An iterative algorithm for low-rank tensor completion problem with sparse noise and missing values

Jianheng Chen

Xiamen University
School of Mathematical Sciences

This is a joint work with Prof. Wen Huang

中国运筹学会第十六届年会（长沙）2023.4.9

厦門大學
数|学|科|学|学|院
SCHOOL OF MATHEMATICAL SCIENCES
XIAMEN UNIVERSITY

Introduction

Background and Applications

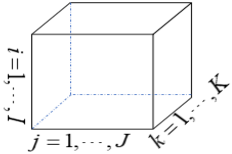

Fig. 1: A third-order tensor: $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$.

Tensor Data

- A tensor is a multidimensional array. Such as a third-order tensor has three indices, as shown in Figure (1). The rapid advance in imaging technology has given rise to the wide presence of high-dimensional data:

- color images;
- video data;
- multispectral/hyperspectral images;
- traffic data;
- $\cdots$

Problem Statement

- **Problem** Tensor completion problem with sparse noise and missing values

$$\min_{\mathcal{L},\mathcal{E}} F(\mathcal{L},\mathcal{E}) = \|\,\mathrm{P}_\Omega(\mathcal{X}-\mathcal{L}-\mathcal{E})\|_F^2 + \lambda\|\mathcal{E}\|_1,$$
$$\text{s.t.}\ \mathrm{rank}_t(\mathcal{L}) = \boldsymbol{r}, \tag{1}$$

where $\mathcal{L} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a low-rank tensor, rank_t denotes a transformed multi-rank,$\mathcal{E} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is sparse, $\Omega \subset \{1,2,\dots,n_1\}\times\{1,2,\dots,n_2\}\times\{1,2,\dots,n_3\}$ is an index set, and P_Ω denotes the projection operator onto Ω, that is,

$$(\mathrm{P}_\Omega(\mathcal{X}))_{ijk} = \begin{cases} \mathcal{X}_{ijk}, & \text{if } (i,j,k)\in\Omega, \\ 0, & \text{otherwise, denoted by } \bar{\Omega}. \end{cases}$$

- Our aim is to recover the underlying low-rank tensor $\mathcal{L}$ from $\mathrm{P}_\Omega(\mathcal{X})$.
- Model (1) relies on the assumptions: the data can be approximated well by a low-rank tensor $\mathcal{L}$, the noise $\mathcal{E}$ is sparse, and the rank of the tensor can be estimated well.

Related Work

- Tensor completion: Tensor completion aims to recover data from partially observed samples.

$$\begin{aligned} &\min_{\mathcal{L}} \operatorname{rank}(\mathcal{L}), \\ &\text{s.t.} \;\; \mathrm{P}_{\Omega}(\mathcal{X}) = \mathrm{P}_{\Omega}(\mathcal{L}), \end{aligned} \tag{2}$$

where the definition of rank varies in literatures.

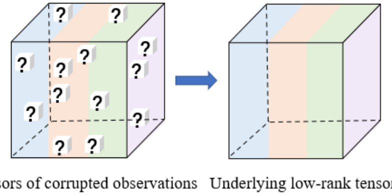

Fig. 2: Tensor completion by low-rank prior on data.

Related Work

- Tensor denoising: All the entries are observed but may be corrupted by noise.

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} \operatorname{rank}(\mathcal{L}) + \lambda\|\mathcal{E}\|_1, \\ &\text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}, \end{aligned} \tag{3}$$

where the definition of rank varies in literatures.

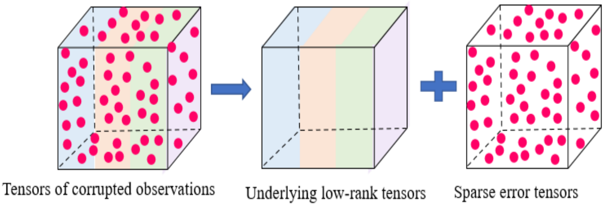

Fig. 3: Tensor denoising by low-rank prior on data and sparsity prior on noise.

Related Work

- Tensor completion and denoising: The natural extension of (3) is

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} \operatorname{rank}(\mathcal{L}) + \lambda\|\mathcal{E}\|_1, \\ &\text{s.t. } \mathrm{P}_\Omega(\mathcal{X}) = \mathrm{P}_\Omega(\mathcal{L}+\mathcal{E}). \end{aligned} \tag{4}$$

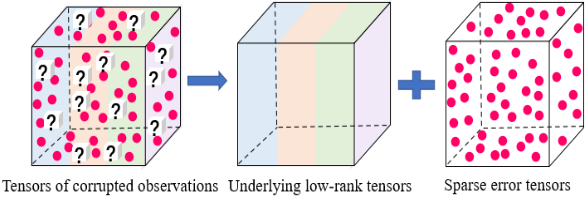

Fig. 4: Tensor completion and denoising by low-rank prior on data and sparsity prior on noise.

Compare the existing models with the proposed model

- The existing models:

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \lambda\|\mathcal{E}\|_1, \\ &\text{s.t.} \;\; \mathrm{P}_\Omega(\mathcal{X}) = \mathrm{P}_\Omega(\mathcal{L}+\mathcal{E}). \end{aligned} \tag{5}$$

- The proposed model:

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} F(\mathcal{L},\mathcal{E}) = \|\,\mathrm{P}_\Omega(\mathcal{X}-\mathcal{L}-\mathcal{E})\|_F^2 + \lambda\|\mathcal{E}\|_1, \\ &\text{s.t.}\, \mathrm{rank}_t(\mathcal{L}) = \boldsymbol{r}. \end{aligned} \tag{6}$$

Compared to (5):

1. Model (6) does not need the constraint $\|\,\mathrm{P}_\Omega(\mathcal{X}) - \mathrm{P}_\Omega(\mathcal{L}+\mathcal{E})\|_F^2 = 0$ to be true, but Model (5) needs.
2. Model (6) needs to estimate the transformed multi-rank of of the tensor $\mathcal{L}$ and uses this fixed rank manifold $\mathcal{M}_{\boldsymbol{r}}$, but Model (5) is different.
3. The parameter λ in Model (6) control the balance between the low-rank tensor error and noise.

Preliminaries

Preliminaries on Tensors

- For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor $\widehat{\mathcal{A}}$ is defined by taking the DCT of all the tubes, that is,

$$\mathrm{vec}(\widehat{\mathcal{A}}(i,j,:)) = \mathrm{dct}(\mathrm{vec}(\mathcal{A}(i,j,:))),\ \forall i,j,$$

  where vec is the vectorization operator that maps the tensor tube to a vector, and the DCT is expressed by dct.

- $$\widehat{\mathcal{A}} = \mathrm{dct}(\mathcal{A}, [], 3).$$

Preliminaries on Tensors

Dedfinition 2.1 (Block diagonal form of third-order tensor)

Let $\bar{\mathcal{A}}$ *be the block diagonal matrix of the tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *in the transform domain, namely,*

$$\bar{\mathcal{A}} = \mathrm{blockdiag}(\widehat{\mathcal{A}}) = \begin{bmatrix} \widehat{\mathcal{A}}^{(1)} & & & \\ & \widehat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \widehat{\mathcal{A}}^{(n_3)} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}.$$

The inverse of $\mathrm{blockdiag}$ *is denoted by* fold, *i.e.,* $\mathrm{fold}(\mathrm{blockdiag}(\widehat{\mathcal{A}})) = \widehat{\mathcal{A}}$.

Preliminaries on Tensors

Dedfinition 2.2 (t_c-product,[KKA15])

The t_c-product $\mathcal{A} * \mathcal{B}$ of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is a tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ which is given by

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \mathrm{idct}[\mathrm{fold}(\mathrm{blockdiag}(\widehat{\mathcal{A}}) \times \mathrm{blockdiag}(\widehat{\mathcal{B}}))],$$

where "$\times$" denotes the usual matrix product. ($\widehat{\mathcal{A}}$ represents the tensor obtained by taking the DCT of all the tubes along the third dimension of $\mathcal{A}$.

Preliminaries on Tensors

Dedfinition 2.3 (t_c-SVD)

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t_c-SVD of $\mathcal{A}$ is given as

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \tag{7}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, that is $\mathcal{U}^\top * \mathcal{U} = \mathcal{I}_{dct}$, $\mathcal{V}^\top * \mathcal{V} = \mathcal{I}_{dct}$, and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a diagonal tensor, respectively.

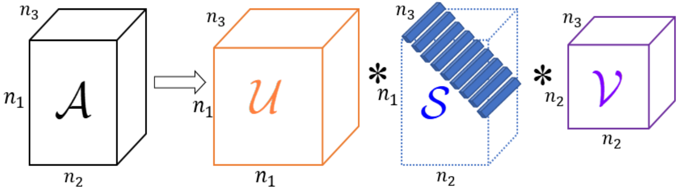

Fig. 5: The t_c-SVD of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Preliminaries on Tensors

Algorithm 1 t_c-SVD for third-order tensors

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$;

1: $\hat{\mathcal{A}} = \mathrm{dct}[\mathcal{A}]$;

2: **for** $i = 0, 1, \ldots, n_3$ **do**

3: $\quad [\mathrm{U}, \mathrm{S}, \mathrm{V}] = \mathrm{SVD}\left(\hat{\mathcal{A}}^{(i)}\right)$;

4: $\quad \hat{\mathcal{U}}^{(i)} = \mathrm{U}, \hat{\mathcal{S}}^{(i)} = \mathrm{S}, \hat{\mathcal{V}}^{(i)} = \mathrm{V}$;

5: **end for**

6: $\mathcal{U} = \mathrm{idct}[\hat{\mathcal{U}}], \mathcal{S} = \mathrm{idct}[\hat{\mathcal{S}}], \mathcal{V} = \mathrm{idct}(\hat{\mathcal{V}}]$;

Output: $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}, \mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$.

Preliminaries on Tensors

Dedfinition 2.4 (transformed multi-rank and tubal rank)

The transformed multi-rank of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *denoted as* $\mathrm{rank}_t(\mathcal{A})$, *is a vector* $\boldsymbol{r} \in \mathbb{R}^{n_3}$ *with its* i*-th entry as the rank of the* i*-th frontal slice of* $\widehat{\mathcal{A}}$, *i.e.,*

$$\mathrm{rank}_t(\mathcal{A}) = \boldsymbol{r} = (r_1, r_2, \cdots, r_{n_3}), \text{ where } r_i = \mathrm{rank}\left(\widehat{\mathcal{A}}^{(i)}\right), \quad i = 1, \ldots, n_3.$$

The tubal rank of a tensor, denoted as $\mathrm{rank}_{ct}(\mathcal{A})$, *is defined as the number of nonzero singular tubes of* $\mathcal{S}$, *where* $\mathcal{S}$ *comes from the* t_c*-SVD of* $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$, *i.e.,*

$$\mathrm{rank}_{ct}(\mathcal{A}) = \#\{i : \mathcal{S}(i,i,:) \neq 0\} = \max_i r_i = r.$$

Preliminaries on Tensors

- The skinny t_c-SVD:
  For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, let $\operatorname{rank}_t(\mathcal{A}) = (r_1, r_2, \cdots, r_{n_3}) = \boldsymbol{r}$, $\operatorname{rank}_{ct} = r$, and $\mathcal{I}_{\boldsymbol{r}} = \operatorname{idct}[\mathcal{I}_d]$, and each frontal slice of $\mathcal{I}_d$ are $\mathcal{I}_d^{(i)} = \begin{pmatrix} \mathcal{I}_{r_i} & 0 \\ 0 & 0 \end{pmatrix} (i = 1, 2, \cdots, n_3)$. Then the skinny t_c-SVD of $\mathcal{A}$ is given as $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$, where $\mathcal{U} \in \mathbb{R}^{n_1\times r\times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2\times r\times n_3}$ satisfy

$$
\begin{aligned}
&\mathcal{U}^\top * \mathcal{U} = \mathcal{I}_{\boldsymbol{r}}, \mathcal{V}^\top * \mathcal{V} = \mathcal{I}_{\boldsymbol{r}}, \operatorname{rank}_t(\mathcal{U}) = \operatorname{rank}_t(\mathcal{V}) = \boldsymbol{r}, \\
&\operatorname{rank}_{ct}(\mathcal{U}) = \operatorname{rank}_{ct}(\mathcal{V}) = r,
\end{aligned}
$$

  and $\mathcal{S} \in \mathbb{R}^{r\times r\times n_3}$ is a diagonal tensor.

A Block Coordinate Descent Algorithm

Iterative form of the proposed problem (1)

- We propose to solve Problem (1) by alternating optimizing over $\mathcal{E}$ and $\mathcal{L}$.
- Given k-th iterate of $\mathcal{L}_k$, the next iterates $\mathcal{E}_{k+1}$ and $\mathcal{L}_{k+1}$ are computed by (approximately) solving

$$\begin{cases} \mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} \| \mathrm{P}_\Omega(\mathcal{X} - \mathcal{L}_k - \mathcal{E})\|_F^2 + \lambda \|\mathcal{E}\|_1, & \text{(8a)} \\ \mathcal{L}_{k+1} \approx \arg\min_{\mathcal{L}\in\mathcal{M}_r} \| \mathrm{P}_\Omega(\mathcal{L} - (\mathcal{X} - \mathcal{E}_{k+1}))\|_F^2, & \text{(8b)} \end{cases}$$

where (8a) is called $\mathcal{E}$-subproblem and (8b) is called $\mathcal{L}$-subproblem.

$\mathcal{E}$-subproblem

- The closed-form solution is available for $\mathcal{E}_{k+1}$.
- We have

$$
\begin{aligned}
\mathcal{E}_{k+1} &= \arg\min_{\mathcal{E}} \| \mathrm{P}_{\Omega}(\mathcal{X} - \mathcal{L}_k - \mathcal{E})\|_F^2 + \lambda\|\mathcal{E}\|_1 \\
&= \arg\min_{\mathcal{E}} \frac{1}{2}\| \mathrm{P}_{\Omega}(\mathcal{E} - (\mathcal{X} - \mathcal{L}_k))\|_F^2 + \frac{\lambda}{2}\| \mathrm{P}_{\Omega}(\mathcal{E})\|_1 + \frac{\lambda}{2}\| \mathrm{P}_{\bar{\Omega}}(\mathcal{E})\|_1 \\
&= \arg\min_{\mathrm{P}_{\Omega}(\mathcal{E})} \frac{1}{2}\| \mathrm{P}_{\Omega}(\mathcal{E}) - \mathrm{P}_{\Omega}(\mathcal{X} - \mathcal{L}_k)\|_F^2 + \frac{\lambda}{2}\| \mathrm{P}_{\Omega}(\mathcal{E})\|_1 \\
&\quad + \arg\min_{\mathrm{P}_{\bar{\Omega}}(\mathcal{E})} \frac{\lambda}{2}\| \mathrm{P}_{\bar{\Omega}}(\mathcal{E})\|_1,
\end{aligned}
\tag{9}
$$

where $\bar{\Omega}$ is an index set including the positions of unknown entries in $\mathcal{X}$.

$\mathcal{E}$-subproblem

- Note that $\mathrm{P}_\Omega(\mathcal{E})$ is the solution of the proximal mapping of one-norm (see [BA17, Lemma 6.5]) and $\mathrm{P}_{\bar{\Omega}}(\mathcal{E})$ must be zero.
- We have

$$(\mathcal{E}_{k+1})_{ijk} = \begin{cases} \left([|\mathcal{X} - \mathcal{L}_k| - \frac{\lambda}{2}]_+ \operatorname{sign}(\mathcal{X} - \mathcal{L}_k)\right)_{ijk}, & \text{if } (i,j,k) \in \Omega, \\ 0, \text{ if } (i,j,k) \in \bar{\Omega}, \end{cases} \tag{10}$$

where $[u]_+$ is denoted by

$$[u]_+ = \begin{cases} u, & \text{if } u \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

$\mathcal{L}$-subproblem

- When solving the $\mathcal{L}$-subproblem, we drop the subscript for simplicity and denote Problem (8b) by
$$\min_{\mathcal{L}\in\mathcal{M}_{\boldsymbol{r}}} f(\mathcal{L}) = \|\mathrm{P}_\Omega(\mathcal{L}-\mathcal{A})\|_F^2, \tag{11}$$
where $\mathcal{A} = \mathcal{X} - \mathcal{E}_{k+1}$.
- Note that the linear conjugate gradient algorithm solves problems in the form of
$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^\top Mx - b^\top x, \tag{12}$$
where $M = M^\top$ is symmetric positive definite.
- Problem (11) can be written in a similar form:
$$\min_{\mathcal{L}\in\mathcal{M}_r} \frac{1}{2}\mathrm{tr}(\mathcal{L}^\top\,\mathrm{P}_\Omega\,\mathcal{L}) - \mathrm{tr}(\mathcal{L}^\top\,\mathrm{P}_\Omega\,\mathcal{A}). \tag{13}$$

Riemannian conjugate gradient algorithm for approximately solving $\mathcal{L}$-subproblem (11) I

Algorithm 2 Riemannian conjugate gradient (RCG) algorithm for approximately solving Problem (11) [SWN20]

Input: $[\mathrm{P}_\Omega(\mathcal{A}), \mathcal{L}^0, \varepsilon]$, where $\mathrm{P}_\Omega(\mathcal{A}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is given, $\varepsilon \geq 0$ is a parameter for stopping criterion, and initial iterate point $\mathcal{L}^0 \in \mathcal{M}_{\boldsymbol{r}}$;

Output: $\mathcal{L}^*$;

1: $i = 0$, $\eta^0 = -\operatorname{grad} f(\mathcal{L}^0)$; $\quad r_0 \leftarrow Mx_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0;$

2: **while** $\|\operatorname{grad} f(\mathcal{L}^i)\| \geq \varepsilon$ **do**

3: Initial step size $\alpha^i = -\frac{\langle \eta^i, \operatorname{grad} f(\mathcal{L}^i)\rangle}{2\langle \eta^i, \mathrm{P}_\Omega(\eta^i)\rangle}$; $\quad \alpha_k \leftarrow -\frac{r_k^\top p_k}{p_k^\top M p_k};$

4: Set $m = 0$;

5: **while** $f(\mathcal{L}^i) - f(\mathrm{R}_{\mathcal{L}^i}(0.5^m \alpha^i \eta^i)) < -0.0001 \times 0.5^m \alpha^i \langle \xi^i, \eta^i \rangle$ **do**

6: $m = m + 1$;

7: **end while**

8: $\mathcal{L}^{i+1} = \mathrm{R}_{\mathcal{L}^i}(0.5^m \alpha^i \eta^i)$; $\quad x_{k+1} \leftarrow x_k + \alpha_k p_k;$

Riemannian conjugate gradient algorithm for approximately solving $\mathcal{L}$-subproblem (11) II

$\beta = \frac{\langle \operatorname{grad} f(\mathcal{L}^i), \mathrm{P}_\Omega(\mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1}))\rangle}{\langle \mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1}), \mathrm{P}_\Omega(\mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1}))\rangle}$; $\quad\beta_{k+1} \leftarrow \frac{r_{k+1}^\top M p_k}{p_k^\top M p_k}$;

if $\frac{|\langle \operatorname{grad} f(\mathcal{L}^i), \mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1})\rangle|}{\|\operatorname{grad} f(\mathcal{L}^i)\|_F \|\mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1})\|_F} > \kappa_1$ or $\|\operatorname{grad} f(\mathcal{L}^i)\|_F > \kappa_2 \|\mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1})\|_F$ **then**

  $\beta = 0$;

end if

Search direction: $\eta^i = -\operatorname{grad} f(\mathcal{L}^i) + \beta \mathcal{T}_{\mathcal{L}^{i-1}\to\mathcal{L}^i}(\eta^{i-1})$;

$r_{k+1} \leftarrow M x_{k+1} - b; \quad p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$;

$i = i + 1$;

end while

Return $\mathcal{L}^* = \mathcal{L}^i$.

- Note that the $\mathcal{L}$-subproblem is not required to be solved exactly.

Algorithm Statement

Algorithm 3 A block coordinate descent (BCD) algorithm for solving Problem (1)

Input: Observed $\mathrm{P}_\Omega(\mathcal{X}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, an estimation of the transformed multi-rank $\boldsymbol{r} > 0$, initial iteration point $\mathcal{L}_0 \in \mathcal{M}_{\boldsymbol{r}}$, parameter $\lambda > 0$ in the objective function, a positive sequence $\{\varepsilon_k\}$ satisfying $\lim_{k\to\infty} \varepsilon_k = 0$;

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Obtain $\mathcal{E}_{k+1}$ by the solution (10) of the $\mathcal{E}$-subproblem (9);
3: Obtain $\mathcal{L}_{k+1}^*$ for solving the $\mathcal{L}$-subproblem (11) by invoking RCG Algorithm 2: $\mathcal{L}_{k+1}^* = Algorithm2[\mathrm{P}_\Omega(\mathcal{X} - \mathcal{E}_{k+1}), \mathcal{L}_k^*, \varepsilon_k]$;
4: **end for**

Convergence Analysis

Assumption 3.1

The transformed multi-rank of the tensor $\mathcal{L}_*^*$ *is* $\boldsymbol{r}$*, i.e.,* $\mathrm{rank}_t(\mathcal{L}_*^*) = \boldsymbol{r}$.

Dedfinition 3.1

The subset ζ *of* $\{(\mathcal{E}, \mathcal{L}) \in \mathbb{R}^{n_1 \times n_2 \times n_3} \times \mathcal{M}_{\boldsymbol{r}}\}$ *is called bounded, if there exists a positive number* N *such that* $\forall(\tilde{\mathcal{E}}, \tilde{\mathcal{L}}) \in \zeta$, $i \in \{1, 2, \cdots, n_1\}$, $j \in \{1, 2, \cdots, n_2\}$, $k \in \{1, 2, \cdots, n_3\}$, *inequalities* $|\tilde{\mathcal{E}}_{ijk}| \leq N$ *and* $|\tilde{\mathcal{L}}_{ijk}| \leq N$ *hold.*

Lemma 3.2

The sublevel sets of the function $F = f + g$ *are bounded, meaning that for any* $\alpha \in \mathbb{R}$*, the sublevel set* $\mathrm{Lev}(F, \alpha) = \{(\mathcal{E}, \mathcal{L}) \in \mathbb{R}^{n_1 \times n_2 \times n_3} \times \mathcal{M}_{\boldsymbol{r}} : F(\mathcal{E}, \mathcal{L}) \leq \alpha\}$ *is bounded. Therefore, the sequence* $\{(\mathcal{E}_k, \mathcal{L}_k)\}$ *generated by Algorithm 3 is bounded.*

Convergence Analysis

Proposition 3.3 ([AMS08, Van13, SWN20])

Suppose $\{\mathcal{L}^i\}$ *be an infinite sequence of iterates generated by RCG Algorithm 2. Then, every accumulation point* $\mathcal{L}^*$ *of* $\{\mathcal{L}^i\}$ *satisfies* $\mathrm{P}_{\mathrm{T}_{\mathcal{L}^*}} \mathrm{P}_\Omega(\mathcal{L}^*) = \mathrm{P}_{\mathrm{T}_{\mathcal{L}^*}} \mathrm{P}_\Omega(\mathcal{X} - \mathcal{E})$.

Theorem 3.4

Suppose Assumption 3.1 holds, and let the sequence $\{(\mathcal{E}_k, \mathcal{L}_k^*)\}$ *generated by BCD Algorithm 3 and* $\{(\mathcal{E}_*, \mathcal{L}_*^*)\}$ *is an accumulation point of the sequence* $\{(\mathcal{E}_k, \mathcal{L}_k^*)\}$. *Then,*

$$\mathcal{E}_* = \arg\min F(\mathcal{E}, \mathcal{L}_*^*), \forall\ \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \tag{14}$$

and

$$\operatorname{grad}_{\mathcal{L}} F(\mathcal{E}_*, \mathcal{L}_*^*) = 0. \tag{15}$$

Implementation Details of BCD Algorithm (3)

$\mathcal{E}$-subproblem and the function of $\mathcal{L}$-subproblem

- The $\mathcal{E}$-subproblem (10) can be compute by

$$\min(0, \mathrm{P}_\Omega(\mathcal{X} - \mathcal{L}_k) + \lambda/2) + \max(0, \mathrm{P}_\Omega(\mathcal{X} - \mathcal{L}_k) - \lambda/2),$$

  and its complexity is $3|\Omega|$ flops.
- Function evaluation: The complexity of the objective function $f(\mathcal{L}) = \|\mathrm{P}_\Omega(\mathcal{L} - \mathcal{A})\|_F^2$, where $\mathcal{A} = \mathcal{X} - \mathcal{E}_{k+1}$, is $4|\Omega|$ flops.
- The set $\mathcal{M}_{\boldsymbol{r}} = \{\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \mathrm{rank}_t(\mathcal{L}) = \boldsymbol{r}\}$ has been well-studied in [SWN20] and it is known to be a manifold.

Manifold geometries:

Tangent space
Riemannian metric
Orthogonal projection to tangent space
Vector transport
Retraction

Tangent space

- Given any $\mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}$, the tangent space at $\mathcal{L}$ is given by

$$\begin{aligned}&\mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}} = \{\mathcal{U} * \mathcal{W} * \mathcal{V}^\top + \mathcal{U}_p * \mathcal{V}^\top + \mathcal{U} * \mathcal{V}_p^\top : \\ &\mathcal{W} \in \mathbb{R}^{r \times r \times n_3}, \mathcal{U}_p \in \mathbb{R}^{n_1 \times r \times n_3}, \mathcal{U}^\top * \mathcal{U}_p = 0, \mathcal{V}_p \in \mathbb{R}^{n_2 \times r \times n_3}, \mathcal{V}^\top * \mathcal{V}_p = 0\}.\end{aligned}$$

  where $\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ denotes the t_c-SVD decomposition of $\mathcal{L}$, $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$, and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$.

- The tangent bundle is defined as the disjoint union of all tangent spaces of $\mathcal{M}_{\boldsymbol{r}}$, that is

$$\mathrm{T}\mathcal{M}_{\boldsymbol{r}} = \{(\mathcal{L}, \xi) | \mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}, \xi \in \mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}\} = \bigcup_{\mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}} \mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}.$$

Metric

- The Euclidean metric on the embedding space $\mathbb{R}^{n_1 \times n_2 \times n_3}$ is

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a_{ijk} b_{ijk}. \tag{16}$$

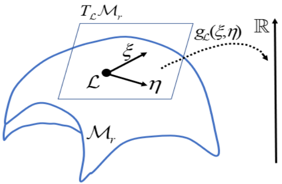

Fig. 6: The Riemannian metric of $\mathcal{M}_{\boldsymbol{r}}$ is endowed with the Euclidean metric (16),i.e., $g_{\mathcal{L}}(\xi, \eta) = \langle \xi, \eta \rangle$, where $\mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}$ and $\xi, \eta \in \mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}$.

Orthogonal projection

- Orthogonal projection:
  Since $\mathcal{L}$ is available as an t_c-SVD, the orthogonal projection onto the tangent space $\mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}$ becomes

$$\mathrm{P}_{\mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}}(\mathcal{Z}) := \mathrm{P}_{\mathcal{U}} * \mathcal{Z} * \mathrm{P}_{\mathcal{V}} + (\mathcal{Z} - \mathrm{P}_{\mathcal{U}} * \mathcal{Z}) * \mathrm{P}_{\mathcal{V}} + \mathrm{P}_{\mathcal{U}} * (\mathcal{Z}^{\top} - \mathrm{P}_{\mathcal{V}} * \mathcal{Z}^{\top})^{\top},$$

  where $\mathrm{P}_{\mathcal{U}} = \mathcal{U} * \mathcal{U}^{\top}$ and $\mathrm{P}_{\mathcal{V}} = \mathcal{V} * \mathcal{V}^{\top}$.
- Riemannian gradient:
  The Riemannian gradient at $\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top}$ is the orthogonal projection of $2\,\mathrm{P}_{\Omega}(\mathcal{L} - (\mathcal{X} - \mathcal{E}_{k+1}))$ onto the tangent space at $\mathcal{L}$.

step size

- step size:
  The initial step size is the exact minimizer along the search direction on the tangent space,

$$\begin{aligned} \alpha^i &= \arg\min_{\alpha} f(\mathcal{L}^i + \alpha\eta^i) \\ &= \arg\min_{\alpha} \| \mathrm{P}_\Omega(\mathcal{L}^i) + \alpha\, \mathrm{P}_\Omega(\eta^i) - \mathrm{P}_\Omega(\mathcal{A}) \|_F^2, \end{aligned} \tag{17}$$

  and is given by

$$\alpha^i = -\frac{\langle \eta^i, \operatorname{grad} f(\mathcal{L}^i) \rangle}{2\langle \eta^i, \mathrm{P}_\Omega(\eta^i) \rangle} = \frac{\langle \mathrm{P}_\Omega(\eta^i), \mathrm{P}_\Omega(\mathcal{A} - \mathcal{L}^i) \rangle}{\langle \mathrm{P}_\Omega(\eta^i), \mathrm{P}_\Omega(\eta^i) \rangle}. \tag{18}$$

Retraction

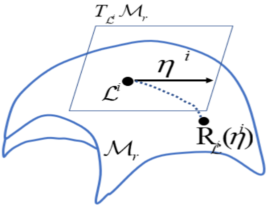

Fig. 7: The Retraction: R.

- For manifold $\mathcal{M}_{\boldsymbol{r}}$, a retraction is given by projection retraction, that is

$$\mathrm{R}_{\mathcal{L}}(\xi) = \arg\min_{\mathcal{Y}\in\mathcal{M}_{\boldsymbol{r}}} \|\mathcal{Y} - (\mathcal{L} + \xi)\|_F.$$

Retraction

- Retraction:
  The retraction can be directly computed by the t_c-SVD of $\mathcal{L}+\xi$.
- Fortunately, the tensor to retract has the particular form

$$\mathcal{L}+\xi=[\mathcal{U} \quad \mathcal{U}_p]*\begin{bmatrix} \mathcal{S}+\mathcal{W} & \mathcal{I} \\ \mathcal{I} & 0 \end{bmatrix}*[\mathcal{V} \quad \mathcal{V}_p]^\top,$$

where $\mathcal{L}=\mathcal{U}*\mathcal{S}*\mathcal{V}^\top\in\mathcal{M}_{\boldsymbol{r}}$ and $\xi=\mathcal{U}*\mathcal{W}*\mathcal{V}^\top+\mathcal{U}_p*\mathcal{V}^\top+\mathcal{U}*\mathcal{V}_p^\top\in\mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}$.
The retraction can be gain by Algorithm 4.

Calculate the Retraction

Algorithm 4 Calculate the retraction by metric projection

Input: Tensor $\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top \in \mathcal{M}_{\boldsymbol{r}}$, tangent vector $\xi = \mathcal{U} * \mathcal{W} * \mathcal{V}^\top + \mathcal{U}_p * \mathcal{V}^\top + \mathcal{U} * \mathcal{V}_p^\top$;

Output: retraction $\mathrm{R}_{\mathcal{L}}(\xi) = \mathrm{P}_{\mathcal{M}_{\boldsymbol{r}}}(\mathcal{L} + \xi) = \mathcal{U}_+ \mathcal{S}_+ \mathcal{V}_+^\top$

$\mathcal{U} \leftarrow \mathrm{dct}(\mathcal{U})$, $\mathcal{S} \leftarrow \mathrm{dct}(\mathcal{S})$, $\mathcal{V} \leftarrow \mathrm{dct}(\mathcal{V})$, $\mathcal{W} \leftarrow \mathrm{dct}(\mathcal{W})$, $\mathcal{U}_p \leftarrow \mathrm{dct}(\mathcal{U}_p)$, $\mathcal{V}_p \leftarrow \mathrm{dct}(\mathcal{V}_p)$; $(2n_1 + 2n_2 + 2r) r n_3 \log n_3$ flops

for $i = 1 : n_3$ **do**

$\quad (\mathcal{Q}_u(:,:,i), \mathcal{R}_u(:,:,i)) \leftarrow \mathrm{qr}(\mathcal{U}_p(:,:,i), 0)$, $(\mathcal{Q}_v(:,:,i), \mathcal{R}_v(:,:,i)) \leftarrow \mathrm{qr}(\mathcal{V}_p(:,:,i), 0)$; $10(n_1 + n_2) \sum_{k=1}^{n_3} r_k^2$ flops

$\quad \bar{\mathcal{S}}(:,:,i) \leftarrow \begin{bmatrix} \mathcal{S}(:,:,i) + \mathcal{W}(:,:,i) & \mathcal{R}_v^\top(:,:,i) \\ \mathcal{R}_u(:,:,i) & 0 \end{bmatrix}$; $\sum_{k=1}^{n_3} r_k^3$ flops

$\quad (\mathcal{U}_s(:,:,i), \mathcal{S}_s(:,:,i), \mathcal{V}_s(:,:,i)) \leftarrow \mathrm{svd}\, \bar{\mathcal{S}}(:,:,i)$;

$\quad \mathcal{S}_+(:,:,i) \leftarrow \mathcal{S}_s(1 : r_i, 1 : r_i, i) + \varepsilon_{mach}$;

$\quad \mathcal{U}_+(:,:,i) \leftarrow [\mathcal{U}(:,:,i) \quad \mathcal{Q}_u] \mathcal{U}_s(:, 1 : r_i, i)$, $\mathcal{V}_+(:,:,i) \leftarrow [\mathcal{V}(:,:,i) \quad \mathcal{Q}_v] \mathcal{V}_s(:, 1 : r_i, i)$; $4(n_1 + n_2) \sum_{k=1}^{n_3} r_k^2$ flops

end for

$\mathcal{U}_+ \leftarrow \mathrm{idct}(\mathcal{U}_+)$, $\mathcal{S}_+ \leftarrow \mathrm{idct}(\mathcal{S}_+)$, $\mathcal{V}_+ \leftarrow \mathrm{idct}(\mathcal{V})_+$; $\quad (n_1 + r + n_2) r n_3 \log n_3$ flops

$\mathcal{U}_+ \mathcal{S}_+ \mathcal{V}_+^\top$. $\quad 2n_1 \sum_{k=1}^{n_3} r_k^2 + 2n_1 n_2 \sum_{k=1}^{n_3} r_k + n_1 n_2 n_3 \log n_3$.

Storage

- Any point $\mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}$ is stored by $(\mathcal{L}, \mathcal{U}, \mathcal{S}, \mathcal{V})$.
  $\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ is the compact t_c-SVD.
  Storing $\mathcal{U}, \mathcal{S}, \mathcal{V}$ for $\mathcal{L}$ reduces the computational cost of the gradient evaluation and retraction evaluation [Van13, HAGH16].
- Any tangent vector $\eta \in \mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}$ is stored by $(\eta, \mathcal{W}, \mathcal{U}_p, \mathcal{V}_p)$.
  A tangent vector $\eta \in \mathrm{T}_{\mathcal{L}}\mathcal{M}_{\boldsymbol{r}}$ at $\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top \in \mathcal{M}_{\boldsymbol{r}}$ can be written as

  $$\eta = \mathcal{U} * \mathcal{W} * \mathcal{V}^\top + \mathcal{U}_p * \mathcal{V}^\top + \mathcal{U} * \mathcal{V}_p^\top,$$

  where $\mathcal{W} \in \mathbb{R}^{r \times r \times n_3}$, $\mathcal{U}_p \in \mathbb{R}^{n_1 \times r \times n_3}$ with $\mathcal{U}^\top * \mathcal{U}_p = 0$, and $\mathcal{V}_p \in \mathbb{R}^{n_2 \times r \times n_3}$ with $\mathcal{V}^\top * \mathcal{V}_p = 0$.

The complexity of BCD Algorithm 3

	$\mathcal{E}$-subproblem	$3\lvert\Omega\rvert$
$\mathcal{L}$-subproblem	Objective function	$4\lvert\Omega\rvert$
	Riemannian gradient	$(2n_1n_2 + 3n_1r + 3n_2r + r^2)n_3\log(n_3)$ $+(6n_1 + 2n_2)\sum_{k=1}^{n_3} r_k^2 + 10n_1n_2\sum_{k=1}^{n_3} r_k$ $+(n_1 + n_2)rn_3 + 2n_1n_2n_3$
	Initial step-size α	$4\lvert\Omega\rvert$
	Retraction	$(3n_1 + 3n_2 + 3r)rn_3\log(n_3)$ $+(16n_1 + 14n_2)\sum_{k=1}^{n_3} r_k^2 + \sum_{k=1}^{n_3} r_k^3$ $+2n_1n_2\sum_{k=1}^{n_3} r_k + n_1n_2n_3\log(n_3)$
	β calculation	$4\lvert\Omega\rvert + (2n_1n_2 + 3n_1r + 3n_2r + r^2)n_3\log(n_3)$ $+(6n_1 + 2n_2)\sum_{k=1}^{n_3} r_k^2 + 10n_1n_2\sum_{k=1}^{n_3} r_k$ $+(n_1 + n_2)rn_3 + 2n_1n_2n_3$
	Search direction	$2rn_3(r + n_1 + n_2)$ $+(r^2 + n_1r + n_2r + n_1n_2)n_3\log n_3$ $+2n_1\sum_{k=1}^{n_3} r_k^2 + 6n_1n_2\sum_{k=1}^{n_3} r_k + 2n_1n_2n_3$
	Total	$(6n_1n_2 + 10n_1r + 10n_2r + 6r^2)n_3\log(n_3)$ $+(30n_1 + 18n_2)\sum_{k=1}^{n_3} r_k^2$ $+28n_1n_2\sum_{k=1}^{n_3} r_k + 15\lvert\Omega\rvert + rn_3(2r + 4n_1 + 4n_2)$ $+6n_1n_2n_3 + \sum_{k=1}^{n_3} r_k^3$

Special Case: Tensor Denoising

- When Ω is the entire set, the tensor completion and denoising problem (1) becomes

$$\min_{\mathcal{L}\in\mathcal{M}_{\mathrm{r}},\mathcal{E}} F(\mathcal{L},\mathcal{E}) = \|\mathcal{X}-\mathcal{L}-\mathcal{E}\|_F^2 + \lambda\|\mathcal{E}\|_1, \tag{19}$$

which is a model for tensor denoising.

Algorithm 5 A block coordinate descent (BCD) algorithm for solving Problem (19)

Input: Observed $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, an estimation of the transformed multi-rank $\boldsymbol{r} > 0$, initial iteration point $\mathcal{L}_0 \in \mathcal{M}_{\boldsymbol{r}}$, parameter $\lambda > 0$ in the objective function;

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Obtain $\mathcal{E}_{k+1}$ by the solution (10) of the $\mathcal{E}$-subproblem;
3: Obtain $\mathcal{L}_{k+1}^*$ by the solution (20) of the $\mathcal{L}$-subproblem;
4: **end for**

- The $\mathcal{L}$-subproblem, in this case, has a closed-form solution, see Theorem 4.1.

Special Case: Tensor Denoising: $\mathcal{L}$-subproblem

Theorem 4.1

Let the t_c-SVD of $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, be given by $\mathcal{Z} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ and $\boldsymbol{r} = (r_1, r_2, \cdots, r_{n_3})$ for $r_i < \min(n_1, n_2)$, $i = 1, 2, \cdots, n_3$, define

$$\mathcal{A}_{\boldsymbol{r}} = \mathcal{U} * \mathcal{S}_{\boldsymbol{r}} * \mathcal{V}^\top,$$

then,

$$\mathcal{A}_{\boldsymbol{r}} = \arg \min_{\mathcal{A} \in \mathcal{M}_{\boldsymbol{r}}} \|\mathcal{Z} - \mathcal{A}\|_F,$$

where $\mathcal{S}_{\boldsymbol{r}}$ is a diagonal tensor with $\hat{\mathcal{S}}_{\boldsymbol{r}}^{(i)} = \begin{pmatrix} \hat{\mathcal{S}}^{(i)}_{1:r_i,1:r_i} & 0 \\ 0 & 0 \end{pmatrix}, (i = 1, 2, \cdots, n_3)$.

- The closed-form solution of the $\mathcal{L}$-subproblem is therefore given by

$$\mathcal{L}_{k+1} = \mathcal{U} * \mathcal{S}_{\boldsymbol{r}} * \mathcal{V}^\top, \tag{20}$$

where $\mathcal{X} - \mathcal{E}_{k+1} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ is the t_c-SVD of $\mathcal{X} - \mathcal{E}_{k+1}$.

Convergence of Special Case

Theorem 4.2

Let $\{(\mathcal{E}_k, \mathcal{L}_k^*)\}$ *be the sequence generated by BCD Algorithm 5 and* $\{(\mathcal{E}_*, \mathcal{L}_*^*)\}$ *be a limit point of the sequence* $\{(\mathcal{E}_k, \mathcal{L}_k^*)\}$. *Then,*

$$\mathcal{E}_* = \arg\min F(\mathcal{E}, \mathcal{L}_*^*), \forall\ \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3},$$

and

$$\mathcal{L}_*^* \in \arg\min F(\mathcal{E}_*, \mathcal{L}), \forall\ \mathcal{L} \in \mathcal{M}_{\boldsymbol{r}}.$$

Numerical Experiments

Measurements

- The peak signal-to-noise ratio (PSNR) is defined as follows

$$\text{PSNR} = 10\log_{10}\frac{n_1n_2n_3(\tilde{\mathcal{L}}_{\max}-\tilde{\mathcal{L}}_{\min})^2}{\|\mathcal{L}_*-\tilde{\mathcal{L}}\|_F^2},$$

  where $\mathcal{L}_*$ is the recovered solution, $\tilde{\mathcal{L}}$ is the ground-truth tensor and $\tilde{\mathcal{L}}_{\max}$ and $\tilde{\mathcal{L}}_{\min}$ are maximal and minimal entries of $\tilde{\mathcal{L}}$, respectively.
- The relative error (Res) is defined by

$$\text{Res} = \frac{\|\mathcal{L}_*-\tilde{\mathcal{L}}\|_F}{\|\tilde{\mathcal{L}}\|_F},$$

  where $\mathcal{L}_*$ is the recovered solution and $\tilde{\mathcal{L}}$ is the ground-truth tensor.

Stopping criterion

- Stopping criterion of the algorithm

$$\frac{\|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F^2}{\|\mathcal{L}_k\|_F^2} \leq \text{tol},$$

  where tol $= 10^{-5}$ for BCD Algorithm 3, tol $= 10^{-6}$ for BCD Algorithm 5.
- The maximum number of iterations: 200.

Test by BCD Algorithm 3 with different ranks

- $rhos = 0.1$, $RS = 0.8$.
- Image named “facade” with different ranks to test by Algorithm 3.

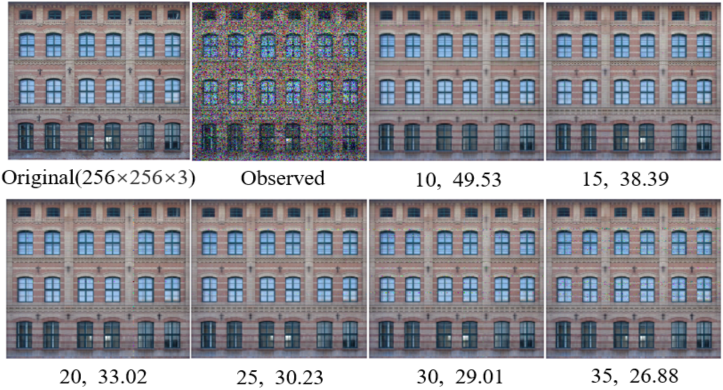

Fig. 8: The original image, the observation image, the recovered images and PSNR values by taking 10,15,20,25,30,35 for the rank respectively.

Algorithm of Rank Estimation

Input: Observed $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the indices of know pixels Ω, a threshold τ_1 for controlling the upper bound of the rank (default $\tau_1 = 0.10$), a threshold τ_2 for estimating the rank (default $\tau_2 = 0.985$);

1: $r_{max} = \tau_1 \min(n_1, n_2)$;

2: $\bar{x} = (\sum_{(i,j,k)\in\Omega} \mathcal{X}_{ijk} - 0.5 * \text{rhos} * \text{SR} * n_1 * n_2 * n_3)/(|\Omega| - \text{rhos} * \text{SR} * n_1 * n_2 * n_3)$, where rhos denotes the noise rate, SR denotes the sampling rate, and $|\Omega|$ denotes the number of entries in Ω;

3: Define $\bar{\mathcal{X}}$ by

$$\bar{\mathcal{X}}_{ijk} = \begin{cases} \mathcal{X}_{ijk}, & \text{if } (ijk) \in \Omega, \\ \bar{x}, & \text{otherwise}; \end{cases} \tag{21}$$

4: Compute $\widetilde{\mathcal{X}} = dct(\bar{\mathcal{X}}, [], 3)$;

5: **for** $i = 1 : n_3$ **do**

6: $\quad [\tilde{U}_i, \tilde{S}_i, \tilde{V}_i] = \text{svd}(\widetilde{\mathcal{X}}(:,:,i))$;

7: $\quad S_i = diag(\tilde{S}_i)$;

8: **end for**

9: $\tilde{r} = \arg\min_r \frac{\sum_{i=1}^{n_3} \sum_{j=1}^{r} (S_i)_j^2}{\sum_{i=1}^{n_3} \sum_{j=1}^{\min(n_1,n_2)} (S_i)_j^2} \geq \tau_2^2$;

Output: $r = r_1 = r_2 = \cdots = r_{n_3} = \min(\tilde{r}, r_{max})$, i.e. an estimation of the transformed multi-rank $\boldsymbol{r}$.

Performance using Color Images

- Compare the BCD algorithm 3 with other tensor completion algorithms on color images.
- Test the first eight color images from Berkeley Dataset[1] with the size $321 \times 481 \times 3$.

TABLE 1 The average results of PSNR and Res values of first eight color images from Berkeley Dataset by different algorithms with rhos $= 0.10$, SR $= 0.60, 0.70, 0.80$.

		SiLRTCTT	STTC	TRLRF	TRALS	TRWOPT	BS-TMac	TNTV (DCT)	TNTV (Data)	BCD
Eight images SR $= 0.80$	PSNR	19.97	20.09	19.96	26.89	26.70	23.06	34.27	34.25	**39.81**
	Res	0.24	0.24	0.24	0.11	0.11	0.16	0.04	0.04	**0.02**
	time(s.)	4.19	27.01	37.03	71.69	41.72	0.40	13.05	12.66	53.19
Eight images SR $= 0.70$	PSNR	20.24	20.44	20.36	26.31	26.11	22.80	32.84	32.88	**37.27**
	Res	0.23	0.23	0.23	0.11	0.12	0.17	0.05	0.05	**0.04**
	time(s.)	5.03	29.23	40.64	69.21	42.43	0.43	13.05	12.68	68.23
Eight images SR $= 0.60$	PSNR	20.44	20.82	20.75	25.58	25.44	22.50	30.96	31.02	**33.69**
	Res	0.23	0.22	0.22	0.12	0.12	0.17	0.07	0.07	**0.06**
	time(s.)	6.60	34.69	39.90	66.30	44.22	0.49	14.21	13.67	97.21

Performance using Synthetic Data I

- The first ground-truth tensor $\tilde{\mathcal{L}}_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is randomly generated by `rand(n1,r,n3)*rand(r,n2,n3)`.
- Set $n_1 = 50, n_2 = 50, n_3 = 3, r = 1$.

TABLE 2 PSNR and Res values of different algorithms for synthetic data with the size $50 \times 50 \times 3$. rhos $= 0.1, 0.2$, SR $= 0.6, 0.7, 0.8$.

	SR	rhos	SiLRTCTT	TRLRF	TRALS	TRWOPT	BS-TMac	TNTV (DCT)	TNTV (Data)	BCD
PSNR	0.8	0.1	22.13	21.81	18.31	23.34	19.81	24.97	24.97	**51.07**
Res			0.22	0.22	0.34	0.19	0.28	0.13	0.13	**0.0078**
time(s.)			**0.0363**	0.2556	1.0988	0.3318	0.0426	0.2748	0.2924	0.4060
PSNR		0.2	18.89	18.99	16.03	20.22	17.40	21.16	21.16	**45.14**
Res			0.32	0.31	0.44	0.27	0.38	0.20	0.20	**0.0155**
time(s.)			**0.0464**	0.8980	0.9104	0.3118	0.0512	0.2293	0.3202	0.5606

Performance using Synthetic Data II

PSNR	0.7	0.1	22.55	21.85	16.61	21.37	19.71	23.60	23.54	**50.63**
Res			0.21	0.22	0.41	0.24	0.29	0.15	0.15	**0.0082**
time(s.)			**0.0535**	0.5965	0.8668	0.3007	0.0744	0.2284	0.3329	0.5360
PSNR		0.2	19.25	19.28	16.42	17.94	17.03	20.47	20.47	**44.75**
Res			0.30	0.30	0.42	0.35	0.40	0.22	0.22	**0.0163**
time(s.)			**0.0570**	0.6051	0.9343	0.3195	0.0755	0.2370	0.2921	0.7185
PSNR	0.6	0.1	22.85	22.04	13.99	20.58	18.73	22.24	22.23	**49.78**
Res			0.20	0.22	0.56	0.26	0.34	0.18	0.18	**0.0091**
time(s.)			**0.0549**	0.5719	0.8309	0.3266	0.0928	0.2268	0.2959	0.6324
PSNR		0.2	19.53	19.48	13.76	17.82	16.75	19.73	19.73	**44.68**
Res			0.29	0.29	0.57	0.36	0.43	0.24	0.24	**0.0164**
time(s.)			**0.0680**	0.6177	0.8749	0.3141	0.1311	0.2530	0.2914	0.8012

Performance using Synthetic Data I

- The second ground-truth tensor $\tilde{\mathcal{L}}_2 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is randomly generated by `1/3*rand(n1,r,n3)*rand(r,n2,n3)`.
- Set $n_1 = 50, n_2 = 50, n_3 = 50, r = 1$.

TABLE 3 PSNR and Res values of different algorithms for synthetic data with the size $50 \times 50 \times 50$. rhos $= 0.1, 0.2$, SR $= 0.6, 0.7, 0.8$.

	SR	rhos	SiLRTCTT	TRLRF	TRALS	TRWOPT	BS-TMac	TNTV (DCT)	TNTV (Data)	BCD
PSNR	0.8	0.1	20.61	20.00	26.74	26.74	16.65	29.98	28.74	**55.70**
Res			0.1486	0.1486	0.0617	0.0603	0.2347	0.0396	0.0445	**0.0026**
time(s.)			0.45	2.50	1.10	4.27	**0.11**	3.05	3.15	7.65
PSNR		0.2	17.53	17.185	25.08	25.03	13.15	26.28	25.65	**50.94**
Res			0.2120	0.2105	0.0796	0.0778	0.3530	0.0602	0.0635	**0.0045**
time(s.)			0.58	5.22	1.83	4.28	**0.12**	3.04	3.43	10.86

Performance using Synthetic Data II

PSNR	0.7	0.1	21.04	20.50	26.79	27.22	16.60	28.46	27.79	**50.89**
Res			0.1415	0.1406	0.0634	0.0619	0.2361	0.0463	0.0501	**0.0045**
time(s.)			0.49	0.89	0.76	4.50	**0.13**	3.03	3.08	9.75
PSNR		0.2	17.94	17.69	25.08	24.38	13.10	25.47	25.30	**49.18**
Res			0.2023	0.1985	0.0819	0.0807	0.3563	0.0658	0.0685	**0.0055**
time(s.)			0.61	5.27	1.06	4.21	**0.14**	3.02	3.09	12.87
PSNR	0.6	0.1	21.52	21.08	25.29	26.45	16.43	27.30	26.86	**47.64**
Res			0.1338	0.1318	0.0997	0.0660	0.2409	0.0545	0.0574	**0.0066**
time(s.)			0.56	5.29	6.02	4.25	**0.16**	3.07	3.14	11.62
PSNR		0.2	18.41	18.24	25.41	24.52	13.16	25.03	24.68	**46.45**
Res			0.1916	0.1859	0.0852	0.0844	0.3515	0.0728	0.0750	**0.0075**
time(s.)			0.68	5.31	1.12	4.21	**0.17**	3.03	3.13	14.91

Special Case: Tensor Denoising about Color Image Recovery

- Compare the BCD algorithm 5 with TRPCA [LFC+20].
- Consider the Berkeley Dataset such as starfish and bird with the size $321 \times 481 \times 3$.
- Set noise rate rhos $= 0.10$ and 0.15.

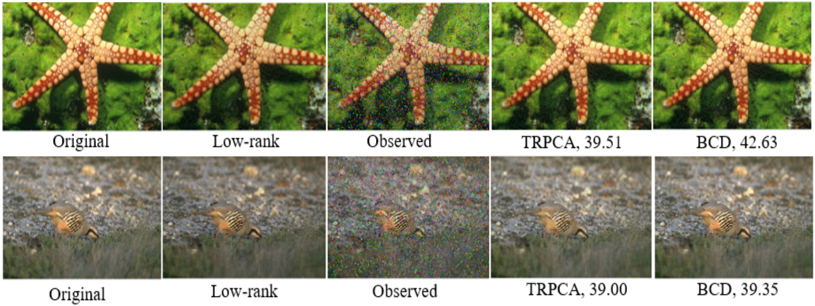

Fig. 9: Original images, original low-rank images, observation images, and the recovered images of TRPCA and BCD, respectively. rhos $= 0.15$.

Special Case: Tensor Denoising about Color Image Recovery

TABLE 4 PSNR and CPU time of different algorithms for two color images from the Berkeley Dataset. rhos $= 0.10, 0.15$.

	rhos		TRPCA	BCD
starfish	0.10	PSNR	44.11	**45.57**
		time(s.)	8.39	**1.94**
	0.15	PSNR	39.51	**42.63**
		time(s.)	8.16	**2.61**
bird	0.10	PSNR	42.46	**43.70**
		time(s.)	8.51	**1.87**
	0.15	PSNR	39.00	**39.35**
		time(s.)	8.20	**2.53**

Conclusion

Conclusion

- We consider the new optimization model on a transformed multi-rank manifold to complete and denoise tensor data. A block coordinate descent algorithm is designed to solve this model.
- When all elements are known, it is a special case, i.e. the tensor denosing problem. We design a simplified block coordinate descent algorithm to solve it.
- Some numerical examples show that the new tensor completion model and algorithm are feasible and effective to solve the low-rank tensor completion problem.

References I

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix mani- folds. *Princeton University Press, Princeton, NJ*, 2008.

[BA17] Beck and Amir. First-order methods in optimization. 2017.

[BPTD17] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Trans Image Process*, 26(5):2466–2479, 2017.

[GF19] Shangqi Gao and Qibin Fan. Robust balancing scheme-based approach for tensor completion. *Neurocomputing*, 330:328–336, 2019.

[HAGH16] Wen Huang, P.-A. Absil, K. A. Gallivan, and Paul Hand. Roptlib: an object-oriented c++ library for optimization on riemannian manifolds. Technical Report FSU16-14.v2, Florida State University, 2016.

[KKA15] E. Kernfeld, M. Kilmer, and S. Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015.

References II

[LFC+20] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2020.

[QBNZ21] D. Qiu, M. Bai, M. K. Ng, and X. Zhang. Robust low-rank tensor completion via rm with total variation regularization. *Neurocomputing*, 435(3), 2021.

[SNZ20] Guangjing Song, M. K. Ng, and X. Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3), 2020.

[SWN20] G. J. Song, X. Z. Wang, and M. K. Ng. Riemannian conjugate gradient descent method for third-order tensor completion. 2020.

[Van13] B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):10.1137/110845768, 2013.

References III

[WAA17] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5698–5706, 2017.

[YCZ+18] Longhao Yuan, Jianting Cao, Xuyang Zhao, Qiang Wu, and Qibin Zhao. Higher-dimension tensor completion via low-rank tensor ring decomposition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1071–1076, 2018.

[YLM+19] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. pages 9151–9158, 2019.

[YLZ+18] Yipeng, Liu, Zhen, Long, Ce, and Zhu. Image completion using low tensor tree rank and total variation minimization. *IEEE Transactions on Multimedia*, 2018.

Thank you for your attention!

Environment and Parameters

Environment

- All the experiments are performed under Windows 10 and MATLAB R2021b running on a laptop (Intel(R) Core(TM) i7, @ 2.80GHz, 16.0G RAM).

Parameters

- In the BCD algorithm 3, the parameter [LFC+20] $\lambda = \frac{1}{\sqrt{\max(n_1,n_2)*n_3}}$.
- In the Riemannian CG Algorithm 2:
  Set $\varepsilon = \frac{1}{2k}$, where k is the kth outer iteration of BCD Algorithm 3;
  Set the maximum number of iterations to 10.
  Set $\kappa_1 = 0.1$ and $\kappa_2 = 1$ for restarting conditions.
- All the hyper-parameters of the algorithms for comparison: default settings in the corresponding papers.
- SiLRTCTT [BPTD17], STTC [YLZ+18], TRLRF [YLM+19], TRWOPT [YCZ+18], TRALS [WAA17], BS-TMac [GF19], TNTV(DCT) [QBNZ21] and TNTV(Data)) [SNZ20, QBNZ21].