Numerische Mathematik (2021) 147:727–757
https://doi.org/10.1007/s00211-021-01185-9

Numerische Mathematik

An approximation method of CP rank for third-order tensor completion

Chao Zeng[1] · Tai-Xiang Jiang[2] · Michael K. Ng [1]

Received: 4 October 2019 / Revised: 14 August 2020 / Accepted: 30 January 2021 /
Published online: 19 February 2021

Abstract
We study the problem of third-order tensor completion based on low CP rank recovery. Due to the NP-hardness of the calculation of CP rank, we propose an approximation method by using the sum of ranks of a few matrices as an upper bound of CP rank. We show that such upper bound is between CP rank and the square of CP rank of a tensor. This approximation would be useful when the CP rank is very small. Numerical algorithms are developed and examples are presented to demonstrate that the tensor completion performance by the proposed method is better than that of existing methods.

Mathematics Subject Classification 15A69 · 90C25 · 90C30 · 65K10

1 Introduction

Tensor completion aims to recover a tensor from partial observations under the assumption of low dimensional structure in the underlying data. For the two-dimensional (matrix) case, this is the low rank matrix completion problem and has been well stud-

T.-X. Jiang's research is supported in part by the National Natural Science Foundation of China (12001446) and the Fundamental Research Funds for the Central Universities (JBK2102001). M. Ng's research is supported in part by the HKRGC GRF 12306616, 12200317, 12300218 and 12300519, and HKU 104005583.

✉ Chao Zeng
zengchao@nankai.edu.cn

Tai-Xiang Jiang
taixiangjiang@gmail.com

Michael K. Ng
mng@maths.hku.hk

1 Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong

2 School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, China

ied in the literature. Given a partially observed matrix $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$, the mathematical formulation of low rank matrix completion problem is given by

$$\min_{\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}} \ \operatorname{rank}(\mathbf{X}) \quad \text{s.t.} \ \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}), \tag{1}$$

where Ω is the set of all index pairs (i, j) of observed entries, and $\mathscr{P}_\Omega$ is the orthogonal projector:

$$\mathscr{P}_\Omega(\mathbf{X}) = \begin{cases} x_{ij}, & \text{if } (i, j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

This problem is NP-hard. It has been shown in [10,11,35] that, under certain conditions, low rank solutions of (1) can be recovered by replacing the objective function rank$(\cdot)$ with a suitable convex relaxation – the nuclear norm of matrices:

$$\min_{\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}} \ \|\mathbf{X}\|_* \quad \text{s.t.} \ \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{M}).$$

On the other hand, tensor completion is quite intractable. The main issue lies in how to define a low dimensional structure of a tensor, or how to extend matrix rank to tensor rank. There are two commonly used definitions of tensor rank: (i) the multilinear rank (see Definition 2.1), which is the ranks of unfolding matrices from a tensor; (ii) the CANDECOMP/PARAFAC (CP) rank (see Definition 2.3), which is the smallest number of the sum of rank-one tensors that can generate the original tensor.

Given a partially observed tensor $\mathcal{T} \in V := \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, tensor completion methods based on multilinear rank aim to minimize the ranks of the unfolding matrices:

$$\min_{\mathcal{X} \in V} \sum_{d=1}^{N} \alpha_d \operatorname{rank}(\mathbf{X}_{(d)}) \quad \text{s.t.} \ \mathscr{P}_\Omega(\mathcal{X}) = \mathscr{P}_\Omega(\mathcal{T}),$$

where α_d's are constants satisfying $\alpha_d \geq 0$ and $\sum_{d=1}^{N} \alpha_d = 1$, and $\mathbf{X}_{(d)}$ is the mode-d unfolding matrix (see (5)). Like (1), this is still an NP-hard problem. Therefore, researchers have proposed several computational models [17,18,33]. For example, some models are based on the sum of nuclear norms of unfolding matrices:

$$\min_{\mathcal{X} \in V} \sum_{d=1}^{N} \alpha_d \|\mathbf{X}_{(d)}\|_* \quad \text{s.t.} \ \mathscr{P}_\Omega(\mathcal{X}) = \mathscr{P}_\Omega(\mathcal{T}) \tag{2}$$

or its variants with the alternating direction method of multipliers (ADMM) [6]; and some models are based on the low rank factorization model [42,44]:

$$\min_{\vec{\mathbf{X}},\vec{\mathbf{Y}},\mathcal{Z}\in V} \sum_{d=1}^{N} \alpha_d \left\|\mathbf{X}_d\mathbf{Y}_d - \mathbf{Z}_{(d)}\right\|^2 \quad \text{s.t.} \;\; \mathscr{P}_\Omega(\mathcal{Z}) = \mathscr{P}_\Omega(\mathcal{T}),$$

where $\mathbf{X}_d \in \mathbb{R}^{I_d\times r_d}$, $\mathbf{Y}_d \in \mathbb{R}^{r_d\times\prod_{\ell\neq d} I_\ell}$ with r_d specified, and $\vec{\mathbf{X}} = (\mathbf{X}_1,\ldots,\mathbf{X}_N)$, $\vec{\mathbf{Y}} = (\mathbf{Y}_1,\ldots,\mathbf{Y}_N)$. In addition, Riemannian optimization [30] is employed for low multilinear rank tensor recovery. However, it has been shown in [49] that unfolding matrices may fail to exploit the tensor structure and may lead to poor tensor recovery performance.

Unlike multilinear rank, CP rank does not need to unfold a tensor into matrices. However, there is no straightforward algorithm to determine the CP rank of a specific given tensor. Indeed, this problem is NP-hard [19,20]. For low CP rank tensor completion, the approach can be formulated as follows [1,23]:

$$\min_{\mathcal{X}\in V,\,\mathrm{rank}(\mathcal{X})\leq R} \left\|\mathscr{P}_\Omega(\mathcal{X}-\mathcal{T})\right\|.$$

We note that the above best CP rank-R approximation problem has no solution in general, see [14]. In [48,51], some algorithms are developed for solving such low rank approximation problem and but there is no guarantee for their performance. In [4], the authors prove that there is a polynomial time algorithm based on the sixth level of the sum-of-squares hierarchy for tensor completion. In [49], the tensor nuclear norm is considered for tensor completion, but this calculation is also NP-hard [16,20].

In the literature, there are some methods based on other tensor rank definitions. The works [27,50] adopt the so-called tubal rank based on the tensor singular value decomposition (t-SVD) proposed in [28]. The t-SVD is suitable for tensors with tube structure. In [38], the researchers study Riemannian optimization for high-dimensional tensor completion based on tensor train rank minimization. This method is suitable for tensors with order higher than three.

In practice, the observed data may be contaminated by noise, and there may not exist low CP rank tensors that satisfy the constraint $\mathscr{P}_\Omega(\mathcal{X}) = \mathscr{P}_\Omega(\mathcal{T})$ exactly. The unconstrained version of low CP rank tensor completion is formulated as follows:

$$\min_{\mathcal{X}\in V} \;\mathrm{rank}(\mathcal{X}) + \frac{\lambda}{2}\left\|\mathscr{P}_\Omega(\mathcal{X}-\mathcal{T})\right\|^2. \tag{3}$$

The main aim of this paper is to study third-order tensor completion by using a suitable upper bound of CP rank. We review the rank invariance property of tensors [14,26], and derive a quantity $\mathscr{R}(X)$ which is between the CP rank of a third-order tensor X and its square. Specifically, $\mathscr{R}(X)$ is the sum of ranks of a few matrices.

Using matrix rank to bound or replace tensor rank is a common operation in tensor completion. In multilinear rank based methods [17,18,33], the ranks of unfolding matrices are used to represent the multilinear rank; the work [34] uses a new unfolding strategy to convert a tensor completion problem into a matrix completion problem; in [25], the authors use square unfoldings to bound CP ranks of even order tensors; the works [24,47] construct a rank-1 equivalence property between a tensor and some special unfolding matrices, based on which the tensor optimization problem is converted into a matrix optimization problem. We can observe that, in all these works, the matrices are unfolding matrices of the original tensor. In this work, the matrices are of the same size as a slice, and not an unfolding of the original tensor.

By replacing rank$(\cdot)$ with such upper bound $\mathscr{R}(X)$ in (3), we present and study a new optimization model:

$$\min_{\mathcal{X}\in V} \mathscr{R}(X) + \frac{\lambda}{2}\|\mathscr{P}_{\Omega}(\mathcal{X}-\mathcal{T})\|^2. \tag{4}$$

The error bound of the solution of (4) can be characterized by the CP rank of the underlying tensor. It follows that if the underlying tensor has a low CP rank, then the solution of (4) will be a very good approximation to the underlying tensor. We develop an iterative algorithm to solve (4) and establish its subsequence convergence. The proposed method has been compared with some state-of-the-art algorithms and shows superior performance both on synthetic data and real-world data.

The rest of this paper is organized as follows. In Sect. 2, we review the rank invariance property of tensors and obtain an upper bound of a third-order tensor CP rank. In Sect. 3, we present a new model for tensor completion and propose an iterative algorithm to solve the model. Section 4 is the convergence analysis of the proposed algorithm. Numerical examples are given in Sect. 5 to illustrate the results of the proposed method. Conclusions are presented in Sect. 6.

2 Tensor rank

2.1 Notation

We use bold-face lowercase letters ($\mathbf{a}, \mathbf{b}, \ldots$) for vectors, bold-face capitals ($\mathbf{A}, \mathbf{B}, \ldots$) for matrices, calligraphic letters ($\mathcal{A}, \mathcal{B}, \ldots$) for tensors. The $(i_1, i_2, \ldots, i_N)$-th entry of an Nth-order tensor $\mathcal{A}$ is denoted by $a_{i_1 i_2 \cdots i_N}$. We use $\mathbf{e}_i$ to denote a vector of suitable length, whose ith entry is 1 and all other entries are 0.

The Frobenius norm of a matrix $\mathbf{A}$ is denoted by $\|\mathbf{A}\|$, the nuclear norm of $\mathbf{A}$ is denoted by $\|\mathbf{A}\|_*$, and the Frobenius norm of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is given by

$$\|\mathcal{A}\| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} a_{i_1 i_2\cdots i_N}^2}.$$

A mode-d fiber is a column vector defined by fixing every index but the dth index, e.g., a matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. A slice is a matrix defined by fixing every index but two indices. The mode-d unfolding of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is denoted by $\mathbf{A}_{(d)}$ and arranges the mode-d fibers to be the columns of the resulting matrix. The $(i_1, i_2, \ldots, i_N)$-th entry is mapped to the matrix entry (i_d, j), where

$$j = 1 + \sum_{k=1,k\neq d}^{N} (i_k - 1)J_k \text{ with } J_k = \prod_{m=1,m\neq d}^{k-1} I_m. \tag{5}$$

The d-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{M}$, denoted by $\mathbf{M} \cdot_d \mathcal{A}$, is a tensor generated by multiplying each mode-d fiber of $\mathcal{A}$ by $\mathbf{M}$. Following [14], we write $\mathbf{M}_1 \cdot_1 \mathbf{M}_2 \cdot_2 \cdots \mathbf{M}_N \cdot_N \mathcal{A}$ more concisely as

$$(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_N) \cdot \mathcal{A}.$$

2.2 Upper bounds of third-order tensor CP rank

There are various ways to generalize the rank concept of matrices for tensors. Unlike matrices, these concepts are not consistent with each other.

Definition 2.1 The d-rank of an Nth-order tensor $\mathcal{A}$, denoted by $r_d(\mathcal{A})$, is the dimension of the vector space spanned by all d-mode fibers.

The N-tuple $\left(r_1(\mathcal{A}), r_2(\mathcal{A}), \ldots, r_N(\mathcal{A})\right)$ is called the multilinear rank of $\mathcal{A}$.

Denote

$$F_d(\mathcal{A}) := \text{span}\,\{v : v \text{ is a mode-}d \text{ fiber of } \mathcal{A}\}.$$

Then

$$r_d(\mathcal{A}) = \dim F_d(\mathcal{A}) = \text{rank}\left(\mathbf{A}_{(d)}\right), \quad d = 1, \ldots, N.$$

For a matrix $\mathbf{A}$, $F_1(\mathbf{A})$ is the space spanned by all columns of $\mathbf{A}$, which is just the range of $\mathbf{A}$.

Definition 2.2 An Nth-order tensor $\mathcal{A}$ is rank-1 if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{A} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \cdots \otimes \mathbf{a}^{(N)},$$

where $\otimes$ represents the vector outer product.

Definition 2.3 The CANDECOMP/PARAFAC (CP) rank of an arbitrary tensor $\mathcal{A}$, denoted by rank$(\mathcal{A})$, is the minimal number of rank-1 tensors that yield $\mathcal{A}$ in a linear combination.

Given an Nth-order tensor $\mathcal{A}$, the following relationship [14, (2.16)] is useful in our discussions:

$$r_d(\mathcal{A}) \leq \text{rank}(\mathcal{A}), \quad d = 1, \ldots, N. \tag{6}$$

As already mentioned in the introduction, the calculation of CP rank is NP-hard. Therefore, the low CP rank tensor completion cannot be realized by (3) directly. Our idea is to find a suitable upper bound of rank($\mathcal{X}$) and utilize it as a substitute of rank($\mathcal{X}$) in (3). There exist some upper bounds of the CP rank in the literature. Given $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, a bound given in [26,32] is as follows:

$$\text{rank}(\mathcal{A}) \leq \frac{\prod_{\ell=1}^{N} r_\ell(\mathcal{A})}{r_d(\mathcal{A})}, \quad \forall d \in \{1, 2, \ldots, N\}. \tag{7}$$

This bound is the product of some d-ranks and cannot be utilized directly in the model. We will explore other bounds in the following parts. First, we have the following rank invariance properties.

Lemma 2.4 (see [32, Proposition 3.1.3.1]) *Let* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\mathbf{M}_d \in \mathbb{R}^{I_d \times J_d}$ *for* $d = 1, \ldots, N$ *and* $\mathcal{S} = \left(\mathbf{M}_1^T, \mathbf{M}_2^T, \cdots, \mathbf{M}_N^T\right) \cdot \mathcal{A}$. *Suppose* $F_1(\mathbf{M}_d) \supseteq F_d(\mathcal{A})$ *for* $d = 1, \ldots, N$. *Then,*

$$rank(\mathcal{S}) = rank(\mathcal{A}), \quad r_d(\mathcal{S}) = r_d(\mathcal{A}), \quad d = 1, \ldots, N.$$

The third-order tensor CP rank can be bounded by the sum of slice matrix ranks. Noting that $\mathcal{A}(i, :, :) = \mathbf{e}_i^T \cdot_1 \mathcal{A}$, $\mathcal{A}(:, j, :) = \mathbf{e}_j^T \cdot_2 \mathcal{A}$ and $\mathcal{A}(:, :, k) = \mathbf{e}_k^T \cdot_3 \mathcal{A}$, we have the following lemma.

Lemma 2.5 (see [31]) *Let* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. *Then,*

$$rank(\mathcal{A}) \leq \min_{1 \leq d \leq 3} \sum_{i=1}^{I_d} rank\left(\mathbf{e}_i^T \cdot_d \mathcal{A}\right). \tag{8}$$

Equation (8) yields an upper bound for the CP rank. However, this bound is too rough: We need to sum the ranks of all slice matrices of the same type. A refined bound is given in the following corollary.

Corollary 2.6 *Let* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ *and* $\mathbf{M}_d \in \mathbb{R}^{I_d \times J_d}$ *for* $d = 1, 2, 3$. *Suppose* $F_1(\mathbf{M}_d) \supseteq F_d(\mathcal{A})$ *for* $d = 1, 2, 3$. *Then,*

$$rank(\mathcal{A}) \leq \min_{1 \leq d \leq 3} \sum_{i=1}^{J_d} rank\left((\mathbf{M}_d \mathbf{e}_i)^T \cdot_d \mathcal{A}\right). \tag{9}$$

Proof By noticing that $\mathbf{M}_3^T \cdot_3 \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$, combining Lemma 2.4 and(8) leads to

$$\text{rank}(\mathcal{A}) = \text{rank}(\mathbf{M}_3^T \cdot_3 \mathcal{A}) \leq \sum_{k=1}^{J_3} \text{rank}\left((\mathbf{M}_3 \mathbf{e}_i)^T \cdot_3 \mathcal{A}\right).$$

The other two inequalities can be shown similarly. □

Remark 2.7 Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $\mathbf{M}_d \in \mathbb{R}^{I_d \times r_d(\mathcal{A})}$ for $d = 1, 2, 3$ and $\mathcal{S} = \left(\mathbf{M}_1^T, \mathbf{M}_2^T, \mathbf{M}_3^T\right) \cdot \mathcal{A}$. Suppose $F_1(\mathbf{M}_d) = F_d(\mathcal{A})$ for $d = 1, 2, 3$. Then by Lemma 2.4 and (8),

$$\operatorname{rank}(\mathcal{A}) \leq \min_{1 \leq d \leq 3} \sum_{i=1}^{r_d(\mathcal{A})} \operatorname{rank}\left(\mathbf{e}_i^T \cdot_d \mathcal{S}\right).$$

This bound is with respect to the Tucker core tensor $\mathcal{S}$. Note that $\mathcal{S} \in \mathbb{R}^{r_1(\mathcal{A}) \times r_2(\mathcal{A}) \times r_3(\mathcal{A})}$ and by Lemma 2.4, $r_d(\mathcal{S}) = r_d(\mathcal{A})$. That is, $\mathcal{S}$ has full multilinear rank. Hence, although $\mathcal{S}$ has a smaller size than $\mathbf{M}_d^T \cdot_d \mathcal{A}$, $d = 1, 2, 3$ in (9), it is not suitable for low rank completion.

Given $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, define

$$B_d(\mathcal{A}) := \left\{\mathbf{M} \in \mathbb{R}^{I_d \times r_d(\mathcal{A})} : F_1(\mathbf{M}) = F_d(\mathcal{A})\right\},$$

i.e., the columns of $\mathbf{M}$ form a basis of $F_d(\mathcal{A})$; and define

$$B(\mathcal{A}) = \left\{\vec{\mathbf{M}} = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3) : \mathbf{M}_d \in B_d(\mathcal{A}), d = 1, 2, 3\right\}.$$

For any $\vec{\mathbf{A}} \in B(\mathcal{A})$, denote $\Gamma_d(\mathcal{A}, \vec{\mathbf{A}}) = \sum_{i=1}^{r_d(\mathcal{A})} \operatorname{rank}((\mathbf{A}_d \mathbf{e}_i)^T \cdot_d \mathcal{A})$. By (9), we have

$$\operatorname{rank}(\mathcal{A}) \leq \min_{1 \leq d \leq 3} \Gamma_d(\mathcal{A}, \vec{\mathbf{A}}), \quad \forall \vec{\mathbf{A}} \in B(\mathcal{A}). \tag{10}$$

We will evaluate how good the upper bound of (10) is. First, we show that the rank of every slice matrix is bounded by some d-rank.

Lemma 2.8 *Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Then*

$$rank\left(\mathbf{e}_i^T \cdot_d \mathcal{A}\right) \leq \min_{\ell \neq d} r_\ell(\mathcal{A}), \quad 1 \leq d \leq 3, 1 \leq i \leq I_d.$$

Proof We prove the case $d = 3$. The other cases can be proved similarly.

Since the columns of $\mathbf{e}_i^T \cdot_3 \mathcal{A}$ are mode-1 fibers and the rows of $\mathbf{e}_i^T \cdot_3 \mathcal{A}$ are mode-2 fibers, $\mathbf{e}_i^T \cdot_3 \mathcal{A}$ is a submatrix of $\mathbf{A}_{(1)}$ and $\left(\mathbf{e}_i^T \cdot_3 \mathcal{A}\right)^T$ is a submatrix of $\mathbf{A}_{(2)}$. It follows that $\operatorname{rank}(\mathbf{e}_i^T \cdot_3 \mathcal{A}) \leq \operatorname{rank}\left(\mathbf{A}_{(1)}\right) = r_1(\mathcal{A})$ and $\operatorname{rank}(\mathbf{e}_i^T \cdot_3 \mathcal{A}) = \operatorname{rank}\left(\left(\mathbf{e}_i^T \cdot_3 \mathcal{A}\right)^T\right) \leq \operatorname{rank}\left(\mathbf{A}_{(2)}\right) = r_2(\mathcal{A})$. $\square$

Combining Lemma 2.4 and Lemma 2.8 gives

$$\operatorname{rank}((\mathbf{A}_d \mathbf{e}_i)^T \cdot_d \mathcal{A}) \leq \min_{\ell \neq d} r_\ell(\mathbf{A}_d^T \cdot_d \mathcal{A}) = \min_{\ell \neq d} r_\ell(\mathcal{A}), \quad 1 \leq d \leq 3, \tag{11}$$

and then

$$\Gamma_d(\mathcal{A}, \vec{\mathbf{A}}) \leq r_d(\mathcal{A}) \min_{\ell \neq d} r_\ell(\mathcal{A}), \quad 1 \leq d \leq 3. \tag{12}$$

Hence, the right side of (10) is less than or equal to $\min\limits_{1\leq d\leq 3} \frac{\prod_{\ell=1}^{3} r_\ell(\mathcal{A})}{r_d(\mathcal{A})}$. That is, (10) is a tighter bound than (7). Actually, the advantage of (10) can be comprehended by Lemma 2.8. A slice matrix is a submatrix of the unfolding matrix. Hence, using ranks of slice matrixes can give a more accurate estimate of the CP rank. Applying (6) on (12) yields the following result:

$$\Gamma_d(\mathcal{A}, \vec{\mathbf{A}}) \leq \operatorname{rank}(\mathcal{A})^2, \quad d = 1, 2, 3. \tag{13}$$

Example 2.9 Consider a $2 \times 2 \times 2$ tensor $\mathcal{A}$ with

$$\mathcal{A}(:, :, 1) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{A}(:, :, 2) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

It can be checked that $r_1(\mathcal{A}) = r_2(\mathcal{A}) = r_3(\mathcal{A}) = 2$ and $\operatorname{rank}(\mathcal{A}) = 2$. Since $\mathcal{A}$ is symmetric,the situations for the three modes are the same. We can only consider the case $d = 3$:

$$\Gamma_3(\mathcal{A}, \vec{\mathbf{A}}) = \operatorname{rank}(\mathbf{C}_1) + \operatorname{rank}(\mathbf{C}_2),$$

where $\mathbf{C}_k = (\mathbf{A}_3\mathbf{e}_k)^T \cdot_3 \mathcal{A}$, $k = 1, 2$. Since $r_3(\mathcal{A}) = 2$, $B_3(\mathcal{A})$ consists of all 2×2 invertible matrices.

If we choose $\mathbf{A}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, then $\mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $\mathbf{C}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, and $\Gamma_3(\mathcal{A}, \vec{\mathbf{A}}) = 2$. If we choose $\mathbf{A}_3 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, then $\mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{C}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, and $\Gamma_3(\mathcal{A}, \vec{\mathbf{A}}) = 3$. If we choose $\mathbf{A}_3 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, then $\mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{C}_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, and $\Gamma_3(\mathcal{A}, \vec{\mathbf{A}}) = 4$.

These results illustrate that (10) is a tighter bound than (7) and (13) is a tight bound for $\Gamma_d(\mathcal{A}, \vec{\mathbf{A}})$.

We want to convert (10) into one expression. For any $\vec{\mathbf{A}} \in B(\mathcal{A})$, define the following function:

$$\mathscr{R}(\mathcal{A}, \vec{\mathbf{A}}) = \sum_{d=1}^{3} \alpha_d \Gamma_d(\mathcal{A}, \vec{\mathbf{A}}),$$

where α_d's are constants satisfying $\alpha_d \geq 0$ and $\sum_{d=1}^{3} \alpha_d = 1$. Combining (10) and (13) gives that

$$\operatorname{rank}(\mathcal{A}) \leq \mathscr{R}(\mathcal{A}, \vec{\mathbf{A}}) \leq \operatorname{rank}(\mathcal{A})^2, \quad \forall \vec{\mathbf{A}} \in B(\mathcal{A}). \tag{14}$$

3 Models and algorithm

We denote by $V := \mathbb{R}^{I_1 \times I_2 \times I_3}$ throughout the remaining part of this paper.

By (14), we can relax (3) as

$$\min_{\mathcal{X}\in V} \mathscr{G}(\mathcal{X}) := \min_{\vec{\mathbf{X}}\in B(\mathcal{X})} \mathscr{R}(\mathcal{X}, \vec{\mathbf{X}}) + \frac{\lambda}{2}\|\mathscr{P}_\Omega(\mathcal{X}-\mathcal{T})\|^2. \tag{15}$$

Denote $\mathscr{G}_1(\mathcal{X}) := \mathrm{rank}(\mathcal{X}) + \frac{\lambda}{2}\|\mathscr{P}_\Omega(\mathcal{X}-\mathcal{T})\|^2$ and $\mathscr{G}_2(\mathcal{X}) := \mathrm{rank}(\mathcal{X})^2 + \frac{\lambda}{2}\|\mathscr{P}_\Omega(\mathcal{X}-\mathcal{T})\|^2$. Also by (14), we have

$$\mathscr{G}_1(\mathcal{X}) \le \mathscr{G}(\mathcal{X}) \le \mathscr{G}_2(\mathcal{X}). \tag{16}$$

Therefore, (15) is a convex combination of the following two models

$$\min_{\mathcal{X}\in V} \mathscr{G}_1(\mathcal{X}), \tag{17}$$

$$\min_{\mathcal{X}\in V} \mathscr{G}_2(\mathcal{X}). \tag{18}$$

Model (17) and (18) are the unconstrained versions of the following two models, respectively,

$$\min_{\mathcal{X}\in V} \mathrm{rank}(\mathcal{X}) \quad \text{s.t. } \mathscr{P}_\Omega(\mathcal{X}) = \mathscr{P}_\Omega(\mathcal{T}), \tag{19}$$

$$\min_{\mathcal{X}\in V} \mathrm{rank}(\mathcal{X})^2 \quad \text{s.t. } \mathscr{P}_\Omega(\mathcal{X}) = \mathscr{P}_\Omega(\mathcal{T}). \tag{20}$$

By noting that (19) is equivalent to (20), both of (17) and (18) are models for low CP rank tensor completion.

Lemma 3.1 *Suppose the sets* arg *min* $\mathscr{G}_1(\mathcal{X})$, arg *min* $\mathscr{G}_2(\mathcal{X})$, arg *min* $\mathscr{G}(\mathcal{X})$ *are nonempty. Denote* $g_1 = \min_{\mathcal{X}} \mathscr{G}_1(\mathcal{X})$. *Then,*

$$0 \le \mathscr{G}_1(\mathcal{X}^*) - g_1 \le \min_{\mathcal{X}\in \arg\min \mathscr{G}_1(\mathcal{X})} rank(\mathcal{X})^2 - rank(\mathcal{X}), \quad \forall \mathcal{X}^* \in \arg\min \mathscr{G}(\mathcal{X}).$$

Proof The inequality $0 \le \mathscr{G}_1(\mathcal{X}^*) - g_1$ follows directly from $g_1 = \min_{\mathcal{X}} \mathscr{G}_1(\mathcal{X})$. For the other inequality, we prove the following relationship:

$$\mathscr{G}_1(\mathcal{X}^*) - g_1 \le g_2 - g_1, \quad \forall \mathcal{X}^* \in \arg\min \mathscr{G}(\mathcal{X}), \tag{21}$$

where $g_2 = \min_{\mathcal{X}} \mathscr{G}_2(\mathcal{X})$. We only need to prove that

$$\mathscr{G}_1(\mathcal{X}^*) \le g_2, \quad \forall \mathcal{X}^* \in \arg\min \mathscr{G}(\mathcal{X}).$$

If $\mathscr{G}_1(\mathcal{X}^*) > g_2$, for any $\mathcal{Y} \in \arg\min \mathscr{G}_2(\mathcal{X})$, we have

$$\mathscr{G}(\mathcal{Y}) \overset{(16)}{\le} \mathscr{G}_2(\mathcal{Y}) = g_2 < \mathscr{G}_1(\mathcal{X}^*) \overset{(16)}{\le} \mathscr{G}(\mathcal{X}^*),$$

which contradicts to $\mathcal{X}^* \in \arg\min \mathcal{G}(\mathcal{X})$.

In addition, since $g_2 \leq \mathcal{G}_2(\mathcal{X})$ for all $\mathcal{X} \in \arg\min \mathcal{G}_1(\mathcal{X})$, we have

$$g_2 - g_1 \leq \mathcal{G}_2(\mathcal{X}) - \mathcal{G}_1(\mathcal{X}), \quad \forall \mathcal{X} \in \arg\min \mathcal{G}_1(\mathcal{X}).$$

It follows that

$$g_2 - g_1 \leq \min_{\mathcal{X} \in \arg\min \mathcal{G}_1(\mathcal{X})} \mathcal{G}_2(\mathcal{X}) - \mathcal{G}_1(\mathcal{X}) = \min_{\mathcal{X} \in \arg\min \mathcal{G}_1(\mathcal{X})} \mathrm{rank}(\mathcal{X})^2 - \mathrm{rank}(\mathcal{X}).$$

Combining the above inequality with (21) completes the proof. □

This lemma gives a bound of the approximation error between (15) and (17). When $\mathrm{rank}(\mathcal{X})$ is small for $\mathcal{X} \in \arg\min \mathcal{G}_1(\mathcal{X})$, (15) approximates (17) well. This is just the target of low CP rank completion.

For $\mathcal{G}(\mathcal{X})$, note that $(\mathbf{X}_1\mathbf{e}_i)^T \cdot_1 \mathcal{X} \in \mathbb{R}^{I_2 \times I_3}$ and $\mathrm{rank}\left((\mathbf{X}_1\mathbf{e}_i)^T \cdot_1 \mathcal{X}\right) \leq \min_{d \neq 1} r_d(\mathcal{X})$. The matrix $(\mathbf{X}_1\mathbf{e}_i)^T \cdot_1 \mathcal{X}$ has low rank. Hence, $\min_{\mathbf{X}_1} \mathrm{rank}\left((\mathbf{X}_1\mathbf{e}_i)^T \cdot_1 \mathcal{X}\right)$ can be relaxed as $\min_{\mathbf{X}_1} \| (\mathbf{X}_1\mathbf{e}_i)^T \cdot_1 \mathcal{X}\|_*$, and (15) can be relaxed as

$$\min_{\mathcal{X} \in V} \min_{\vec{\mathbf{X}} \in B(\mathcal{X})} \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{r_d(\mathcal{X})} \left\|(\mathbf{X}_d\mathbf{e}_i)^T \cdot_d \mathcal{X}\right\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{T})\|^2. \tag{22}$$

3.1 The algorithm for low CP rank completion

Problem (22) is difficult to solve, because and $r_d(\mathcal{X})$, $B(\mathcal{X})$ depend on $\mathcal{X}$. Our idea is to use a not too bad estimate $\mathcal{X}^0$ to obtain a not too bad estimate of $F_d(\mathcal{X})$ and $r_d(\mathcal{X})$, avoiding the dependence of $r_d(\mathcal{X})$, $B(\mathcal{X})$ on $\mathcal{X}$. Define the following function:

$$\mathcal{F}(\mathcal{X}, \vec{\mathbf{X}}) = \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^0} \left\|(\mathbf{X}_d\mathbf{e}_i)^T \cdot_d \mathcal{X}\right\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{T})\|^2,$$

where $R_d^0 = r_d(\mathcal{X}^0)$. Then (22) is converted into the following model

$$\min_{\mathcal{X} \in W(\mathcal{X}^0)} \min_{\vec{\mathbf{X}} \in B(\mathcal{X}^0)} \mathcal{F}(\mathcal{X}, \vec{\mathbf{X}}), \tag{23}$$

where

$$W(\mathcal{A}) := \{\mathcal{Z} \in V : F_d(\mathcal{Z}) \subseteq F_d(\mathcal{A}), d = 1, 2, 3\}.$$

The model (23) is still difficult to solve. One point is, if $\mathbf{X}_d$ does not have a good structure, then $\min_{\mathcal{X}} \mathcal{F}(\mathcal{X}, \vec{\mathbf{X}})$ is difficult to solve. In Sect. 3.2, we will see that the requirement that $\mathbf{X}_d$ satisfies $\mathbf{X}_d^T\mathbf{X}_d = \mathbf{I}$ is crucial for the calculation of $\min_{\mathcal{X}} \mathscr{F}(\mathcal{X}, \vec{\mathbf{X}})$. Therefore, we only consider some special $\mathbf{X}_d$.

Given $\mathcal{A} \in V$, define

$$Q_d(\mathcal{A}) := \left\{\mathbf{M} \in \mathbb{R}^{I_d \times r_d(\mathcal{A})} : \mathbf{M}^T\mathbf{M} = \mathbf{I} \text{ and } F_1(\mathbf{M}) = F_d(\mathcal{A})\right\},$$

i.e., the columns of $\mathbf{M}$ form an orthonormal basis of $F_d(\mathcal{A})$; and define

$$Q(\mathcal{A}) = \left\{\vec{\mathbf{M}} = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3) : \mathbf{M}_d \in Q_d(\mathcal{A}), d = 1, 2, 3\right\}.$$

An ideal strategy is to consider the following model:

$$\min_{\mathcal{X}\in W(\mathcal{X}^0)} \min_{\vec{\mathbf{X}}\in Q(\mathcal{X}^0)} \mathscr{F}(\mathcal{X}, \vec{\mathbf{X}}). \tag{24}$$

A natural thought is to follow the framework of the alternating minimization, i.e., two-block coordinate descent method [43], to solve this problem. That is, starting with some given initial point, we generate a sequence $\{(\mathcal{X}^{n+1}, \vec{\mathbf{X}}^n)\}_{n\in\mathbb{N}}$ via the scheme

$$\vec{\mathbf{X}}^n \in \arg \min_{\vec{\mathbf{X}}\in Q(\mathcal{X}^0)} \mathscr{F}(\mathcal{X}^n, \vec{\mathbf{X}}) \tag{25}$$

$$\mathcal{X}^{n+1} \in \arg \min_{\mathcal{X}\in W(\mathcal{X}^0)} \mathscr{F}(\mathcal{X}, \vec{\mathbf{X}}^n). \tag{26}$$

The problem (26) is convex and not very difficult to solve. For (25), because the computations of the three entries of $\vec{\mathbf{X}}^n$ are mutually independent and similar to each other, we only need to consider the computation of one entry. We write it as a formal optimization problem as follows:

$$\min_{\mathbf{M}\in Q_1(\mathcal{A})} \mathscr{T}(\mathbf{M}) := \sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\|_*, \tag{27}$$

where $\mathcal{A} \in V$ is a given tensor. This is an optimization problem with generalized orthogonality constraints. See [15]. This type of problem is difficult because the constraints are not only non-convex but numerically expensive to preserve during iterations [41]. The case is even worse for (27), because the computation of the subgradient $\frac{\partial \mathscr{T}}{\partial \mathbf{M}}$[1] is rather expensive.

Therefore, we consider finding a closed form quasi-minimizer as a compromise. For a matrix $\mathbf{A} \in \mathbb{R}^{I\times J}$, suppose $\text{rank}(\mathbf{A}) = r$. Let $\mathbf{A} = \mathbf{U}6\mathbf{V}^T$ be the compact singular

[1] Denote $\mathbf{Y} = \sum_{\ell=1}^{I_1} m_{\ell j}\mathcal{A}(\ell, :, :)$. Let the SVD of $\mathbf{Y}$ be $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. The chain rule gives

$$\frac{\partial \mathscr{T}}{\partial \mathbf{M}}(i, j) = \left\{\text{tr}\left((\mathbf{U}\mathbf{V}^T + \mathbf{W})^T\mathcal{A}(i, :, :)\right) : \mathbf{W} \in \mathbb{R}^{I_2\times I_3}, \mathbf{U}^T\mathbf{W} = 0, \mathbf{W}\mathbf{V} = 0, \|\mathbf{W}\|_2 \le 1\right\},$$

where $\|\mathbf{W}\|_2$ is the spectral norm of $\mathbf{W}$ and $\text{tr}(\cdot)$ is the trace of a matrix.

value decomposition (SVD) of $\mathbf{A}$, where $\mathbf{U} \in \mathbb{R}^{I\times r}$, $\mathbf{6} \in \mathbb{R}^{r\times r}$, $\mathbf{V} \in \mathbb{R}^{J\times r}$. We call $\mathbf{U}$ the *compact left singular matrix* of $\mathbf{A}$. This matrix has the following property.

Proposition 3.2 *Let $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$, and $\mathbf{U}_1 \in \mathbb{R}^{I_1\times r_1(\mathcal{A})}$ be the compact left singular matrix of $\mathbf{A}_{(1)}$. Then for any $\mathbf{M} \in Q_1(\mathcal{A})$, one has*

$$\|\mathbf{A}_{(1)}\|_* = \sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{U}_1\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\| \le \sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\|. \tag{28}$$

Proof First, we prove the following claim: For any $\mathbf{C} \in \mathbb{R}^{I\times J}$, one has

$$\|\mathbf{C}\|_* \le \sum_{i=1}^{I} \|\mathbf{C}(i,:)\|. \tag{29}$$

Since the dual norm of the nuclear norm is the spectral norm, it follows that

$$\|\mathbf{C}\|_* = \max_{\|\mathbf{Y}\|_2\le 1} \mathrm{tr}(\mathbf{Y}^T\mathbf{C}),$$

where $\|\mathbf{Y}\|_2$ is the spectral norm of $\mathbf{Y}$ and $\mathrm{tr}(\cdot)$ is the trace of a matrix. Therefore, there exists a matrix $\mathbf{B}$ with $\|\mathbf{B}\|_2 \le 1$ such that

$$\begin{aligned}\|\mathbf{C}\|_* &= \|\mathbf{C}^T\|_* = \mathrm{tr}(\mathbf{B}^T\mathbf{C}^T) = \mathrm{tr}(\mathbf{C}\mathbf{B}) = \sum_{i=1}^{I} (\mathbf{C}\mathbf{B})_{ii} \\ &\le \sum_{i=1}^{I} \|(\mathbf{C}\mathbf{B})\,(i,:)\| = \sum_{i=1}^{I} \|(\mathbf{C}(i,:))\,\mathbf{B}\| \le \sum_{i=1}^{I} \|\mathbf{B}\|_2\,\|\mathbf{C}(i,:)\| \le \sum_{i=1}^{I} \|\mathbf{C}(i,:)\|,\end{aligned}$$

which proves the claim (29).

By the definition of unfolding matrix, we can verify that

$$\left\|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\| = \left\|\left(\mathbf{M}^T\mathbf{A}_{(1)}\right)(i,:)\right\|, \quad \forall i = 1,\ldots,r_1(\mathcal{A}).$$

Let the nonzero singular values of $\mathbf{A}_{(1)}$ be $\sigma_1,\ldots,\sigma_{r_1(\mathcal{A})}$. Then

$$\sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{U}_1\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\| = \sum_{i=1}^{r_1(\mathcal{A})} \left\|\left(\mathbf{U}_1^T\mathbf{A}_{(1)}\right)(i,:)\right\| = \sum_{i=1}^{r_1(\mathcal{A})} \sigma_i = \|\mathbf{A}_{(1)}\|_*. \tag{30}$$

For any $\mathbf{M} \in Q_1(\mathcal{A})$, there exists an orthogonal matrix $\hat{\mathbf{M}} \in \mathbb{R}^{I_1\times I_1}$ such that $\hat{\mathbf{M}}(:,1:r_1(\mathcal{A})) = \mathbf{M}$. Then $(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A} = 0$ for $i = r_1(\mathcal{A})+1,\ldots,I_3$ and

$$\sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\| = \sum_{i=1}^{I_1} \left\|\left(\hat{\mathbf{M}}\mathbf{e}_i\right)^T \cdot_1 \mathcal{A}\right\|$$

$$= \sum_{i=1}^{I_1} \left\| \left(\hat{\mathbf{M}}^T \mathbf{A}_{(1)} \right) (i,:) \right\| \overset{(29)}{\geq} \left\| \hat{\mathbf{M}}^T \mathbf{A}_{(1)} \right\|_* = \|\mathbf{A}_{(1)}\|_*.$$

Combining the above inequality with (30) gives (28). □

Denote by $\mathbf{M}_i := (\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}, i = 1, \ldots, r_1(\mathcal{A})$ and $\mathbf{v}(\mathbf{M}) := \left(\|\mathbf{M}_1\|, \ldots, \|\mathbf{M}_{r_1(\mathcal{A})}\| \right)$. Then $\sum_{i=1}^{r_1(\mathcal{A})} \|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\| = \|\mathbf{v}(\mathbf{M})\|_{\ell_1}$, where $\|\cdot\|_{\ell_1}$ is the ℓ_1 norm. Inequality (28) implies that

$$\|\mathbf{v}(\mathbf{U}_1)\|_{\ell_1} \leq \|\mathbf{v}(\mathbf{M})\|_{\ell_1}, \quad \forall \mathbf{M} \in Q_1(\mathcal{A}). \tag{31}$$

Note that

$$\|\mathbf{v}(\mathbf{M})\| = \|\mathcal{A}\|, \quad \forall \mathbf{M} \in Q_1(\mathcal{A}). \tag{32}$$

The mapping $\mathcal{A} \mapsto \mathbf{M}^T \cdot_1 \mathcal{A}$ can be regarded as a rearrangement of the energy of $\mathcal{A}$: A smaller ℓ_1 norm means the energy of $\mathbf{v}(\mathbf{U}_1)$ is more concentrated in general.

Corollary 3.3 *Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times r_1(\mathcal{A})}$ be the compact left singular matrix of $\mathbf{A}_{(1)}$. Denote by $R = min\{r_2(\mathcal{A}), r_3(\mathcal{A})\}$. Then for any $\mathbf{M} \in Q_1(\mathcal{A})$, one has*

$$\sum_{i=1}^{r_1(\mathcal{A})} \left\| (\mathbf{U}_1\mathbf{e}_i)^T \cdot_1 \mathcal{A} \right\|_* \leq \sqrt{R} \sum_{i=1}^{r_1(\mathcal{A})} \left\| (\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A} \right\|_*.$$

Proof Denote by $\mathbf{u}(\mathbf{M}) := \left(\|\mathbf{M}_1\|_*, \ldots, \|\mathbf{M}_{r_1(\mathcal{A})}\|_* \right)$. Then $\sum_{i=1}^{r_1(\mathcal{A})} \left\|(\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A}\right\|_* = \|\mathbf{u}(\mathbf{M})\|_{\ell_1}$. By (11) and the fact that $\|\mathbf{C}\| \leq \|\mathbf{C}\|_* \leq \sqrt{\mathrm{rank}(\mathbf{C})}\|\mathbf{C}\|$ for any matrix $\mathbf{C}$, we have

$$\|\mathbf{u}(\mathbf{M})\|_{\ell_1} \leq \sqrt{R}\|\mathbf{v}(\mathbf{M})\|_{\ell_1} \leq \sqrt{R}\|\mathbf{u}(\mathbf{M})\|_{\ell_1}. \tag{33}$$

It follows that, for all $\mathbf{M} \in Q_1(\mathcal{A})$,

$$\|\mathbf{u}(\mathbf{U}_1)\|_{\ell_1} \overset{(33)}{\leq} \sqrt{R}\|\mathbf{v}(\mathbf{U}_1)\|_{\ell_1} \overset{(31)}{\leq} \sqrt{R}\|\mathbf{v}(\mathbf{M})\|_{\ell_1} \overset{(33)}{\leq} \sqrt{R}\|\mathbf{u}(\mathbf{M})\|_{\ell_1}.$$

□

Corollary 3.4 *Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Denote by $R = min\{r_2(\mathcal{A}), r_3(\mathcal{A})\}$. Then for any $\mathbf{M}, \mathbf{G} \in Q_1(\mathcal{A})$, one has*

$$\sum_{i=1}^{r_1(\mathcal{A})} \left\| (\mathbf{M}\mathbf{e}_i)^T \cdot_1 \mathcal{A} \right\|_* \leq \sqrt{Rr_1(\mathcal{A})} \sum_{i=1}^{r_1(\mathcal{A})} \left\| (\mathbf{G}\mathbf{e}_i)^T \cdot_1 \mathcal{A} \right\|_*.$$

Proof Noting that $\|\mathbf{x}\| \leq \|\mathbf{x}\|_{\ell_1} \leq \sqrt{I}\|\mathbf{x}\|$ for any $\mathbf{x} \in \mathbb{R}^I$, we have

$$\|\mathbf{u}(\mathbf{M})\|_{\ell_1} \overset{(33)}{\leq} \sqrt{R}\|\mathbf{v}(\mathbf{M})\|_{\ell_1} \leq \sqrt{Rr_1(\mathcal{A})}\|\mathbf{v}(\mathbf{M})\| \overset{(32)}{=} \sqrt{Rr_1(\mathcal{A})}\|\mathcal{A}\|.$$

On the other hand,

$$\|\mathbf{u}(\mathbf{G})\|_{\ell_1} \overset{(33)}{\geq} \|\mathbf{v}(\mathbf{G})\|_{\ell_1} \geq \|\mathbf{v}(\mathbf{G})\| = \|\mathcal{A}\|.$$

Combining the above two inequalities yields the result. □

Corollary 3.3 implies that $\mathbf{U}_1$ is a quasi-minimizer of (27). We propose the following **a**pproximation method for low **CP** rank **c**ompletion (ACPC) of third-order tensors.

ACPC of third-order tensors

1. Input $\alpha_1, \alpha_2, \alpha_3, \lambda$ and initial value $\mathcal{X}^0$, $R_d^0 = r_d(\mathcal{X}^0)$, a compact left singular matrix $\mathbf{U}_d^0$ of $\mathbf{X}_{(d)}^0$, where $d = 1, 2, 3$.
2. For each $n = 0, 1, \ldots$, compute $\mathcal{X}^{n+1}$ by solving

$$\min_{\mathcal{X} \in W^n} \mathcal{E}^n(\mathcal{X}) := \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^n} \left\| \left(\mathbf{U}_d^n \mathbf{e}_i\right)^T \cdot_d \mathcal{X} \right\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{T})\|^2, \quad (34)$$

where $W^n = W(\mathcal{X}^n)$, $R_d^n = r_d(\mathcal{X}^n)$, and $\mathbf{U}_d^n$ is a compact left singular matrix of $\mathbf{X}_{(d)}^n$.

The initial value $\mathcal{X}^0$ is crucial for ACPC. The criterion for finding $\mathcal{X}^0$ is that the algorithm is fast and the output is not so bad. The low multilinear rank tensor completion (2) solved by HaLRTC [33] satisfies such criterion. We will use the initialization by HaLRTC in our experiments. The parameter α_d's can be set based on the estimate of the multilinear rank of the original tensor. A simple strategy is to set $\alpha_d = \frac{R_d^0}{R_1^0 + R_2^0 + R_3^0}$.

3.2 Implementation of ACPC

Since (34) is a convex optimization problem, we can solve it by the ADMM.

For each $\mathcal{X} \in W^n$, since $F_3(\mathcal{X}) \subseteq F_3(\mathcal{X}^n) = F_1(\mathbf{U}_3^n)$, there exists $\mathcal{S}_3 \in \mathbb{R}^{I_1 \times I_2 \times R_3^n}$ such that $\mathcal{X} = \mathbf{U}_3^n \cdot_3 \mathcal{S}_3$. Then,

$$\mathbf{U}_3^{n\,T} \cdot_3 \mathcal{X} = \left(\mathbf{U}_3^{n\,T}\mathbf{U}_3^n\right) \cdot_3 \mathcal{S}_3 = \mathcal{S}_3. \quad (35)$$

Therefore, by introducing an auxiliary variable $\vec{\mathcal{S}} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$, where $\mathcal{S}_1 \in \mathbb{R}^{R_1^n \times I_2 \times I_3}$, $\mathcal{S}_2 \in \mathbb{R}^{I_1 \times R_2^n \times I_3}$ and $\mathcal{S}_3 \in \mathbb{R}^{I_1 \times I_2 \times R_3^n}$, (34) is reformulated as

$$\begin{aligned} \min_{\vec{\mathcal{S}}, \mathcal{X} \in V} \quad & \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^n} \left\| \mathbf{e}_i^T \cdot_d \mathcal{S}_d \right\|_* + \frac{\lambda}{2} \| \mathscr{P}_\Omega(\mathcal{X} - \mathcal{T}) \|^2 \\ \text{s.t.} \quad & \mathcal{X} = \mathbf{U}_d^n \cdot_d \mathcal{S}_d, \quad d = 1, 2, 3. \end{aligned}$$

The augmented Lagrangian function for the above problem is defined as

$$\begin{aligned} \mathscr{L}(\vec{\mathcal{S}}, \mathcal{X}, \vec{\mathcal{Y}}) = & \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^n} \left\| \mathbf{e}_i^T \cdot_d \mathcal{S}_d \right\|_* + \sum_{d=1}^{3} \langle \mathcal{Y}_d, \mathcal{X} - \mathbf{U}_d^n \cdot_d \mathcal{S}_d \rangle \\ & + \frac{\beta_d}{2} \| \mathcal{X} - \mathbf{U}_d^n \cdot_d \mathcal{S}_d \|^2 + \frac{\lambda}{2} \| \mathscr{P}_\Omega(\mathcal{X} - \mathcal{T}) \|^2, \end{aligned}$$

where $\vec{\mathcal{Y}} = (\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3)$ is the Lagrange multiplier. The iterative scheme of ADMM for (34) reads

$$\begin{cases} \vec{\mathcal{S}}^{t+1} \in \arg\min_{\vec{\mathcal{S}}} \mathscr{L}(\vec{\mathcal{S}}, \mathcal{X}^t, \vec{\mathcal{Y}}^t); \ \mathcal{X}^{t+1} \in \arg\min_{\mathcal{X} \in V} \mathscr{L}(\vec{\mathcal{S}}^{t+1}, \mathcal{X}, \vec{\mathcal{Y}}^t); & (36) \\ \mathcal{Y}_d^{t+1} = \mathcal{Y}_d^t + \delta \beta_d (\mathcal{X}^{t+1} - \mathbf{U}_d^n \cdot_d \mathcal{S}_d^{t+1}), & (37) \end{cases}$$

where $\delta > 0$ is the step length. The two subproblems in the above algorithm are calculated as follows.

1. The $\vec{\mathcal{S}}$-subproblem (36): For $d = 1$, we can simplify (36) as

$$\mathcal{S}_1^{t+1} \in \arg\min_{\mathcal{S}_1} \alpha_1 \sum_{i=1}^{R_1^n} \| \mathbf{e}_i^T \cdot_1 \mathcal{S}_1 \|_* + \frac{\beta_1}{2} \left\| \mathbf{U}_1^n \cdot_1 \mathcal{S}_1 - \left(\mathcal{X}^t + \frac{1}{\beta_1} \mathcal{Y}_1^t \right) \right\|^2. \quad (38)$$

Let $\hat{\mathcal{S}}_1 \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ satisfy $\hat{\mathcal{S}}_1(1 : R_1^n, I_2, I_3) = \mathcal{S}_1$ and $\hat{s}_{ijk} = 0, \ \forall i > R_1^n$; and $\hat{\mathbf{U}}_1 \in \mathbb{R}^{I_1 \times I_1}$ be orthogonal and $\hat{\mathbf{U}}_1(:, 1 : R_1^n) = \mathbf{U}_1^n$. Then (38) is equivalent to

$$\begin{aligned} \mathcal{S}_1^{t+1} & \in \arg\min_{\mathcal{S}_1} \alpha_1 \sum_{i=1}^{R_1^n} \| \mathbf{e}_i^T \cdot_1 \mathcal{S}_1 \|_* + \frac{\beta_1}{2} \left\| \hat{\mathbf{U}}_1 \cdot_1 \hat{\mathcal{S}}_1 - \left(\mathcal{X}^t + \frac{1}{\beta_1} \mathcal{Y}_1^t \right) \right\|^2 \\ & = \arg\min_{\mathcal{S}_1} \alpha_1 \sum_{i=1}^{R_1^n} \| \mathbf{e}_i^T \cdot_1 \mathcal{S}_1 \|_* + \frac{\beta_1}{2} \left\| \hat{\mathcal{S}}_1 - \hat{\mathbf{U}}_1^T \cdot_1 \left(\mathcal{X}^t + \frac{1}{\beta_1} \mathcal{Y}_1^t \right) \right\|^2. \end{aligned}$$

Note that

$$\left\| \hat{\mathcal{S}}_1 - \hat{\mathbf{U}}_1^T \cdot_1 \left(\mathcal{X}^t + \frac{1}{\beta_1} \mathcal{Y}_1^t \right) \right\|^2 = \left\| \mathcal{S}_1 - \mathbf{U}_1^{nT} \cdot_1 \left(\mathcal{X}^t + \frac{1}{\beta_1} \mathcal{Y}_1^t \right) \right\|^2$$

$$+ \left\| \mathcal{X}^t + \frac{1}{\beta_1}\mathcal{Y}_1^t \right\|^2 - \left\| \mathbf{U}_1^{nT} \cdot_1 \left(\mathcal{X}^t + \frac{1}{\beta_1}\mathcal{Y}_1^t \right) \right\|^2.$$

Let $\mathcal{Z}_1^t = \mathbf{U}_1^{nT} \cdot_1 \left(\mathcal{X}^t + \frac{1}{\beta_1}\mathcal{Y}_1^t \right)$. Then, for $i = 1, \ldots, R_1^n$,

$$\mathcal{S}_1^{t+1}(i, :, :) \in \arg\min_{\mathcal{S}_1} \left\{ \alpha_1 \|\mathbf{e}_i^T \cdot_1 \mathcal{S}_1\|_* + \frac{\beta_1}{2} \left\| \mathbf{e}_i^T \cdot_1 \mathcal{S}_1 - \mathbf{e}_i^T \cdot_1 \mathcal{Z}_1^t \right\|^2 \right\}.$$

Given $\mathbf{M} \in \mathbb{R}^{I \times J}$ and $\gamma > 0$, let $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{M}$. We define the following *shrinkage* operator

$$\mathscr{D}_\gamma(\mathbf{M}) := \mathbf{U}(\boldsymbol{\Sigma} - \gamma \mathbf{I})_+ \mathbf{V}^T,$$

where $(a)_+ := \max\{a, 0\}$. Then, it follows from [9, Theorem 2.1] that

$$\mathcal{S}_1^{t+1}(i, :, :) = \mathscr{D}_{\alpha_1/\beta_1}\left(\mathbf{e}_i^T \cdot_1 \mathcal{Z}_1^t \right), \quad i = 1, \ldots, R_1^n.$$

Similarly, we can obtain the formulae of $\mathcal{S}_2^{t+1}$ and $\mathcal{S}_3^{t+1}$. We list a unified result here:

$$\mathbf{e}_i^T \cdot_d \mathcal{S}_d^{t+1} = \mathscr{D}_{\alpha_d/\beta_d}\left(\mathbf{e}_i^T \cdot_d \mathcal{Z}_d^t \right), \quad d = 1, 2, 3, i = 1, \ldots, R_d^n,$$

where $\mathcal{Z}_d^t = \mathbf{U}_d^{nT} \cdot_d \left(\mathcal{X}^t + \frac{1}{\beta_d}\mathcal{Y}_d^t \right)$.

2. The $\mathcal{X}$-subproblem (37): We can simplify (37) as

$$\mathcal{X}^{t+1} \in \arg\min_{\mathcal{X} \in V} \left\{ \sum_{d=1}^{3} \left(\langle \mathcal{Y}_d^t, \mathcal{X} \rangle + \frac{\beta_d}{2} \left\| \mathcal{X} - \mathbf{U}_d^n \cdot_d \mathcal{S}_d^{t+1} \right\|^2 \right) + \frac{\lambda}{2} \|\mathscr{P}_\Omega(\mathcal{X} - \mathcal{T})\|^2 \right\}.$$

This is a least squares problem. By noting $\mathscr{P}_\Omega^* = \mathscr{P}_\Omega$, where $\mathscr{P}_\Omega^*$ is the adjoint of $\mathscr{P}_\Omega$ in V, and $\mathscr{P}_\Omega^2 = \mathscr{P}_\Omega$, the corresponding normal equation of the above problem is

$$\left(\frac{\lambda}{\beta} \mathscr{P}_\Omega + \mathscr{I} \right) \mathcal{X} = \frac{\lambda}{\beta} \mathscr{P}_\Omega(\mathcal{T}) + \frac{1}{\beta} \sum_{d=1}^{3} \beta_d \mathbf{U}_d^n \cdot_d \mathcal{S}_d^{t+1} - \frac{1}{\beta} \sum_{d=1}^{3} \mathcal{Y}_d^t,$$

where $\beta = \beta_1 + \beta_2 + \beta_3$. By [46, (2.2)], we have $(\frac{\lambda}{\beta}\mathscr{P}_\Omega + \mathscr{I})^{-1} = \mathscr{I} - \frac{\lambda}{\lambda+\beta}\mathscr{P}_\Omega$. Then

$$\mathcal{X}^{t+1} = \frac{\lambda}{\lambda + \beta} \mathscr{P}_\Omega(\mathcal{T}) + \frac{1}{\beta} \left(\mathscr{I} - \frac{\lambda}{\lambda + \beta} \mathscr{P}_\Omega \right) \left(\sum_{d=1}^{3} \beta_d \mathbf{U}_d^n \cdot_d \mathcal{S}_d^{t+1} - \sum_{d=1}^{3} \mathcal{Y}_d^t \right).$$

4 Convergence analysis of ACPC

The coordinate descent method for nonconvex problems has attracted a lot of attention in recent years. One of the best results on global convergence is about proximal alternating minimization for the so-called Kurdyka-Łojasiewicz (KL) functions. See [2,5,45]. The coordinate descent method has also been used in tensor computations, e.g., the alternating least squares (ALS) for computing CP decompositions [29] and the alternating linear schemes for computing tensor train decompositions [21]. The local convergence properties of these two algorithms are established via the contraction principle in [36,39], respectively.

Since $\vec{\mathbf{U}}^n := (\mathbf{U}_1^n, \mathbf{U}_2^n, \mathbf{U}_3^n)$ in ACPC is only a quasi-minimizer of the function $\vec{\mathbf{X}} \mapsto \mathscr{F}(\mathcal{X}^n, \vec{\mathbf{X}})$ and it is difficult to estimate the distance between $\vec{\mathbf{U}}^n$ and the set of the critical points of the function $\vec{\mathbf{X}} \mapsto \mathscr{F}(\mathcal{X}^n, \vec{\mathbf{X}})$, we cannot obtain the global or local convergence of ACPC. However, based on the properties of the feasible domain, we can establish its subsequence convergence.

By definition, we have $\mathcal{X}^n \in W^n$ and any $\mathcal{A} \in W^n$ satisfies

$$\mathrm{rank}_d(\mathcal{A}) \le R_d^n, \quad d = 1, 2, 3. \tag{39}$$

It can be verified that W^n is a linear space and $\dim W^n \ge 1$ is equivalent to $R_d^n \ge 1$ for $d = 1, 2, 3$. If $\dim W^n = 0$, then $\mathcal{X}^m = 0$ for $m \ge n$ and the convergence holds trivially. Therefore, we suppose $\dim W^n \ge 1$ in the following discussion.

Lemma 4.1 *The function $\mathscr{E}^n(\mathcal{X})$ in (34) is coercive on W^n. Hence, the solution of (34) always exists.*

Proof By (35), for each $\mathcal{X} \in W^n$, there exists $\mathcal{S}_3 \in \mathbb{R}^{I_1 \times I_2 \times R_3^n}$ such that $\mathcal{X} = \mathbf{U}_3^n \cdot_3 \mathcal{S}_3$, $\|\mathcal{X}\| = \|\mathcal{S}_3\|$ and $\mathbf{U}_3^{nT} \cdot_3 \mathcal{X} = \mathcal{S}_3$. Therefore, the function $\mathcal{X} \mapsto \sum_{i=1}^{R_3^n} \left\| \left(\mathbf{U}_3^n \mathbf{e}_i\right)^T \cdot_3 \mathcal{X} \right\|_* = \sum_{i=1}^{R_3^n} \left\| \mathbf{e}_i^T \cdot_3 \mathcal{S}_3 \right\|_*$ is coercive and then $\mathscr{E}^n(\mathcal{X})$ is coercive on W^n. □

4.1 Basic convergence properties

Lemma 4.2 *For $d = 1, 2, 3$, the sequence $\{R_d^n\}$ satisfies $R_d^n \ge R_d^{n+1}$, and $\{R_d^n\}$ will converge within finite steps.*

Proof It follows from the algorithm that $\mathcal{X}^{n+1} \in W^n$. By (39), we have $R_d^{n+1} = \mathrm{rank}_d(\mathcal{X}^{n+1}) \le R_d^n$. Since $R_d^n \in \{1, \ldots, I_d\}$, it follows that $\{R_d^n\}$ will converge within finite steps. □

Lemma 4.3 *The subspace sequence $\{F_d(\mathcal{X}^n)\}$ satisfies $F_d(\mathcal{X}^n) \supseteq F_d(\mathcal{X}^{n+1})$ and $\{F_d(\mathcal{X}^n)\}$ will converge within finite steps in $\mathbb{R}^{I_d}$, where $d = 1, 2, 3$.*

Proof It follows from $\mathcal{X}^{n+1} \in W^n$ and the definition of W^n that $F_d(\mathcal{X}^{n+1}) \subseteq F_1(\mathbf{U}_d^n) = F_d(\mathcal{X}^n)$. Since $\dim F_d(\mathcal{X}^n) = R_d^n$, $\{F_d(\mathcal{X}^n)\}$ will converge within finite steps in $\mathbb{R}^{I_d}$. □

The following corollary is immediate from the above lemma.

Corollary 4.4 *The space sequence* $\{W^n\}$ *satisfies* $W^n \supseteq W^{n+1}$ *and* $\{W^n\}$ *will converge within finite steps in* V.

For the coordinate descent method, the decreasing of the objective function is a basic property. For ACPC, we can only obtain the following result on the objective functions.

Lemma 4.5 *Denote by* $R^n = \max\{R_1^n, R_2^n, R_3^n\}$. *Then*

$$\mathscr{E}^n(\mathcal{X}^{n+1}) \leq \sqrt{R^n}\mathscr{E}^{n-1}(\mathcal{X}^n).$$

Proof By the algorithm and Corollary 3.3, we have $\mathscr{E}^n(\mathcal{X}^{n+1}) \leq \mathscr{E}^n(\mathcal{X}^n) \leq \sqrt{R^n}\mathscr{E}^{n-1}(\mathcal{X}^n)$. □

4.2 Convergence behavior of $\{(\mathcal{X}^{n+1}, \vec{\mathbf{U}}^n)\}$

By Lemmas 4.2, 4.3 and Corollary 4.4, there exists $M > 0$ such that

$$R_d^n = R_d^M,\ \ F_d(\mathcal{X}^n) = F_d(\mathcal{X}^M) \text{ and } W^n = W^M, \quad \forall n \geq M, d = 1, 2, 3.$$

Denote $\bar{R}_d = R_d^M$, $\bar{F}_d = F_d(\mathcal{X}^M)$ and $\bar{W} = W^M$. Define the following function:

$$\overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}) = \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{\bar{R}_d} \left\| (\mathbf{U}_d \mathbf{e}_i)^T \cdot_d \mathcal{X} \right\|_* + \frac{\lambda}{2} \|\mathscr{P}_\Omega(\mathcal{X} - \mathcal{T})\|^2,$$

and the following set

$$\bar{Q} = \left\{\vec{\mathbf{Y}} = (\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3) : \mathbf{Y}_d^T \mathbf{Y}_d = \mathbf{I}, F_1(\mathbf{Y}_d) = \bar{F}_d, d = 1, 2, 3\right\}.$$

By Lemma 4.1,

$$\mathscr{H}(\vec{\mathbf{U}}) := \min_{\mathcal{X} \in \bar{W}} \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}})$$

is well defined on $\bar{Q}$. Then, when $n \geq M$,

$$\overline{\mathscr{F}}(\mathcal{X}^{n+1}, \vec{\mathbf{U}}^n) = \mathscr{E}^n(\mathcal{X}^{n+1}) = \min_{\mathcal{X} \in \bar{W}} \mathscr{E}^n(\mathcal{X}) = \min_{\mathcal{X} \in \bar{W}} \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}^n) = \mathscr{H}(\vec{\mathbf{U}}^n). \quad (40)$$

First, we prove some properties of $\bar{Q}$ and $\mathscr{H}(\vec{\mathbf{U}})$.

Lemma 4.6 *The set* $\bar{Q}$ *is compact.*

Proof Define

$$\bar{Q}_d = \left\{ \mathbf{Y} \in \mathbb{R}^{I_d \times \bar{R}_d} : \mathbf{Y}^T\mathbf{Y} = \mathbf{I}, F_1(\mathbf{Y}) = \bar{F}_d \right\}.$$

Then $\bar{Q}$ is the Cartesian product of $\bar{Q}_1, \bar{Q}_2, \bar{Q}_3$. We only need to prove that $\bar{Q}_d$ is compact.

Let $\bar{\mathbf{Y}}$ be a fixed element of $\bar{Q}_d$. By the properties of orthonormal basis, $\mathbf{Y} \in \bar{Q}_d$ if and only if there exists an orthogonal matrix $\mathbf{N} \in \mathbb{R}^{\bar{R}_d \times \bar{R}_d}$, such that $\mathbf{Y} = \bar{\mathbf{Y}}\mathbf{N}$. Define

$$\hat{Q}_d = \left\{ \mathbf{N} \in \mathbb{R}^{\bar{R}_d \times \bar{R}_d} : \mathbf{N}^T\mathbf{N} = \mathbf{I} \right\}.$$

That is, $\hat{Q}_d$ is the orthogonal group and compact [22, p. 85]. Then, $\bar{Q}_d$ is the image of $\hat{Q}_d$ under the linear transformation: $\mathbf{N} \mapsto \bar{\mathbf{Y}}\mathbf{N}$. Therefore, the compactness of $\bar{Q}_d$ follows from the compactness of $\hat{Q}_d$. $\square$

Lemma 4.7 *The function $\mathscr{H}(\vec{\mathbf{U}})$ is continuous on $\bar{Q}$.*

Proof Since $\overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}})$ is defined by the sum of some norms, it is continuous. For a given $\vec{\mathbf{M}}^0 \in \bar{Q}$ and every $\varepsilon > 0$, there exists a $\delta > 0$, such that when $\vec{\mathbf{M}}^1 \in \bar{Q}$ satisfies $\|\vec{\mathbf{M}}^1 - \vec{\mathbf{M}}^0\|^2 = \|\mathbf{M}_1^1 - \mathbf{M}_1^0\|^2 + \|\mathbf{M}_2^1 - \mathbf{M}_2^0\|^2 + \|\mathbf{M}_3^1 - \mathbf{M}_3^0\|^2 < \delta$, we have

$$\left| \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{M}}^0) - \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{M}}^1) \right| < \varepsilon, \quad \forall \mathcal{X} \in V. \tag{41}$$

Suppose $\mathscr{H}(\vec{\mathbf{M}}^0) = \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^0)$, $\mathscr{H}(\vec{\mathbf{M}}^1) = \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^1, \vec{\mathbf{M}}^1)$, where $\widetilde{\mathcal{X}}^0, \widetilde{\mathcal{X}}^1 \in \bar{W}$. It follows that

$$\begin{aligned} &\overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^0) \leq \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^1, \vec{\mathbf{M}}^0) \overset{(41)}{<} \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^1, \vec{\mathbf{M}}^1) \\ &+\varepsilon \leq \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^1) + \varepsilon \overset{(41)}{<} \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^0) + 2\varepsilon. \end{aligned}$$

Therefore,

$$-\varepsilon < \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^1, \vec{\mathbf{M}}^1) - \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^0) < \varepsilon,$$

which is

$$|\mathscr{H}(\vec{\mathbf{M}}^1) - \mathscr{H}(\vec{\mathbf{M}}^0)| = \left| \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^1, \vec{\mathbf{M}}^1) - \overline{\mathscr{F}}(\widetilde{\mathcal{X}}^0, \vec{\mathbf{M}}^0)) \right| < \varepsilon$$

and completes the proof. $\square$

The following corollary follows from Lemma 4.6.

Corollary 4.8 *The sequence $\{\vec{\mathbf{U}}^n\}$ is bounded, and there exists $M > 0$ such that when $n \geq M$, the columns of $\mathbf{U}_d^n$ form an orthonormal basis of $\bar{F}_d$, $d = 1, 2, 3$.*

Theorem 4.9 *The sequence $\{\mathcal{X}^n\}$ is bounded and has at least one convergent subsequence with a limit $\mathcal{X}^*$ satisfying that there exists $\vec{\mathbf{U}}^* \in \bar{Q}$ such that*

$$\overline{\mathscr{F}}(\mathcal{X}^*, \vec{\mathbf{U}}^*) = \min_{\mathcal{X}\in\bar{W}} \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}^*).$$

Proof First, we prove that $\{\mathcal{X}^n\}$ has at least one convergent subsequence.

By Corollary 4.8, there exists a convergent subsequence $\{\vec{\mathbf{U}}^{n_k}\}$ of $\{\vec{\mathbf{U}}^n\}$:

$$\lim_{n_k\to\infty} \vec{\mathbf{U}}^{n_k} = \vec{\mathbf{U}}^*, \tag{42}$$

where $\vec{\mathbf{U}}^* \in \bar{Q}$. It follows from (40) and Lemma 4.7 that

$$\lim_{n_k\to\infty} \overline{\mathscr{F}}(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^{n_k}) = \lim_{n_k\to\infty} \mathscr{H}(\vec{\mathbf{U}}^{n_k}) = \mathscr{H}(\vec{\mathbf{U}}^*).$$

Combining the continuity of $\overline{\mathscr{F}}$ and (42) yields

$$\lim_{n_k\to\infty} \overline{\mathscr{F}}(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^*) = \mathscr{H}(\vec{\mathbf{U}}^*). \tag{43}$$

Hence, the sequence $\{\overline{\mathscr{F}}(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^*)\}$ is convergent and bounded. By Lemma 4.1, the function $\mathcal{X} \mapsto \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}^*)$ is coercive on $\bar{W}$. Therefore, $\{\mathcal{X}^{n_k+1}\}$ is bounded and there exists a convergent subsequence $\{\mathcal{X}^{n_l}\}$: $\lim_{n_l\to\infty} \mathcal{X}^{n_l} = \mathcal{X}^*$ such that

$$\overline{\mathscr{F}}(\mathcal{X}^*, \vec{\mathbf{U}}^*) = \lim_{n_l\to\infty} \overline{\mathscr{F}}(\mathcal{X}^{n_l}, \vec{\mathbf{U}}^*) \overset{(43)}{=} \mathscr{H}(\vec{\mathbf{U}}^*) = \min_{\mathcal{X}\in\bar{W}} \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}^*).$$

Now we prove $\{\mathcal{X}^n\}$ is bounded. If $\{\mathcal{X}^n\}$ is not bounded, then there exists a subsequence $\{\mathcal{X}^{n_i}\}$ satisfying $\lim_{n_i\to\infty} \|\mathcal{X}^{n_i}\| = \infty$. Consider the sequence $\{\overline{\mathscr{F}}(\mathcal{X}^{n_i}, \vec{\mathbf{U}}^{n_i-1})\}$. We have $\overline{\mathscr{F}}(\mathcal{X}^{n_i}, \vec{\mathbf{U}}^{n_i-1}) = \mathscr{H}(\vec{\mathbf{U}}^{n_i-1})$. Like the discussion above, there exists a subsequence $\{\mathcal{X}^{n_j}\}$ of $\{\mathcal{X}^{n_i}\}$, such that

$$\lim_{n_j\to\infty} \overline{\mathscr{F}}(\mathcal{X}^{n_j}, \vec{\mathbf{U}}') = \mathscr{H}(\vec{\mathbf{U}}'), \tag{44}$$

where $\vec{\mathbf{U}}'$ is a limit point of $\{\vec{\mathbf{U}}^{n_i-1}\}$. Since $\lim_{n_j\to\infty} \|\mathcal{X}^{n_j}\| = \infty$, (44) contradicts the coercivity of the function $\mathcal{X} \mapsto \overline{\mathscr{F}}(\mathcal{X}, \vec{\mathbf{U}}')$ on $\bar{W}$, which completes the proof. $\square$

Now we explore the relationship between the limit points of the sequence $\{(\mathcal{X}^{n+1}, \vec{\mathbf{U}}^n)\}$ and the solution of the original problem (24). If $R_d^n < R_d^0$ for some $n \geq 1$ and $d \in \{1, 2, 3\}$, then $\bar{Q} \subsetneq Q(\mathcal{X}^0)$ and $\bar{W} \subsetneq W(\mathcal{X}^0)$, entailing difficulties in constructing the relationship. We assume that $R_d^n = R_d^0$ holds for all $n \geq 1$ and $d = 1, 2, 3$. Then (40) becomes

$$\mathscr{F}(\mathcal{X}^{n+1}, \vec{\mathbf{U}}^n) = \mathscr{E}^n(\mathcal{X}^{n+1}) = \min_{\mathcal{X}\in W(\mathcal{X}^0)} \mathscr{E}^n(\mathcal{X}) = \min_{\mathcal{X}\in W(\mathcal{X}^0)} \mathscr{F}(\mathcal{X}, \vec{\mathbf{U}}^n).$$

Theorem 4.10 *Suppose that $R_d^n = R_d^0$ for all $n \geq 1$ and $d = 1, 2, 3$. Denote by $\rho = \max_{1 \leq k \neq \ell \leq 3} R_k^0 R_\ell^0$. Let $(\underline{\mathcal{X}}, \vec{\underline{\mathbf{X}}})$ be any minimizer of (24). Then any limit point $(\mathcal{X}^*, \vec{\mathbf{U}}^*)$ of $\{(\mathcal{X}^{n+1}, \vec{\mathbf{U}}^n)\}$ satisfies*

$$\mathscr{F}(\mathcal{X}^*, \vec{\mathbf{U}}^*) - \mathscr{F}(\underline{\mathcal{X}}, \underline{\vec{\mathbf{U}}}) \leq (\sqrt{\rho} - 1) \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^0} \left\| \left(\underline{\mathbf{X}}_d \mathbf{e}_i\right)^T \cdot_d \underline{\mathcal{X}} \right\|_* .$$

Proof Assume that there is a subsequence $\{(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^{n_k})\}$ such that $(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^{n_k}) \to (\mathcal{X}^*, \vec{\mathbf{U}}^*)$ as $k \to \infty$. Then

$$\mathscr{F}(\mathcal{X}^{n_k+1}, \vec{\mathbf{U}}^{n_k}) \leq \mathscr{F}(\mathcal{X}, \vec{\mathbf{U}}^{n_k}), \quad \forall \mathcal{X} \in W(\mathcal{X}^0).$$

Letting $k \to \infty$, we obtain

$$\mathscr{F}(\mathcal{X}^*, \vec{\mathbf{U}}^*) \leq \mathscr{F}(\mathcal{X}, \vec{\mathbf{U}}^*), \quad \forall \mathcal{X} \in W(\mathcal{X}^0).$$

Substituting $\mathcal{X}$ by $\underline{\mathcal{X}}$ in the above inequality, we obtain

$$\begin{aligned}
&\mathscr{F}(\mathcal{X}^*, \vec{\mathbf{U}}^*) - \mathscr{F}(\underline{\mathcal{X}}, \underline{\vec{\mathbf{U}}}) \leq \mathscr{F}(\underline{\mathcal{X}}, \vec{\mathbf{U}}^*) - \mathscr{F}(\underline{\mathcal{X}}, \vec{\underline{\mathbf{X}}}) \\
=&\sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^0} \left(\left\| \left(\mathbf{U}_d^* \mathbf{e}_i\right)^T \cdot_d \underline{\mathcal{X}} \right\|_* - \left\| \left(\underline{\mathbf{X}}_d \mathbf{e}_i\right)^T \cdot_d \underline{\mathcal{X}} \right\|_* \right) \\
\leq&(\sqrt{\rho} - 1) \sum_{d=1}^{3} \alpha_d \sum_{i=1}^{R_d^0} \left\| \left(\underline{\mathbf{X}}_d \mathbf{e}_i\right)^T \cdot_d \underline{\mathcal{X}} \right\|_* ,
\end{aligned}$$

where the last inequality follows from Corollary 3.4. □

5 Numerical experiments

We present some numerical results to show the performance of ACPC. All experiments are performed on MATLAB R2016a with Tensor Toolbox, version 3.0 [3] on a laptop (Intel Core i5-6300HQ CPU at 2.30GHz, 8.00G RAM).

Given a tensor with size $I_1 \times I_2 \times I_3$, the sample ratio (SR) is defined as

$$\mathrm{SR} := \frac{|\Omega|}{I_1 I_2 I_3},$$

where $|\Omega|$ is the number of entries of Ω. The test tensors include synthetic and real-world data. We use the relative error (RErr) to evaluate the results:

$$\mathrm{RErr} := \frac{\|\mathcal{X} - \mathcal{X}^*\|}{\|\mathcal{X}^*\|},$$

where $\mathcal{X}^*$ is the original tensor and $\mathcal{X}$ is the recovered result.

We use the initialization by HaLRTC for all experiments. We set $\alpha_d = \frac{R_d^0}{R_1^0+R_2^0+R_3^0}$ and $\lambda = 1000$ for all experiments. The inner iterations of ACPC, i.e., the iterations of ADMM presented in Sect. 3.2 for solving (34), are terminated whenever

$$\frac{\|\mathcal{X}^{n,t+1} - \mathcal{X}^{n,t}\|}{\max(\|\mathcal{X}^{n,t}\|, 1)} \leq tol,$$

where $\mathcal{X}^{n,t+1}$ is the solution of (37), or the iteration number reaches 500. The outer iterations of ACPC are terminated whenever

$$\frac{\|\mathcal{X}^{n+1} - \mathcal{X}^{n}\|}{\max(\|\mathcal{X}^{n}\|, 1)} \leq Tol,$$

or the iteration number reaches N. The tolerances tol, Tol and the maximum outer iteration number N will be specified below.

We will test the convergence behaviour of ACPC in Sect. 5.1, and compare ACPC with HaLRTC [33], TMac [44], FBCP [51] and TNN [50] on synthetic data in Sect. 5.2 and real-world data in Sect. 5.3, where HaLRTC and TMac are based on multilinear rank, FBCP is based on CP rank and TNN is based on tubal rank.

The synthetic tensors are generated based on the Tucker decomposition [13] and the CP decomposition [29]. To generate a Tucker decomposition based tensor $\mathcal{A} \in \mathbb{R}^{I\times I\times I}$, we first generate one tensor $\mathcal{C}$ and three matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ by the following MATLAB script

$$\mathcal{C} = \texttt{randn}(r, r, r), \quad \mathbf{A}_d = \texttt{randn}(I, r),\ d = 1, 2, 3,$$

where r is a fixed integer. Then $\mathcal{A}$ with multilinear rank (r, r, r) is given by $\mathcal{A} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)\cdot\mathcal{C}$. To generate a CP decomposition based tensor $\mathcal{B} \in \mathbb{R}^{I\times I\times I}$, we first generate three matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ by the following MATLAB script

$$\mathbf{B}_d = \texttt{randn}(I, r),\ d = 1, 2, 3,$$

where r is a fixed integer. Then $\mathcal{B}$ with $\text{rank}(\mathcal{B}) = r$ is given by

$$\mathcal{B} = \sum_{\ell=1}^{r} \lambda_\ell\, \mathbf{B}_1(:, \ell) \otimes \mathbf{B}_2(:, \ell) \otimes \mathbf{B}_3(:, \ell), \tag{45}$$

where λ_ℓ is a fixed weight.

5.1 Convergence behaviour of ACPC

We test two synthetic tensors: a Tucker decomposition based tensor $\mathcal{A} \in \mathbb{R}^{100\times100\times100}$ with multilinear rank $(10, 10, 10)$; and a CP decomposition based tensor $\mathcal{B} \in \mathbb{R}^{100\times100\times100}$ with $\text{rank}(\mathcal{B}) = 10$ and $\lambda_\ell = 1$ for $\ell = 1, \ldots, 10$ in (45).

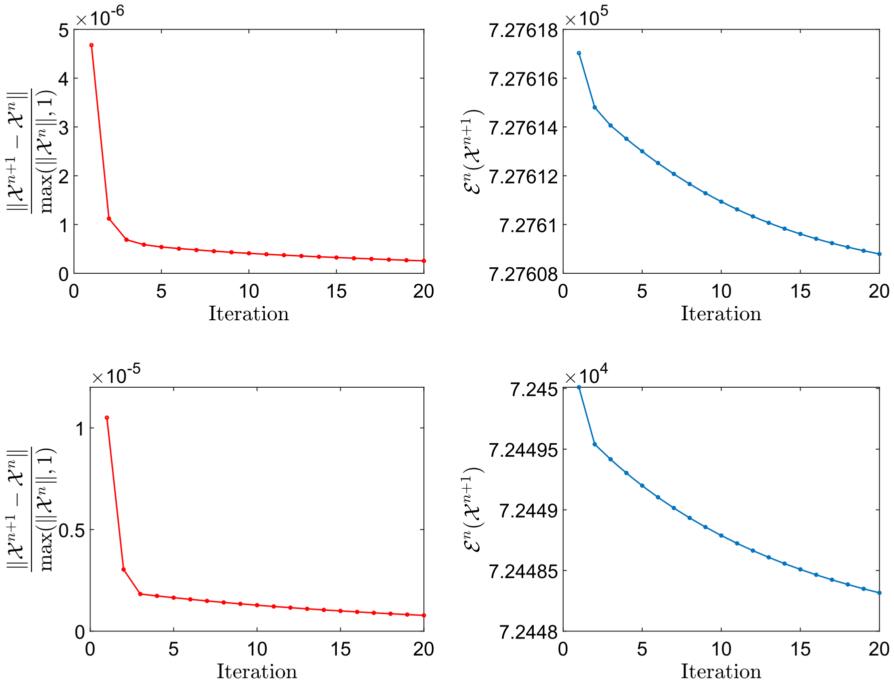

Fig. 1 The values of $\frac{\|\mathcal{X}^{n+1}-\mathcal{X}^n\|}{\max(\|\mathcal{X}^n\|,1)}$ and $\mathscr{E}^n(\mathcal{X}^{n+1})$ versus the outer iteration number in ACPC. The first row is for tensor $\mathcal{A}$, and the second row is for $\mathcal{B}$

The inner tolerance tol is set as 10^{-6}. Because we want to show the convergence behaviour of ACPC, the outer iterations are terminated only when the iteration number reaches the maximum outer iteration number, which is set as 20, and Tol is ignored. Figure 1 shows $\frac{\|\mathcal{X}^{n+1}-\mathcal{X}^n\|}{\max(\|\mathcal{X}^n\|,1)}$ and $\mathscr{E}^n(\mathcal{X}^{n+1})$ versus the outer iteration number in ACPC. From the curves, we can observe that ACPC has a good performance in convergence. The sequence of $\{\mathscr{E}^n(\mathcal{X}^{n+1})\}$ is monotonically decreasing for these two tensors.

5.2 Synthetic data

We compare ACPC with HaLRTC, TMac, FBCP and TNN on four types of synthetic tensors: Tucker decomposition based tensors with size $50\times50\times50$ and multilinear rank $(5, 5, 5)$; Tucker decomposition based tensors with size $100\times100\times100$ and multilinear rank $(10, 10, 10)$; CP decomposition based tensors with size $50 \times 50 \times 50$ and with $r = 5, \lambda_\ell = \ell^2$ in (45); CP decomposition based tensors with size $100 \times 100 \times 100$ and with $r = 10, \lambda_\ell = \ell^2$ in (45).

Remark 5.1 Most methods for fitting CP decompositions, including the ALS, perform not very well on tensors with power-law increasing weights in the form (45). This appears to be related to the condition numbers of factor matrices versus rank-1 tensors, as discussed in [8,40][7, Sect. 2]. Also in [7], the authors propose a Riemannian optimization method to tackle this issue. Because FBCP is a method generalized from the ALS, it has an overwhelming advantage over other methods if the original tensor can be fitted very well by the ALS. Hence we consider CP decomposition based tensors with power-law increasing weights like [44].

We set the inner tolerance $tol = 10^{-6}$, the outer tolerance $Tol = 10^{-5}$ and the maximum outer iteration number $N = 5$. The tolerances of all the other methods are set as 10^{-6}. We generate ten tensors for each type of synthetic tensors and report the average results. Table 1 shows the RErrs and CPU time (measured in seconds) on synthetic tensors without noise and Table 2 shows the results on synthetic tensors with Gaussian noise of zero mean and standard deviation 10^{-3}. Here, the CPU time of ACPC includes the time for generating the initial value $\mathcal{X}^0$.

For RErr, we can observe that ACPC has the best performance on all cases. HaLRTC has a stable performance on all cases and the results are not so bad. TMac performs well on the Tucker decomposition based tensors, while performs badly on the CP decomposition based tensors. FBCP performs not so well on some Tucker decomposition based tensors and is sensitive to noise. TNN performs badly when the sample ratio is low.

For CPU time, HaLRTC or TMac has the best performance on all cases. The CPU time of ACPC includes the initialization time by HaLRTC, but the gap between ACPC time and HaLRTC time is not so great on most cases. In addition, ACPC time is much shorter than FBCP time and TNN time on most cases.

5.3 Real-world data

We test three real-world data: a hyperspectral image[2] with size $200 \times 200 \times 89$; an MRI[3] with size $152 \times 188 \times 121$; and a video[4] with size $144 \times 176 \times 100$. The

For ACPC, we set the inner tolerance $tol = 10^{-5}$, the outer tolerance $Tol = 10^{-4}$ and the maximum outer iteration number $N = 3$. The tolerances of all the other methods are set as 10^{-5}.

We compare ACPC with HaLRTC, TMac, FBCP and TNN. The results for different methods under different SRs are presented in Table 3. The comparisons show that ACPC is the best-performing method in terms of relative error. As for the CPU time, the ACPC time is much longer than HaLRTC and is close to the TNN time, but much shorter than the FBCP time.

The visual comparisons of these methods are shown in Figs. 2, 3, and 4. We show one slice of the recovered results of different methods. From all visual comparisons, we can observe that ACPC has the best performance.

[2] The data are available at http://peterwonka.net/Publications/code/LRTC_Package_Ji.zip and have been used in [44].

[3] The data are from BrainWeb [12] and available at http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html.

[4] The data are from the video trace library [37] and available at http://trace.eas.asu.edu/yuv/.

Table 1 Comparison results for different methods under different SRs: synthetic tensors tensors without noise

Tensor	SR		HaLRTC	TMac	FBCP	TNN	ACPC
$50 \times 50 \times 50$ multilinear rank = (5,5,5)	20%	RErr	2.51e–5	2.04e–5	1.14e–1	4.61e–1	**1.65e–5**
		CPU	**0.8**	3.6	4.8	14.0	1.2
	40%	RErr	1.40e–5	1.36e–5	1.44e–5	3.58e–2	**7.81e–6**
		CPU	**0.4**	0.5	6.2	6.6	1.1
	60%	RErr	6.78e–6	5.84e–6	1.03e–5	2.40e–5	**5.33e–6**
		CPU	0.3	**0.2**	7.3	2.8	0.7
$100 \times 100 \times 100$ multilinear rank = (10,10,10)	20%	RErr	2.33e–5	1.23e–5	6.85e–2	4.28e–1	**4.23e–6**
		CPU	6.4	**2.6**	29.8	89.2	21.6
	40%	RErr	1.05e–5	9.05e–6	9.74e–5	3.94e–5	**1.77e–6**
		CPU	2.8	**1.3**	37.8	73.4	6.9
	60%	RErr	1.03e–5	7.52e–6	7.41e–6	2.01e–5	**1.27e–6**
		CPU	7.7	**0.8**	39.3	180.1	10.8
$50 \times 50 \times 50$ CP rank = 5	20%	RErr	3.71e–5	1.63e–1	2.37e–5	3.16e–1	**5.28e–6**
		CPU	**2.8**	3.2	5.3	5.8	8.8
	40%	RErr	1.03e–5	1.19e–1	8.32e–6	1.55e–2	**1.85e–6**
		CPU	**0.5**	3.7	7.0	7.7	2.1
	60%	RErr	9.32e–6	1.00e–1	1.48e–6	2.54e–5	**1.35e–6**
		CPU	**1.2**	4.3	10.6	3.3	1.9
$100 \times 100 \times 100$ CP rank = 10	20%	RErr	3.12e–5	1.02e–1	1.70e–4	2.56e–1	**1.55e–6**
		CPU	16.4	**13.0**	27.9	33.1	26.9
	40%	RErr	1.69e–5	6.64e–2	3.15e–5	3.82e–5	**8.30e–7**
		CPU	**7.8**	16.0	40.3	153.8	14.1
	60%	RErr	8.77e–6	5.42e–2	3.05e–6	2.16e–5	**4.80e–7**
		CPU	**7.8**	61.1	89.6	51.6	14.9

The best results are highlighted in boldface

Table 2 Comparison results for different methods under different SRs: synthetic tensors tensors with noise

Tensor	SR		HaLRTC	TMac	FBCP	TNN	ACPC
$50 \times 50 \times 50$ multilinear rank = (5,5,5)	20%	RErr	7.75e–5	2.31e–5	2.45e–1	4.98e–1	**1.99e–5**
		CPU	**0.7**	3.6	4.5	4.6	2.6
	40%	RErr	6.53e–5	2.07e–5	3.34e–5	3.92e–2	**9.31e–6**
		CPU	0.8	**0.4**	5.9	7.7	1.8
	60%	RErr	5.67e–5	1.28e–5	2.56e–5	9.45e–5	**6.05e–6**
		CPU	0.3	**0.2**	7.1	3.5	1.0
$100 \times 100 \times 100$ multilinear rank = (10,10,10)	20%	RErr	2.60e–5	1.39e–5	2.06e–1	4.52e–1	**4.62e–6**
		CPU	6.6	**3.1**	26.5	87.8	22.4
	40%	RErr	2.23e–5	1.06e–5	2.30e–4	6.18e–5	**2.61e–6**
		CPU	2.8	**1.3**	37.4	157.4	5.5
	60%	RErr	1.74e–5	8.73e–6	1.36e–5	3.83e–5	**1.92e–6**
		CPU	8.2	**0.8**	38.9	58.2	10.5
$50 \times 50 \times 50$ CP rank = 5	20%	RErr	6.42e–5	2.39e–1	1.14e–4	3.55e–1	**5.88e–6**
		CPU	4.1	**0.5**	4.3	5.2	15.6
	40%	RErr	2.25e–5	1.71e–1	9.52e–5	2.77e–2	**2.35e–6**
		CPU	0.6	**0.2**	6.2	8.9	2.7
	60%	RErr	1.12e–5	1.38e–1	2.27e–5	3.94e–5	**1.91e–6**
		CPU	1.2	**0.2**	8.5	6.4	1.7
$100 \times 100 \times 100$ CP rank = 10	20%	RErr	3.21e–5	1.74e–1	8.94e–4	2.85e–1	**1.79e–6**
		CPU	13.9	**1.4**	30.8	34.8	30.5
	40%	RErr	2.02e–5	1.01e–1	5.33e–5	4.78e–5	**1.05e–6**
		CPU	**6.1**	16.2	48.3	233.3	10.0
	60%	RErr	1.59e–5	8.81e–2	1.78e–5	2.78e–5	**4.90e–7**
		CPU	3.8	**2.6**	70.0	36.9	4.8

The best results are highlighted in boldface

Table 3 Comparison results for different methods under different SRs: real-world data

Data	SR		HaLRTC	TMac	FBCP	TNN	ACPC
Hyperspectral image	10%	RErr	7.23e–2	6.22e–2	5.65e–2	7.46e–2	**3.26e–2**
		CPU	**20**	38	1777	131	313
	20%	RErr	2.88e–2	4.05e–2	3.18e–2	5.18e–2	**1.55e–2**
		CPU	**33**	54	5039	315	449
	30%	RErr	2.01e–2	2.96e–2	2.46e–2	4.06e–2	**1.25–2**
		CPU	**23**	27	4701	394	461
	40%	RErr	1.58e–2	2.32e–2	2.09e–2	3.29e–2	**1.14–2**
		CPU	**6**	7	2627	109	144
MRI	10%	RErr	2.70e–1	6.09e–1	1.42e–1	1.93e–1	**1.16e–1**
		CPU	**22**	25	2015	98	135
	20%	RErr	1.66e–1	2.71e–1	9.77e–2	1.03e–1	**6.87e–2**
		CPU	18	**17**	3007	102	163
	30%	RErr	1.10e–1	1.17e–1	7.79e–2	6.80e–2	**4.85–2**
		CPU	**15**	16	2947	60	101
	40%	RErr	7.47e–2	5.90e–2	5.49e–2	4.99e–2	**3.70–2**
		CPU	**5**	6	1997	55	99
Video	10%	RErr	1.31e–1	4.81e–1	1.20e–1	1.28e–1	**8.36e–2**
		CPU	10	**9**	2033	64	106
	20%	RErr	8.67e–2	1.79e–1	8.61e–2	6.97e–2	**5.59e–2**
		CPU	**8**	12	1775	60	99
	30%	RErr	6.17e–2	5.28e–2	6.21e–2	5.04e–2	**4.12–2**
		CPU	**8**	11	1502	58	92
	40%	RErr	4.53e–2	4.12e–2	4.72e–2	3.93e–2	**3.10–2**
		CPU	**6**	8	1344	52	76

The best results are highlighted in boldface

5.4 Summary

The results show that ACPC has a significant improvement over HaLRTC, which is used as initialization, in terms of relative error. The key reason is that ACPC converts the tensor completion problem into a series of slice matrix completion problems, rather than unfolding matrix completion problems. As already mentioned in the introduction, it has been shown in [49] that unfolding matrices may fail to exploit the tensor structure and may lead to poor tensor recovery performance. On the other hand, a slice matrix is a submatrix of the unfolding matrix[5] and can give a more accurate estimate of the rank information.

As for the CPU time, ACPC performs very well on synthetic data but consumes much more time than HaLRTC on real-world data. This is because for synthetic data, the multilinear rank of the initialization is rather small, while for real-world data, the

[5] To be more accurate, the slices that we utilize are submatrices of the unfolding matrix from the original tensor after some linear transform. See Corollary 2.6.

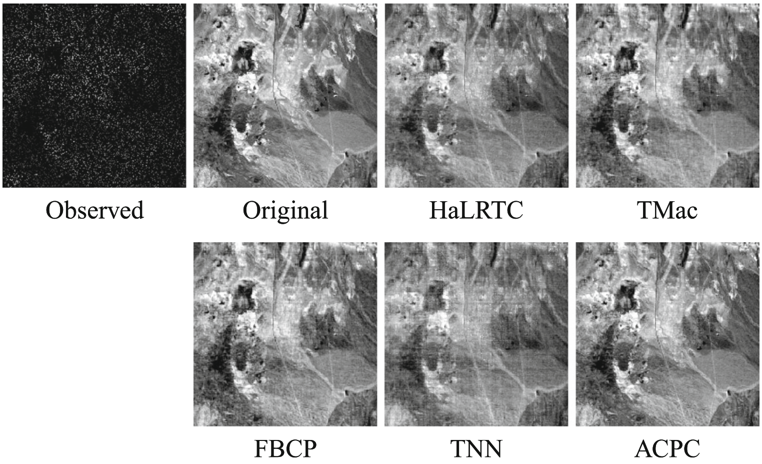

Fig. 2 Recovered results for the hyperspectral image with 10% SR

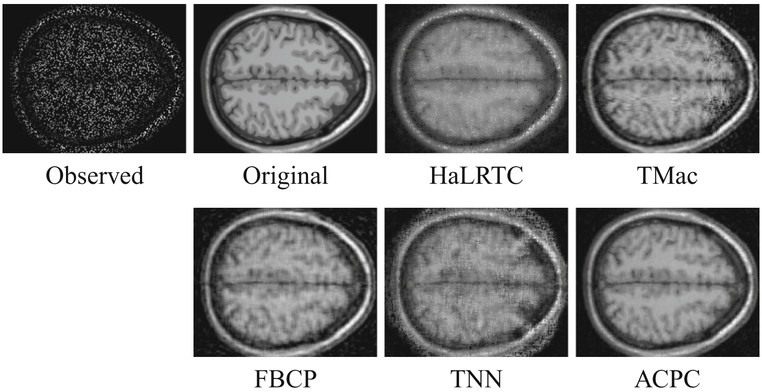

Fig. 3 The recovered results recovered results for the MRI with 20% SR

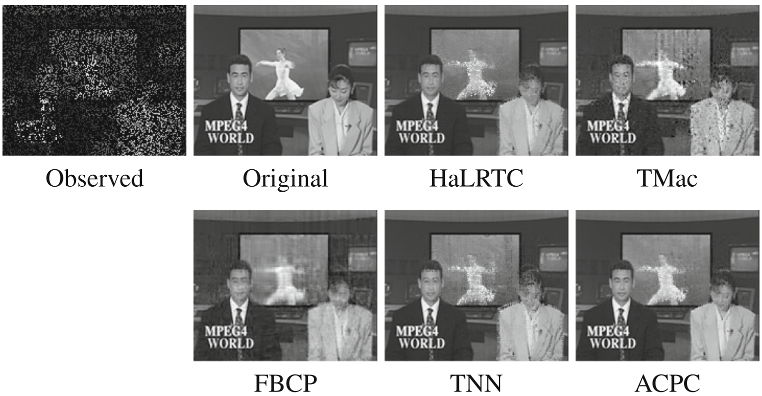

Fig. 4 Recovered results for the video with 30% SR

multilinear rank of the initialization is not very small. A greater multilinear rank of the initialization means that R_d^n is greater in (34) and thus increases the CPU time. To handle this issue, one strategy is to set a smaller maximum outer iteration number. In our experience, $N = 1$ or 2 is enough for most real-world data. Another strategy is that we can use a truncated left singular matrix of $\mathbf{X}_d^0$ as the initial $\mathbf{U}_d^0$ for ACPC.

6 Conclusions

We review the rank invariance properties of tensors. Based on these properties, we obtain an upper bound of CP rank for third-order tensors, which is the sum of ranks of a few matrices. By replacing the CP rank with this bound, the problem of low CP rank tensor completion is converted into the problem of low rank matrix completion problem. The new problem can be implemented by the ADMM easily. The subsequence convergence is also established. We test numerical examples on both synthetic and real-world data. The results show the advantage of the proposed method over some state-of-the-art algorithms.

A considerable problem is how to extend this work to tensors with order higher than three. The core issue is how to extend (9) for higher-order tensors. For a third-order tensor, we use the sum of the ranks of some matrices, which are second-order tensors, to bound the CP rank. For an Nth-order tensor, it is natural to use the sum of the CP ranks of some $(N-1)$st-order tensors to bound the CP rank. However, the CP ranks of $(N-1)$st-order tensors also need to be bounded, leading to a complicated bound for the original Nth-order tensor. The consequence is that this strategy is not practical for higher-order tensors. This problem will be explored in the future.

Acknowledgements We are extremely grateful to two anonymous referees for their valuable feedback, which improved this paper significantly.

References

1. Ashraphijuo, M., Wang, X.: Fundamental conditions for low-CP-rank tensor completion. J. Mach. Learn. Res. **18**(1), 2116–2145 (2017)
2. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Łojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
3. Bader, B.W., Kolda, T.G. et al.: MATLAB Tensor Toolbox Version 3.0-dev. https://www.tensortoolbox.org (2017)
4. Barak, B., Moitra, A.: Noisy tensor completion via the sum-of-squares hierarchy. In: *Conference on Learning Theory*, pp. 417–445 (2016)
5. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**(1–2), 459–494 (2014)
6. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
7. Breiding, P., Vannieuwenhoven, N.: A Riemannian trust region method for the canonical tensor rank approximation problem. SIAM J. Optim. **28**(3), 2435–2465 (2018)
8. Breiding, P., Vannieuwenhoven, N.: The condition number of join decompositions. SIAM J. Matrix Anal. Appl. **39**(1), 287–309 (2018)

9. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
10. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717 (2009)
11. Candès, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inf. Theory **56**(5), 2053–2080 (2010)
12. Cocosco, C.A., Kollokian, V., Kwan, R.K.-S., Pike, G.B., Evans, A.C.: Brainweb: Online interface to a 3D MRI simulated brain database. In NeuroImage, Citeseer (1997)
13. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
14. De Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**(3), 1084–1127 (2008)
15. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)
16. Friedland, S., Lim, L.-H.: Nuclear norm of higher-order tensors. Math. Comput. **87**(311), 1255–1281 (2018)
17. Gandy, S., Recht, B., Yamada, I.: Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Prob. **27**(2), 025010 (2011)
18. Goldfarb, D., Qin, Z.: Robust low-rank tensor recovery: models and algorithms. SIAM J. Matrix Anal. Appl. **35**(1), 225–253 (2014)
19. Håstad, J.: Tensor rank is NP-complete. J. Algorithms **11**(4), 644–654 (1990)
20. Hillar, C.J., Lim, L.-H.: Most tensor problems are NP-hard. J. ACM (JACM) **60**(6), 45 (2013)
21. Holtz, S., Rohwedder, T., Schneider, R.: The alternating linear scheme for tensor optimization in the tensor train format. SIAM J. Sci. Comput. **34**(2), A683–A713 (2012)
22. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (2012)
23. Jain, P., Oh, S.: Provable tensor factorization with missing data. In: *Advances in Neural Information Processing Systems*, pp. 1431–1439 (2014)
24. Jiang, B., Ma, S., Zhang, S.: Tensor principal component analysis via convex optimization. Math. Program. **150**(2), 423–457 (2015)
25. Jiang, B., Ma, S., Zhang, S.: Low-M-rank tensor completion and robust tensor PCA. IEEE J. Sel. Top. Signal Process. **12**(6), 1390–1404 (2018)
26. Jiang, B., Yang, F., Zhang, S.: Tensor and its Tucker core: the invariance relationships. Numer. Linear Algebra Appl. **24**(3), e2086 (2017)
27. Jiang, Q., Ng, M.: Robust low-tubal-rank tensor completion via convex optimization. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 2649–2655 (2019)
28. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. SIAM J. Matrix Anal. Appl. **34**(1), 148–172 (2013)
29. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
30. Kressner, D., Steinlechner, M., Vandereycken, B.: Low-rank tensor completion by Riemannian optimization. BIT Numer. Math. **54**(2), 447–468 (2014)
31. Kruskal, J.B.: Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra Appl. **18**(2), 95–138 (1977)
32. Landsberg, J.M.: Tensors: Geometry and Applications, vol. 128. American Mathematical Society, Providence (2012)
33. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 208–220 (2013)
34. Mu, C., Huang, B., Wright, J., Goldfarb, D.: Square deal: lower bounds and improved relaxations for tensor recovery. In: *International conference on machine learning*, pp. 73–81 (2014)
35. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
36. Rohwedder, T., Uschmajew, A.: On local convergence of alternating schemes for optimization of convex problems in the tensor train format. SIAM J. Numer. Anal. **51**(2), 1134–1162 (2013)
37. Seeling, P., Reisslein, M.: Video transport evaluation with H. 264 video traces. IEEE Commun. Surv. Tutor. **14**(4), 1142–1165 (2011)
38. Steinlechner, M.: Riemannian optimization for high-dimensional tensor completion. SIAM J. Sci. Comput. **38**(5), S461–S484 (2016)

39. Uschmajew, A.: Local convergence of the alternating least squares algorithm for canonical tensor approximation. SIAM J. Matrix Anal. Appl. **33**(2), 639–652 (2012)
40. Vannieuwenhoven, N.: Condition numbers for the tensor rank decomposition. Linear Algebra Appl. **535**, 35–86 (2017)
41. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. **142**(1–2), 397–434 (2013)
42. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math. Program. Comput. **4**(4), 333–361 (2012)
43. Wright, S.J.: Coordinate descent algorithms. Math. Program. **151**(1), 3–34 (2015)
44. Xu, Y., Hao, R., Yin, W., Su, Z.: Parallel matrix factorization for low-rank tensor completion. Inverse Problems Imag. **9**(2), 601–624 (2015)
45. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imag. Sci. **6**(3), 1758–1789 (2013)
46. Yang, J., Yuan, X.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math. Comput. **82**(281), 301–329 (2013)
47. Yang, Y., Feng, Y., Huang, X., Suykens, J.A.: Rank-1 tensor properties with applications to a class of tensor optimization problems. SIAM J. Optim. **26**(1), 171–196 (2016)
48. Yokota, T., Zhao, Q., Cichocki, A.: Smooth PARAFAC decomposition for tensor completion. IEEE Trans. Signal Process. **64**(20), 5423–5436 (2016)
49. Yuan, M., Zhang, C.-H.: On tensor completion via nuclear norm minimization. Found. Comput. Math. **16**(4), 1031–1068 (2016)
50. Zhang, Z., Aeron, S.: Exact tensor completion using t-SVD. IEEE Trans. Signal Process. **65**(6), 1511–1526 (2017)
51. Zhao, Q., Zhang, L., Cichocki, A.: Bayesian CP factorization of incomplete tensors with automatic rank determination. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1751–1763 (2015)