A Nonconvex Method to Low-rank Matrix Completion

HAIZHEN HE\textsuperscript{1,2}, ANGANG CUI\textsuperscript{3}, HONG YANG\textsuperscript{3}, AND MENG WEN\textsuperscript{4}

\textsuperscript{1}School of International Education, Yulin University, Yulin 719000, China
\textsuperscript{2}Endicott College, Woosong University, Daejeon 34606, Republic of Korea
\textsuperscript{3}School of Mathematics and Statistics, Yulin University, Yulin 719000, China
\textsuperscript{4}School of Science, Xi’an Polytechnic University, Xi’an 710048, China

Corresponding author: Angang Cui (e-mail: cuiangang@yulinu.edu.cn).

The work was supported by the National Natural Science Foundation of China (11871392, 12001416), the Natural Science Basic Research Program of Shaanxi (2021JM-002), the Science Foundation of the Education Department of Shaanxi Province (21JK1008), and the Doctoral Research Project of Yulin University (21GK04).

ABSTRACT In recent years, the problem of recovering a low-rank matrix from partial entries, known as low-rank matrix completion problem, has attracted much attention in many applications. However, it is a NP-hard problem due to the nonconvexity nature of the matrix rank function. In this paper, a rank Laplace function is studied to recover the low-rank matrices. Firstly, we propose an iterative Laplace thresholding algorithm to solve the regularized Laplace low-rank matrix completion problem. Secondly, some other iterative thresholding algorithms are designed to recover the low-rank matrices. Finally, we provide a series of numerical simulations to test the proposed algorithms on some low-rank matrix completion and image inpainting problems, and the results show that our algorithms perform better than some state-of-art methods in recovering the low-rank matrices.

INDEX TERMS Matrix completion, Matrix rank Laplace function, Iterative Laplace thresholding algorithm, Iterative difference Laplace thresholding algorithm.

I. INTRODUCTION

In recent years, the low-rank matrix completion problem has attracted much attention in many applications such as video denoising [1], [2], image inpainting [3], [4], system identification [5], and so on. In fact, for a low-rank matrix \( M \in \mathbb{R}^{m \times n} \) one would like to know as precisely as possible. However, the only information available about \( M \) is a sampled set of entries \( M_{i,j} \), \((i, j) \in \Omega\), where \( \Omega \subset \{1, 2, \cdots, m\} \times \{1, 2, \cdots, n\} \) is the set of indices of samples. The goal of low-rank matrix completion problem is to recover the low-rank matrix from its partial entries \( M_{i,j} \), \((i, j) \in \Omega\). In mathematics, the low-rank matrix completion problem can be modeled as the following minimization problem:

\[
\min_{X \in \mathbb{R}^{m \times n}} \text{rank}(X) \quad \text{s.t.} \quad X_{i,j} = M_{i,j}, \; \forall \; (i, j) \in \Omega. \quad (1)
\]

Throughout this paper, without loss of generality, we assume that \( m \leq n \). For convenience, if we summarize the information available via \( P_\Omega(M) \), where \( P_\Omega : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n} \) is a sampling operator that keeps the entries in \( \Omega \) unchanged and those outside \( \Omega \) zeros, i.e.,

\[
[P_\Omega(X)]_{i,j} = \begin{cases} X_{i,j}, & \text{if } (i, j) \in \Omega; \\ 0, & \text{if } (i, j) \notin \Omega. \end{cases} \quad (2)
\]

the low-rank matrix completion problem (2) can be reformulated as

\[
\min_{X \in \mathbb{R}^{m \times n}} \text{rank}(X) \quad \text{s.t.} \quad P_\Omega(X) = P_\Omega(M). \quad (3)
\]

Unfortunately, due to the nonconvexity nature of the matrix rank function \( \text{rank}(X) \), problem (3) is NP-hard [6], and cannot be solved efficiently by the algorithms. In fact, all known algorithms for exactly solving it are doubly exponential in theory and in practice. To avoid such a difficulty, some researchers introduced the convex surrogate function of \( \text{rank}(X) \), namely, nuclear-norm \( \|X\|_* \) of \( X \), to relax the rank of matrix \( X \), which leads to the following nuclear-norm low-rank matrix completion problem [6]–[12]:

\[
\min_{X \in \mathbb{R}^{m \times n}} \|X\|_* \quad \text{s.t.} \quad P_\Omega(X) = P_\Omega(M), \quad (4)
\]

where \( \|X\|_* = \sum_{i=1}^{n} \sigma_i(X) \) denotes the matrix nuclear norm and \( \sigma_i(X) \) is the \( i \)-th largest singular value of matrix \( X \in \mathbb{R}^{m \times n} \). As a convex relaxation for the NP-hard problem (3), problem (4) and its regularized version possess many algorithmic advantages. The representative algorithms include semidefinite programming and interior point SDP.
singer value thresholding (SVT) algorithms [7], accelerated proximal gradient (APG) algorithm [14], proximal point algorithms [15], fixed point and Bregman iterative algorithms [8], linearized augmented Lagrangian method and alternating direction method [16], and so on. Although the problem (4) and its regularized version possess many algorithmic advantages, nuclear-norm always performs not very well in recovering the low-rank matrices. In fact, nuclear-norm always needs more observations to recover a real low-rank matrix.

In this way, the idea of using the rank Laplace function defined in (6) to relax the rank function $\text{rank}(X)$ comes very naturally. Using the rank Laplace function $|||X|||_\gamma$ to relax the rank function $\text{rank}(X)$ in problem (3) leads to the following Laplace low-rank matrix completion problem:

$$\min |||X|||_\gamma \text{ s.t. } P_\Omega(X) = P_\Omega(M). \quad (8)$$

Usually, solving problem (8) can be transformed into solving its regularized version which is described as the following regularized Laplace low-rank matrix completion problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} ||P_\Omega(X) - P_\Omega(M)||^2_F + \lambda |||X|||_\gamma \right\}, \quad (9)$$

where $\lambda > 0$ is the regularized parameter.

The rest of the paper is organized as follows. In Section II, we present some useful notions and crucial preliminary results that are used in this paper. In Section III, an iterative Laplace thresholding algorithm is proposed to solve problem (9). In Section IV, three types of iterative difference Laplace thresholding algorithms are studied for low-rank matrix completion. In Section V, we provide a series of numerical simulations to test our algorithms on some low-rank matrix completion and image inpainting problems. We conclude this paper in Section VI.

II. NOTIONS AND PRELIMINARY RESULTS

In this section, we present some useful notions and preliminary results that are used in this paper.

A. NOTIONS

The space of $m \times n$ real matrices is denoted by $\mathbb{R}^{m \times n}$. Given any matrices $X, Y \in \mathbb{R}^{m \times n}$, $(X, Y) = \text{Tr}(Y^\top X)$ denotes the inner product of matrices $X$ and $Y$, where $\text{Tr}(Y^\top)$ denotes the trace of the matrix $Y^\top X$. The Frobenius norm of matrix $X \in \mathbb{R}^{m \times n}$ is denoted by $|||X|||_F = \sqrt{\text{Tr}(X^\top X)}$. The singular value decomposition (SVD) of matrix $X \in \mathbb{R}^{m \times n}$ is defined as

$$X = U[\text{diag}((\sigma_i(X))_{1 \leq i \leq m}), 0_{m \times (n-m)}]V^\top,$$

where $U$ is a $m \times m$ unitary matrix, $V$ is a $n \times n$ unitary matrix, $\text{diag}((\sigma_i(X))_{1 \leq i \leq m}) \in \mathbb{R}^{m \times n}$ is a diagonal matrix of the singular values of matrix $X$ arranged in descending order $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_m(X) \geq 0$, and $0_{m \times (n-m)} \in \mathbb{R}^{m \times (n-m)}$ is a $m \times (n-m)$ zero matrix.

B. PRELIMINARY RESULTS

For any $t \in \mathbb{R}$, consider the following minimization problem

$$h_t(\gamma) = \arg \min_{\theta \geq 0} \left\{ \frac{1}{2} (x-t)^2 + \lambda (1 - e^{-\gamma t}) \right\}, \quad (10)$$

where $\gamma > 0$ and $\lambda > 0$. For any $x \geq 0$, the derivative of $\theta_t(x)$ is given by

$$[\theta_t(x)]' = x - t + \frac{\lambda}{\gamma} e^{-\gamma t}. \quad (11)$$

Similar to [17, Theorem 1], we can get the following key lemma.
Theorem 1. Suppose $\gamma > \sqrt{\lambda}$ and $x = g_{\lambda, \gamma}(t)$ is the implicit function determined by the equation $[h_{\lambda}(x)]^2 = 0$. Let $x^*$ be the global optimal solution of problem (10). Then we can get that 

$$x^* = h_{\lambda}(t) := \begin{cases} 
    h_{\lambda, \gamma}(t), & t > t_{\lambda, \gamma} \\
    0, & t \leq t_{\lambda, \gamma}
\end{cases}$$  \hspace{1cm} (12)$$

where $t_{\lambda, \gamma} = \frac{4}{\gamma}$.

Definition 1. (Laplace thresholding function) Suppose $t \in \mathbb{R}$, for any $\lambda > 0$, the function $h_{\lambda}(t)$ defined in (12) is called the Laplace thresholding function.

Definition 2. (Vector Laplace thresholding operator) Suppose $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$, for any $\lambda > 0$, the vector Laplace thresholding operator $H_{\lambda}(x)$ is defined as

$$H_{\lambda}(x) := (h_{\lambda}(x_1), h_{\lambda}(x_2), \ldots, h_{\lambda}(x_n))^T.$$ \hspace{1cm} (13)

Definition 3. (Rank Laplace thresholding operator) Let $X = U[\text{diag}((\sigma_i(X))_{1 \leq i \leq m}), 0_{m \times (n-m)}]^T \in \mathbb{R}^{m \times n}$. For any $\lambda > 0$, we defined the rank Laplace thresholding operator $G_{\lambda} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ as

$$G_{\lambda}(X) := X[\text{diag}(H_{\lambda}((\sigma_i(X))_{1 \leq i \leq m}), 0_{m \times (n-m)})]^T.$$ \hspace{1cm} (14)

Definition 4. (See [18], [19]) $T : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}_+$ is a smooth function of type $C^{1,1}$, i.e., the gradient is Lipschitz continuous,

$$||\nabla T(X) - \nabla T(Z)||_F \leq L(T)||X - Z||_F$$ \hspace{1cm} (15)

for every $X, Z \in \mathbb{R}^{m \times n}$, where $L(T) > 0$ is the Lipschitz constant of $\nabla T$.

Lemma 1. (See [18]-[20]) Let $T : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}_+$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant $L(T)$. Then, for any $\mu \geq L(T)$,

$$T(X) \leq T(Z) + (X - Z, \nabla T(Z)) + \frac{\mu}{2}||X - Z||_F^2$$ \hspace{1cm} (16)

for every $X, Z \in \mathbb{R}^{m \times n}$.

Moreover, we also need the definition of the singular value shrinkage operator [7], and it simply applies the soft thresholding operator [21], [22] defined on vector to the singular values of a matrix.

Definition 5. Let $Y = U[\text{diag}((\sigma_i(Y))_{1 \leq i \leq m}), 0_{m \times (n-m)}]^T \in \mathbb{R}^{m \times n}$, for any $\lambda > 0$, suppose that

$$D_{\lambda}(Y) = \arg \min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2}||X - Y||_F^2 + \lambda||X||_1 \right\},$$ \hspace{1cm} (17)

then the soft thresholding operator $D_{\lambda}(\cdot)$ can be expressed as

$$D_{\lambda}(Y) = U[\text{diag}((\sigma_i(Y) - \lambda_i)_+), 0_{m \times (n-m)}]^T,$$ \hspace{1cm} (18)

where $\nu_+$ is the positive part of $\nu$, and $\nu_+ = \max(\nu, 0)$.

III. ITERATIVE LAPLACE THRESHOLDING ALGORITHM FOR LOW-RANK MATRIX COMPLETION

In this section, the proximal gradient descent algorithm is utilized to solve problem (9) and generate an iterative thresholding algorithm for low-rank matrix completion.

Let

$$G(X) = \frac{1}{2}||P_{\Omega}(X) - P_{\Omega}(M)||_F^2.$$ \hspace{1cm} (19)

Then the objective function

$$F(X) = \frac{1}{2}||P_{\Omega}(X) - P_{\Omega}(M)||_F^2 + \lambda||X||_Y,$$ \hspace{1cm} (20)

of problem (9) can be rewritten as

$$F(X) = G(X) + \lambda||X||_Y.$$ \hspace{1cm} (21)

Clearly, $G(X)$ is a smooth function of type $C^{1,1}$. Therefore, by Lemma 1, for any $Z \in R^{m \times n}$, we have

$$G(X) \leq G(Z) + \langle X - Z, \nabla G(Z) \rangle + \frac{\mu}{2}||X - Z||_F^2,$$ \hspace{1cm} (22)

where $\mu \geq 1$.

In this paper, we use the function

$$P(X, Z) = G(Z) + \langle X - Z, \nabla G(Z) \rangle + \frac{\mu}{2}||X - Z||_F^2 - \frac{1}{2\mu}||\nabla G(Z)||_F^2,$$ \hspace{1cm} (23)

to approximate the objective function of problem (9). The function $P(X, Z)$ can also be seen as the linearization of $G(X)$ at $Z$ and add the term $\lambda||X||_Y$, and it can be rewritten as

$$P(X, Z) = \frac{1}{2}||X - (Z - \frac{1}{\mu}\nabla G(Z))||_F^2 + \lambda||X||_Y + g(Z).$$ \hspace{1cm} (24)

By above approximation, problem (9) can be approximated as the following minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} P(X, Z).$$ \hspace{1cm} (25)

Ignoring constant terms $g(Z)$ and $-\frac{1}{2\mu}||\nabla g(Z)||_F^2$, solving problem (24) is equivalent to solving the following minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2}||X - B_{\mu}(Z)||_F^2 + \frac{\lambda}{\mu}||X||_Y \right\},$$ \hspace{1cm} (26)

where $B_{\mu}(Z) = Z - \frac{1}{\mu}(P_{\Omega}(Z) - P_{\Omega}(M))$.

We evaluate the global optimal solution of problem (24) iteratively by setting $Z = X^k$, which leads to the following iterative scheme:

$$X^{k+1} = \arg \min_{X \in \mathbb{R}^{m \times n}} P(X, X^k).$$ \hspace{1cm} (27)

This means that, in each iteration, if the following minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2}||X - B_{\mu}(X^k)||_F^2 + \frac{\lambda}{\mu}||X||_Y \right\}$$ \hspace{1cm} (28)

is solved.
can be solved, we can easily generate an iterative thresholding algorithm to solve problem (9).

According to Theorem 1 and Definition 3, problem (27) permits a thresholding representation theory for its global optimal solution.

**Theorem 2.** For any $\gamma > \sqrt{\frac{2}{\mu}}$, the global optimal solution $X^{k+1}$ of the problem (27) can be expressed as

$$X^{k+1} = \mathcal{H}_\frac{\lambda}{\mu}(B_\mu(X^k)),$$

where $\mathcal{H}_\frac{\lambda}{\mu}(\cdot)$ is obtained by replacing $\lambda$ with $\frac{\lambda}{\mu}$ in $\mathcal{H}_\lambda(\cdot)$.

**Proof.** By von Neumann’s trace inequality, we have the inequality

$$\frac{1}{2}||X - B_\mu(X^k)||_F^2 \\
= \frac{1}{2} \left( \text{Tr}(X^T X) - 2\text{Tr}(B_\mu(X^k)^T X) + \text{Tr}((B_\mu(X^k))^T(B_\mu(X^k))) \right) \\
= \frac{1}{2} \left( \sum_{i=1}^{m} \sigma_i^2(X) - 2\sum_{i=1}^{m} \sigma_i(B_\mu(X^k))\sigma_i(X) + \sum_{i=1}^{m} \sigma_i^2(B_\mu(X^k)) \right)$$

$$\geq \frac{1}{2} \left( \sum_{i=1}^{m} \sigma_i^2(X) - 2 \sum_{i=1}^{m} \sigma_i(B_\mu(X^k))\sigma_i(X) + \sum_{i=1}^{m} \sigma_i^2(B_\mu(X^k)) \right)$$

Notice that the equality holds if the matrix $X$ admits the singular value decomposition

$$X = U_B[\text{diag}(\sigma_i(X))]_{1\leq i \leq r}, 0_{m \times (m-r)}](V_B)^T,$$

where $U_B$ and $V_B$ are the left and right orthonormal matrices in the SVD of matrix $B_\mu(X^k)$. Therefore, solving problem (27) reduces to solve the following minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \sum_{i=1}^{m} \left[ \frac{1}{2}(\sigma_i(X) - \sigma_i(B_\mu(X^k)))^2 + \frac{\lambda}{\mu}(1 - e^{-\frac{\sigma_i(X)}{\gamma}}) \right].$$

The objective function in (30) is separable, solving the minimization problem (30) is equivalent to solving the following $n$ subproblems, for $i \in \{1, 2, \cdots, n\}$,

$$\min_{X \in \mathbb{R}^{m \times n}} \left[ \frac{1}{2}(\sigma_i(X) - \sigma_i(B_\mu(X^k)))^2 + \frac{\lambda}{\mu}(1 - e^{-\frac{\sigma_i(X)}{\gamma}}) \right].$$

By Theorem 1, for $\gamma > \sqrt{\frac{2}{\mu}}$, the global optimal solution of problem (31) can be expressed as

$$\sigma_i(X^{k+1}) = h_\frac{\lambda}{\mu}(\sigma_i(B_\mu(X^k))),$$

where $h_\frac{\lambda}{\mu}(\cdot)$ is obtained by replacing $\lambda$ with $\frac{\lambda}{\mu}$ in $g_\lambda(\cdot)$. Therefore, it follows that

$$X^{k+1} = \mathcal{H}_\frac{\lambda}{\mu}(B_\mu(X^k))$$

is the global optimal solution of problem (27). \hfill $\square$

Initialized $X^0 \in \mathbb{R}^{m \times n}$, iteration (28) describes an iterative thresholding algorithm to solve problem (9). In this paper, we call this method the iterative Laplace thresholding algorithm (ILTA), and we depict it in Algorithm 1.

**Algorithm 1 Iterative Laplace thresholding algorithm (ILTA)**

**Input:** $\lambda > 0, \mu > 1, \gamma > \sqrt{\frac{2}{\mu}}$;

**Initialize:** $X^0 \in \mathbb{R}^{m \times n}, k = 0$;

**while** not converged, **do**

- $B_\mu(X^k) = X^k - (\mathcal{P}_\Omega(X^k) - \mathcal{P}_\Omega(M))$;
- $X^{k+1} = \mathcal{H}_\frac{\lambda}{\mu}(B_\mu(X^k))$;
- $k = k + 1$;

**end while**

**Output:** $X^{opt}$

Similar to [23], Proposition 4.1], we can get the following convergence results for ILTA.

**Theorem 3.** Suppose $F(X^0) < +\infty, \lambda > 0, \mu > 1$ and $\gamma > \sqrt{\frac{2}{\mu}}$. Let $\{X^k\}$ be the sequence generated by ILTA. Then the following two statements hold:

1. The sequence $\{F(X^k)\}$ is monotonically decreasing and converges to $F(X^*)$, where $\{X^*\}$ is any accumulation point of the sequence $\{X^k\}$.
2. The sequence $\{X^k\}$ is asymptotically regular, i.e.,

$$\lim_{k \to \infty} \|X^{k+1} - X^k\|_F^2 = 0.$$

Moreover, the sequence $\{X^k\}$ is bounded and has at least one accumulation point.

Although the ILTA has been generated for low-rank matrix completion in this section, the closed form rank Laplace thresholding operator $\mathcal{H}_\frac{\lambda}{\mu}(\cdot)$ in ILTA is not available. In fact, $[\theta(\cdot)]^\prime = 0$ is a transcendental equation and it has no explicit solution, which means that there is no explicit expression for $\theta(\cdot)$ and ILTA in fact is not used to recover the matrices in practical applications. In the following, we will study some new algorithms for problem (9) to recover the low-rank matrices, which can avoid the confusion caused by the unavailability of rank Laplace thresholding operator $\mathcal{H}_\frac{\lambda}{\mu}(\cdot)$ in ILTA.

**IV. ITERATIVE DIFFERENCE LAPLACE THRESHOLDING ALGORITHMS FOR LOW-RANK COMPLETION**

In this section, different from the ILTA, three new algorithms for problem (9) are studied to recover the low-rank matrices.

**A. ITERATIVE DIFFERENCE LAPLACE THRESHOLDING ALGORITHM**

Applying the idea of DCA [24], we transform the objective function

$$M(X) = \frac{1}{2}||X - B_\mu(X^k)||_F^2 + \frac{\lambda}{\mu}(1 - e^{-\frac{\sigma_i(X)}{\gamma}})$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
of problem (27) into a difference between convex function
\[
W(X, X^k) = \frac{1}{2} ||X - B_\mu(X^k)||^2_F + \alpha \lambda \mu ||X||_s
\]
and
\[
N(X) = \alpha \lambda \mu ||X||_s - \frac{1}{\mu} \langle X, S^k \rangle,
\]
where \(\alpha > 0\). In this way, problem (27) can be rewritten as
\[
\min_{X \in \mathbb{R}^{m \times n}} \left\{ W(X, X^k) - N(X) \right\}.
\]
(33)
At the current point \(X^k\) of iteration \(k\), replacing the function \(N(X)\) with its affine minorization defined by
\[
N_k(x) = N(X^k) + \langle X - X^k, S^k \rangle,
\]
(34)
at the neighborhood of \(X^k\), where \(\partial N(\cdot)\) represents the subdifferential of \(N(\cdot)\) and \(S^k \in \partial N(X^k)\), we can get the following convex minimization problem of the form
\[
\min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} ||X - B_\mu(X^k)||^2_F + \alpha \lambda \mu ||X||_s - \langle X, S^k \rangle \right\}
\]
which generates the algorithm for solving the problem (27), i.e.,
\[
X^{k+1} = \arg \min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} ||X - B_\mu(X^k)||^2_F + \alpha \lambda \mu ||X||_s - \langle X, S^k \rangle \right\}.
\]
(35)
It is easy to verify that the global optimal solution \(X^{k+1}\) to the problem (35) can be expressed as
\[
X^{k+1} = D_{\alpha \lambda \mu}(T(X^k)),
\]
(36)
where \(T(X^k) = B_\mu(X^k) + S^k\) and \(D_{\alpha \lambda \mu}(\cdot)\) is obtained by replacing \(\lambda\) with \(\alpha \lambda \mu\) in \(D_{\alpha}(\cdot)\).

**Remark 1.** Let \(X^k = U^k[\text{diag}((\sigma_i(X^k)))_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top\) be the SVD of matrix \(X^k \in \mathbb{R}^{m \times n}\). Then, we can set
\[
S^k \in \partial \mathcal{L}(X^k)
\]
\[
= \partial(\alpha \lambda \mu ||X^k||_s - \frac{\lambda}{\mu} ||X^k||_{\text{F}}^2)
\]
\[
= U^k\left[\text{diag}((\alpha \lambda \mu - \frac{\lambda}{\mu} \|\| e^{-\frac{\sigma_i^2}{\mu^2}})_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top\right].
\]
In this paper, we select
\[
S^k = U^k[[\text{diag}((\alpha \lambda \mu - \frac{\lambda}{\mu} \|\| e^{-\frac{\sigma_i^2}{\mu^2}})_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top].
\]
The more knowledge for the subdifferential of the matrix function can be seen in [25].

For \(k = 0, 1, 2, \ldots\), if we threshold the singular values of matrix \(D_{\alpha \lambda \mu}(T(X^k))\) by the threshold value \(t_{\alpha \lambda \mu} = \frac{\lambda}{\mu^2}\), we can get the iteration
\[
X^{k+1} = H_{\frac{\lambda}{\mu^2}}(D_{\alpha \lambda \mu}(T(X^k))).
\]
(37)
In this paper, we use iteration (37) to recover the low-rank matrices, and we call iteration (37) the iterative difference Laplace thresholding algorithm (IDLTA). The IDLTA for low-rank matrix completion is depicted in Algorithm 2.

**Algorithm 2 Iterative difference Laplace thresholding algorithm (IDLTA)**

**Input:** \(\gamma > 0, \alpha > 0, \lambda > 0, \mu > 1\);  
**Initialize:** \(X^0 \in \mathbb{R}^{m \times n}, k = 0\);  
**while** not converged, **do**  
\[
B_\mu(X^k) = X^k - \frac{1}{\gamma}(P_{\Omega}(X^k) - P_{\Omega}(M));
\]
Compute the SVD of \(X^k\):
\[
X^k = U^k[\text{diag}((\sigma_i(X^k)))_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top;
\]
\[
S^k = U^k[[\text{diag}((\alpha \lambda \mu - \frac{\lambda}{\mu} \|\| e^{-\frac{\sigma_i^2}{\mu^2}})_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top;
\]
\[
T(X^k) = B_\mu(X^k) + S^k;
\]
\[
X^{k+1} = H_{\frac{\lambda}{\mu^2}}(D_{\alpha \lambda \mu}(T(X^k)));
\]
\(k = k + 1;\)
**end while**

**Output:** \(X^{opt}\)

**B. ADAPTIVE ITERATIVE DIFFERENCE LAPLACE THRESHOLDING ALGORITHM**

It is well known that the quality of the solution of a regularized problem depends seriously on the setting of the regularized parameter \(\lambda\). In general, there is no optimal rule for the selection of proper regularized parameter \(\lambda\) in a regularized problem. In IDLTA, we set
\[
\lambda = \lambda_k = \mu \gamma \sigma_k + 1(X^k)
\]
(38)
in each iteration, where \(\gamma\) is the matrix rank of the global optimal solution to the problem (9). When doing so, the IDLTA will be adaptive and free from the choice of the regularized parameter \(\lambda\) in each iteration. In this paper, we call IDLTA with parameter-setting strategy (38) the adaptive iterative difference Laplace thresholding algorithm (AIDLTA), and we depict it in Algorithm 3.

**Algorithm 3 Adaptive iterative difference Laplace thresholding algorithm (AIDLTA)**

**Input:** \(\gamma > 0, \alpha > 0, \lambda > 0, \mu > 1\);  
**Initialize:** \(X^0 \in \mathbb{R}^{m \times n}, k = 0\);  
**while** not converged, **do**  
\[
B_\mu(X^k) = X^k - \frac{1}{\gamma}(P_{\Omega}(X^k) - P_{\Omega}(M));
\]
Compute the SVD of \(X^k\):
\[
X^k = U^k[\text{diag}((\sigma_i(X^k)))_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top;
\]
\[
\lambda = \mu \gamma \sigma_k + 1(X^k);
\]
\[
S^k = U^k[[\text{diag}((\alpha \lambda \mu - \frac{\lambda}{\mu} \|\| e^{-\frac{\sigma_i^2}{\mu^2}})_{1 \leq i \leq m}],[0_{m,n-m}](V^k)^\top;
\]
\[
T(X^k) = B_\mu(X^k) + S^k;
\]
\[
X^{k+1} = H_{\frac{\lambda}{\mu^2}}(D_{\alpha \lambda \mu}(T(X^k)));
\]
\(k = k + 1;\)
**end while**

**Output:** \(X^{opt}\)
C. FAST ADAPTIVE ITERATIVE DIFFERENCE LAPLACE THRESHOLDING ALGORITHM

In this section, we discuss how to accelerate the proposed AIDLTA for low-rank matrix completion. Inspired by the modified fast iterative shrinkage-thresholding algorithm (FISTA) proposed in [26], which is generated to minimize the problems of the sum of a convex function with Lipschitz continuous gradient and a convex function that its proximity operator is efficiently calculated, we proposed a fast version of AIDLTA for low-rank matrix completion.

When the modified FISTA is applied to solve problem (9) for low-rank matrix completion, it generates the following iterative scheme:

\[
\begin{align*}
    t_k &= \frac{p+\sqrt{q+q^2}}{2}, \\
    Y^k &= X^k + \left( \frac{s_k}{q_k} \right)(X^k - X^{k-1}), \\
    X^{k+1} &= \mathcal{H}_\mu(B_p(Y^k)), \\
\end{align*}
\]

where \( p, q > 0 \) and \( \varsigma \in (0, 4) \).

Combining the iterative scheme of (39) into AIDLTA, a fast version of AIDLTA can be given by

\[
\begin{align*}
    t_k &= \frac{p+\sqrt{q+q^2}}{2}, \\
    Y^k &= X^k + \left( \frac{s_k}{q_k} \right)(X^k - X^{k-1}), \\
    T(Y^k) &= B_p(Y^k) + S_k, \\
    X^{k+1} &= \mathcal{H}_\mu(D_{\alpha p}(T(Y^k))),
\end{align*}
\]

where \( p, q > 0 \), \( \varsigma \in (0, 4) \) and \( S_k \) is defined in Remark 1. We call iteration (40) the fast adaptive iterative difference Laplace thresholding algorithm (FAIDLTA), and we depict it in Algorithm 4.

**Algorithm 4** Fast adaptive iterative difference Laplace thresholding algorithm (FAIDLTA)

**Input:** \( \gamma > 0, \mu > 1, p > 0, q > 0, \varsigma \in (0, 4), t_{-1} = 1 \);

**Initialize:** \( X^0 \in \mathbb{R}^{m \times n}, X^{-1} = X^0, k = 0 \);

while not converged, do

- \( t_k = \frac{p+\sqrt{q+q^2}}{2} \),
- \( Y^k = X^k + \left( \frac{s_k}{q_k} \right)(X^k - X^{k-1}) \),
- \( B_p(Y^k) = Y^k - \frac{1}{\mu}(P_\Omega(Y^k) - P_\Omega(M)) \),
- Compute the SVD of \( X^k \),
- \( X^k = U_k^P \sigma_k^{(1)}(U_k^T) \),
- \( \lambda = \mu \gamma \sigma_k^{(1)}(X^k) \),
- \( S_k = U_k^P \sigma_k^{(1)}(U_k^T) \),
- \( T(Y^k) = B_p(Y^k) + S_k \),
- \( X^{k+1} = \mathcal{H}_\mu(D_{\alpha p}(T(Y^k))) \),
- \( k = k + 1 \);

end while

**Output:** \( X^{\text{opt}} \)

It should be noted that there is no analysis for the convergence of FAIDLTA, however, as shown in the numerical experiments, the convergence speed of FAIDLTA is indeed faster than AIDLTA.

V. NUMERICAL SIMULATIONS

In this section, we present some numerical simulations to test the performance of the proposed algorithms and compare them with some state-of-art methods. Some completion of random matrices and grayscale image inpainting problems are considered in our numerical simulations. In these numerical experiments, we only consider to test the performance of AIDLTA and FAIDLTA, and compared them with TS1-s2 [4], SVP [27] and APG [14].

In all numerical simulations, we set \( \mu = 1.5, \alpha = 0.1 \) and \( \gamma = 2 \) in AIDLTA, and set \( \mu = 1.5, \alpha = 0.1, \gamma = 2, p = 0.01, q = 0.2 \) and \( \varsigma = 4 \) in FAIDLTA. The stopping criterion is defined as

\[
\frac{\|X^{k+1} - X^k\|_F}{\max\{\|X^k\|_F, 1\}} \leq 10^{-8},
\]

and the accuracy of the generated solution \( X^{\text{opt}} \) is measured by the relative error (RE), defined as

\[
\text{RE} = \frac{\|X^{\text{opt}} - M\|_F}{\|M\|_F}.
\]

The numerical simulations are all performed in MATLAB on a personal computer (Intel(R) Core(TM)i5-9500 CPU @ 3.00GHZ, 8.00GB RAM)

A. COMPLETION OF RANDOM MATRICES

In this subsection, we present the numerical simulations to test the performance of AIDLTA and FAIDLTA for the completion of randomly generated matrices of size \( n \times n \) and rank \( r \), and compared them with TS1-s2, SVP and APG. In numerical simulations, the matrix \( M \in \mathbb{R}^{n \times n} \) of rank \( r \) is generated as a product \( M_1M_2 \), where \( M_1 \in \mathbb{R}^{n \times p} \) and \( M_2 \in \mathbb{R}^{r \times n} \) have independently identically distributed Gaussian entries. The index set \( \Omega \) of the given \( s \) entries is sampled uniformly at random. The sampling ratio is measured by \( sr = s/n^2 \). The freedom ratio is defined as \( fr = s/(r(2n-r)) \), which is one quantity that helps to quantify the difficulty of a recovery problem. If \( fr < 1 \), there is always an infinite number of matrices of rank \( r \) with the observed entries, so it is impossible to recover the original low-rank matrix in this situation.

The numerical results of the algorithms for the completion of random matrices are reported in Table 1 and Table 2. We can see that AIDLTA and FAIDLTA perform better than TS1-s2, SVP and APG, and the convergence speed of FAIDLTA is faster than AIDLTA.

B. IMAGE INPAINTING

In this subsection, we demonstrate the performance of AIDLTA and FAIDLTA on the image inpainting problems, and compared them with TS1-s2, SVP and APG. These algorithms are tested on two standard 256×256 grayscale images (Peppers and Cameraman). We use the SVD to obtain their approximated images with rank \( r = 30 \). The original images and their corresponding approximated low-rank images are displayed in Figure 2 and Figure 3, respectively.
TABLE 1. Numerical results of algorithms for completion of random matrices with different rank r but fixed n, sr = 0.40.

| Problem      | AIDLTA | FAIDLTA | TS1-s2 | SVP  | APG  |
|--------------|--------|---------|--------|------|------|
| (200, 14, 2.96) | 5.54e-08 | 4.02   | 3.23e-08 | 1.40 | 1.29e-07 | 1.00 |
| (200, 16, 2.60) | 6.23e-08 | 4.67   | 4.65e-08 | 1.50 | 1.09e-07 | 1.14 |
| (200, 18, 2.32) | 6.65e-08 | 5.48   | 4.75e-08 | 1.68 | 2.09e-07 | 1.32 |
| (200, 20, 2.10) | 7.05e-08 | 6.38   | 5.46e-08 | 1.85 | 2.26e-07 | 1.53 |
| (200, 22, 1.92) | 7.87e-08 | 7.65   | 6.18e-08 | 2.12 | 2.76e-07 | 1.80 |
| (200, 24, 1.77) | 9.71e-08 | 10.04  | 9.26e-08 | 2.52 | 4.44e-07 | 2.35 |
| (200, 26, 1.64) | 1.13e-07 | 13.43  | 1.06e-07 | 3.10 | 5.83e-07 | 2.87 |
| (200, 28, 1.53) | 1.21e-07 | 16.30  | 1.11e-07 | 3.46 | 6.73e-07 | 3.54 |
| (200, 30, 1.44) | 1.26e-07 | 18.55  | 1.19e-07 | 3.84 | 7.46e-07 | 4.24 |
| (200, 32, 1.35) | 1.44e-07 | 23.82  | 1.33e-07 | 4.45 | 9.53e-07 | 5.22 |
| (200, 34, 1.28) | 1.80e-07 | 35.85  | 1.81e-07 | 6.12 | 1.51e-06 | 8.00 |

TABLE 2. Numerical results of algorithms for completion of random matrices with different n and rank r, sr = 0.40.

| Problem      | AIDLTA | FAIDLTA | TS1-s2 | SVP  | APG  |
|--------------|--------|---------|--------|------|------|
| (300, 50, 1.30) | 1.58e-07 | 67.46  | 1.56e-08 | 12.12 | 1.09e-06 | 14.84 |
| (400, 60, 1.44) | 1.18e-07 | 80.87  | 1.04e-07 | 17.51 | 6.19e-07 | 18.87 |
| (500, 70, 1.53) | 1.01e-07 | 95.91  | 8.71e-08 | 23.16 | 4.61e-07 | 22.82 |
| (600, 80, 1.60) | 9.30e-08 | 121.91 | 7.76e-08 | 31.00 | 3.78e-07 | 29.88 |
| (700, 90, 1.66) | 8.68e-08 | 157.68 | 6.95e-08 | 40.72 | 3.42e-07 | 37.52 |
| (800, 100, 1.70) | 8.29e-08 | 197.98 | 6.78e-08 | 53.94 | 3.08e-07 | 48.09 |
| (900, 110, 1.74) | 7.99e-08 | 237.41 | 6.58e-08 | 61.34 | 2.83e-07 | 61.95 |
| (1000, 120, 1.77) | 7.71e-08 | 335.07 | 5.72e-08 | 95.44 | 2.68e-07 | 81.37 |
| (1100, 130, 1.79) | 7.58e-08 | 439.97 | 5.85e-08 | 122.75 | 2.57e-07 | 107.82 |
| (1200, 140, 1.82) | 7.42e-08 | 613.04 | 5.71e-08 | 178.33 | 2.49e-07 | 144.23 |
| (1300, 150, 1.83) | 7.32e-08 | 808.15 | 5.67e-08 | 237.19 | 2.35e-07 | 188.39 |

TABLE 3. Numerical results of algorithms for image inpainting problems

| Image                  | AIDLTA | FAIDLTA | TS1-s2 | SVP  | APG  |
|------------------------|--------|---------|--------|------|------|
| (Peppers, 30, 2.03)    | 8.78e-08 | 35.25  | 9.89e-08 | 7.78 | 5.58e-07 | 5.82 |
| (Cameraman, 30, 2.03)  | 1.45e-07 | 77.03  | 1.71e-07 | 12.25 | 1.63e-06 | 14.43 |
| (Peppers, 30, 1.81)    | 1.16e-07 | 48.13  | 1.39e-07 | 9.38 | 1.20e-06 | 9.35 |
| (Cameraman, 30, 1.81)  | 1.77e-07 | 149.73 | 2.23e-07 | 18.18 | 3.11e-06 | 26.49 |

The numerical results of the algorithms for image inpainting problems are reported in Table 3. We can see that AIDLTA and FAIDLTA perform better than TS1-s2, SVP and APG in image inpainting problems, and the convergence speed of FAIDLTA is faster than AIDLTA. For sr = 0.40, we display the recovered low-rank Peppers and Cameraman images via the five algorithms in Figure 4 and Figure 5, respectively.

VI. CONCLUSION

In this paper, we use a rank Laplace function to approximate-ly substitute the rank function in the NP-hard low-rank matrix completion problem, and design some algorithms to recover the low-rank matrices. Numerical results on some low-rank
matrix completion problems show that our algorithms perform better than some state-of-art methods in recovering the low-rank matrices.

REFERENCES

[1] P. Chen and D. Suter, Recovering the missing components in a large noisy low-rank matrix: application to SFM, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1051–1063, 2004.
[2] H. Ji, C. Liu, Z. Shen, and Y. Xu, Robust video denoising using low rank matrix completion, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1791-1798, 2010.
[3] Y. Yu, J. Peng, and S. Yue, A new nonconvex approach to low-rank matrix completion with application to image inpainting, Multidimensional Systems and Signal Processing, vol. 30, no. 1, pp. 145–174, 2019.
[4] S. Zhang, P. Yin, and J. Xiong, Transformed Schatten-1 iterative thresholding algorithms for low rank completion, Communications in Mathematical Sciences, vol. 15, no. 3, pp. 839–862, 2017.
[5] Z. Liu and L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM Journal on Matrix Analysis and Applications, vol. 31, no. 3, pp. 1235–1256, 2009.
[6] E. Candès and B. Recht, Exact matrix completion via convex optimization, Communications of the ACM, vol. 55, no. 6, pp. 111–119, 2012.
[7] J. Cai, E. J. Candès, and Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization, vol. 20, no. 4, pp. 1956–1982, 2010.
[8] S. Ma, D. Goldfarb, and L. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, Mathematical Programming, vol. 128, pp. 321–353, 2011.
[9] E. J. Candès and Y. Plan, Matrix completion with noise, Proceedings of the IEEE, vol. 98, no. 6, pp. 925–936, 2010.
[10] E. J. Candès and T. Tao, The power of convex relaxation: near-optimal matrix completion, IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2053-2080, 2010.
[11] B. Recht, M. Fazel, and P. A. Parrilo, Guaranteed minimum-rank solution of linear matrix equations via nuclear norm minimization, SIAM Review, vol. 52, no. 3, pp. 471–501, 2010.
[12] X. Zeng, L. Shen, Y. Xu and J. Lu, Matrix completion via minimizing an approximate rank, Analysis and Applications, vol. 17, no. 5, pp. 689–713, 2019.
[13] R. H. Tütüncü, K. C. Toh, and M. J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3. Mathematical Programming, vol. 95, pp. 189–217, 2003.
[14] K. C. Toh and S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific Journal of Optimization, vol. 6, no. 3, pp. 615–640, 2010.
[15] Y. J. Liu, D. Sun, and K. C. Toh, An implementable proximal point algorithmic framework for nuclear norm minimization, Mathematical Programming, vol. 133, pp. 399–436, 2012.
[16] J. Yang and X. Yuan, Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization, Mathematics of Computation, vol. 82, no. 281, pp. 301–329, 2013.
[17] Q. Wang, G Qu, and G. Han, A thresholding algorithm for sparse recovery via Laplace norm, Signal, Image and Video Processing, vol. 13, pp. 389-395, 2019.
[18] C. Lu, J. Tang, S. Yan, Z. Lin, Generalized nonconvex nonsmooth low-rank minimization, 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4130–4137, 2014.
[19] C. Lu, J. Tang, S. Yan, Z. Lin, Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm, IEEE Transactions on Image Processing, vol. 25, no. 2, pp. 829–839, 2016.
[20] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183–202, 2009.
[21] I. Daubechies, M. Defrise, and C. D. Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Communications on Pure and Applied Mathematics, vol. 57, pp. 1413–1457, 2004.
[22] S. Foucart and H. Rauhut, A mathematical introduction to compressive sensing, Springer Science+Business Media, New York, 2013.
[23] D. Lazzaro, A nonconvex approach to low-rank matrix completion using convex optimization, Numerical Linear Algebra with Applications, vol. 23, pp. 801–824, 2016.
[24] H. A. Le Thi and T. Pham Dinh, DC programming and DCA: thirty years of developments, Mathematical Programming, vol. 169, pp. 5–68, 2018.
[25] A. S. Lewis, The convex analysis of unitarily invariant matrix functions, Journal of Convex Analysis, vol. 2, no. 1/2, pp. 173–183, 1995.
[26] J. Liang, T. Luo, and C. B. Schönlieb, Improving “Fast iterative shrinkage-thresholding algorithm”: faster, smarter and greedier, arXiv:1811.01430v4, 2021.
[27] P. Jain, R. Meka, and I. Dhillon, Guaranteed rank minimization via singular value projection, Proceedings of the Neural Information Processing Systems Conference (NIPS), pp. 937-945, 2010.

HAIZHEN HE received the M.Sc. degree in foreign linguistics and applied linguistics from Xi’an Polytechnic University, Xi’an, China, in 2019. She is currently a Teaching Assistant with the School of International Education, Yulin University, and she is currently pursuing the Ph.D. degree with the Management studies, Woosong University, Daejeon, Republic of Korea. Her current research interests include strategic management, human resource management and non-convex optimization algorithms.

ANGANG CUI received the M.Sc. degree in applied mathematics from Xi’an Polytechnic University, Xi’an, China, in 2013, and the Ph.D. degree in mathematics from Xi’an Jiaotong University, Xi’an, China, in 2020. He is currently a Lecturer with the School of Mathematics and Statistics, Yulin University. His current research interests include machine learning theory and sparse information processing.

HONG YANG received the M.Sc. degree in mathematics from Yan’an University, Yan’an, China, in 2004. He is currently an Associate Professor of School of Mathematics and Statistics, Yulin University, Yulin, China. His current research interests include non-smooth semi-infinite programming and machine learning theory.

MENG WEN received the B.S. degrees in mathematics from Xianyang Normal University, Xianyang, China, in 2010, the M.Sc. degrees in mathematics from Hubei Normal University, Huangshi, China, in 2013, and the Ph.D. degree in mathematics from Xi’an Jiaotong University, Xi’an, China, in 2018. She is currently an Associate Professor with the School of Science, Xi’an Polytechnic University. Her research interests include sparse optimization models, and algorithms with applications in image processing and image reconstruction.