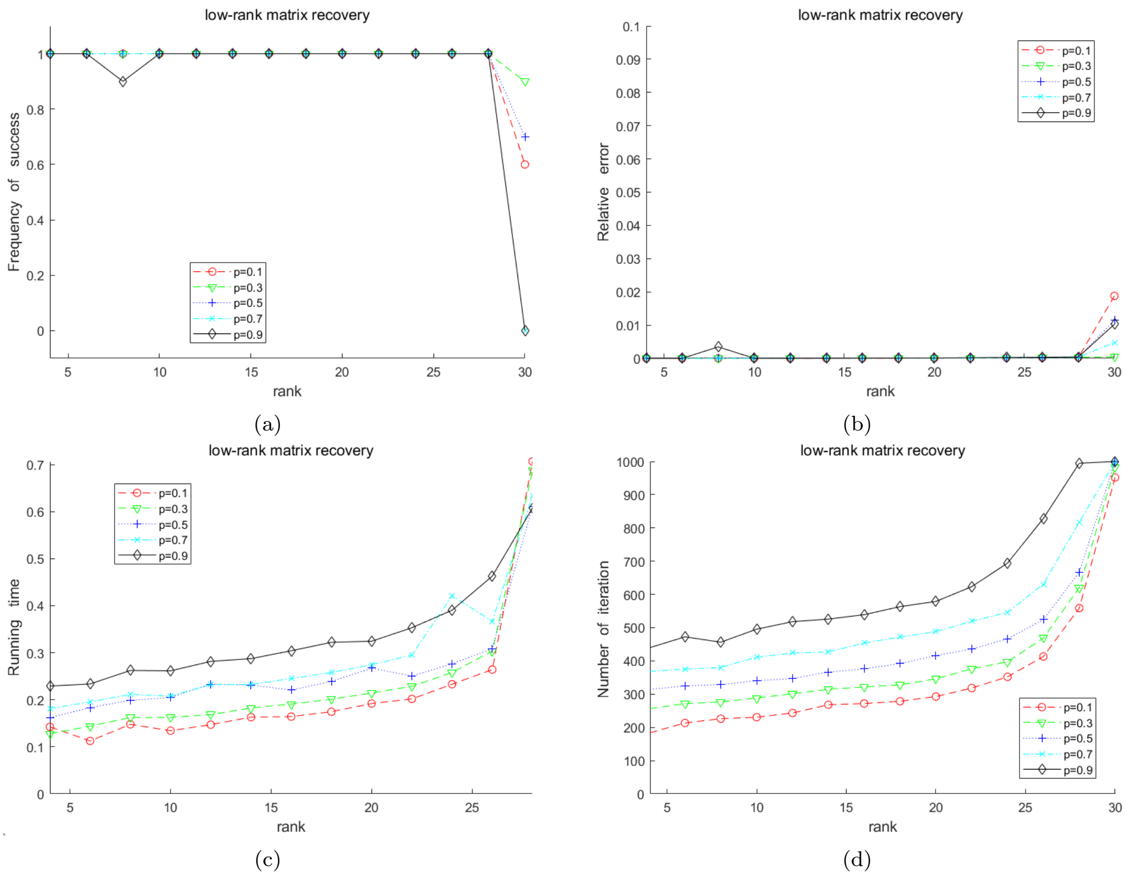


Figure 4. 1

From Figure 1, we can see that among the five values, $p = 0.1$ and $p = 0.3$ cost the least number of iterations and running time for almost all the ranks, while $p = 0.3$ possesses the highest frequency of success and the lowest relative error. This shows that the parameter $p = 0.3$ is the best strategy for SVHT algorithm. Thus, we always use $p = 0.3$ in the following test.

4.1.2 Comparison of four competitive algorithms for random matrices

In this part, we compare the above mentioned four competitive algorithms for randomly generated low-rank matrix completion problems. For a comprehensive comparison, the considered instances are multifarious: including noiseless problems ($\sigma = 0$) and noised problems ($\sigma = 0.01$), small-scale problems and large-scale problems, as well as problems with various sampling rates. In this experiment, for each small-scale test problem, i.e., $m < 3000$, we run 5 instances and record the average results, while for each large-scale test problem, i.e., $m \geq 3000$, we run 2 instances and record the average results. The numerical results of average relative error and average CPU time for this experiment are displayed in Table 4.1.

Table 4.1: Numeric results for randomly generated matrix completion problem

Problems		Methods	$\sigma = 0$		$\sigma = 0.01$	
			Rel_err	Time	Rel_err	Time
m=n=1000	r=30 SR=0.177	SVHT	**1.10e-04**	3.27	1.35e-03	17.01
		WNNM	1.91e-04	22.77	3.34e-03	161.41
		LMaFit	1.25e-03	**0.81**	**1.08e-03**	**0.47**
		SVT	3.74e-01	175.71	3.81e-01	130.35
	r=50 SR=0.205	SVHT	**1.75e-04**	4.77	**1.64e-03**	17.98
		WNNM	2.43e-04	26.38	2.68e-03	127.30
		LMaFit	9.43e-01	**1.02**	9.43e-03	**1.11**
		SVT	5.21e-01	69.94	5.19e-01	74.88
	r=100 SR=0.266	SVHT	**3.27e-04**	14.07	**1.57e-03**	24.95
		WNNM	4.12e-04	67.34	2.19e-03	218.32
		LMaFit	3.02e-01	**2.36**	1.36e-01	**1.91**
		SVT	6.17e-01	38.07	6.16e-01	33.46
m=n=2000	r=30 SR=0.125	SVHT	**1.33e-04**	**12.47**	**1.19e-03**	**65.22**
		WNNM	1.51e-04	77.48	2.34e-03	842.52
		LMaFit	Out of memory		Out of memory	
		SVT	2.36e-01	12065.00	2.39e-01	7296.56
	r=50 SR=0.153	SVHT	**1.30e-04**	13.51	**1.16e-03**	82.14
		WNNM	1.83e-04	97.89	2.33e-03	769.87
		LMaFit	9.57e-04	**3.67**	9.62e-02	**6.37**
		SVT	3.77e-01	9172.60	3.79e-01	2123.98
	r=100 SR=0.214	SVHT	**1.29e-04**	20.20	**1.06e-03**	117.75
		WNNM	2.32e-04	760.45	1.79e-03	996.61
		LMaFit	1.62e-01	**5.19**	9.40e-01	**6.61**
		SVT	4.86e-01	3285.44	4.87e-01	1055.26
m=n=3000	r=50 SR=0.126	SVHT	**1.30e-04**	29.56	**1.01e-03**	186.21
		WNNM	1.58e-04	312.32	2.23e-03	1933.69
		LMaFit	Out of memory		Out of memory	
		SVT	2.93e-01	**0.03**	2.91e-01	12886.48
	r=100 SR=0.154	SVHT	**1.82e-04**	69.89	**1.01e-03**	230.12
		WNNM	2.24e-04	434.49	1.77e-03	2422.82
		LMaFit	9.60e-02	11.50	9.58e-03	11.33
		SVT	5.15e-01	5143.31	5.16e-01	5126.43
	r=150 SR=0.180	SVHT	**2.73e-04**	80.51	**1.02e-03**	294.55
		WNNM	2.79e-04	798.35	1.57e-03	2786.30
		LMaFit	1.42e-01	**14.33**	1.43e-01	**15.05**
		SVT	5.90e-01	3398.58	5.89e-01	3543.63
m=n=5000	r=100 SR=0.087	SVHT	**2.25e-04**	**187.70**	**1.08e-03**	**535.12**
		WNNM	2.34e-04	1592.18	1.84e-03	6250.21
		LMaFit	Out of memory		Out of memory	
		SVT	6.22e-01	26665.22	6.23e-01	26527.10
	r=150 SR=0.089	SVHT	**3.37e-04**	567.36	**1.40e-03**	699.78
		WNNM	3.42e-04	2814.89	1.85e-03	6280.64
		LMaFit	2.99e-01	**37.09**	1.09e-01	**50.68**
		SVT	7.49e-01	17386.02	7.56e-01	17392.59
	r=200 SR=0.118	SVHT	**3.55e-04**	1304.85	**1.15e-03**	1064.08
		WNNM	3.58e-04	3520.59	1.61e-03	8873.66
		LMaFit	8.94e-02	**142.86**	7.61e-01	**98.26**
		SVT	7.20e-01	14586.77	7.19e-01	14639.66

From Table 4.1, we can see that although LMaFit algorithm is always the fastest one among the four algorithms, its accuracy is not very good, even occurs "out of memory" for some instances with very low sampling ratios. While our SVHT algorithm not only possesses the highest accuracy, but also has faster speed than WNNM algorithm and SVT algorithm. For example, for $m = n = 5000, r = 150$, in the case of noiseless, SVHT algorithm only needs 8.9% samplings and the time of 567.36 seconds but can recover the matrix with the relative error reaching 3.37e-04, where the speed is 4.96 times of WNNM algorithm and 30.64 times of SVT algorithm, meanwhile in the case of noise, SVHT algorithm only needs 8.9% samplings and the time of 699.78 seconds but can recover the matrix with the relative error reaching 1.40e-03, where the speed is 8.97 times of WNNM algorithm, 24.85 times of SVT algorithm. In the case of noiseless, the time for SVT algorithm even exceeds a terrifying 17,000 seconds, but its relative error only attains 7.56e-01. Obviously, the numerical results show the great advantages of SVHT algorithm in recovery the randomly generated low-rank matrices.

4.2 Application to Image Inpainting

In this subsection, we use the USC-SIPI image database to evaluate our algorithm, and the 2D/3D images are downloaded from http://sipi.usc.edu/database/.

Firstly, we randomly select three two-dimensional images of size $512 * 512$ from this database. The original images are not low-tank in nature, we construct low-rank images based on the randomly and noisy sampling from the original images as their low-rank approximations. For this experiment, we set sampling ratio SR= 0.3, Gaussian noise with standard deviation $Gn = 10^{-2}$ and the recoved rank $r = 50$. We use the following four indexes: peak signal to noise ratio (PSNR) [13], structural similarity (SSIM) [40], root mean square error (RMSE) [40] and CPU time to evaluate the numerical performances of the compared algorithms for image inpainting, where

$$\mathrm{PSNR} := 10\log_{10}\left(\frac{mn * 255^2}{\|M - X^*\|_F^2}\right) \quad \text{and} \quad \mathrm{RMSE} := \sqrt{\frac{\|X^* - M\|_F^2}{mn}}.$$

Obviously, the higher PSNR and SSIM values and smaller RMSE and CPU time values represents the better recovery performance. Numerical results of this experiment are displayed in Figure 4.2 and Table 4.2.

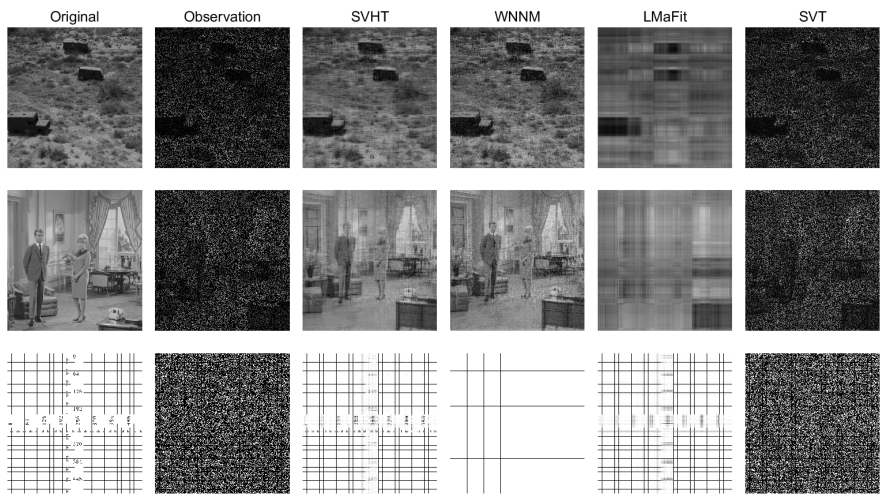


Figure 4. 2. Image recovery results with Gaussian noise ($Gn = 10^{-2}$)

Table 4.2: Image recovery results of all the compared algorithms

Image \ Method		SVHT	WNNM	LMaFit	SVT
Truck	PSNR	**24.2104**	21.7740	21.4179	10.9148
	SSIM	**0.4970**	0.3845	0.3634	0.0877
	RMSE	**15.7044**	20.7893	21.6593	72.5776
	Time	8.6082	22.1494	**0.0645**	0.8427
Couple	PSNR	**24.0853**	22.4996	21.3421	8.1418
	SSIM	**0.6047**	0.5189	0.5228	0.0439
	RMSE	**15.9321**	19.1231	21.8493	99.8740
	Time	8.4387	20.9571	**0.1067**	0.5997
Ruler	PSNR	**20.7958**	10.8142	19.4680	2.4527
	SSIM	**0.8820**	0.4114	0.8686	0.1035
	RMSE	**23.2676**	73.4227	27.1105	192.2675
	Time	7.5871	1.1386	**0.2051**	0.3859

From Figure 4.2, we can see that the images restored by SVHT algorithm are relatively clearer than other algorithms, which shows that SVHT algorithm can effectively recover the images in the case of noise. In order to compare the recovery effect of the four algorithms more clearly, we display PSNR, SSIM, RMSE and CPU time of the four algorithms in Table 4.2. We can see that although SVHT algorithm is slower than LMaFit algorithm and SVT algorithm, but outperforms them and WNNM algorithm in terms of PSNR, SSIM and RMSE. While SVHT algorithm has higher PNSR and SSIM values and smaller RMSE and CPU time values than WNNM algorithm for most of the cases. In summary, SVHT algorithm performs the best among the four algorithms.

Secondly, we apply the four algorithms to recover three-dimensional images. We select four three-dimensional images of size $256*256*3$ for this experiment, whose entries denote the pixels of the corresponding images. We sample 30% pixels of each image and add Gaussian noise with standard deviation $Gn = 10^{-2}$ as observation data, based on which we reconstruct an image of rank 30 by each algorithm as low-rank approximation to each original image, that is, SR= 30% and $r = 30$. The recovered images are shown in Figure 4.3 and the numeric results of PSNR, SSIM, RMSE and Time are reported in Table 4.3.

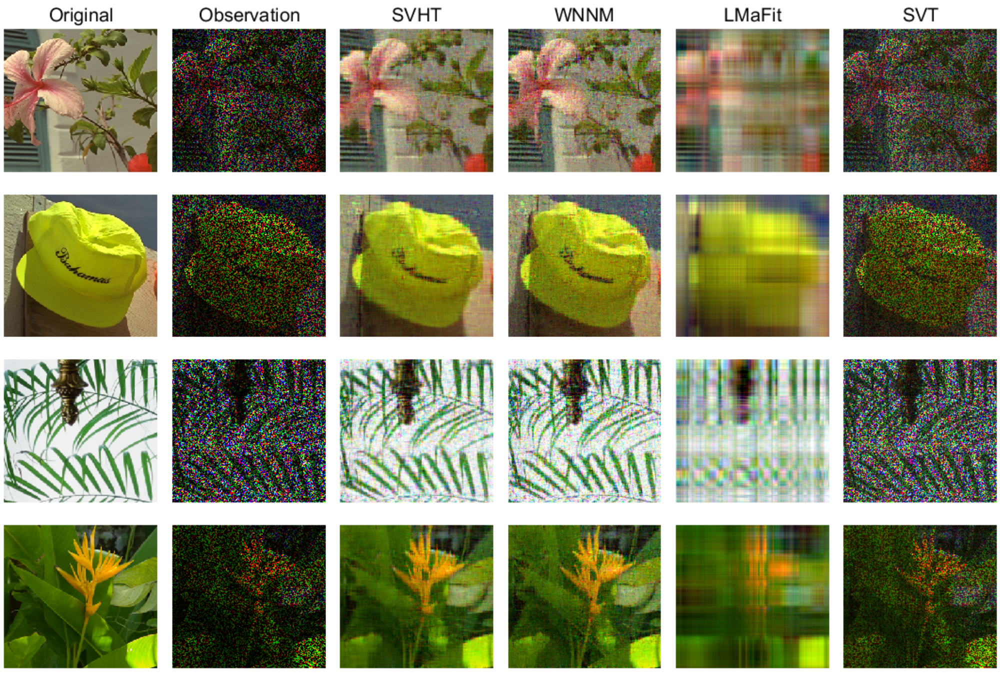


Figure 4. 3. Recovery results for 3D images

Table 4.3: Numeric results for 3D images recovery of the four compared algorithms

Image \ Method		SVHT	WNNM	LMaFit	SVT
Flower	PSNR	**21.2013**	19.1316	18.5224	10.2418
	SSIM	**0.4326**	0.3313	0.3544	0.1213
	RMSE	**22.2063**	28.1811	30.2289	78.4243
	Time	2.3871	4.7171	**0.0177**	0.5562
Hat	PSNR	**23.4241**	21.5304	20.3392	10.2523
	SSIM	**0.4960**	0.3475	0.4900	0.1772
	RMSE	**17.1923**	21.3806	24.5233	78.3292
	Time	2.2722	3.8314	**0.0445**	1.5539
Leaves	PSNR	**15.7829**	13.7406	12.5215	5.5154
	SSIM	**0.4553**	0.3736	0.2180	0.1225
	RMSE	**41.4376**	52.4217	60.3208	135.1353
	Time	2.0403	3.6539	**0.0079**	0.2139
Plants	PSNR	**24.2323**	22.2863	20.6571	14.2017
	SSIM	**0.5584**	0.3782	0.5107	0.3324
	RMSE	**15.6649**	19.5985	23.6420	49.7113
	Time	2.2751	2.8642	**0.0517**	2.2638

From Figure 4.3, we can also see that the images recovered by SVHT algorithm is clearer than that recovered by the other three algorithms. From Table 4.3, we can see that for most of the cases, SVHT algorithm still outputs the highest PNSR, the highest SSIM, the lowest RMSE and the third fastest speed. Although LMaFit algorithm and SVT algorithm are faster than SVHT algorithm, the quality of their recovered images are poor.

4.3 Application to MRI Volume Dataset

We adopt the MRI volume dataset to test our algorithm further. We selected the 50th slice of the CThead and MRbrain which are downloaded from http://graphics.stanford.edu/data/voldata and of size 256*256. Similarly, we set sampling rate SR= 0.3, Gaussian noise with standard deviation $Gn = 10^{-2}$ and the recovered rank $r = 30$. Numerical results of this experiment are displayed in Figures 4.4 and 4.5.

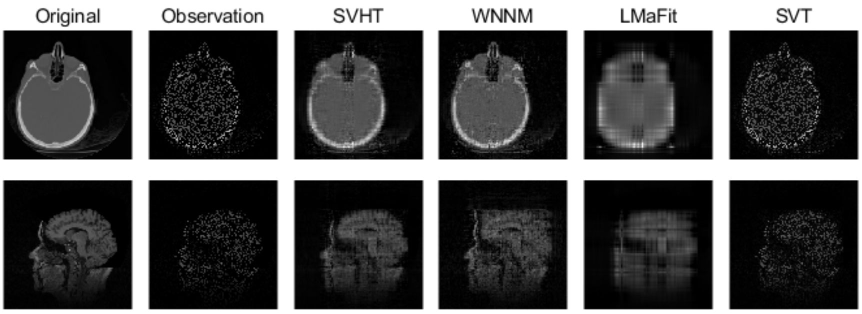
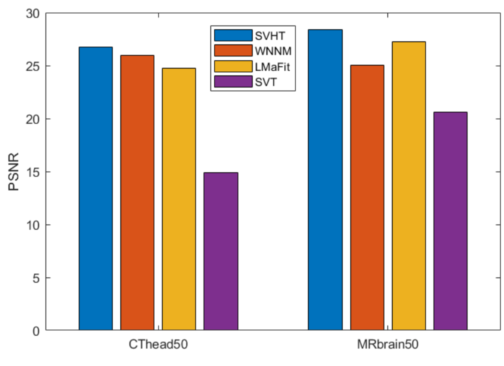
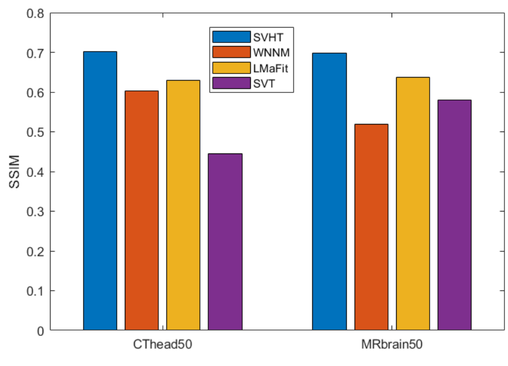


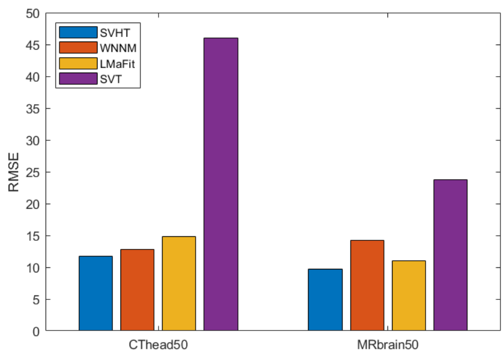
Figure 4. 4. Image recovery results with Gaussian noise ($Gn = 10^{-2}$))



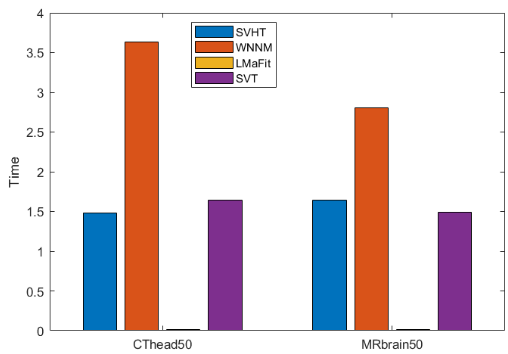
(a)



(b)



(c)



(d)

Figure 4. 5. Image recovery results of all the compared algorithms

At first, we can see from Figure 4.4 that the images recovered by SVHT algorithm are more detailed. Secondly, from the histogram 4.5, our stone columns in PSNR and SSIM are higher and in RMSE and Time are lower, which indicates that our results are closer to the real images.

To summarize the numeric experiments on random data and real image data, the results show that SVHT algorithm works very well in almost all the cases, and it is very competitive compared with the three state-of-the-art algorithms.

5 Conclusion

In this paper, we studied the low-rank matrix optimization problem by relaxing the rank of matrix to the ℓ_p $(0 < p < 1)$ matrix quasi-norm. Based on the locally $\ell_{1/2}$ approximation of ℓ_p $(0 < p < 1)$ matrix quasi-norm and the closed-form solution of the $\ell_{1/2}$ subproblem, we proposed a singular value half thresholding algorithm for the ℓ_p regularized matrix optimization problem. By constructing Lipschitz and non-Lipschitz approximate functions of the objective function, we proved that any accumulation point of the sequence generated by SVHT algorithm is a first-order stationary point of the problem. In numerical experiments, we test SVHT algorithm by low-rank matrix completion problem on simulation data, USC-SIPI image data and MRI volume image data. Extensive numerical results show that SVHT algorithm is very competitive for low-rank matrix optimization problems in comparison with some popular algorithms.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (12261020), the Guizhou Provincial Science and Technology Program (ZK[2021]009), the Foundation for Selected Excellent Project of Guizhou Province for High-level Talents Back from Overseas ([2018]03), and the Research Foundation for Postgraduates of Guizhou Province (YJSCXJH[2020]085).

References

1. Cai, J., Candès, E., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim., 20(4), 1956-1982 (2010)
2. Candès, E., Wakin, M., Boyd, S.: Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl., 14, 877-905 (2008)
3. Cao, W., Sun, J., Xu, Z.: Fast image deconvolution using closed-form thresholding formulas of $\ell_q(q = 1/2, 2/3)$ regularization. J. Vis. Commun. Image. Represent., 24 (1), 31-41 (2013)
4. Chartrand, R., Yin,W.: Iteratively reweighted algorithms for compressive sensing. In: 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 3869-3872 (2008)
5. Chen, C.H., He, B.S., Yuan, X.M.: Matrix completion via alternating direction methods. IMA J. Numer. Anal., 32, 227-245 (2012)
6. Chen, X., Zhou, W.: Convergence of reweighted ℓ_1 minimization algorithm for ℓ_2-ℓ_p minimization. Comput. Optim. Appl., 59(1-2), 47-61 (2014)
7. Daubechies, I., DeVore, R., Fornasier, M., Gunturk, C.S.: Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure. Appl. Math., 63, 1-38 (2010)
8. Efron, B., Hastie, T., Johnstone, I.: Least angle regression. Ann Statist., 32, 407-499 (2004)
9. Fazel, M., Hindi, H., Boyd, S.: A rank minimization heuristic with application to minimum order system approximation. In: Proceedings of the American Control Conference (ACC), 4734-4739 (2001)

10. Fazel, M., Hindi, H., Boyd, S.: Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In: Proceedings of the American Control Conference (ACC), 2156-2162 (2003)
11. Fornasier, M., Rauhut, H., Ward, R.: Low-rank matrix recovery via iteratively reweighted least squares minimization. SIAM J. Optim., 21, 1614-1640 (2011)
12. Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., Zhang, L.: Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vis., 121(2), 183-208 (2017)
13. He, L., Wang, Y., Xiang, Z.: Support driven wavelet frame-based image deblurring, Inf. Sci., 479, 250-269 (2019)
14. Jain, P., Meka, R., Dhillon, I.: Guaranteed rank minimization via singular value projection. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS)., 937-945 (2010)
15. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE Trans. Inform. Theory., 56(6), 2980-2998 (2010)
16. Lai, M., Wang, J.: An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of underdetermined linear systems. SIAM J. Optim., 21, 82-101 (2010)
17. Lai, M., Xu, Y., Yin, W.: Improved iteratively rewighted least squares for unconstrained smoothed ℓ_p minimization. SIAM J. Numer. Anal., 51(2), 927-957 (2013)
18. Lee, K., Bresler, Y.: ADMiRA: atomic decomposition for minimum rank approximation, IEEE Trans. Inf. Theory., 56(9), 4402-4416 (2010)
19. Lewis, A.S., Sendov, H.S.: Nonsmooth analysis of singular values. Part II: applications. Set Valued Anal., 13, 243-264 (2005)
20. Li, Y.F., Shang, K., Huang, Z.H.: A singular value p-shrinkage thresholding algorithm for low rank matrix recovery. Comput. Optim. Appl., 73, 453-476 (2019).
21. Li, Y.F., Zhang, Y.J., Huang, Z.H.: A reweighted nuclear norm minimization algorithm for low rank matrix recovery. J. Comput. Appl. Math., 263, 338-350 (2014)
22. Liu, T., Lu, Z., Chen, X., Dai, Y.H.: An exact penalty method for semidefinite-box-constrained low-rank matrix optimization problems. IMA J. Numer. Anal., 40(1), 563-586 (2020)
23. Liu, Y., Sun, D., Toh, K.-C.: An implementable proximal point algorithmic framework for nuclear norm minimization. Math. Program., 133(1-2), 399-436 (2012)
24. Lu, Z .: Iterative Reweighted minimization methods for l_p regularized unconstrained nonlinear programming. Math. Program., 147(1-2), 277-307 (2014)
25. Lu Z, Zhang Y, Li X. Penalty decomposition methods for rank minimization. Optim. Method Softw., 30: 531-558 (2015)
26. Lu, Z., Zhang, Y., Lu, J.: ℓ_p regularized low-rank approximation via iterative reweighted singular value minimization. Comput. Optim. Appl., 68, 619-642 (2017)
27. Lu, Y., Zhang, L., Wu, J.: A smoothing majorization method for ℓ_2^2-ℓ_p^p matrix minimization. Optim. Methods Softw., 30(4), 682-705 (2015)
28. Marjanovic, G., Solo, V.: On ℓ_q optimization and matrix completion. IEEE Trans. Signal Process., 60(11), 5714-5724 (2012)
29. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Math. Program., 128(1-2), 321-353 (2011)
30. Mohan, K., Fazel, M.: Iterative reweighted algorithms for matrix rank minimization. J. Mach. Learn. Res., 13(1), 3441-3473 (2012)
31. Mohan, K., Fazel, M.: Iterative reweighted least squares for matrix rank minimization. In: 48th Annual Allerton Conference on Communication, Control, and Computing, 653-661 (2010)
32. Peng, D., Xiu, N., Yu, J.: $S_{1/2}$ regularization methods and fixed point algorithms for affine rank minimization problems. Comput. Optim. Appl., 67, 543-569 (2017)
33. Peng, D., Xiu, N., Yu, J.: Global optimality conditions and fixed point continuation algorithm for non-Lipschitz ℓ_p regularized matrix minimization. Sci. China Math., 61, 1139-1152 (2018)
34. Recht, B., Fazel, M., Parrilo, P.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev., 52(3), 471-501 (2010)
35. Rohde, A., Tsybakov, A.: Estimation of high-dimensional low-rank matrices. Ann. Statist., 39, 887-930 (2011)
36. Shwartz, S., Gonen, A., Shamir, O.: Large-scale convex minimization with a low rank constraint. In: Proceedings of the 28th International Conference on Machine Learning (ICML), 329-336 (2011)
37. Skelton, R., Iwasaki, T., Grigoriadis, K.: A Unified Algebraic Approach to Linear Control Design. Abingdon: Taylor and Francis, 1998
38. Toh, K.C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac. J. Optim., 6(3), 615-640 (2010)

39. Wang, H., Liu, X., Chen, X., Yuan, Y.X.: SNIG property of matrix low-rank factorization model. J. Comput. Math., 36(3), 374-390 (2018)
40. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process., 13(4), 600-612 (2004)
41. Wang, Z., Gao, C., Luo, X.: Accelerated inexact matrix completion algorithm via closed-form q-thresholding ($q = 1/2, 2/3$) operator. Int. J. Mach. Learn. Cybern., 11(10), 2327-2339 (2020)
42. Wang, Z., Gao, C., Luo, X.: Fast and efficient algorithm for matrix completion via closed-form 2/3-thresholding operator. Neurocomputing, 330, 212-222 (2019)
43. Wang, Z., Lai, M., Lu, Z., Fan, W., Davulcu, H., Ye, J.: Rank-one matrix pursuit for matrix completion. In: Proceedings of the 31st international conference on Machine learning (ICML), 91-99 (2014)
44. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math. Program. Comput., 4(4), 333-361 (2012)
45. Xiu, X., Kong, L., Li, Y., Qi, H.: Iterative reweighted methods for ℓ_1-ℓ_p minimization. Comput. Optim. Appl., 70(1), 201-219 (2018)
46. Xu, Z., Chang, X., Xu, F., Zhang, H.: $\ell_{1/2}$ regularization: a thresholding representation theory and a fast solver. IEEE Trans. Neural Netw. Learn. Syst., 23(7), 1013-1027 (2012)
47. Yang, J., Yuan, X.M.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math. Comput., 82(281), 301-329 (2013)