应 用 数 学
MATHEMATICA APPLICATA
2019, 32(4): 887-899

An ℓ-Step Modified Augmented Lagrange Multiplier Algorithm for Completing a Toeplitz Matrix

WEN Ruiping(温瑞萍), LI Shuzhen(李姝贞)
(Key Laboratory of Engineering & Computing Science, Shanxi Provincial Department of Education /Department of Mathematics, Taiyuan Normal University, Jinzhong 030619, China)

Abstract: Based on the modified augmented Lagrange multiplier (MALM) algorithm for Toeplitz matrix completion (TMC) proposed by WANG et al. (2016), we put forward an accelerated technique to MALM algorithm, which will reduce the extra load coming from data communication. It is drawn that an ℓ-step modified augmented Lagrange multiplier algorithm. Meanwhile, we demonstrate the convergence theory of the new algorithm. Finally, numerical experiments show that the ℓ-step modified augmented Lagrange multiplier (ℓ-MALM) algorithm is more effective than the MALM algorithm.
Key words: Toeplitz matrix; Matrix completion; Augmented Lagrange multiplier; Data communication
CLC Number: O241.6 **AMS(2000)Subject Classification:** 15A83; 90C25
Document code: A **Article ID:** 1001-9847(2019)04-0887-13

1. Introduction

Completing an unknown low-rank or approximately low-rank matrix from a sampling of its entries is a challenging problem applied in many fields of information science. For example, machine learning[1−2], control theory[11], image inpainting[4], computer vision[14] etc. Matrix completion (MC) problem first introduced by Candès and Recht[6] is from portion of the observed matrix elements to fill a low-rank matrix as precisely as possible, it is a hot spot of researching in recent years. The famous recommendation system of the Netflix[3] is the typical application of matrix completion. Mathematical language is expressed as follows:

$$\begin{aligned} \min_{A\in\mathbb{R}^{m\times n}} \quad & \operatorname{rank}(A) \\ \text{s.t.} \quad & \mathcal{P}_\Omega(A)=\mathcal{P}_\Omega(M), \end{aligned} \tag{1.1}$$

where the matrix $M\in\mathbb{R}^{m\times n}$ is the underlying matrix to be reconstructed, $\Omega\subset\{1,2,\cdots,m\}\times\{1,2,\cdots,n\}$ represents the random subset of indices for the known entries to the sampling

* **Received date:** 2018-10-17
Foundation item: Supported by NSF of China(11371275), NSF of Shanxi Province(201601D011004), and the SYYJSKC-1803
Biography: WEN Ruiping, female, Han, Shanxi, professor, major in numerical algebra and its applications.

matrix M, and $\mathcal{P}_{\Omega}$ is associated sampling orthogonal projection operator on Ω. The purpose of the optimization problem is to reduce the rank of matrix obtained by filling missing elements as far as possible. In other words, the model (1.1) optimizes the structure of the matrix.

Although matrix completion needs solving the global solution of the non-convex objective problem, there are still many effective algorithms which are applicable to some specific matrix. Many researchers have suggested that the most of a low-rank matrix can be accurately completed according to the known entries at specified accuracy. In a low dimensional linear subspace, the meaning of precision is that we can realize the reasonable matrix completion under the hypothesis of the lowest-rank data matrix.

However, the optimization problem (1.1) is an NP-hard problem, in theory and practice, the computational complexity of the existing algorithms is a double exponential function about the matrix dimension. As a result, the computations are cumbersome and the data take up a large amount of computer memory. Therefore, the solving of the problem (1.1) is usually expensive by the existing algorithms. Hence, using the relation between the rank and the nuclear norm of a matrix, Candès and Recht [6] made the equivalent form of the problem (1.1) as follows:

$$\begin{aligned} \min_{A\in\mathbb{R}^{m\times n}} \quad & \|A\|_* \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(A)=\mathcal{P}_{\Omega}(M), \end{aligned} \tag{1.2}$$

where $M\in\mathbb{R}^{m\times n}$ is the underlying matrix, $\|A\|_*=\sum\limits_{k=1}^{r}\sigma_k(A)$, $\sigma_k(A)$ denotes the k-th largest singular value of the r-rank matrix $A\in\mathbb{R}^{m\times n}$, namely $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_k\geq\cdots\geq\sigma_r>0$. The problem (1.1) is transformed into a convex optimization problem which is easy to solve.

Academic circles have made abundant research results in solving the optimization problem (1.2) such as the augmented Lagrange multiplier (ALM) algorithm[9], the singular value theresholding (SVT) algorithm as well as its variants[5, 8, 19], the accelerated proximal gradient (APG) algorithm[15], and the modified augmented Lagrange multiplier (MALM) algorithm[17] etc. On the practical problems, the sampling matrix often has a special structure, for instance, the Toeplitz and Hankel matrices. Many scholars have conducted in-depth research on the special structure, property and application of the Toeplitz and Hankel matrix in recent years[10,12−13,16,18]. As we well know, an $n\times n$ Toeplitz matrix can be expressed as the following form:

$$T=\begin{pmatrix} t_0 & t_1 & \cdots & t_{n-2} & t_{n-1} \\ t_{-1} & t_0 & \cdots & t_{n-3} & t_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{-n+2} & t_{-n+3} & \cdots & t_0 & t_1 \\ t_{-n+1} & t_{-n+2} & \cdots & t_{-1} & t_0 \end{pmatrix}\in\mathbb{R}^{n\times n}, \tag{1.3}$$

which is determined by $2n-1$ entries.

As for MALM algorithm, the intuitive idea is to switch iteration matrix into Toeplitz structure at each step by an operator. To implement this idea, a heavy data has to be moved at each iterate step. However, there is a cost, sometimes relatively great, associated with the moving of data. The control of memory traffic is crucial to performance in many computers.

In order to reduce the traffic jam of data, an ℓ-step modified ALM algorithm is proposed in this paper. Compared with the MALM algorithm, the new algorithm saves computation cost and reduces data communication. Two aspects are taken into account, which result in a more practical or economic implementing. The new algorithm not only overcomes the slowness produced by the computing of singular value decomposition in ALM algorithm, but also saves the data congestion caused by the data communication in MALM algorithm. Compared with the CPU of the MALM algorithm, we can see that the CPU of the ℓ-MALM algorithm is reduced to 45%.

Here are some of the necessary notations and preliminaries. $\mathbb{R}^{m\times n}$ denotes the set of $m\times n$ real matrix. The nuclear norm of a matrix A is denoted by $\|A\|_*$, the Frobenius norm by $\|A\|_F$ is the sum of absolute values of the matrix entries of A. A^{T} is used to denote the transpose of a matrix $A\in\mathbb{R}^{n\times n}$, rank$(A)$ stands for the rank of A and tr(A) represents the trace of A. The standard inner product between two matrices is denoted by $\langle X,Y\rangle=\mathrm{tr}(X^{\mathrm{T}}Y)$. $\Omega\subset\{-n+1,\cdots,n-1\}$ is the indices set of observed diagonals of a Toeplitz matrix $M\in\mathbb{R}^{n\times n}$, $\bar{\Omega}$ is the complementary set of Ω. For a Toeplitz matrix $A\in\mathbb{R}^{n\times n}$, the vector vec$(A,l)$ denotes a vector reshaped from the l-th diagonal of A, $l=-n+1,\cdots,n-1$, $\mathcal{P}_\Omega$ is the orthogonal projector on Ω, satisfying

$$\mathrm{vec}(\mathcal{P}_\Omega(A),l)=\begin{cases}\mathrm{vec}(A,l), & l\in\Omega\\ \mathbf{0}, & l\notin\Omega\end{cases}\quad(\mathbf{0}\text{ is a zero-vector}).$$

Definition 1.1(Singular value decomposition (SVD)) The singular value decomposition of a matrix $A\in\mathbb{R}^{m\times n}$ of r-rank is defined as follows:

$$A=U\Sigma_rV^{\mathrm{T}},\quad \Sigma_r=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r),$$

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

Definition 1.2 (Singular value thresholding operator)[5] For each $\tau\ge0$, the singular value thresholding operator $\mathcal{D}_\tau$ is defined as follows:

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

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

Definition 1.3 The matrices

$$T_l=(t_{ij})_{n\times n}=\begin{cases}1, & i-j=l\\ 0, & i-j\ne l\end{cases},\quad l=-n+1,\cdots,n-1, \tag{1.4}$$

are called the basis of the Toeplitz matrix space.

It is clear that a Toeplitz matrix $T\in\mathbb{R}^{n\times n}$, shown in (1.3), can be rewritten as a linear combination of these basis matrices, that is,

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

Definition 1.4(Toeplitz structure smoothing operator) For any matrix $A=(a_{ij})\in\mathbb{R}^{n\times n}$, the Toeplitz structure mean operator $\mathcal{T}$ is defined as follows:

$$\mathcal{T}(A):=\sum_{l=-n+1}^{n-1}\tilde{a}_lT_l, \tag{1.5}$$

where $\tilde{a}_l=\mathrm{mean}(\mathrm{vec}(A,l)),l=-n+1,\cdots,n-1$, namely $\tilde{a}_l$ represents the mean value of all components of vec(A,l). It is evident that $\mathcal{T}(A)$ is a Toeplitz matrix derived from the matrix

A. In other words, $\forall A \in \mathbb{R}^{n\times n}$ can be switched into a Toeplitz structure via the smoothing operator $\mathcal{T}(\cdot)$.

The rest of the paper is organized as follows. After we review briefly the ALM, MALM algorithms and the dual approach, an ℓ-step modified ALM algorithm will be proposed in Section 2. Next, the convergence analysis is given in Section 3. Then the numerical results are provided to show the effectiveness of the ℓ-MALM in Section 4. Finally, the paper ends with a conclusion in Section 5.

2. Relative 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 way as RPCA, an equivalent problem of (1.2) can be considered as follows. In terms of estimating the low-dimensional subspace, the purpose of the mathematical model 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) to minimize the difference between matrix A and M, generating the following constraint optimization problem model:

$$\begin{array}{ll}\min\limits_{A,E\in\mathbb{R}^{m\times n}} & \|A\|_* \\ \text{s.t.} & A+E=M, \mathcal{P}_\Omega(E)=0,\end{array} \tag{2.1}$$

where E will compensate for the unknown entries of M, the unknown entries of $M \in \mathbb{R}^{m\times n}$ are simply set as zeros. And $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear operator that keeps the entries in Ω unchanged and sets those outside Ω (say, in $\overline{\Omega}$) zeros. Then we introduce the algorithm for solving problem (2.1).

Ⅰ The augmented Lagrange multiplier (ALM) algorithm

It is famous that partial augmented Lagrangian function of the problem (2.1) is

$$\mathcal{L}(A,E,Y,\mu) = \|A\|_* + \langle Y, M-A-E\rangle + \frac{\mu}{2}\|M-A-E\|_F^2. \tag{2.2}$$

Hence, the augmented Lagrange multiplier (ALM) algorithm[9] is designed as follows.

Algorithm 2.1 Step 0 Give Ω, sampled matrix $D = \mathcal{P}_\Omega(M)$, $\mu_0 > 0$, $\rho > 1$. Give also two initial matrices $Y_0 = 0$, $E_0 = 0$. $k := 0$;

Step 1 Compute the SVD of the matrix $(D - E_k + \mu_k^{-1}Y_k)$,

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

Step 2 Set

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

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

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

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 Step 4;

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.

Remark It is reported that the ALM algorithm performs better both in theory and algorithms than the others that with a Q-linear convergence speed globally. It is of much better numerical behavior, and it is also of much higher accuracy. However, the algorithm has a disadvantage of the penalty function: the matrix sequences $\{X_k\}$ generated by Algorithm 2.1 are not feasible. Hence, the accepted solutions are not feasible.

Ⅱ The dual algorithm

The dual algorithm proposed in [7] tackles the problem (2.1) via its dual. That is, one first solves the dual problem

$$\begin{aligned} \max_{Y} \quad & \langle M, Y \rangle \\ \text{s.t.} \quad & J(Y) \leq 1, \end{aligned} \tag{2.3}$$

for the optimal Lagrange multiplier Y, where

$$J(Y) = \max(\|Y\|_2, \lambda^{-1}\|Y\|_\infty). \tag{2.4}$$

A steepest ascend algorithm constrained on the surface $\{Y|J(Y) = 1\}$ can be adopted to solve (2.3) , where the constrained steepest ascend direction is obtained by projecting M onto the tangent cone of the convex body $\{Y|J(Y) \leq 1\}$. It turns out that the optimal solution to the primal problem (2.1) can be obtained during the process of finding the constrained steepest ascend direction.

Ⅲ The modified augmented Lagrange multiplier (MALM) algorithm

In this pant, we mention a mean-value technique for TMC problem[17]. The problem can be expressed as the following convex programming,

$$\begin{aligned} \min_{A,E\in\mathbb{R}^{n\times n}} \quad & \|A\|_* \\ \text{s.t.} \quad & A + E = \mathcal{P}_\Omega(M), \mathcal{P}_\Omega(E) = 0, \end{aligned} \tag{2.5}$$

where $A, M \in \mathbb{R}^{n\times n}$ are both real Toeplitz matrices, $\Omega \subset \{-n+1, \cdots, n-1\}$. Let $D = \mathcal{P}_\Omega(M)$. Then the partial augmented Lagrangian function is

$$\mathcal{L}(A, E, Y, \mu) = \|A\|_* + \langle Y, D - A - E\rangle + \frac{\mu}{2}\|D - A - E\|_F^2, \tag{2.6}$$

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

Algorithm 2.2 Step 0 Give Ω, sampled matrix D, $\mu_0 > 0$, $\rho > 1$. Give 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

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

Step 2 Set

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

and compute $\tilde{a}_l = \text{mean}(\text{vec}(X_{k+1}, l))$, $l = -n+1, \cdots, n-1$, and

$$A_{k+1} = \mathcal{T}(X_{k+1}) = \sum_{l=-n+1}^{n-1} \tilde{a}_l T_l, E_{k+1} = \mathcal{P}_{\bar{\Omega}}(D - A_{k+1} + \mu_k^{-1}Y_k);$$

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 Step 4;

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.

Remark It is reported that MALM algorithm performs better that of much higher accuracy. Compared with the ALM, APGL, and SVT algorithms, the MALM algorithm is advantageous over the other three algorithms on the time costed by the SVD for smoothing at each iterate.

As we know, the saving of the SVD time is at the expense of data communication. Sometimes, this is not worth the candle. This motivated us to put up with the following algorithm.

Ⅳ The ℓ-step modified augmented Lagrange multiplier (ℓ-MALM) algorithm

To reduce the workload of data being moved at each iteration step, we propose a new accelerated algorithm for the TMC problem, which is smoothing once the diagonal elements of the iteration matrix by (1.5) for every ℓ steps. The technique saves computation cost and reduces the data communication. It turns out that the iteration matrices keep a Toeplitz structure, which ensure the fast SVD of Toeplitz matrices can be utilized.

Algorithm 2.3 (ℓ-MALM algorithm)

Input: Ω, sampled matrix D, $Y_{0,0} = 0$, $E_{0,0} = 0$; parameters $\mu_0 > 0$, $\rho > 1$, ℓ, ϵ_1, ϵ_2. Let $k := 0$, $q := 1$, $q = 1, 2, \cdots, \ell - 1$.

Repeat:

Step 1 $\ell - 1$ iterations.

1) Compute the SVD of the matrix $(D - E_{k,q} + \mu_{k,q}^{-1} Y_{k,q})$ using the Lanczos method

$$[U_{k,q}, \Sigma_{k,q}, V_{k,q}] = \text{lansvd}(D - E_{k,q} + \mu_{k,q}^{-1} Y_{k,q});$$

2) Set

$$X_{k+1,q+1} = U_{k,q} \mathcal{D}_{\mu_{k,q}^{-1}}(\Sigma_{k,q}) V_{k,q}^{\mathrm{T}},$$
$$E_{k+1,q+1} = \mathcal{P}_{\bar{\Omega}}(D - X_{k+1,q+1} + \mu_{k,q}^{-1} Y_{k,q});$$

3) If $\|D - X_{k+1,q+1} - E_{k+1,q+1}\|_F / \|D\|_F < \epsilon_1$ and $\mu_{k,q} \|E_{k+1,q+1} - E_{k,q}\|_F / \|D\|_F < \epsilon_2$, stop; otherwise, go to Step 1 4);

4) Set $Y_{k+1,q+1} = Y_{k,q} + \mu_{k,q}(D - X_{k+1,q+1} - E_{k+1,q+1})$, $\mu_{k+1,q+1} = \rho \mu_{k,q}$; otherwise, go to Step 1 1);

Step 2 ℓ-th smoothing.

1) Compute

$$[U_{k,\ell}, \Sigma_{k,\ell}, V_{k,\ell}] = \text{lansvd}(D - E_{k,\ell} + \mu_{k,\ell}^{-1} Y_{k,\ell});$$

2) $X_{k+1,\ell} = U_{k,\ell} \mathcal{D}_{\mu_{k,\ell}^{-1}}(\Sigma_{k,\ell}) V_{k,\ell}^{\mathrm{T}}, E_{k+1,\ell} = \mathcal{P}_{\bar{\Omega}}(D - X_{k+1,\ell} + \mu_{k,\ell}^{-1} Y_{k,\ell});$
Compute for smoothing $\tilde{a}_l = \text{mean}(\text{vec}(X_{k+1,\ell}, l))$, $l = -n + 1, \cdots, n - 1$,

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

Update $E_{k+1,\ell} = \mathcal{P}_{\bar{\Omega}}(D - A_{k+1,\ell} + \mu_{k,\ell}^{-1} Y_{k+1,\ell});$

Step 3 If $\|D - A_{k+1,\ell} - E_{k+1,\ell}\|_F / \|D\|_F < \epsilon_1$ and $\mu_{k,\ell} \|E_{k+1,\ell} - E_{k,\ell}\|_F / \|D\|_F < \epsilon_2$, stop; otherwise, go to Step 4;

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

Remark Clearly, this algorithm is an acceleration of the MALM algorithm in [17]. When $\ell = 1$, it becomes the MALM scheme.

3. Convergence Analysis

We provided first some lemmas in the following.

Let $(\ddot{A}, \ddot{E})$ be the solution of the model (2.5) and $\ddot{Y}$ be that of the optimal problem (2.3).

Lemma 3.1 [6] Let $A \in \mathbb{R}^{m\times n}$ be an arbitrary matrix and $U\Sigma V^{\mathrm{T}}$ be its SVD. Then the set of subgradients of the nuclear norm of A is provided by

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

Lemma 3.2 [9] If μ_k is nondecreasing then each term of the following series is nonnegative and the series is convergent, that is,

$$\sum_{k=1}^{+\infty} {\mu_k}^{-1}(\langle Y_{k+1}-Y_k, E_{k+1}-E_k\rangle + \langle A_{k+1}-\ddot{A}, \hat{Y}_{k+1}-\ddot{Y}\rangle + \langle E_{k+1}-\ddot{E}, Y_{k+1}-\ddot{Y}\rangle) < +\infty. \quad (3.1)$$

Lemma 3.3 [9] 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 $\{Y_{k,q}\}$ generated by Algorithm 2.3 is bounded.

Proof Let $B = \mu_{k,q}(D - E_{k,q} + \mu_{k,q}^{-1}Y_{k,q} - X_{k+1,q+1})$, $\mathcal{T}(B) = \sum\limits_{l\in\Omega} \tilde{b}_l T_l$, defined as (1.5).

First of all, we indicate that $Y_{k,q}, E_{k,q}, k = 1, 2, \cdots, q = 1, 2, \cdots, \ell-1$ are all Toeplitz matrices. Evidently, $Y_{0,0} = 0, E_{0,0} = 0$ are both smoothed into Toeplitz matrices. Suppose that after $Y_{k,q}, E_{k,q}$ are both Toeplitz matrices, so is $E_{k+1,q+1} = \mathcal{P}_{\bar{\Omega}}(D - A_{k+1,q+1} + \mu_{k,q}^{-1}Y_{k,q})$. Thus, $Y_{k+1,q+1}$ is a Toeplitz matrix also from the Step 4 in Algorithm 2.3.

$$\begin{aligned} Y_{k+1,q+1} =& Y_{k,q} + \mu_{k,q}(D - A_{k+1,q+1} - E_{k+1,q+1}) \\ =& Y_{k,q} + \mu_{k,q}(D - A_{k+1,q+1} - E_{k,q}) + \mu_{k,q}(E_{k,q} - E_{k+1,q+1}). \end{aligned}$$

And,

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

It is clear that by Steps 1-2 in Algorithm 2.3,

$$D - E_{k,q} + \mu_{k,q}^{-1}Y_{k,q} = \check{U}_{k,q}\check{\Sigma}_{k,q}\check{V}_{k,q}^{\mathrm{T}} + \tilde{U}_{k,q}\tilde{\Sigma}_{k,q}\tilde{V}_{k,q}^{\mathrm{T}},$$

where $\check{U}_{k,q}, \check{V}_{k,q}$ are the singular vectors associated with singular values that are more than $\frac{1}{\mu_{k,q}}$ and $\tilde{U}_{k,q}, \tilde{V}_{k,q}$ are those associated with singular values that are not more than $\frac{1}{\mu_{k,q}}$, the elements of the diagonal matrix $\check{\Sigma}_{k,q}$ are more than $\frac{1}{\mu_{k,q}}$ and those of the diagonal matrix $\tilde{\Sigma}_{k,q}$ are not more than $\frac{1}{\mu_{k,q}}$. Hence, it is drawn that $X_{k+1,q+1} = \check{U}_{k,q}(\check{\Sigma}_{k,q} - \mu_{k,q}^{-1}I)\check{V}_{k,q}^{\mathrm{T}}$ and

$$\begin{aligned} \|B\|_F =& \|\mu_{k,q}(D - E_{k,q} + \mu_{k,q}^{-1}Y_{k,q} - X_{k+1,q+1})\|_F \\ =& \|\mu_{k,q}(\mu_{k,q}^{-1}\check{U}_{k,q}\check{V}_{k,q}^{\mathrm{T}} + \tilde{U}_{k,q}\tilde{\Sigma}_{k,q}\tilde{V}_{k,q}^{\mathrm{T}})\|_F \\ =& \|\check{U}_{k,q}\check{V}_{k,q}^{\mathrm{T}} + \mu_{k,q}\tilde{U}_{k,q}\tilde{\Sigma}_{k,q}\tilde{V}_{k,q}^{\mathrm{T}}\|_F \\ \le& \sqrt{n}. \end{aligned}$$

Hence we can obtain that $Y_{k,q} + \mu_{k,q}(D - A_{k+1,q+1} - E_{k,q}) \in \partial\|A_{k+1,q+1}\|_*$ from Lemmas 3.2 and 3.3. It is known that for $A_{k+1,q+1} = U\Sigma V^{\mathrm{T}}$ by Lemma 3.1,

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

We have also,

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

Therefore, the following inequalities can be obtained:

$$\|Y_{k,q} + \mu_{k,q}(D - A_{k+1,q+1} - E_{k,q})\|_F \le \sqrt{n},$$

and

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

It is clear that the sequence $\{Y_{k,q}\}$ is bounded.

Theorem 3.1 Suppose that $\langle A_{k+1,q+1} - A_{k,q}, D - A_{k+1,q+1} - E_{k,q}\rangle \ge 0$, then the sequence $\{A_{k,q}\}$ converges to the solution of (2.5) when $\mu_{k,q} \to \infty$ and $\sum\limits_{k=1}^{+\infty} \mu_{k,q}^{-1} = +\infty$.

Proof It is true that

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

since $\mu_{k,q}^{-1}(Y_{k+1,q+1} - Y_{k,q}) = D - A_{k+1,q+1} - E_{k+1,q+1}$ and Lemma 3.4. Let $(\ddot{A}, \ddot{E})$ be the solution of (2.5). Then $A_{k+1,q+1}, Y_{k+1,q+1}, E_{k+1,q+1}, k = 1, 2, \cdots$, are all Toeplitz matrices from $\ddot{A} + \ddot{E} = D$. We prove first that

$$\begin{aligned}
&\|E_{k+1,q+1} - \ddot{E}\|_F^2 + \mu_{k,q}^{-2}\|Y_{k+1,q+1} - \ddot{Y}\|_F^2 \\
=&\|E_{k,q} - \ddot{E}\|_F^2 - \|E_{k+1,q+1} - E_{k,q}\|_F^2 + \mu_{k,q}^{-2}\|Y_{k,q} - \ddot{Y}\|_F^2 \\
&- \mu_{k,q}^{-2}\|Y_{k+1,q+1} - Y_{k,q}\|_F^2 - 2\mu_{k,q}^{-1}\langle A_{k+1,q+1} - \ddot{A}, \hat{Y}_{k+1,q+1} - \ddot{Y}\rangle, \qquad (3.2)
\end{aligned}$$

where $\hat{Y}_{k+1,q+1} = Y_{k,q} + \mu_{k,q}(D - A_{k+1,q+1} - E_{k,q})$, $\ddot{Y}$ is the optimal solution to the dual problem (2.3).

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

We obtain the following result through the same analysis,

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

Then

$$\min_{A\in\mathbb{R}^{m\times n}} \quad \mu\|A\|_* + \tfrac{1}{2}\|\mathcal{P}_{\Omega}(A - M)\|_F^2 \qquad (3.3)$$

holds true.

The series $\sum\limits_{k=1}^{\infty} \mu_{k,q}^{-1}\langle A_{k,q} - \ddot{A}, \hat{Y}_{k,q} - \ddot{Y}\rangle$ is convergent and $\|E_{k,q} - \ddot{E}\|^2 + \mu_{k,q}^{-2}\|Y_{k,q} - \ddot{Y}\|^2$ is nonincreasing from $\|A\|_*$ is a convex function and $\hat{Y}_{k+1,q+1} \in \partial\|A\|_*$, $\langle A_{k+1,q+1} - \ddot{A}, \hat{Y}_{k+1,q+1} - \ddot{Y}\rangle \ge 0$.

On the other hand, the following is true by Algorithm 2.3:

$$\langle Y_{k,q}, E_{k,q} - \ddot{E}\rangle = 0, \text{and} \langle \ddot{Y}, E_{k,q} - \ddot{E}\rangle = 0.$$

Moreover, along the same idea of Theorem 2 in [9], it is obtained that $\ddot{A}$ is the solution of (2.5).

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 (1.5). 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 $\sum\limits_{i,j}(x_{ij} - \tilde{x}_{ij}) = 0$, $i, j = 1, 2, \cdots, n$. Since Y is a Toeplitz matrix, and $y_l = y_{ij}$, $l = i - j$, $i, j = 1, 2, \cdots, n$. Then

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

Theorem 3.3 In Algorithm 2.3, $A_{k,q}$ is a Toeplitz matrix derived by $X_{k,q}$. Then

$$\|A_{k,q} - \ddot{A}\|_F < \|X_{k,q} - \ddot{A}\|_F,$$

where $\ddot{A}$ is the solution of (2.5).

Proof

$$\begin{aligned}\|X_{k,q} - \ddot{A}\|_F^2 =&\|X_{k,q} - A_{k,q} + A_{k,q} - \ddot{A}\|_F^2\\ =&\langle X_{k,q} - A_{k,q}, X_{k,q} - A_{k,q}\rangle + 2\langle X_{k,q} - A_{k,q}, A_{k,q} - \ddot{A}\rangle + \langle A_{k,q} - \ddot{A}, A_{k,q} - \ddot{A}\rangle\\ =&\|X_{k,q} - A_{k,q}\|_F^2 + \|A_{k,q} - \ddot{A}\|_F^2.\end{aligned}$$

Thus, $\|A_{k,q} - \ddot{A}\|_F < \|X_{k,q} - \ddot{A}\|_F$.

4. Numerical Experiments

In this section, some original numerical results of two algorithms (MALM, ℓ-MALM) are presented for the $n \times n$ matrices with different ranks. We conducted numerical experiments on the same and modest workstation. By analyzing and comparing iteration numbers (IT), computing time in second (time(s)), deviation (error 1, error 2) and RATIO which are defined in the following, we can see that the ℓ-MALM algorithm proposed by this paper is far more effective than the MALM algorithm.

$$\text{error 1} = \frac{\|A + E - D\|_F}{\|D\|_F}, \quad \text{error 2} = \frac{\|A - M\|_F}{\|M\|_F},$$

$$\text{RATIO} = \frac{\text{the CPU of the } \ell\text{-MALM algorithm}}{\text{the CPU of the MALM algorithm}} \times 100\%.$$

In our experiments, $M \in \mathbb{R}^{n\times n}$ represents the Toeplitz matrix. We select the sampling density $p = m/(2n - 1)$, where m is the number of the observed diagonal entries of M, then $0 \le m \le 2n - 1$. With regard to the ℓ-MALM algorithm, we set the parameters $\tau_0 = 1/\|D\|_2$, $\delta = 1.2172 + \frac{1.8588m}{n^2}$, $\epsilon_1 = 10^{-9}$, $\epsilon_2 = 5 \times 10^{-6}$ and $\ell = 3$ as a rule of thumb. The parameters of the MALM algorithm take the same as the ℓ-MALM algorithm.

The experimental results of two algorithms are shown in Tables 4.1-4.4. From the tables, two algorithms can successfully calculate the approximate solution of prescriptive stop condition for all the test matrices M. And our ℓ-MALM algorithm in computing time is far less than that of the MALM algorithm. In particular, compared with the CPU of the MALM algorithm, we can find that the CPU of the ℓ-MALM algorithm is reduced to 45%. The "RATIO" in Table 4.5 can show this effectiveness.

Table 4.1 Comparison between MALM and ℓ-MALM for $p = 0.6$.

n	rank(M)	Algorithm	IT	time(s)	error 1	error 2
500	10	MALM	41	5.6849	6.2296e-10	2.0730e-07
		ℓ-MALM	42	4.3861	4.9434e-10	5.2983e-08
800	10	MALM	51	15.6330	8.7995e-10	1.1466e-06
		ℓ-MALM	51	11.3258	7.3819e-10	5.8595e-09
1000	10	MALM	54	19.9837	6.9070e-10	4.9044e-09
		ℓ-MALM	54	16.3703	8.7574e-10	2.2620e-09
1500	10	MALM	60	42.4736	8.0852e-10	1.7307e-08
		ℓ-MALM	60	33.3273	9.5563e-10	3.7126e-09
2000	10	MALM	68	73.3195	8.5026e-10	3.2041e-08
		ℓ-MALM	66	59.5894	9.5805e-10	3.4934e-09
2500	10	MALM	71	108.6437	8.3229e-10	4.7617e-08
		ℓ-MALM	69	87.2279	8.3959e-10	1.4779e-09
3000	10	MALM	72	149.1788	8.2695e-10	5.9963e-05
		ℓ-MALM	75	128.7656	4.1858e-10	4.8568e-08
4000	20	MALM	69	265.5661	7.5063e-10	3.0654e-06
		ℓ-MALM	69	223.4528	6.9301e-10	1.6058e-08
5000	20	MALM	69	364.0319	9.8621e-10	2.0684e-07
		ℓ-MALM	72	319.3049	6.9864e-10	1.4224e-08
8000	25	MALM	76	985.8195	5.5826e-10	5.2859e-08
		ℓ-MALM	75	808.5147	5.7117e-10	1.9585e-06

Table 4.2 Comparison between MALM and ℓ-MALM for $p = 0.5$.

n	rank(M)	Algorithm	IT	time(s)	error 1	error 2
500	10	MALM	42	6.1728	4.9915e-10	5.2160e-07
		ℓ-MALM	42	4.5122	7.3772e-10	4.1083e-09
800	10	MALM	53	17.0730	7.4750e-10	6.6172e-06
		ℓ-MALM	51	11.4760	4.1454e-10	1.4190e-09
1000	10	MALM	55	21.8193	9.3316e-10	3.2335e-07
		ℓ-MALM	54	16.2247	8.8878e-10	1.9908e-09
1500	10	MALM	63	47.7943	7.9557e-10	3.1867e-07
		ℓ-MALM	63	38.0657	4.9309e-10	2.8648e-09
2000	10	MALM	68	73.0738	7.7554e-10	3.5540e-08
		ℓ-MALM	66	60.4303	8.0263e-10	1.2006e-09
2500	10	MALM	74	110.7135	9.4926e-10	8.5401e-08
		ℓ-MALM	72	89.4616	8.5799e-10	3.4276e-09
3000	10	MALM	76	144.7241	8.6824e-10	3.4200e-08
		ℓ-MALM	75	123.6858	6.6697e-10	1.2446e-09
4000	20	MALM	70	296.2682	7.4389e-10	7.4712e-05
		ℓ-MALM	69	220.3868	5.0967e-10	6.5335e-09
5000	20	MALM	72	417.4897	8.5359e-10	1.1646e-06
		ℓ-MALM	75	377.9499	7.1757e-10	7.1559e-09
8000	25	MALM	75	1.0022e+03	9.5530e-10	2.7559e-08
		ℓ-MALM	75	832.5463	4.2152e-10	1.7814e-09

Table 4.3 Comparison between MALM and ℓ-MALM for $p = 0.4$.

n	rank(M)	Algorithm	IT	time(s)	error 1	error 2
500	10	MALM	44	6.2275	8.8258e-10	6.7251e-09
		ℓ-MALM	42	4.7537	7.4998e-10	1.2431e-08
800	10	MALM	52	17.0830	6.2871e-10	7.5038e-06
		ℓ-MALM	54	11.5776	4.3061e-10	2.7773e-08
1000	10	MALM	55	21.9272	8.8685e-10	4.4925e-08
		ℓ-MALM	59	17.8789	8.7309e-10	6.2657e-09
1500	10	MALM	64	54.1298	7.0666e-10	7.0586e-06
		ℓ-MALM	63	39.2845	5.0556e-10	2.1187e-09
2000	10	MALM	69	80.1405	6.3821e-10	1.5618e-06
		ℓ-MALM	69	65.1278	5.7268e-10	2.7199e-09
2500	10	MALM	75	127.2009	8.0848e-10	7.4884e-06
		ℓ-MALM	72	93.8748	5.8434e-10	1.7110e-09
3000	10	MALM	75	143.6459	7.7719e-10	1.0123e-07
		ℓ-MALM	75	124.6447	4.6348e-10	1.6521e-09
4000	20	MALM	70	293.9415	8.0333e-10	6.2254e-05
		ℓ-MALM	69	232.3938	5.8045e-10	2.3264e-08
5000	20	MALM	72	449.3883	8.4229e-10	4.8407e-05
		ℓ-MALM	75	360.8994	4.4362e-10	1.2475e-07
8000	25	MALM	76	1.0158e+03	9.3894e-10	2.0435e-06
		ℓ-MALM	78	896.5192	3.5295e-10	6.0870e-08

Table 4.4 Comparison between MALM and ℓ-MALM for $p = 0.3$.

n	rank(M)	Algorithm	IT	time(s)	error 1	error 2
500	10	MALM	43	9.4693	8.8863e-10	4.2793e-06
		ℓ-MALM	42	4.2637	8.6663e-10	1.8090e-07
800	10	MALM	53	20.7760	7.4045e-10	1.7890e-05
		ℓ-MALM	51	12.1585	9.3578e-10	1.7124e-07
1000	10	MALM	57	35.3704	7.5894e-10	2.9163e-05
		ℓ-MALM	57	21.3350	7.4972e-10	2.4146e-05
1500	10	MALM	65	61.1234	7.7663e-10	3.0042e-05
		ℓ-MALM	66	40.9089	4.2967e-10	1.3964e-09
2000	10	MALM	67	73.5681	8.1474e-10	2.2219e-06
		ℓ-MALM	71	67.0528	7.8603e-10	1.2435e-09
2500	10	MALM	72	119.7323	9.3469e-10	3.7511e-05
		ℓ-MALM	75	97.7843	6.1088e-10	5.6465e-09
3000	10	MALM	71	165.0568	9.3024e-10	6.1620e-05
		ℓ-MALM	75	125.0663	5.8542e-10	3.1692e-09
4000	20	MALM	72	360.7658	8.9501e-10	7.9403e-06
		ℓ-MALM	72	280.7485	5.9043e-10	8.1067e-07
5000	20	MALM	72	425.5066	9.4621e-10	2.6905e-06
		ℓ-MALM	72	365.5450	7.6720e-10	3.5167e-05
8000	25	MALM	77	1.5448e+03	8.1698e-10	4.3347e-04
		ℓ-MALM	75	899.7484	9.0788e-10	3.1049e-08

Table 4.5 The values of RATIO.

$p=0.6$	n	500	800	1000	1500	2000	2500	3000	4000	5000	8000
	rank(M)	10	10	10	10	10	10	10	20	20	25
	RATIO (%)	77.15	72.45	81.92	78.47	81.27	80.29	86.32	84.14	87.71	82.01
$p=0.5$	n	500	800	1000	1500	2000	2500	3000	4000	5000	8000
	rank(M)	10	10	10	10	10	10	10	20	20	25
	RATIO (%)	73.10	67.22	74.36	79.64	82.70	80.80	85.40	74.39	90.53	83.07
$p=0.4$	n	500	800	1000	1500	2000	2500	3000	4000	5000	8000
	rank(M)	10	10	10	10	10	10	10	20	20	25
	RATIO (%)	76.33	67.77	81.54	72.57	81.27	73.80	86.77	79.06	80.31	88.26
$p=0.3$	n	500	800	1000	1500	2000	2500	3000	4000	5000	8000
	rank(M)	10	10	10	10	10	10	10	20	20	25
	RATIO (%)	45.03	58.52	60.32	66.93	91.14	81.67	75.77	77.82	85.91	58.24

5. Conclusion

As is known to all, matrix completion is usually to recover a matrix from a subset of the elements of a matrix by taking advantage of low rank structure matrix interdependencies between the entries. It is well-known but NP-hard in general. In recent years, Toeplitz matrix completion has attracted widespread attention and TMC is one of the most important completion problems. In order to solve such problems, we put forward an ℓ-step modified augmented Lagrange multiplier (ℓ-MALM) algorithm based on the MALM algorithm, and corresponding with the theory of the convergence of the ℓ-MALM algorithm are established. Theoretical analysis and numerical results have shown that the ℓ-MALM algorithm is effective for solving TMC problem. The ℓ-MALM algorithm overcomes the original ALM algorithm both singular value decomposition of tardy, and surmounts the property of the extra load of the MALM algorithm. The reason is that data communication congestion is far more expensive than computing. Compared with the CPU of the MALM algorithm, we can see that the CPU of our ℓ-MALM algorithm is reduced to 45%. Therefore, ℓ-MALM algorithm has better convergence rate for solving TMC problem than the MALM algorithm (tables 4.1-4.5).

Acknowledgments The authors gratefully acknowledge the anonymous referees and Professor ZZ Bai (academy of mathematics and systems science, Chinese academy of sciences) for their helpful comments and suggestions which greatly improved the original manuscript of this paper.

References:

[1] AMIT Y, FINK M, SREBRO N et al. Unconvering shared structures in malticalass classification[C]//GHAHRAMANI Z.Proceeding of the 24th international conference on Machine Learning. New York: ACM, 2007: 17-24.

[2] ARGYRIOU A, EVGENIOU T, PONTIL M. Multi-task feature learning[J]. Adv. Neural Inf. Process. Syst., 2007, 19: 41-48.

[3] BENNETT J, LANNING S. The netflix prize[C]//SAN J. Proceedings of KDD Cup and Workshop. California: ACM, 2007,3-6.

[4] BERTALMIO M, SAPIRO G, CASELLES V et al. Multi-task feature learing, image inpainting[J]. Comput. Gr., 2000, 34: 417-424.

[5] CAI Jianfeng, CANDÈS E J, SHEN Zuowei. A singular value thresholding method for matrix completion[J]. SIAM J. Optim., 2010, 20(4): 1956-1982.
[6] CANDÈS E J, RECHT B. Exact matrix completion via convex optimization[J]. Found. Comput. Math., 2009, 9(6): 717-772.
[7] CHEN Minming, GANESH A, LIN Zhouchen et al. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix[J]. J. Mar. Biol. Assoc. UK, 2009, 56(3): 707-722.
[8] HU Yao, ZHANG Debing, LIU Jun et al. Accelerated singular value thresholding for matrix completion[C]. In Proceedings of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing: 2012.
[9] LIN Zhouchen, CHEN Minming, MA Yi. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices[D]. UIUC Technicial Report UIUL-ENG-09-2214, 2010.
[10] LUK F T, QIAO Sanzheng. A fast singular value algorithm for Hankel matrices[J]. Linear Algebra Appl., 2000, 316: 171-182.
[11] MESBAHI M, PAPAVASSILOPOULOS G P. On the rank minimization problem over a positive semidefinite linear matrix inequality[J]. IEEE Trans. Autom. Control, 1997, 42(2): 239-243.
[12] SEBERT F, ZOU Yiming, YING L. Toeplitz block matrices in compressed sensing and their applications in imaging[C]//FILIPOVIC M, ROSIC I. Proceedings of the International Conference on IEEE Information Technology and Applications in Biomedicine, 2008: 47-50.
[13] SHAW A K, POKALA S, KUMARESAN R. Toeplitz and Hankel approximation using structured approach[C]//THOMSON C. Acoustics, Speech and Signal Processing of the IEEE International Conference, 1998, 4: 2349-2352.
[14] TOMASI C, KANADE T. Shape and motion from image streams under orthography: a factorization method[J]. International J. Computer Vision, 1992, 9(2): 137-154.
[15] TOH K C, YUN S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems[J]. Pacific J. Optim., 2010, 6: 615-640.
[16] WANG Chuanlong, Li Chao. A mean value algorithm for Toeplitz matrix completion[J]. Appl. Math. Lett., 2015, 41: 35-40.
[17] WANG Chuanlong, LI Chao, WANG Jin. A modified augmented larange multiplier algorithm for toeplitz matrix completion[J]. Adv. Comput. Math., 2016, 42: 1209-1224.
[18] WANG Chuanlong, LI Chao, WANG Jin. Comparisons of several algorithms for Toeplitz matrix recovery[J]. Comput. Math. Appl., 2016, 71(1): 133-146.
[19] WEN Ruiping, YAN Xihong. A new gradient projection method for matrix completion[J]. Appl. Math. Comput., 2015, 258(1): 537-544.

Toeplitz矩阵填充的ℓ-步修正增广拉格朗日乘子算法

温瑞萍, 李姝贞
(太原师范学院工程科学计算山西省高等学校重点实验室/数学系, 山西 晋中 030619)

摘要: 基于Toeplitz矩阵填充(TMC)的修正增广拉格朗日乘子(MALM)算法, 本文给出此算法的一种加速策略, 提出Toeplitz矩阵填充的ℓ-步修正增广拉格朗日乘子算法. 该方法通过削减原MALM算法中每一步迭代的频繁数据传输, 提高算法的运行效率. 同时也证明了新算法的收敛性. 最后以数值实验表明ℓ-步修正增广拉格朗日乘子算法比原MALM算法更有效.

关键词: Toeplitz矩阵; 矩阵填充; 增广拉格朗日乘子; 数据传输