Applied Mathematics and Computation 355 (2019) 299–310

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Toeplitz matrix completion via smoothing augmented Lagrange multiplier algorithm

Rui-Ping Wen*, Shu-Zhen Li, Fang Zhou

Key Laboratory of Engineering & Computing Science, Shanxi Provincial Department of Education/Department of Mathematics, Taiyuan Normal University, China

ARTICLE INFO

MSC:
65F50
65F10
15A83

Keywords:
Toeplitz matrix
Augmented Lagrange multiplier
Matrix completion
Smoothing

ABSTRACT

Toplitz matrix completion (TMC) is to fill a low-rank Toeplitz matrix from a small subset of its entries. Based on the augmented Lagrange multiplier (ALM) algorithm for matrix completion, in this paper, we propose a new algorithm for the TMC problem using the smoothing technique of the approximation matrices. The completion matrices generated by the new algorithm are of Toeplitz structure throughout iteration, which save computational cost of the singular value decomposition (SVD) and approximate well the solution. Convergence results of the new algorithm are proved. Finally, the numerical experiments show that the augmented Lagrange multiplier algorithm with smoothing is more effective than the original ALM and the accelerated proximal gradient (APG) algorithms.

1. Introduction and preliminaries

Matrix completion (MC) problem proposed by Candès and Recht [12] is to fill a matrix from some of its observed entries. Recovering an unknown low-rank or approximately low-rank matrix from a sampling of its entries is a challenging problem with applications in many fields of applied science and engineering areas such as model reduction [24], machine learning [1,2], control [27], pattern recognition [15], imaging inpainting [4], computer vision [33], and so on. Despite matrix completion requiring the global solution of a non-convex objective, there are many computational algorithms which are effective for a broad class of matrices.

Through the summary of many of the current situation, the researchers indicated that most low-rank matrices can be recovered from a sampling of its entries. Explicitly seeking the lowest rank matrix consistent with the known entries is mathematically expressed as:

$$\begin{array}{ll} \min\limits_{A\in\mathbb{R}^{m\times n}} & \operatorname{rank}(A) \\ \text{subject to} & \mathcal{P}_\Omega(A) = \mathcal{P}_\Omega(M), \end{array} \tag{1}$$

where the matrix $M \in \mathbb{R}^{m\times n}$ is the underlying matrix to be reconstructed, Ω is a random subset of indices for the known entries, and $\mathcal{P}_\Omega$ is the associated sampling orthogonal projection operator which acquires only the entries indexed by $\Omega \subset \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$.

The problem (1), in general, is non-convex and is NP-hard [17] due to the rank objective. A few algorithms have been presented to solve (1) while the rank of the matrix M was known or can be estimated [24,25]. Vandereycken [34] applied the

* Corresponding author.
E-mail address: wenrp@163.com (R.-P. Wen).

https://doi.org/10.1016/j.amc.2019.02.027

Riemannian optimization to the problem by minimizing the least square distance on the sampling set over the Riemannian manifold of the matrix M. Then a Riemannian geometry method and a Riemannian trust-region method were given by Mishra et al. [28] and Boumal and Absil [6], respectively. The hard thresholding algorithm as well as its variants were introduced by Blanchard et al. [5], Foucart [19], Jain et al. [20,21], and Kyríllidis and Cevher [22] proposed the alternating minimization method, Tanner and Wei [31] presented the alternating steepest descent (ASD) method, and Wen and Liu [38] improved the ASD method to the two-stage iteration algorithms.

As we well known, Candés and Recht [12] replaced the rank objective in (1) with its convex relaxation in 2009, the nuclear norm $\|A\|_*$ which is the sum of all singular values of the matrix A, that is

$$\begin{array}{ll}\min\limits_{A\in\mathbb{R}^{m\times n}} & \|A\|_* \\ \text{subject to} & \mathcal{P}_\Omega(A)=\mathcal{P}_\Omega(M).\end{array} \tag{2}$$

In order to solve the optimization problem (2), numerous methods have been designed. For example, the singular value theresholding (SVT) method as well as its variants [7,18,37], an accelerated proximal gradient (APG) method [32], the augmented Lagrange multiplier (ALM) method [23] etc. Numerous details derivations on MC problem can be refer to the [5–13,16,18–23,25,28,31,32,34–39] and the references given therein.

In this study, we focus on the Toeplitz matrix completion (TMC) problem, one of the most important MC problems. The problem has attracted a lot of attention in recent years. Many researchers have studied the approximation of the Toeplitz and Hankel matrices: see for example the nuclear norm minimization for the low-rank Hankel matrix reconstruction problem under the random Gaussian sampling model investigated in [9]. Chen and Chi [14] studied nuclear norm minimization for the low-rank Hankel matrix completion problem. Cai et al. [8] developed a fast non-convex algorithm for low-rank Hankel matrix completion by minimizing the distance between low-rank matrices and Hankel matrices with partial known anti-diagonals, and an accelerated variant has been developed in [8] using Nesterov's memory technique as inspired by FISTA [3]. An iterative hard thresholding (IHT) and a fast IHT (FIHT) algorithms have been came up with in [10] for efficient reconstruction of spectrally sparse signals via low-rank Hankel matrix completion. By utilizing the low-rank structure of the Hankel matrix corresponding to a spectrally sparse signal x, Cai et al. [11] introduced a computationally efficient algorithm for the spectral compressed sensing problem. Sebert et al. consider Toeplitz block matrices as sensing matrices whose elements are drawn from the same distributions in [29]. Shaw et al. [30] presented algorithms for least-squares approximation of Toeplitz and Hankel matrices from noise corrupted or ill-composed matrices, which may not have correct structural or rank properties. Wang and Li [35] proposed a mean value algorithm to force the Toeplitz structure and provided a comparison of several algorithms for the Toeplitz matrix recovery in [36]. Fazel et al. [16] proposed also the reconstruction of low-rank Hankel matrices via nuclear norm minimization for system identification realization. Further, Ying et al. [39] have exploited the low-rank tensor structure of the signal when developing recovery problems for multidimensional spectrally sparse signal recovery problems. For details, one can refer to the above mentioned algorithms and references given therein.

A Toeplitz matrix $T\in\mathbb{R}^{n\times n}$ is of the form

$$T=\begin{pmatrix} t_0 & t_1 & \cdots & t_{n-2} & t_{n-1}\\ t_{-1} & t_0 & \cdots & t_{n-3} & t_{n-2}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ t_{-n+2} & t_{-n+3} & \cdots & t_0 & t_1\\ t_{-n+1} & t_{-n+2} & \cdots & t_{-1} & t_0\end{pmatrix}, \tag{3}$$

which is determined by $2n-1$ entries, that is the first row and the first column. This leads the special study both from theoretical and algorithmic aspects on the TMC problem. That is also the motivation of this paper.

In this paper, we are interested in the augmented Lagrange multiplier algorithm with smoothing (SALM) for the TMC problem. By smoothing the diagonal entries of iteration matrices, the sequence matrices generated by the new algorithm hold on a Toeplitz structure, in which the fast algorithm [26] of the singular value decomposition (SVD) were used. And we also establish the convergence theory. Experimental results show that the new algorithm is up to dozens of times faster than the standard APG and the ALM algorithms, and its precision is also higher.

The rest of the paper is organized as follows. After we provide some notations and preliminaries in this section, we review briefly the standard APG and the ALM algorithms as well as the dual approach in Section 2. In Section 3, the algorithm of augmented Lagrange multiplier with smoothing is proposed and its convergence theory is established. Then, numerical experiments are shown and compared in Section 4. Finally, we end the paper with the concluding remarks in Section 5.

Here are some necessary notations and preliminaries. $\mathbb{R}^{m\times n}$ is used to denote the set of $m\times n$ real matrices. The nuclear norm of a matrix A is denoted by $\|A\|_*$, the Frobenius norm by $\|A\|_F$, the Euclidean norm by $\|A\|_2$, and $\|A\|_\infty$ is maximum absolute value of the matrix entries of the matrix A (also known as infinite norm of A). The transpose of a matrix $A\in\mathbb{R}^{n\times n}$ is A^T, and tr(A) represents the trace of a matrix A. The standard inner product between two matrices is denoted by $\langle X, Y\rangle=\text{tr}(X^TY)$. $\Omega\subset\{-n+1,\dots,n-1\}$ is the set of indices of observed diagonals of a Toeplitz matrix $M\in\mathbb{R}^{n\times n}$, and $\bar{\Omega}$ is the complementary set of Ω. For a Toeplitz matrix A, the vector diag(A, l) denotes the lth diagonal of A, $l\in\{-n+1,\dots,n-1\}$, and $\mathcal{P}_\Omega$ is the orthogonal projector on Ω, satisfying $\text{diag}(\mathcal{P}_\Omega(A),l)=\begin{cases}\text{diag}(A,l), & l\in\Omega\\ \mathbf{0}, & l\notin\Omega\end{cases}$ ($\mathbf{0}$ is a zero-vector).

The singular value decomposition (SVD) of a r-rank matrix $A \in \mathbb{R}^{m\times n}$ is:

$$A = U\Sigma_r V^T, \quad \Sigma_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r),$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ are column orthogonal matrices, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

Definition 1.1. [7] (Singular value thresholding operator) For each $\tau \geq 0$, the singular value thresholding operator $\mathcal{D}_\tau$ is defined as follows:

$$\mathcal{D}_\tau(A) := U\mathcal{D}_\tau(\Sigma)V^T, \quad \mathcal{D}_\tau(\Sigma) = \mathrm{diag}(\{\sigma_i - \tau\}_+),$$

where $A = U\Sigma_r V^T \in \mathbb{R}^{m\times n}$, $\{\sigma_i - \tau\}_+ = \begin{cases} \sigma_i - \tau, & \text{if} \quad \sigma_i > \tau \\ 0, & \text{if} \quad \sigma_i \leq \tau \end{cases}.$

Definition 1.2. The matrices

$$T_l = (t_{ij})_{n\times n} = \begin{cases} 1, & i-j=l \\ 0, & i-j \neq l \end{cases}, l = -n+1, \ldots, n-1 \tag{4}$$

are called the basis of the Toeplitz matrices space.

It is clear that any Toeplitz matrix $T \in \mathbb{R}^{n\times n}$, shown in (3), can be written as a linear combination of the basis matrices in Definition 1.2, that is,

$$T = \sum_{l=-n+1}^{n-1} t_l T_l.$$

Definition 1.3 (Toeplitz structure smoothing operator)**.** For any matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, the Toeplitz structure smoothing operator $\mathcal{T}$ is defined as follows:

$$\mathcal{T}(A) := \sum_{l=-n+1}^{n-1} \tilde{a}_l T_l, \tag{5}$$

where $\tilde{a}_l = \frac{{}_lA^{\min} + {}_lA^{\max}}{2}$, $l \in \{-n+1, \ldots, n-1\}$ with

$${}_lA^{\min} = \min_{i,j\in\{1,2,\ldots,n\}} \{a_{ij}, i-j=l\},$$
$${}_lA^{\max} = \max_{i,j\in\{1,2,\ldots,n\}} \{a_{ij}, i-j=l\}.$$

It is clear that $\mathcal{T}(A)$ is a Toeplitz matrix derived from the matrix A. That is to say, any $A \in \mathbb{R}^{n\times n}$ can be changed into a Toeplitz matrix via the smoothing operator $\mathcal{T}(\cdot)$. Here and in the sequel, $\mathcal{T}(\cdot)$ stands for the Toeplitz matrix derived from the corresponding matrix by the smoothing operator (5).

2. Related algorithms

Since the matrix completion problem is closely connected to the robust principal component analysis (RPCA) problem, then it can be formulated in the same by the RPCA. An equivalent problem of (2) can be considered as follows.

As E will compensate for the unknown entries of M, the unknown entries of M are simply set as zeros. Suppose that the given data are arranged as the columns of a large matrix $M \in \mathbb{R}^{m\times n}$. The mathematical model for estimating the low-dimensional subspace is to find a low-rank matrix $A \in \mathbb{R}^{m\times n}$ (as long as the error matrix E is sufficiently sparse, relative to the rank of A), such that the discrepancy between A and M is minimized, leading to the following constrained optimization:

$$\begin{aligned} \min_{A,E\in\mathbb{R}^{m\times n}} \quad & \|A\|_*, \\ \text{subject to} \quad & A + E = M, \ \mathcal{P}_\Omega(E) = 0. \end{aligned} \tag{6}$$

where $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear operator that keeps the entries in Ω unchanged and sets those outside Ω (say, in $\overline{\Omega}$) zeros.

There has been a lot of algorithms for solving the optimization (6). In this section, for the goal of completing comparison subsequently, we briefly review and introduce some algorithms for matrix completion problem.

2.1. The augmented Lagrange multipliers algorithm

The partial augmented Lagrange function of (6) is

$$\mathcal{L}(A, E, Y, \mu) = \|A\|_* + \langle Y, M - A - E\rangle + \frac{\mu}{2}\|M - A - E\|_F^2.$$

Then the augmented Lagrange multipliers algorithm is summarized in the following Algorithm 2.1.

It is reported that the algorithm of augmented Lagrange multipliers have been applied to the MC problem. It is of much better numerical behavior, and it is also of much higher accuracy.

Algorithm 2.1 ALM algorithm [23].

Step 0: Given a sampled set Ω, a sampled matrix $N = \mathcal{P}_\Omega(M)$, parameters $\mu_0 > 0$, $\rho > 1$.Given also two initial matrices $Y_0 = 0$, $E_0 = 0$, $k := 0$;

Step 1: Solve $A_{k+1} = \arg\min_A \mathcal{L}(A, E_k, Y_k, \mu_k)$, compute the SVD of the matrix $(N - E_k + \mu_k^{-1} Y_k)$,

$$[U_k, \Sigma_k, V_k] = svd(N - E_k + \mu_k^{-1} Y_k);$$

Step 2: Set

$$A_{k+1} = U_k \mathcal{D}_{\mu_k^{-1}}(\Sigma_k) V_k^T,$$

solves $E_{k+1} = \arg\min_{\mathcal{P}_\Omega(E)=0} \mathcal{L}(A_{k+1}, E, Y_k, \mu_k)$,

$$E_{k+1} = \mathcal{P}_{\bar{\Omega}}(N - A_{k+1} + \mu_k^{-1} Y_k);$$

Step 3: If $\|N - A_{k+1} - E_{k+1}\|_F / \|N\|_F < \epsilon_1$ and $\mu_k \|E_{k+1} - E_k\|_F / \|N\|_F < \epsilon_2$,
stop; otherwise, go to Step 4;

Step 4: Set $Y_{k+1} = Y_k + \mu_k(N - A_{k+1} - E_{k+1})$,
If $\mu_k \|E_{k+1} - E_k\|_F / \|N\|_F < \epsilon_2$, set $\mu_{k+1} = \rho\mu_k$; otherwise, go to Step 1.

2.2. The accelerated proximal gradient algorithm

The relaxed version of the RPCA problem can be rewritten as the following unconstrained convex problem

$$\min_{A \in \mathbb{R}^{m\times n}} \quad \mu\|A\|_* + \tfrac{1}{2}\|\mathcal{P}_\Omega(A - M)\|_F^2. \tag{7}$$

where μ is a small positive scalar. And then the accelerated proximal gradient algorithm [32] was designed as follows to solve (7).

Algorithm 2.2 APG algorithm [32].

Step 0: Given $\mu_0 > 0$, $\bar{\mu} > 0$, L_f, $\zeta \in (0, 1)$. Given also $X_0 = X_{-1} = 0$, $Y_0 = 0$, $t_0 = t_{-1} = 1$, $k := 0$;

Step 1: Compute the SVD of the matrix $H_k = Y_k - \frac{1}{L_f}(\mathcal{P}_\Omega(Y_k) - \mathcal{P}_\Omega(M))$,

$$[U_k, \Sigma_k, V_k] = svd(H_k);$$

Step 2: Set

$$A_{k+1} = U_k \mathcal{D}_{\frac{\mu_k}{L_f}}(\Sigma_k) V_k^T,$$

Step 3: If $\|A_{k+1} - A_k\|_F / \|A_{k+1}\|_F < \epsilon$, stop; otherwise, go to Step 4;

Step 4: Compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$, $\mu_{k+1} = \max(\zeta\mu_k, \bar{\mu})$;
Set $Y_k = A_k + \frac{t_{k-1}-1}{t_k}(A_k - A_{k-1})$, go to Step 1.

It is suggested that the accelerated proximal gradient algorithm with a fast method, such as PROPACK, for computing partial singular value decomposition is simple and suitable for solving large-scale matrix completion problems when the solution matrix has low-rank. Three techniques, namely, linesearch-like, continuation, and truncation techniques, have been developed to accelerate the convergence of the original APG algorithm.

2.3. The dual algorithm

The dual algorithm proposed in [13] tackles the problem (6) via its dual. That is, one first solves the dual problem

$$\begin{aligned} \max_Y \quad & \langle M, Y\rangle \\ \text{subject to} \quad & J(Y) \le 1 \end{aligned} \tag{8}$$

for the optimal Lagrange multiplier Y, where

$$J(Y) = \max(\|Y\|_2, \lambda^{-1}\|Y\|_\infty). \tag{9}$$

A steepest ascend algorithm constrained on the surface $\{Y|J(Y) = 1\}$ can be adopted to solve (8), where the constrained steepest ascend direction is obtained by projecting M onto the tangent cone of the convex body $\{Y|J(Y) \le 1\}$. It turns out that the optimal solution to the primal problem (6) can be obtained during the process of finding the constrained steepest ascend direction.

3. The augmented Lagrange multiplier with smoothing

In this section, we develop a smoothing augmented Lagrange multiplier (SALM) algorithm for the TMC problem, and provide its convergence results.

Based on the special structure and properties of the Toeplitz matrices, we use a smoothing technique in the original ALM algorithm. It turns out that the iteration matrices stick a Toeplitz structure, which ensures that the fast SVD of the Toeplitz matrices can be utilized. Therefore, we present a new fast algorithm for the TMC problem.

The problem admits the following convex programming,

$$\begin{array}{ll} \min\limits_{A,E\in\mathbb{R}^{n\times n}} & \|A\|_*, \\ \text{subject to} & A+E=\mathcal{P}_\Omega(M),\ \ \mathcal{P}_\Omega(E)=0, \end{array} \tag{10}$$

where $A, M \in \mathbb{R}^{n\times n}$ are both real Toeplitz matrices, $\Omega \subset \{-n+1,\dots,n-1\}$.

Let $D=\mathcal{P}_\Omega(M)$. Then the partial augmented Lagrangian function is

$$\mathcal{L}(A,E,Y,\mu)=\|A\|_*+\langle Y, D-A-E\rangle+\frac{\mu}{2}\|D-A-E\|_F^2, \tag{11}$$

where $Y\in\mathbb{R}^{n\times n}$.

In kth iteration, we are to smooth the approximation matrix X_{k+1} by using the Toeplitz structure smoothing operator $\mathcal{T}$, lead to the matrix A_{k+1} stick in consequence a Toeplitz structure. A merit of smoothing is that the fast SVD scheme can be utilized to reduce the computation.

The rest of this section will analyze the convergence of Algorithm 3.1. First of all, we give several useful lemmas.

Algorithm 3.1 (SALM algorithm).

Step 0: Given Ω, sampled matrix D, $\mu_0>0, \rho>1$. Given also two initial matrices $Y_0=0$, $E_0=0$, $k:=0$;
Step 1: Compute the SVD of the matrix $(D-E_k+\mu_k^{-1}Y_k)$ using the Lanczos method

$$[U_k,\Sigma_k,V_k]=lansvd(D-E_k+\mu_k^{-1}Y_k);$$

Step 2: Set

$$X_{k+1}=U_k\mathcal{D}_{\mu_k^{-1}}(\Sigma_k)V_k^T,$$

compute for smoothing $\tilde{a}_l=\frac{{}_lX_{k+1}^{\min}+{}_lX_{k+1}^{\max}}{2}$, $l\in\{-n+1,\dots,n-1\}$,
and

$$A_{k+1}=\mathcal{T}(X_{k+1})=\sum_{l=-n+1}^{n-1}\tilde{a}_lT_l,$$

$$E_{k+1}=\mathcal{P}_{\bar{\Omega}}(D-A_{k+1}+\mu_k^{-1}Y_k);$$

Step 3: If $\|D-A_{k+1}-E_{k+1}\|_F/\|D\|_F<\epsilon_1$ and $\mu_k\|E_{k+1}-E_k\|_F/\|D\|_F<\epsilon_2$,
stop; otherwise, go to next Step;
Step 4: Set $Y_{k+1}=Y_k+\mu_k(D-A_{k+1}-E_{k+1})$.
If $\mu_k\|E_{k+1}-E_k\|_F/\|D\|_F<\epsilon_2$, set $\mu_{k+1}=\rho\mu_k$; otherwise, go to Step 1.

Let $(\ddot{A},\ddot{E})$ be the solution of the sequel (10) and $\ddot{Y}$ be the optimal solution of the sequel problem (8).

Lemma 3.1 [12]**.** *Let $A\in\mathbb{R}^{m\times n}$ be an arbitrary matrix and $U\Sigma V^T$ be its SVD. Then the set of subgradients of the nuclear norm of A is given by*

$$\partial\|A\|_*=\{UV^T+W: W\in\mathbb{R}^{m\times n}, U^TW=0, WV=0, \|W\|_2\le 1\}.$$

Lemma 3.2 [23]**.** *If μ_k is nondecreasing, then each entry of the following series is nonnegative and their sum is finite, i.e.,*

$$\sum_{k=1}^{+\infty}\mu_k{}^{-1}(\langle Y_{k+1}-Y_k, E_{k+1}-E_k\rangle+\langle A_{k+1}-\ddot{A},\hat{Y}_{k+1}-\ddot{Y}\rangle+\langle E_{k+1}-\ddot{E}, Y_{k+1}-\ddot{Y}\rangle)<+\infty. \tag{12}$$

Lemma 3.3 [23]**.** *The sequences $\{\ddot{Y}_k\}$, $\{Y_k\}$ and $\{\hat{Y}_k\}$ are all bounded, where $\hat{Y}_k=Y_{k+1}+\mu_{k-1}(D-A_k-E_{k-1})$.*

Lemma 3.4. *The sequence $\{Y_k\}$ generated by Algorithm 3.1 is bounded.*

Proof. Let $B=\mu_k(D-E_k+\mu_k^{-1}Y_k-X_{k+1})$, $\mathcal{T}(B)=\sum_{l\in\Omega}\tilde{b}_lT_l$, defined as (5).

First of all, we indicate that $Y_k, E_k, k = 1, 2, \ldots,$ are Toeplitz matrices. Clearly, $Y_0 = 0, E_0 = 0$ are both Toeplitz matrices. Suppose that if Y_k, E_k are both Toeplitz matrices, so is $E_{k+1} = \mathcal{P}_{\bar{\Omega}}(D - A_{k+1} + \mu_k^{-1}Y_k)$. Thus, Y_{k+1} is a Toeplitz matrix also from the step 4 in Algorithm 3.1.

$$\begin{aligned} Y_{k+1} &= Y_k + \mu_k(D - A_{k+1} - E_{k+1}) \\ &= Y_k + \mu_k(D - A_{k+1} - E_k) + \mu_k(E_k - E_{k+1}). \end{aligned}$$

Also,

$$\begin{aligned} \mu_k(E_k - E_{k+1}) &= \mu_k \mathcal{P}_{\bar{\Omega}}(E_k - (D - A_{k+1} + \mu_k^{-1}Y_k)) \\ &= \mu_k \mathcal{P}_{\bar{\Omega}}(E_k - (D - \mathcal{T}(X_{k+1}) + \mu_k^{-1}Y_k)) \\ &= \mu_k \mathcal{P}_{\bar{\Omega}}(E_k - (D - \mathcal{T}(D - E_k + \mu_k^{-1}Y_k - \mu_k^{-1}B) + \mu_k^{-1}Y_k)) \\ &= \mu_k \mathcal{P}_{\bar{\Omega}}(E_k - (D - (D - E_k + \mu_k^{-1}Y_k - \mu_k^{-1}\mathcal{T}(B)) + \mu_k^{-1}Y_k)) \\ &= \mathcal{P}_{\bar{\Omega}}\mathcal{T}(B). \end{aligned}$$

It is clear that by Steps 1 and 2 in Algorithm 3.1,

$$D - E_k + \mu_k^{-1}Y_k = \check{U}_k\check{\Sigma}_k\check{V}_k^T + \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^T,$$

where $\check{U}_k, \check{V}_k$ are the singular vectors associated with singular values that are more than $\frac{1}{\mu_k}$ and $\tilde{U}_k, \tilde{V}_k$ are those associated with singular values that are less than or equal to $\frac{1}{\mu_k}$, the elements of the diagonal matrix $\check{\Sigma}_k$ are larger than $\frac{1}{\mu_k}$ and those of the diagonal matrix $\tilde{\Sigma}_k$ are less than or equal to $\frac{1}{\mu_k}$. Hence, it is drawn that $X_{k+1} = \check{U}_k(\check{\Sigma}_k - \frac{1}{\mu_k}I)\check{V}_k^T$ and

$$\begin{aligned} \|B\|_F &= \|\mu_k(D - E_k + \mu_k^{-1}Y_k - X_{k+1})\|_F \\ &= \|\mu_k(\mu_k^{-1}\check{U}_k\check{V}_k^T + \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^T)\|_F \\ &= \|\check{U}_k\check{V}_k^T + \mu_k\tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^T\|_F \\ &\le \sqrt{n}. \end{aligned}$$

We can obtain hence that $Y_k + \mu_k(D - A_{k+1} - E_k) \in \partial\|A_{k+1}\|_*$ from Lemmas 3.2 and 3.3.

For $A_{k+1} = U\Sigma V^T$, it is known that by Lemma 3.1,

$$\partial\|A_{k+1}\|_* = \{UV^T + W : W \in \mathbb{R}^{n\times n}, U^TW = 0, WV = 0, \|W\|_2 \le 1\}.$$

We have also,

$$\|UV^T + W\|_F^2 = \mathrm{tr}((UV^T + W)^T(UV^T + W)) = \mathrm{tr}(VV^T + W^TW) \le n.$$

Therefore, the following inequalities can be obtained

$$\|Y_k + \mu_k(D - A_{k+1} - E_k)\|_F \le \sqrt{n},$$

and

$$\|\mathcal{P}_{\bar{\Omega}}(\mathcal{T}(B))\|_F \le \|\mathcal{T}(B))|_F \le \|B\|_F \le \sqrt{n}.$$

Then it is clear that the sequence $\{Y_k\}$ is bounded. □

Theorem 3.1. *Suppose that* $\langle A_{k+1} - A_k, D - A_{k+1} - E_k\rangle \ge 0$, *then the sequence* $\{A_k\}$ *converges to the solution of (10) when* $\mu_k \to \infty$ *and* $\sum_{k=1}^{+\infty}\mu_k^{-1} = +\infty$.

Proof. It is true that

$$\lim_{k\to\infty}(D - A_{k+1} - E_{k+1}) = 0$$

since $\mu_k^{-1}(Y_{k+1} - Y_k) = D - A_{k+1} - E_{k+1}$ and Lemma 3.4.

Let $(\ddot{A}, \ddot{E})$ be the solution of (10). Then $A_{k+1}, Y_{k+1}, E_{k+1}, k = 1, 2, \ldots,$ are all Toeplitz matrices from $\ddot{A} + \ddot{E} = D$. We prove first that

$$\begin{aligned} &\|E_{k+1} - \ddot{E}\|_F^2 + \mu_k^{-2}\|Y_{k+1} - \ddot{Y}\|_F^2 \\ &\quad = \|E_k - \ddot{E}\|_F^2 - \|E_{k+1} - E_k\|_F^2 + \mu_k^{-2}\|Y_k - \ddot{Y}\|_F^2 \\ &\quad -\mu_k^{-2}\|Y_{k+1} - Y_k\|_F^2 - 2\mu_k^{-1}\langle A_{k+1} - \ddot{A}, \hat{Y}_{k+1} - \ddot{Y}\rangle, \end{aligned} \tag{13}$$

where $\hat{Y}_{k+1} = Y_k + \mu_k(D - A_{k+1} - E_k)$, $\ddot{Y}$ is the optimal solution to the dual problem (8).

$$\begin{aligned} \|E_k - \ddot{E}\|_F^2 &= \|\mathcal{P}_{\bar{\Omega}}(E_k - \ddot{E})\|_F^2 \\ &= \|\mathcal{P}_{\bar{\Omega}}(E_{k+1} - \ddot{E} - E_{k+1} + E_k)\|_F^2 \end{aligned}$$

$$= \|\mathcal{P}_{\bar{\Omega}}(E_{k+1} - \ddot{E})\|_F^2 + \|\mathcal{P}_{\bar{\Omega}}(E_{k+1} - E_k)\|_F^2 - 2\langle \mathcal{P}_{\bar{\Omega}}(E_{k+1} - \ddot{E}), \mathcal{P}_{\bar{\Omega}}(E_{k+1} - E_k)\rangle$$
$$= \|E_{k+1} - \ddot{E}\|_F^2 + \|E_{k+1} - E_k\|_F^2 + 2\mu_k^{-1}\langle \mathcal{P}_{\bar{\Omega}}(A_{k+1} - \ddot{A}), \hat{Y}_{k+1} - \ddot{Y}\rangle.$$

We obtain the following result with the same analysis,

$$\begin{aligned}\mu_k^{-2}\|Y_k - \ddot{Y}\|_F^2 &= \mu_k^{-2}\|\mathcal{P}_\Omega(Y_k - \ddot{Y})\|_F^2 \\ &= \mu_k^{-2}\|\mathcal{P}_\Omega(Y_{k+1} - \ddot{Y})\|_F^2 + \mu_k^{-2}\|\mathcal{P}_\Omega(Y_{k+1} - Y_k)\|_F^2 - 2\mu_k^{-1}\langle \mathcal{P}_\Omega(Y_{k+1} - \ddot{Y}), \mathcal{P}_\Omega(Y_{k+1} - Y_k)\rangle \\ &= \mu_k^{-2}\|Y_{k+1} - \ddot{Y}\|_F^2 + \mu_k^{-2}\|Y_{k+1} - Y_k\|_F^2 + 2\mu_k^{-1}\langle \mathcal{P}_\Omega(A_{k+1} - \ddot{A}), \hat{Y}_{k+1} - \ddot{Y}\rangle.\end{aligned}$$

Then (7) holds true.

The sum $\sum_{k=1}^{\infty} \mu_k^{-1}\langle A_k - \ddot{A}, \hat{Y}_k - \ddot{Y}\rangle < +\infty$ and $\|E_k - \ddot{E}\|^2 + \mu_k^{-2}\|Y_k - \ddot{Y}\|^2$ is nonincreasing since $\|A\|_*$ is a convex function and $\hat{Y}_{k+1} \in \partial\|A\|_*$, $\langle A_{k+1} - \ddot{A}, \hat{Y}_{k+1} - \ddot{Y}\rangle \geq 0$. On the other hand, the following are true by Algorithm 3.1:

$$\langle Y_k, E_k - \ddot{E}\rangle = 0, \text{ and } \langle \ddot{Y}, E_k - \ddot{E}\rangle = 0.$$

Therefore, using the same idea of Theorem 2 in [23], it is obtained that $\ddot{A}$ is the solution of (10). □

Theorem 3.2. *Let $X = (x_{ij}) \in \mathbb{R}^{n\times n}$, $\mathcal{T}(X) = (\tilde{x}_{ij}) \in \mathbb{R}^{n\times n}$ be the Toeplitz matrix derived from X, introduced in (5). Then for all Toeplitz matrix $Y = (y_{ij}) \in \mathbb{R}^{n\times n}$, $\langle X - \mathcal{T}(X), Y\rangle = 0$.*

Proof. By the definition of $\mathcal{T}(X)$, we have $\sum_{i,j}(x_{ij} - \tilde{x}_{ij}) = 0$, $i, j = 1, \ldots, n$. Since Y is a Toeplitz matrix, and $y_l = y_{ij}$, $l = i - j$, $i, j = 1, 2, \ldots, n$. Then

$$\begin{aligned}\langle X - \mathcal{T}(X), Y\rangle &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij} - \tilde{x}_{ij})y_{ji} \\ &= \sum_{l=-n+1}^{n-1}(y_l \sum_{i-j=l}(x_{ij} - \tilde{x}_{ij})) \\ &= 0.\end{aligned}$$

□

Table 1
Comparison between APG, ALM and SALM for $p = 0.5$. Corresponding to each triplet $\{n, r, m/(2n-1)\}$, the MC problem was solved for the same data matrices M and $\mathcal{P}_\Omega(M)$ using the three different algorithms.

n	Rank (M)	Algorithm	IT	Time (s)	Error 1	Error 2	SP
$p = 0.5$							
		SALM	43	3.7728	8.4054e−10	3.7720e−08	1
500	10	ALM	46	15.904	8.9941e−10	2.9914e−03	4.2
		APG	101	10.159	8.9089e−06	3.6341e−04	2.7
		SALM	55	11.755	6.5826e−10	7.2567e−06	1
800	10	ALM	58	151.73	8.3419e−10	8.0959e−03	12.9
		APG	433	93.874	9.5902e−06	7.9012e−04	8.0
		SALM	57	11.871	8.2316e−10	4.4203e−09	1
1000	10	ALM	62	51.747	9.1452e−10	2.5361e−04	4.4
		APG	120	35.553	9.7347e−06	1.9991e−04	3.0
		SALM	62	24.182	9.9261e−10	3.7873e−09	1
1500	10	ALM	72	191.07	7.6265e−10	5.5638e−03	7.9
		APG	371	218.53	9.8653e−06	1.0021e−03	9.0
		SALM	68	43.288	8.9880e−10	7.7282e−09	1
2000	10	ALM	77	218.28	8.6379e−10	1.0700e−04	5.0
		APG	84	86.686	9.6895e−06	2.5169e−04	2.0
		SALM	72	117.13	8.1302e−10	2.3732e−08	1
2500	10	ALM	81	274.72	9.6323e−10	6.1304e−04	2.3
		APG	129	317.45	9.9785e−06	3.6951e−04	2.7
		SALM	74	160.34	7.5571e−10	8.9361e−09	1
3000	10	ALM	84	369.94	8.7250e−10	0.0016	2.3
		APG	234	828.91	9.5890e−06	0.0011	5.2
		SALM	81	333.37	9.7993e−10	1.1101e−04	1
4000	10	ALM	88	740.90	9.7141e−10	0.0011	2.2
		APG	213	1237.4	9.9670e−06	3.8087e−04	3.7
		SALM	86	501.75	9.4728e−10	1.1109e−05	1
5000	10	ALM	88	5255.9	8.4510e−10	0.0166	10.5
		APG	372	2955.5	9.8703e−06	5.5241e−04	5.9

Table 2
Comparison between APG, ALM and SALM for $p = 0.4$. Corresponding to each triplet $\{n, r, m/(2n-1)\}$, the MC problem was solved for the same data matrices M and $\mathcal{P}_\Omega(M)$ using the three different algorithms.

n	Rank (M)	Algorithm	IT	Time (s)	Error 1	Error 2	SP
$p = 0.4$							
		SALM	43	2.7308	6.7554e−10	5.6193e−09	1
500	10	ALM	46	105.41	8.9265e−10	3.5316e−02	38.6
		APG	116	10.976	9.0127e−06	5.7461e−04	4.0
		SALM	55	17.890	8.8774e−10	1.4611e−06	1
800	10	ALM	58	269.38	8.0161e−10	2.2739e−02	15.1
		APG	646	132.89	9.9574e−06	1.5520e−03	7.5
		SALM	57	12.849	8.4685e−10	1.2379e−08	1
1000	10	ALM	62	181.54	9.4945e−10	8.4332e−04	14.1
		APG	169	48.272	9.6239e−06	4.2304e−04	3.8
		SALM	62	25.562	9.6824e−10	8.3985e−09	1
1500	10	ALM	71	488.11	9.9396e−10	4.4316e−03	19.1
		APG	347	195.98	9.9600e−06	1.024e−03	7.7
		SALM	69	53.573	9.2787e−10	2.8387e−10	1
2000	10	ALM	77	287.58	9.4627e−10	1.5280e−04	5.4
		APG	80	154.54	9.0339e−06	2.8566e−04	2.9
		SALM	72	107.76	9.2031e−10	1.3326e−08	1
2500	10	ALM	84	879.65	8.3073e−10	3.1000e−03	8.2
		APG	146	342.16	9.8999e−06	8.5012e−04	3.2
		SALM	75	152.32	9.1238e−10	2.6371e−08	1
3000	10	ALM	85	586.27	8.5129e−10	0.0020	3.9
		APG	274	883.33	9.8645e−06	4.7709e−04	5.8
		SALM	78	299.66	9.8893e−10	1.5646e−08	1
4000	10	ALM	90	3504.8	8.8429e−10	0.0032	11.7
		APG	336	2115.5	9.9528e−06	4.7201e−04	7.1
		SALM	85	502.50	7.6911e−10	6.4065e−07	1
5000	10	ALM	88	8248.7	9.8558e−10	0.0098	16.4
		APG	454	3754.3	9.9072e−06	5.3508e−04	7.5

Table 3
Comparison between APG, ALM and SALM for $p = 0.3$. Corresponding to each triplet $\{n, r, m/(2n-1)\}$, the MC problem was solved for the same data matrices M and $\mathcal{P}_\Omega(M)$ using the three different algorithms.

n	Rank (M)	Algorithm	IT	Time (s)	Error 1	Error 2	SP
$p = 0.3$							
		SALM	41	28.634	6.0753e−10	1.4547e−07	1
500	10	ALM	46	112.06	7.1818e−10	7.7333e−02	3.9
		APG	503	69.074	9.9989e−06	8.5125e−04	2.4
		SALM	54	14.071	8.2150e−10	1.9203e−05	1
800	10	ALM	58	179.34	8.0852e−10	1.5020e−02	12.8
		APG	840	179.77	9.9586e−06	1.2379e−02	12.8
		SALM	59	16.732	7.8692e−10	1.8151e−08	1
1000	10	ALM	62	415.11	8.6558e−10	3.1824e−03	24.8
		APG	128	38.207	9.8932e−06	2.1673e−04	2.4
		SALM	62	24.764	9.0766e10	1.5073e−08	1
1500	10	ALM	71	988.61	9.8991e−10	4.5513e−02	39.9
		APG	718	424.97	9.9334e−06	1.9452e−03	17.2
		SALM	73	58.250	7.6945e−10	8.0801e−08	1
2000	10	ALM	77	550.36	9.1992e−10	2.2785e−04	9.5
		APG	107	177.284	8.728e−06	5.5930e−04	3.0
		SALM	76	134.894	8.4687e−10	2.0536e−07	1
2500	10	ALM	84	1286.3	8.9607e−10	1.3000e−03	9.5
		APG	364	828.91	9.5890e−06	1.1000e−03	6.1
		SALM	76	172.48	9.3651e−10	1.4683e−08	1
3000	10	ALM	85	2125.3	9.0453e−10	6.6000e−03	12.3
		APG	225	698.96	9.9330e−06	1.1000e−03	4.1
		SALM	82	400.28	8.1824e−10	2.7405e−07	1
4000	10	ALM	89	7473.5	9.3736e−10	4.1000e−03	18.7
		APG	487	3267.5	9.9985e−06	1.8000e−03	8.2
		SALM	86	647.29	8.0991e−10	1.6480e−04	1
5000	10	ALM	89	21099	8.4209e−08	1.9600e−02	32.6
		APG	408	4589.9	9.9351e−06	8.5115e−04	7.1

Table 4
Comparison between APG, ALM and SALM for $p = 0.25$. Corresponding to each triplet $\{n, r, m/(2n-1)\}$, the MC problem was solved for the same data matrices M and $\mathcal{P}_\Omega(M)$ using the three different algorithms.

n	Rank (M)	Algorithm	IT	Time (s)	Error 1	Error 2	SP
$p = 0.25$							
		SALM	46	15.171	5.7522e−10	2.5222e−05	1
500	10	ALM	46	41.875	9.3628e−10	1.6700e−02	2.8
		APG	206	39.334	9.9424e−06	7.7864e−04	2.5
		SALM	57	100.80	6.9817e−10	3.5101e−03	1
800	10	ALM	58	202.06	8.3957e−10	2.3285e−02	2
		APG	634	224.81	9.9160e−06	1.7676e−03	2.3
		SALM	60	279.07	9.1120e−10	9.5276e−05	1
1000	10	ALM	62	742.88	8.9951e−10	1.5600e−02	2.7
		APG	402	649.54	9.9749e−06	3.7883e−04	2.4
		SALM	68	54.736	8.0617e−10	1.7536e−05	1
1500	10	ALM	72	1065.9	7.7569e−10	4.2624e−02	19.5
		APG	660	380.75	9.9091e−06	2.7153e−03	7.0
		SALM	74	185.26	9.1436e−10	3.4410e−04	1
2000	10	ALM	77	895.75	9.1992e−10	3.5288e−04	4.8
		APG	175	264.58	9.4263e−06	4.6172e−04	1.5
		SALM	77	149.27	8.0392e−10	4.6381e−05	1
2500	10	ALM	84	1779.0	9.2647e−10	3.5000e−03	11.9
		APG	630	1383.9	9.9718e−06	8.3793e−04	9.3
		SALM	75	165.72	9.2929e−10	1.0924e−07	1
3000	10	ALM	85	4780.5	8.7486e−10	4.5000e−03	28.8
		APG	566	1890.5	9.9326e−06	1.6000e−03	11.4
		SALM	81	1884.2	9.1987e−10	9.3658e−04	1
4000	10	ALM	89	7540.5	8.8706e−10	3.8000e−03	4.0
		APG	391	2752.9	9.9568e−06	3.8000e−03	1.5
		SALM	83	601.72	9.7970e−10	1.7010e−05	1
5000	10	ALM	89	24398	8.5580e−10	4.3703e−02	40.5
		APG	633	6457.7	9.9100e−06	1.7038e−03	10.7

Table 5
Comparison between APG, ALM and SALM for $p = 0.2$. Corresponding to each triplet $\{n, r, m/(2n-1)\}$, the MC problem was solved for the same data matrices M and $\mathcal{P}_\Omega(M)$ using the three different algorithms.

n	Rank (M)	Algorithm	IT	Time (s)	Error 1	Error 2	SP
$p = 0.2$							
		SALM	45	20.362	20.362e−9	7.0326e−05	1
500	10	ALM	46	46.083	8.5668e−10	0.3604	2.3
		APG	712	45.567	9.9803e−06	6.1540e−04	2.2
		SALM	57	56.325	9.3656e−8	6.2835e−05	1
800	10	ALM	58	77.126	7.8797e−10	0.0171	1.4
		APG	747	114.18	9.9764e−06	0.0036	2.0
		SALM	61	60.877	8.2954e−10	2.2868e−04	1
1000	10	ALM	62	323.24	9.1712e−10	0.1836	5.3
		APG	603	124.10	9.7879e−06	6.3217e−04	2.0
		SALM	70	144.96	8.1523e−10	7.8311e−04	1
1500	10	ALM	72	664.16	8.0154e−10	0.1251	4.6
		APG	820	361.40	9.9819e−06	1.4000e−03	2.5
		SALM	74	275.95	8.9601e−10	8.8021e−05	1
2000	10	ALM	77	1521.3	8.3083e−10	0.0534	5.5
		APG	388	453.67	9.7411e−06	4.6751e−04	1.6
		SALM	79	101.90	9.1750e−10	1.4283e−05	1
2500	10	ALM	85	1843.9	8.2902e−10	0.2060	18.1
		APG	539	446.93	9.9985e−06	8.4510e−04	4.4
		SALM	81	394.63	9.5989e−10	5.8763e−05	1
3000	10	ALM	85	2357.4	8.3047e−10	0.0163	6.0
		APG	643	956.27	9.8914e−06	0.0018e−03	2.4
		SALM	85	377.24	9.9307e−10	4.4664e−04	1
4000	10	ALM	89	3427.2	8.5920e−10	0.0083	9.1
		APG	436	1241.1	9.9715e−06	8.1030e−04	3.3
		SALM	87	556.30	9.7556e−10	3.7391e−04	1
5000	10	ALM	91	1870.3	9.6191e−10	0.0387	3.4
		APG	579	2370.8	9.9033e−06	0.0025	4.3

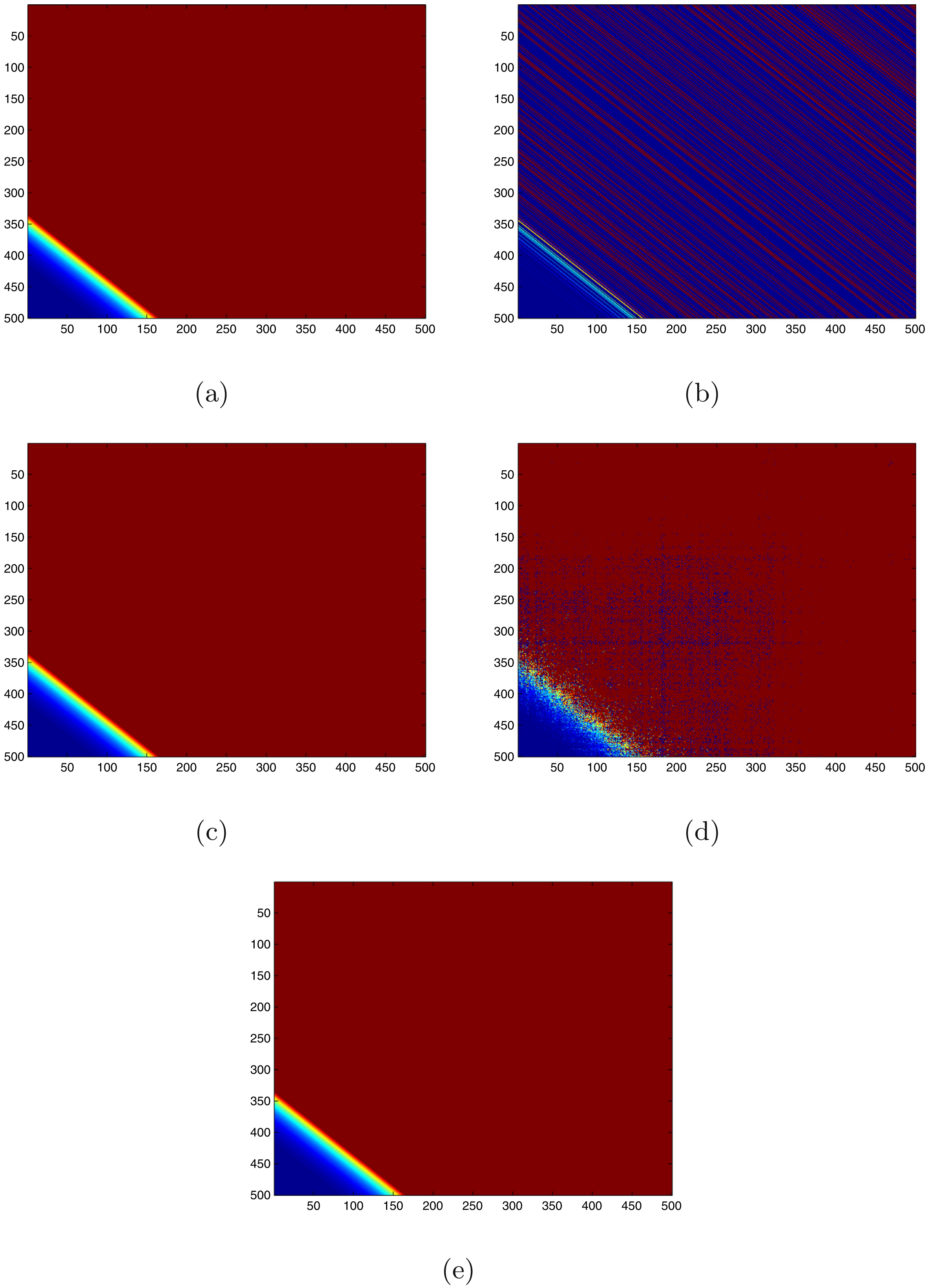

Fig. 1. (a) Original image, (b) damaged image, (c) inpainted image by the SALM algorithm, (d) inpainted image by the ALM algorithm, (e) inpainted image by the APG algorithm.

Theorem 3.3. *In Algorithm 3.1, A_k is a Toeplitz matrix generated by X_k. Then*

$$\|A_k - \ddot{A}\|_F < \|X_k - \ddot{A}\|_F,$$

where $\ddot{A}$ is the solution of (10).

Proof.

$$\begin{aligned}\|X_k - \ddot{A}\|_F^2 &= \|X_k - A_k + A_k - \ddot{A}\|_F^2 \\ &= \langle X_k - A_k, X_k - A_k\rangle + 2\langle X_k - A_k, A_k - \ddot{A}\rangle \\ &\quad + \langle A_k - \ddot{A}, A_k - \ddot{A}\rangle \\ &= \|X_k - A_k\|_F^2 + \|A_k - \ddot{A}\|_F^2.\end{aligned}$$

Thus, $\|A_k - \ddot{A}\|_F < \|X_k - \ddot{A}\|_F$. □

4. Numerical experiments

This section shows a brief comparison of the three algorithms (APG, ALM and SALM) for the Toplitz matrix completion problem. We conduct tests of randomly drawn $n \times n$ matrices from the model (10). Then through numerical experiments, we evaluate the performance of our SALM algorithm with the ALM algorithm and the APG algorithm (APG with line search) by analyzing iteration numbers (IT), computing time in second (time (s)), deviation (error 1, error 2) as well as the speed-up (SP) defined in the following. SP=$\frac{\text{time of the other algorithms}}{\text{time of the SALM algorithm}}$, error1=$\frac{\|X_{k+1}-X_k\|_F}{\|X_{k+1}\|_F}$, error 2= $\frac{\|X_{k+1}-M\|_F}{\|M\|_F}$, (for APG Algorithm) error 1 =$\frac{\|A+E-D\|_F}{\|D\|_F}$, error 2= $\frac{\|A-M\|_F}{\|M\|_F}$. (for ALM and SALM Algorithms)

In our tests, M denotes the real Toeplitz matrices, we suggest the sampling density $p = m/(2n-1)$, where m is the diagonal number of observed entries. Since the special structure of a Toeplitz matrix, we have $0 \le m \le 2n-1$. For Algorithm 3.1, we empirically set the parameters $\tau_0 = 1/\|D\|_2$, $\delta = 1.2172 + \frac{1.8588m}{n^2}$, $\epsilon_1 = 10^{-9}$, $\epsilon_2 = 5 \times 10^{-6}$. The parameters of the ALM algorithm are the same as Algorithm 3.1. For Algorithm 2.2 (APG), we suggest $\mu_0 = \|\mathcal{P}_\Omega(M)\|_2$, $\bar{\mu} = 10^{-4}\mu_0$, $\zeta = 0.8$, $L_f = 1$, $\epsilon = 1 \times 10^{-5}$.

Tests are conducted for $m = n \in \{500, 800, 1000, 1500, 2000, 2500, 3000, 4000, 5000\}$, $p \in \{0.2, 0.25, 0.3, 0.4, 0.5\}$ and the rank of the underlying matrix to be reconstructed $r(M) = 10$.

The experimental results are reported in the following Tables 1–5. Besides, we apply our SALM algorithm to a simply image processing using one of the experiment data. This simulation can be seen in Fig. 1 (After zooming in, a significant difference among subfigures can be found.).

From the tables, we can see that all algorithms can successfully compute approximate solutions satisfying the prescribed stopping criterion for all tested matrices of M, while our SALM algorithm significantly outperforms the APG and the ALM algorithms in terms of both number of iteration steps and computing time. In particular, we see that the convergent speed of the SALM algorithm is faster than that of the ALM and APG algorithms by the values of speed-up, up to 40.5.

5. Concluding remarks

Matrix completion involves recovering a matrix from a subset of its entries by exploiting interdependency between the entries typically through low-rank structure. It is well-known but NP-hard in general. Toeplitz matrix completion is one of the most important MC problems and has attracted widespread attention in recent years. As one feasible way for solving this kind of problem, we develop an augmented Lagrange multiplier algorithm with smoothing (SALM) for the Toeplitz matrix completion problem, and correspondingly, establish the convergence theory of the SALM algorithm. Both theoretical analyses and numerical computations have shown that the SALM algorithm is an effective algorithm for solving the TMC problem whose approximation matrices may hold on Toeplitz structure throughout the iteration by using the smoothing operator. It is worth pointing out that the SALM algorithm exhibits better convergence speed than the APG and ALM algorithms for solving the TMC problem.

Acknowledgments

The authors are very much indebted to the editor and anonymous referees for their helpful comments and suggestions which greatly improved the original manuscript of this paper.

This work is supported by NSF of China (11371275), NSF of Shanxi Province (201601D011004) and the SYYJSKC-1803.

References

[1] Y. Amit, M. Fink, N. Srebro, S. Ullman, Unconvering shared structures in malticalass classification, in: Proceeding of the Twenty-Fourth International Conference on Machine Learning, ACM, 2007, pp. 17–24.
[2] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, Adv. Neural Inf. Process. Syst. 19 (2007) 41–48.
[3] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009) 183–202.

[4] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Multi-task feature learning, image inpainting, Comput. Gr. 34 (2000) 417–424.
[5] J. Blanchard, J. Tanner, K. Wei, CGIHT: Conjugate Grandient Iterative Hard Thresholding for Compressed Sensing and Matrix Completion, University of Oxford, 2014. Numerical Analysis Group Preprint 14/08
[6] N. Boumal, P.A. Absil, RTRMC: a Riemannian trust-region method for low-rank matrix completion, in: J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. pereira, K.Q. Weinberger (Eds.), Proceedings of the Advances in Neural Information Processing Systems, NIPS, 24, 2011, pp. 406–414.
[7] J.F. Cai, E.J. Candès, Z. Shen, A singular value thresholding method for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[8] J.F. Cai, S. Liu, W. Xu, A fast algorithm for restruction of spectrally sparse signals in super-resolution, in: Proceedings of the SPIE Optical Engineering + Applications, International Society for Optics and Photonics, 2015, p. 95970A.
[9] J.F. Cai, X. Qu, W. Xu, G.B. Ye, Robust recovery of complex exponential signals from random gaussian projections via low-rank Hankel matrix reconstruction, Appl. Comput. Harmon. Anal. 41 (2) (2016) 470–490.
[10] J.F. Cai, T. Wang, K. Wei, Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank Hankel matrix completion, Appl. Comput. Harmon. Anal. 46 (1) (2019) 94–121.
[11] J.F. Cai, T. Wang, K. Wei, Spectral compressed sensing via projected gradient descent, SIAM J. Optim. 28 (3) (2018) 2625–2653.
[12] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[13] M. Chen, A. Ganesh, Z. Lin, Y. Ma, J. Wright, L. Wu, Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix, J. Mar. Biol. Assoc. UK 56 (3) (2009) 707–722.
[14] Y. Chen, Y. Chi, Robust spectral compressed sensing via structured matrix completion, IEEE Trans. Inf. Theory 60 (10) (2014) 6576–6601.
[15] L. Eldén, Matrix Methods in Data Mining and Pattern Recognization, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.
[16] M. Fazel, T.K. Pong, D. Sun, P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, SIAM J. Matrix Anal. Appl. 34 (2013) 946–977.
[17] N.J. Harvey, D.R. Karger, S. Yekhanin, The complexity of matrix completion, in: Proceeding of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2006, pp. 1103–1111.
[18] Y. Hu, D.B. Zhang, J. Liu, J.P. Ye, X.F. He, Accelerated singular value thresholding for matrix completion, in: Proceeding of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012.
[19] S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, SIAM J. Numer. Anal. 49 (6) (2011) 2543–2563.
[20] P. Jain, R. Meka, I. Dhillon, Guaranteed rank minimization via singular value projection, in: Proceeding of the Neural Information Processing Systems Conference, 2010, pp. 937–945.
[21] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing (STOC), 2013, pp. 665–674.
[22] A. Kyríllidis, V. Cevher, Matrix recipes for hard thresholding methods, J. Math. Imaging Vis. 48 (2) (2014) 235–265.
[23] Z. Lin, M. Chen, L. Wu, Y. Ma, The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices, UIUC Technical Report UIUL-ENG-09-2214, 2010.
[24] Z. Liu, L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM J. Matrix Anal. Appl. 31 (2009) 1235–1256.
[25] Z. Lu, Y. Zhang, X. Li, Penalty decomposition methods for rank minimization, Optim. Methods Softw. 30 (3) (2015) 531–558.
[26] F.T. Luk, S. Qiao, A fast singular value algorithm for Hankel matrices, in: V. Olshevsky (Ed.), Fast Algorithms for Structured Matrices: Theory and Applications (Contemporary Mathematics), 323, American Mathematical Society, 2003.
[27] M. Mesbahi, G.P. Papavassilopoulos, On the rank minimization problem over a positive semidefinite linear matrix inequality, IEEE Trans. Autom. Control 42 (1997) 239–243.
[28] B. Mishra, K.A. Apuroop, R. Sepulchre, A Riemannian Geometry for Low-Rank Matrix Completion, 2013, arXiv:1306.2672
[29] F. Sebert, Y.M. Zou, L. Ying, Toeplitz block matrices in compressed sensing and their applications in imaging, in: Proceedings of the International Conference on IEEE Information Technology and Applications in Biomedicine, 2008, pp. 47–50, doi:10.1109/ITAB.2008.4570587.
[30] A.K. Shaw, S. Pokala, R. Kumaresan, Toeplitz and Hankel approximation using structured approach. acoustics, in: Proceedings of the IEEE International Conference on Speech and Signal Processing, 2349, 1998, pp. 12–15.
[31] J. Tanner, K. Wei, Low-rank matrix completion by alternating steepest descent methods, Appl. Comput. Harmon. Anal. 40 (2016) 417–429.
[32] K.C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific J. Optim. 6 (2010) 615–640.
[33] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, Int. J. Comput. Vis. 9 (1992) 137–154.
[34] B. Vandereycken, Low-rank matrix completion by Riemannian optimization, SIAM J. Control Optim. 23 (2) (2013) 1214–1236.
[35] C.L. Wang, C. Li, A mean value algorithm for Toeplitz matrix completion, Appl. Math. Lett. 41 (2015) 35–40.
[36] C.L. Wang, C. Li, J. Wang, Comparisons of several algorithms for Toeplitz matrix recovery, Comput. Math. Appl. 71 (1) (2016) 133–146.
[37] R.P. Wen, X.H. Yan, A new gradient projection method for matrix completion, Appl. Math. Comput. 258 (2015) 537–544.
[38] R.P. Wen, L.X. Liu, The two-stage iteration algorithms based on the shortest distance for low-rank matrix completion, Appl. Math. Comput. 314 (2017) 133–141.
[39] J. Ying, H. Lu, Q. Wei, J.F. Cai, D. Guo, J. Wu, Z. Chen, X. Qu, Hankel matrix nuclear norm regularized tensor completion for n-dimensional exponential signal, IEEE Trans. Signal Process. 65 (2017) 3702–3717.