Handling Slice Permutations Variability in Tensor Recovery

Jingjing Zheng[1,2], **Xiaoqin Zhang**[2*], **Wenzhe Wang**[2], **Xianta Jiang**[1]

[1] Department of Computer Science, Memorial University of Newfoundland, Newfoundland and Labrador, Canada
[2] College of Computer Science and Artificial Intelligence, Wenzhou University, Zhejiang, China
jjzheng233@gmail.com, zhangxiaoqinnan@gmail.com, woden3702@gmail.com, xiantaj@mun.ca

Abstract

This work studies the influence of slice permutations on tensor recovery, which is derived from a reasonable assumption about algorithm, *i.e.* changing data order should not affect the effectiveness of the algorithm. However, as we will discussed in this paper, this assumption is not satisfied by tensor recovery under some cases. We call this interesting problem as Slice Permutations Variability (SPV) in tensor recovery. In this paper, we discuss SPV of several key tensor recovery problems theoretically and experimentally. The obtained conclusion shows that there is a huge gap between results by tensor recovery using tensor with different slices sequences. To overcome SPV in tensor recovery, a novel tensor recovery algorithm by Minimum Hamiltonian Circle for SPV (TRSPV) is developed which exploits a low dimensional subspace structures within data tensor more exactly. To our best knowledge, this is the first work to discuss SPV in tensor recovery and give an effective solution for it. The experimental results demonstrate the effectiveness of the proposed algorithm in eliminating SPV in tensor recovery.

Introduction

With the growing explosion of high-dimensional data such as images and videos, the problem of exploiting low-dimensional structures in such high-dimensional data has become increasingly important in computer vision and pattern recognition (Candes and Plan 2010; Candès and Recht 2009; Candès et al. 2011; Chandrasekaran et al. 2009; Xu, Caramanis, and Sanghavi 2012; Wright et al. 2009; Eckart and Young 1936; Wold, Esbensen, and Geladi 1987; Zhou et al. 2010). Since most visual data including color images and video are in the form of tensor, dealing with such tensor data has attracted more and more attention recently. And lots of low rank tensor recovery methods are proposed (Gandy, Recht, and Yamada 2011; Lu et al. 2019; Zhang et al. 2014a, 2019; Zheng et al. 2019; Zhang et al. 2021; Yang et al. 2020; Cai et al. 2021; Lu, Peng, and Wei 2019), which basically assumes the tensor data approximately lie on a low-dimensional linear subspace, and to recover a low rank tensor form the high-dimensional data tensor with various of perturbation. These methods have been widely used in various fields such as color images and video processing (Tan et al. 2014; Dian, Li, and Fang 2019; Wei et al. 2018), data dimension reduction (Luo et al. 2015), etc.

A key problem for tensor recovery is to define the tensor rank. Unlike matrix rank, there are several ways to define a tensor rank. For example, Kolda and Bader (Kolda and Bader 2009) have adopt the minimum number of tensor rank-one decomposition (CP decomposition) of the given tensor as the rank of tensor (CP rank), which corresponds to one equivalent definition of matrix rank *i.e.* matrix rank of a matrix is equal to the minimum number of rank-one decomposition of the given matrix. Unfortunately, because the computing of CP rank is a NP-hard problem, the application of the CP rank in tensor recovery has been greatly restricted. In addition, due to the breakthroughs in low rank matrix recovery, the method based on Tucker decomposition (the unfolding matrices of the tensor) is more popular than the one based on CP rank. For example, in (Gandy, Recht, and Yamada 2011), the rank of the tensor (Tucker rank) is defined as the sum of the ranks of the different unfolding matrices. Besides, since the corresponding tensor rank minimization problem is NP-hard problem, Gandy *et al.* utilizes the sum of nuclear norms of the different unfolding matrices (SNN) instead of the sum of ranks for tensor recovery. However, as stated in (Lu et al. 2019), SNN is not the convex envelope of the sum of the ranks. Therefore, a weighted sum of the ranks of the unfolding matrices is considered in (Liu et al. 2012).

Recently, tensor recovery method based on tensor-tensor product (t-product) has received more and more attention because of its effectiveness (Hu et al. 2016; Zhang et al. 2014b). Based on t-product, tensor tubal rank is proposed, which utilizes tensor Singular Value Decomposition (t-SVD) based on t-product. Assuming $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ ($*$ stands for t-product) is the t-SVD of $\mathcal{A}$, tensor tubal rank of $\mathcal{A}$ is defined as the number of non-zero singular tubes of $\mathcal{S}$. Since tensor tubal rank is non-convex and discrete which leads to NP-hard problem. A convex norm, tensor nuclear norm (TNN) (Zhang et al. 2014b), was applied to solve the tensor completion problem which aims to recover a low rank tensor from tensor data with missing entries. Later, Lu *et al.* (Lu et al. 2019) proposed tensor average rank of tensor (corresponding to the rank of block circulant matrix of the tensor), and proved the tensor nuclear norm is the convex envelope of the tensor average rank within the unit ball of the tensor spectral norm.

* Corresponding author

Based on that, Tensor Robust Principal Component Analysis (TRPCA) problem with recovery guarantee is studied, which extends the Robust PCA (Wright et al. 2009) to the tensor case, and aims to exactly recover the low rank tensor from tensor data with gross corruptions. However, for tensor data with large scale size, tensor nuclear norm based method often costs much computation because of computing t-SVD. To alleviate this issue, low rank tensor factorization strategy based on t-product is proposed (Zhou et al. 2017), which factorizes tensor data into the product of two tensors with much smaller size and avoiding the computing of t-SVD of data tensor.

Although tensor recovery based on t-product is effective and wildly used, there are still some limitations: as shown in Fig.1, rearranging frontal slices sequence order of tensor will have significant influence on the effectiveness of tensor recovery, in which $\hat{\mathcal{X}}^*$ is obtained by arranging the low rank approximation of $\hat{\mathcal{Y}}$ ($\hat{\mathcal{Y}}$ is obtained by rearranging $\mathcal{Y}$ in randomly frontal slices sequence order) in original frontal slices sequence order. Note that the gap of two mean PSNR (Peak Signal to Noise Ratio) results even achieve 3dB. We called this phenomenon as Slice Permutations Variability (SPV) in tensor recovery.

This paper focuses on this new problem which has not been explored so far to the best of our knowledge. Our contributions are three-fold:

- We study SPV and Slice Permutations Invariance (SPI) of tensor recovery theoretically and experimentally for the first time. A tensor recovery algorithm has SPI, *i.e.* whatever how to change the slice order of data tensor, the solution of the algorithm will not be changed. We prove that the tensor recovery algorithm has SPI property under certain conditions.
- When the conditions are not met, to make tensor recovery more stable for slice permutations on data tensor, we propose a tensor recovery algorithm for SPV (TRSPV) to solve a basic problem (Tensor Principal Component Analysis) in tensor recovery. In the proposed algorithm, we find better sequence of tensor slice by solving a Minimum Hamiltonian Circle problem. Based on the new sequence obtained by the proposed algorithm, we can extract the intrinstic low-dimensional structure of high-dimensional tensor data more exactly.
- We conduct experiments to examine SPV of TRPCA, the goal of which is to recovery a low rank tensor from a high-dimensional data tensor with chaos slices sequence despite both small entry-wise noise and gross sparse errors. An extension of TRSPV, Robust Principal Component Analysis for SPV (TRPCA-SPV), is proposed to deal with this problem. The experimental results show a much better performance of TRPCA-SPV compared with the existing state-of-the-art tensor recovery algorithms, and a hug gap between the results by TRPCA-SPV and TRPCA.

Notations and Preliminaries

Notations

Here, we summarize some definitions and symbols used in this paper relating to matrices, tensor and sets in Table 1.

Table 1: Notations.

Notations	Descriptions	Notations	Descriptions
$\mathbb{R}$	real field	$\mathscr{A}$	sets
$\mathbb{C}$	complex field	$\lvert\mathscr{A}\rvert$	number of elements of $\mathscr{A}$
a	scalars	$\mathcal{A}$	tensors
A	matrices	$\mathcal{A}_{i,j,k}$	(i,j,k)-th element in $\mathcal{A}$
$A_{i,j}$	(i,j)-th element of matrix A	$\mathcal{A}_{i,j,:}$	(i,j)-th tube
$\sigma_i(\mathrm{A})$	i-th singular value of matrix A	$\mathcal{A}_{i,:,:}$	i-th horizontal slice
$\sigma(\mathrm{A})$	$(\sigma_1(\mathrm{A}),\sigma_2(\mathrm{A}),\ldots,\sigma_r(\mathrm{A}))^T$	$\mathcal{A}_{:,i,:}$	i-th lateral slice
rank(A)	rank function of A	$\mathcal{A}_{:,:,i}$	i-th frontal slice
$\Vert A\Vert_F$	$\sqrt{\sum_{i,j}A_{i,j}^2}$	$\Vert\mathcal{A}\Vert_1$	$\sum_{i,j,k}\lvert\mathcal{A}_{i,j,k}\rvert$
$\Vert A\Vert_*$	nuclear norm of A	$\Vert\mathcal{A}\Vert_F$	$\sqrt{\sum_{i,j,k}\mathcal{A}_{i,j,k}^2}$
A^T	conjugate transpose of A	$\bar{\mathcal{A}}$	$\mathrm{fft}(\mathcal{A},[],3)$
$A\longrightarrow B$	B can be obtained by elementary row or column transformations of A		
$(S-\tau)_+$	A matrix, each element of $(S-\tau)_+$ is $\max(S_{i,j}-\tau,0)$		

In addition, we follow the definitions of $\mathrm{unfold}(\cdot)$, $\mathrm{fold}(\cdot)$, $\mathrm{bcirc}(\cdot)$ and $\mathrm{bdiag}(\cdot)$ from (Lu et al. 2019):

$$\mathrm{unfold}(\mathcal{A})=\begin{pmatrix}\mathcal{A}_{:,:,1}\\ \mathcal{A}_{:,:,2}\\ \vdots\\ \mathcal{A}_{:,:,n_3}\end{pmatrix},\ \mathrm{fold}(\mathrm{unfold}(\mathcal{A}))=\mathcal{A},$$

$$\mathrm{bcirc}(\mathcal{A})=\begin{pmatrix}\mathcal{A}_{:,:,1} & \mathcal{A}_{:,:,n_3} & \cdots & \mathcal{A}_{:,:,2}\\ \mathcal{A}_{:,:,2} & \mathcal{A}_{:,:,1} & \cdots & \mathcal{A}_{:,:,3}\\ \vdots & \vdots & \ddots & \vdots\\ \mathcal{A}_{:,:,n_3} & \mathcal{A}_{:,:,n_3-1} & \cdots & \mathcal{A}_{:,:,1}\end{pmatrix},$$

$$\mathrm{bdiag}(\mathcal{A})=\begin{pmatrix}\mathcal{A}_{:,:,1} & & & \\ & \mathcal{A}_{:,:,2} & & \\ & & \ddots & \\ & & & \mathcal{A}_{:,:,n_3}\end{pmatrix}.$$

Preliminary definitions and results

Definition 1. *(t-product) (Kilmer and Martin 2011) Let $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{B}\in\mathbb{R}^{n_2\times l\times n_3}$. Then the* t*-product $\mathcal{A}*\mathcal{B}$ is defined to be a tensor of size $n_1\times l\times n_3$,*

$$\mathcal{A}*\mathcal{B}=\mathrm{fold}(\mathrm{bcirc}(\mathcal{A})\cdot\mathrm{unfold}(\mathcal{B})). \tag{1}$$

Definition 2. *(f-diagonal tensor) (Kilmer and Martin 2011) Tensor $\mathcal{A}$ is called f-diagonal if each of its frontal slices is a diagonal matrix.*

Definition 3. *(Identity tensor) (Kilmer and Martin 2011) The tensor $\mathcal{I}\in\mathbb{R}^{n\times n\times n_3}$ is the tensor with the first frontal slice being the identity matrix , and other frontal slices being all zeros.*

Definition 4. *(Conjugate transpose) (Lu et al. 2019) The conjugate transpose of a tensor $\mathcal{A}\in\mathbb{C}^{n_1\times n_2\times n_3}$ is the tensor $\mathcal{A}^T\in\mathbb{C}^{n_2\times n_1\times n_3}$ obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slice through positions 2 to n_3.*

Definition 5. *(Orthogonal tensor) (Kilmer and Martin 2011) A tensor $\mathcal{Q}\in\mathbb{C}^{n\times n\times n_3}$ is orthogonal if it satisfies $\mathcal{Q}^T*\mathcal{Q}=\mathcal{Q}*\mathcal{Q}^T=\mathcal{I}$.*

(a) (b) (c) (d)

Figure 1: Color video ('bus') (modeled as a tensor $\mathcal{Y} \in \mathbb{R}^{144\times176\times90}$) can be approximated by low tubal rank tensor . Here, only first frame of visual results in (a)-(b) are presented. (a) The first frame of original video (b) approximation by tensor $\mathcal{X}^* \in \mathbb{R}^{144\times176\times90}$ with tubal rank $r = 30$. (MPSNR=32.45dB) (c) approximation by tensor $\hat{\mathcal{X}}^* \in \mathbb{R}^{144\times176\times90}$ with tubal rank $r = 30$. (MPSNR=29.27dB) (d) MSE results of $\mathcal{X}^*$ and $\hat{\mathcal{X}}^*$ comparison for different r.

Theorem 1. *(t-SVD) (Lu et al. 2019) Let $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$. Then it can be factorized as $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{U} \in \mathbb{R}^{n_1\times n_1\times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2\times n_2\times n_3}$ are orthogonal, and $\mathcal{S} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a f-diagonal tensor.*

Definition 6. *(Tensor tubal rank) (Lu et al. 2019) For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the tensor tubal rank of $\mathcal{A}$, denoted by $\mathrm{rank_t}(\mathcal{A})$, is defined as the number of non-zero singular tubes of $\mathcal{S}$, where $\mathcal{S}$ is from the t-SVD of $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. We can write $\mathrm{rank_t}(\mathcal{A}) = |\{i|\mathcal{S}(i,i,:) \neq \boldsymbol{0}\}| = |\{i|\mathcal{S}(i,i,1) \neq 0\}|$. Denote $\sigma(\mathcal{S}) = (\mathcal{S}(1,1,1), \mathcal{S}(2,2,1), ..., \mathcal{S}(r,r,1))^T$,in which $r = \mathrm{rank}_t(\mathcal{A})$.*

Definition 7. *(Tensor nuclear norm) (Lu et al. 2019) Let $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ be the t-SVD of $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$. Tensor nuclear norm of $\mathcal{A}$ is defined as $\|\mathcal{A}\|_* = \langle \mathcal{S}, \mathcal{I}\rangle = \sum_{i=1}^{r} \mathcal{S}(i,i,1)$, where $r = \mathrm{rank}_t(\mathcal{A})$.*

Definition 8. *(Tensor average nuclear norm) (Lu et al. 2019) For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the tensor average nuclear norm is defined as $\|\mathcal{A}\|_{*,a} = \frac{1}{n_3}\|\mathrm{bcirc}(\mathcal{A})\|_*$.*

Definition 9. *(Zhang 2017) $P \in \mathbb{R}^{n\times n}$ is a permutation matrix if each row and each column of P has unique non-zero entries 1.*

Definition 10. *(Bondy, Murty et al. 1976) Let $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $\boldsymbol{C} = \{i_1, i_2, ..., i_{n_3}, i_1\}$ is a circle on $\mathcal{A}$ which composed of 1, 2, 3,..., n_3. And we regard $\{i_1, i_2, ..., i_{n_3}, i_1\}$, $\{i_2, i_3, ..., i_{n_3}, i_1, i_2\}, ..., \{i_{n_3}, i_1, ..., i_{n_3-2}, i_{n_3-1}, i_{n_3}\}$ as the same circle.*

Definition 11. *Let $\boldsymbol{C}_k = \{i_1, i_2, ..., i_{n_k}, i_1\}$ is a circle on $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ which composed of 1, 2, 3,..., n_k. And we call $\boldsymbol{Or}_k = \{i_1, i_2, ..., i_{n_k}\}$ is obtained an ordered array by $\boldsymbol{C}_k$. Define $\boldsymbol{Or}(i)$ is the i-th number of the ordered array, $(\mathcal{A} \circ \mathcal{P}^{(k)}_{\boldsymbol{Or}_k})(k = 1,2,3)$ are horizontal slice permutations, lateral slice permutations and frontal slice permutations of $\mathcal{A}$ according to $\boldsymbol{Or}_k$,* i.e. $(\mathcal{A} \circ \mathcal{P}^{(1)}_{\boldsymbol{Or}_1})_{i,:,:} = \mathcal{A}_{\boldsymbol{Or}_1(i),:,:}$, $(\mathcal{A} \circ \mathcal{P}^{(2)}_{\boldsymbol{Or}_2})_{:,i,:} = \mathcal{A}_{:,\boldsymbol{Or}_2(i),:}$ *and* $(\mathcal{A} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}_3})_{:,:,i} = \mathcal{A}_{:,:,\boldsymbol{Or}_3(i)}$ *for $i = 1, 2, 3, ..., n_k$. (If there is no danger of ambiguity, these are abbreviated to $(\mathcal{A} \circ \mathcal{P}^{(k)})(k = 1, 2, 3)$.)*

Definition 12. *Let $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $\boldsymbol{C} = \{i_1, i_2, ..., i_{n_3}, i_1\}$ is a circle on $\mathcal{A}$ which composed of 1, 2, 3,..., n_3. We call $\boldsymbol{C}(i_s, i_t) = \{i_s, i_{s+1}, ..., i_t\}$ as a walk from i_s to i_t on $\boldsymbol{C}$, and $\boldsymbol{C}^{-1}(i_s, i_t) = \{i_t, i_{t-1}, ..., i_s\}$ as inverse of walk $\boldsymbol{C}(i_s, i_t)$. Assume $\boldsymbol{C}(i_1, i_l) = \{i_1, i_2, ..., i_l\}$ and $\boldsymbol{C}(i_l, i_{l+k}) = \{i_l, i_{l+1}, ..., i_{l+k}\}$ are two walks on circle $\boldsymbol{C}$, mark $\boldsymbol{C}(i_1, i_l) \bigcup \boldsymbol{C}(i_l, i_{l+k}) = \{i_1, i_2, ..., i_l, i_{l+1}, ..., i_{l+k}\}$.*

Definition 13. *Let $\boldsymbol{C} = \{i_1, i_2, ..., i_{n_3}, i_1\}$ is a circle on $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ which composed of 1, 2, 3,..., n_3., and $W(\mathcal{A})$ is a weight matrix in which $W_{i,j}(\mathcal{A}) = \|\mathcal{A}_{:,:,i} - \mathcal{A}_{:,:,j}\|_F$ is weight of $\mathcal{A}_{:,:,i}$ and $\mathcal{A}_{:,:,j}$ for $i \neq j$, and $W_{i,j}(\mathcal{A}) = \infty$ for $i = j$. Mark $\boldsymbol{w}(\mathcal{A}, \boldsymbol{C}) = \sum_{k=1}^{n_3-1} W_{i_k,i_{k+1}}(\mathcal{A}) + W_{i_{n_3},i_1}(\mathcal{A})$, $\boldsymbol{C}^*(\mathcal{A}) = \arg\min_{\boldsymbol{C}} \boldsymbol{w}(\mathcal{A}, \boldsymbol{C})$ and $c^*(\mathcal{A}) = \min_{\boldsymbol{C}} \boldsymbol{w}(\mathcal{A}, \boldsymbol{C})$.*

SPI of tensor recovery

SPI of the sum of nuclear norms

For matrix recovery, as we all known, singular values of the matrix will not be affected by any row or column transformations on matrix, which means it does not make any influence on the effectiveness of matrix recovery to rearrange the data sequence. And we call it as row or column transformations invariance in matrix recovery (Property 1 and Theorem 2). Therefore, for tensor recovery based on the unfolding matrices of the tensor, SPV is satisfied naturally (Property 2 and Theorem 3). Please refer to the supplementary material of this paper for the detailed proof of these conclusions.

Property 1. *For $A \in \mathbb{R}^{n_1\times n_2}$, then nuclear norm satisfies row (or column) permutations invariance,* i.e. $\|PA\|_* = \|A\|_*$ *for any permutation matrix $P \in \mathbb{R}^{n_1\times n_1}$ (or $\|AP\|_* = \|A\|_*$ for any permutation matrix $P \in \mathbb{R}^{n_2\times n_2}$).*

Theorem 2. *For $Y \in \mathbb{R}^{n_1\times n_2}$, $\mathcal{D}_\tau(Y) = P^{-1}\mathcal{D}_\tau(PY)$ for any permutation matrix $P \in \mathbb{R}^{n_1\times n_1}$ (and $\mathcal{D}_\tau(Y) = \mathcal{D}_\tau(YP)P^{-1}$ for any permutation matrix $P \in \mathbb{R}^{n_2\times n_2}$), where $\mathcal{D}_\tau(Y) = \arg\min_X \frac{1}{2}\|Y - X\|_F^2 + \tau\|X\|_*$, and P^{-1} is inverse operator of P.*

Property 2. *For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, then $\sum_{i=1}^{3} \alpha_i \|(\mathcal{A} \circ \mathcal{P}^{(k)})_{(i)}\|_* = \sum_{i=1}^{3} \alpha_i \|\mathcal{A}_{(i)}\|_*$ for any slice permutations $\mathcal{P}^{(k)})_{(i)}$* i.e. $(k = 1, 2, 3)$, *where $\mathcal{A}_{(i)}$ represents the mode-i unfolding matrix of $\mathcal{A}$, $\mathcal{A} \circ \mathcal{P}^{(k)}(k = 1, 2, 3)$ stands for the result by perform horizontal slice permutations, lateral slice permutations and frontal slice permutations on $\mathcal{A}$, respectively.*

Theorem 3. *$\mathcal{S}_\tau(\mathcal{Y}) = \mathcal{S}_\tau(\mathcal{Y} \circ \mathcal{P}^{(k)}) \circ (\mathcal{P}^{(k)})^{-1} (k = 1, 2, 3)$, where $\mathcal{S}_\tau(\mathcal{Y}) = \arg\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \tau \sum_{i=1}^{3} \frac{1}{3}\|\mathcal{X}_{(i)}\|_*$, and $(\mathcal{P}^{(k)})^{-1}$ is an inverse operator of $\mathcal{P}^{(k)}$.*

SPI of tensor nuclear norm

In this part, we study the SPI of tensor nuclear norm. And we have the following conclusions. Please refer to the supplementary material of this paper for the detailed proof of these conclusions.

Property 3. *(Horizontal SPI of tensor nuclear norm) Tensor nuclear norm satisfies HSPI (Horizontal SPI),* i.e. *$\|\mathcal{A}\|_* = \|\mathcal{A} \circ \mathcal{P}^{(1)}\|_*$, for any horizontal slice permutations $\mathcal{P}^{(1)}$.*

Property 4. *(Lateral SPI of tensor nuclear norm) tensor nuclear norm satisfies LSPI (Lateral SPI),* i.e. *$\|\mathcal{A}\|_* = \|\mathcal{A} \circ \mathcal{P}^{(2)}\|_*$, for any lateral slices permutations $\mathcal{P}^{(2)}$.*

Property 5. *For same circle $\boldsymbol{C}^1 = \{i_1, i_2, ..., i_{n_3}, i_1\}$ and $\boldsymbol{C}^2 = \{i_k, i_{k+1}, ..., i_{n_3}, ..., i_{k-1}, i_k\}$,*

$$\|\mathcal{A} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}^1}\|_* = \|\mathcal{A} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}^2}\|_*,$$

where $\boldsymbol{Or}^1 = \{i_1, i_2, ..., i_{n_3}\}$ is obtained by $\boldsymbol{C}^1$, and $\boldsymbol{Or}^2 = \{i_k, i_{k+1}, ..., i_{n_3}, ..., i_{k-1}\}$ is obtained by $\boldsymbol{C}^2$.

The symbols and definitions used in Property 5 are explained in Definitions 10-11.

Theorem 4. *For same circle $\boldsymbol{C}^1 = \{i_1, i_2, ..., i_{n_3}, i_1\}$ and $\boldsymbol{C}^2 = \{i_k, i_{k+1}, ..., i_{n_3}, ..., i_{k-1}, i_k\}$,*

$$\mathcal{D}_\tau(\mathcal{Y} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}^1}) \circ \mathcal{P}^{(3)^{-1}}_{\boldsymbol{Or}^1} = \mathcal{D}_\tau(\mathcal{Y} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}^2}) \circ \mathcal{P}^{(3)^{-1}}_{\boldsymbol{Or}^2} \quad (2)$$

where $\mathcal{D}_\tau(\mathcal{A}) = \arg\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{A} - \mathcal{X}\|_F^2 + \tau\|\mathcal{X}\|_*$, $\boldsymbol{Or}^1 = \{i_1, i_2, ..., i_{n_3}\}$ is obtained by $\boldsymbol{C}^1$, and $\boldsymbol{Or}^2 = \{i_k, i_{k+1}, ..., i_{n_3}, ..., i_{k-1}\}$ is obtained by $\boldsymbol{C}^2$.

Property 6. *For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, if $n_3 \leq 3$, then tensor nuclear norm satisfies frontal slice permutations invariance (FSPI),* i.e. *$\|\mathcal{A}\|_* = \|\mathcal{A} \circ \mathcal{P}^{(3)}_{\boldsymbol{Or}}\|_*$ for any frontal slice permutations $\boldsymbol{P}^{(3)}_{\boldsymbol{Or}}$.*

Theorem 5. *For $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, if $n_3 \leq 3$, then*

$$\mathcal{D}_\tau(\mathcal{Y}) = \mathcal{D}_\tau(\mathcal{Y} \circ \mathcal{P}^{(k)}) \circ \mathcal{P}^{(k)^{-1}} \quad (3)$$

for $k = 1, 2, 3$.

Although, for $n_3 > 3$, we have taken an example which contradicts SPI of tensor recovery utilizing tensor-tensor product (see Fig. 1). By Theorem 5, it can be seen that tensor nuclear norm based tensor recovery satisfies slice permutations invariance for $n_3 \leq 3$.

Tensor recovery for SPV

In the following, we consider the case of $n_3 > 3$.

Tensor principal component analysis for SPV

Consider the following key problem:

$$\min_{\mathcal{X}, \mathcal{P}^{(3)}_{\mathbf{Or}}} \frac{1}{2}\|\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}} - \mathcal{X}\|_F^2 + \tau\|\mathcal{X}\|_*. \quad (4)$$

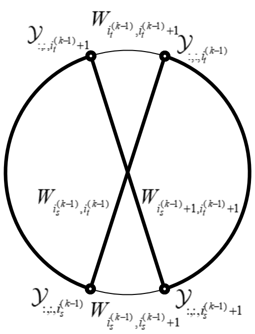


Figure 2

Since $\|\mathcal{X}\|_* = \|\mathcal{X}\|_{*,a}$ (Lu et al. 2019), therefore (4) can be converted to

$$\min_{\mathcal{X}, \mathcal{P}^{(3)}_{\mathbf{Or}}} \frac{1}{2}\|\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}} - \mathcal{X}\|_F^2 + \tau\|\mathcal{X}\|_{*,a} \quad (5)$$

$$= \min_{\mathcal{X}, \mathcal{P}^{(3)}_{\mathbf{Or}}} \frac{1}{2n_3}\|\mathrm{bcirc}(\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}}) - \mathrm{bcirc}(\mathcal{X})\|_F^2 + \frac{\tau}{n_3}\|\mathrm{bcirc}(\mathcal{X})\|_*. \quad (6)$$

From

$$\mathrm{bcirc}(\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}})$$
$$= \begin{pmatrix} \mathcal{Y}_{:,:,\mathbf{Or}(1)} & \mathcal{Y}_{:,:,\mathbf{Or}(n_3)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(2)} \\ \mathcal{Y}_{:,:,\mathbf{Or}(2)} & \mathcal{Y}_{:,:,\mathbf{Or}(1)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Y}_{:,:,\mathbf{Or}(n_3)} & \mathcal{Y}_{:,:,\mathbf{Or}(n_3-1)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(1)} \end{pmatrix}$$
$$\longrightarrow \begin{pmatrix} \mathcal{Y}_{:,:,\mathbf{Or}(1)} & \mathcal{Y}_{:,:,\mathbf{Or}(2)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(n_3)} \\ \mathcal{Y}_{:,:,\mathbf{Or}(2)} & \mathcal{Y}_{:,:,\mathbf{Or}(3)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Y}_{:,:,\mathbf{Or}(n_3)} & \mathcal{Y}_{:,:,\mathbf{Or}(1)} & \cdots & \mathcal{Y}_{:,:,\mathbf{Or}(n_3-1)} \end{pmatrix},$$

it can be seen that $\mathrm{bcirc}(\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}})$ will be approximated to a more lower rank matrix and get a better low rank eastimation of $\mathcal{Y}$ when adjacent $\mathcal{Y}_{:,:,\mathbf{Or}(i)}$ and $\mathcal{Y}_{:,:,\mathbf{Or}(i+1)}$ are more similar (mark $\mathcal{Y}_{:,:,\mathbf{Or}(n_3+1)} = \mathcal{Y}_{:,:,\mathbf{Or}(1)}$ for convenience). Therefore, we convert (4) to the following problem:

$$\arg\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} \circ \mathcal{P}^{(3)}_{\mathbf{Or}^*} - \mathcal{X}\|_F^2 + \tau\|\mathcal{X}\|_*, \quad (7)$$

where $\mathbf{Or}^*$ is obtained by $\mathbf{C}^*(\mathcal{Y})$. Therefore we solve (4) via Algorithm 2 approximately by Theorem 6[1]. The symbols and definitions used in Algorithm 2 are explained in Definitions 12-13.

Theorem 6. *(Lu et al. 2019) Tensor nuclear minimize problem:*

$$\mathcal{D}_\tau(\mathcal{Y}) = \arg\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \tau\|\mathcal{X}\|_*, \quad (8)$$

where $\mathcal{D}_\tau(\mathcal{Y})$ can be obtained by Algorithm 1.

[1] It is worth noting that we convert (4) to a Minimum Hamiltonian circle problem.

Algorithm 1: Tensor Singular Value Thresholding (t-SVT)

Input: $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\tau > 0$ as defined in (8).
Output: $\mathcal{D}_\tau(\mathcal{Y})$.
Compute $\bar{\mathcal{Y}} = \text{fft}(\mathcal{Y}, [], 3)$;
Perform matrix SVT on each frontal slice of $\bar{\mathcal{Y}}$ by
for $i = 1, ..., \lfloor \frac{n_3+1}{2} \rfloor$ **do**
  $[U, S, V] = \text{SVD}(\bar{Y}^{(i)})$;
  $\bar{W}^{(i)} = U(S - \tau)_+ V^*$;
end
for $i = \lfloor \frac{n_3+1}{2} \rfloor + 1, \cdots, n_3$ **do**
  $\bar{W}^{(i)} = conj(\bar{W}^{(n_3-i+2)})$;
end
$\mathcal{D}_\tau(\mathcal{Y}) = \text{ifft}(\bar{\mathcal{W}}, [], 3).$;

A key point to $\mathcal{T}_\tau(\mathcal{Y})$ is to find $\mathbf{C}^*(\mathcal{Y})$. And a simplest idea for getting $\mathbf{C}^*(\mathcal{Y})$ is that, when we get $\mathbf{C}^{(k-1)}$, we can make appropriate modifications for the circle $\mathbf{C}^{(k)}$ to get another circle $\mathbf{C}^{(k)}$ with a smaller $\mathbf{w}(\mathcal{Y}, \mathbf{C}^{(k)})$ as Fig. 2 (Bondy, Murty et al. 1976). Repeat the above process until $\mathbf{C}^{(k)}$ convergence to $\mathbf{C}^*(\mathcal{Y})$.

TRPCA for SPV

Consider the following problem:

$$(\mathcal{L}^*, \mathcal{S}^*, \mathcal{P}^{(3)}_{\mathbf{Or}^*}) = \min_{\mathcal{L}, \mathcal{S}, \mathcal{P}^{(3)}_{\mathbf{Or}}} \|\mathcal{L}\|_* + \lambda \|\mathcal{S}\|_1$$
$$s.t. (\mathcal{P} - \mathcal{S}) \circ \mathcal{P}^{(3)}_{\mathbf{Or}} = \mathcal{L}, \quad (9)$$

where $\mathcal{P}^{(3)}_{\mathbf{Or}}$ is a frontal slice permutation, $\mathcal{L}$ is low-rank, and $\mathcal{S}$ is sparse. And Algorithm 3 based on alternating direction method (ADM) (Bertsekas 1997) is proposed for solving (9). It is worth noting that, for fixed $\mathcal{P}^{(3)}_{\mathbf{Or}}$, (9) degenerate to TRPCA (which means (9) can exactly recover the low-rank and sparse components from their sum for the fixed $\mathcal{P}^{(3)}_{\mathbf{Or}}$.).

Experimental results

This section include three parts: in the first two parts, we compare the proposed algorithm (TRPCA-SPV) with several existing state-of-the-art tensor recovery methods (including RPCA[2](Candès et al. 2011), SNN[3](Gandy, Recht, and Yamada 2011), Liu's work [3](called Liu for short)(Candes and Plan 2010) and TRPCA[3] (Lu et al. 2019)) on image sequence recovery task and image classification task to evaluate the effectiveness of the algorithms to alleviate SPV problem on tensor recovery. And the third part is conducted in order to evaluate the performance of TRPCA-SPV with different values of the parameter κ.

[2] https://github.com/dlaptev/RobustPCA
[3] https://github.com/canyilu/LibADMM-toolbox

Algorithm 2: Tensor recovery for SPV (TRSPV)

Input: $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and Iternum.
Output: $\mathbf{C}^*(\mathcal{Y})$ and $\mathcal{T}_\tau(\mathcal{Y})$
Compute weight matrix W;
Initialize circle $\mathbf{C}^{(0)} = \{i_1^{(0)}, i_2^{(0)}, ..., i_{n_3}^{(0)}, i_1^{(0)}\}$, and $k = 0$;
while $k \leq$ Iternum **do**
  $k = k + 1$;
  if *there are different* $i_s^{(k-1)}, i_t^{(k-1)}, i_s^{(k-1)} + 1, i_t^{(k-1)} + 1$ *in* $\mathbf{C}^{(k-1)}$ *which make* $W_{i_s^{(k-1)}, i_t^{(k-1)}}(\mathcal{Y}) + W_{i_s^{(k-1)}+1, i_t^{(k-1)}+1}(\mathcal{Y}) < W_{i_s^{(k-1)}, i_s^{(k-1)}+1}(\mathcal{Y}) + W_{i_t^{(k-1)}, i_t^{(k-1)}+1}(\mathcal{Y})$ **then**
    $\mathbf{C}^{(k)} = \{i_t^{(k-1)}, i_s^{(k-1)}\} \bigcup \mathbf{C}^{(k-1)^{-1}}(i_{t+1}^{(k-1)}, i_s^{(k-1)}) \bigcup \{i_{t+1}^{(k-1)}, i_{s+1}^{(k-1)}\} \bigcup \mathbf{C}^{(k-1)}(i_{s+1}^{(k-1)}, i_t^{(k-1)})$;
  else
    $\mathbf{C}^{(k)} = \mathbf{C}^{(k-1)}$;
    break;
  end
end
Obtain $\mathbf{C}^*(\mathcal{Y}) = \mathbf{C}^{(k)}$, and compute $\mathcal{T}_\tau(\mathcal{Y}) = \mathcal{D}_\tau(\mathcal{Y}^{\mathbf{Or}^*})$, where $\mathbf{Or}^*$ obtained by $\mathbf{C}^*(\mathcal{Y})$;

Image sequence recovery

In this part, all five methods are tested on two hyperspectral image databases including Pavia University [4] and Botswana[4].

Each image with dimension of $n_1 \times n_2$ is contaminated by the mixture of zero mean Gaussian noise and random valued impulse noise, in which standard deviations of zero mean Gaussian δ is set as $\delta = 5 : 10 : 25$ and random-valued impulse noise with density level c is set as $c = 0.05 : 0.1 : 0.25$. For Pavia University, we empirically set $\lambda = 1/\sqrt{\max(n_1, n_2)}$ for RPCA (which deal with each band separately), $\lambda = [\frac{240}{3}, \frac{240}{3}, \frac{240}{3}]$ for SNN, and $\lambda = 330 \times [0.2, 0.1, 0.7]$ for Liu. For Botswana, we empirically set $\lambda = 0.9/\sqrt{\max(n_1, n_2)}$ for RPCA (which deal with each band separately), $\lambda = [\frac{340}{3}, \frac{340}{3}, \frac{340}{3}]$ for SNN, and $\lambda = 370 \times [0.3, 0.1, 0.6]$ for Liu. For TRPCA, the parameter λ is tuned to $\lambda = 0.9/\sqrt{\max(n_1, n_2)n_3}$ and $\lambda = 0.8/\sqrt{\max(n_1, n_2)n_3}$ for Pavia University and Botswana respectively, in which n_3 is the number of spectral bands. For TRPCA-SPV, the parameter λ is tuned to $\lambda = 0.9/\sqrt{\max(n_1, n_2)n_3}$ for the two databases.

The Mean Peak Signal-To-Noise Ratio (MPSNR) value $\frac{1}{n_3}\sum_{i=1}^{n_3} \text{PSNR}_i$ is used to evaluate the methods, where PSNR_i is the Peak Signal-To-Noise Ratio (PSNR) result of i-th restored band. From Table 2, there are some observations as following: TRPCA-SPV outperforms the compared methods by a wide margin in most of cases. Specifically, for Pavia University, TRPCA-SPV outperforms other methods

[4] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensin_Scenes

Algorithm 3: TRPCA for SPV (TRPCA-SPV)

Initialize: $\mathcal{L}^{(0)} = \mathcal{S}^{(0)} = \mathcal{Q}^{(0)} = \mathcal{Y}^{(0)} = \mathbf{0}$, $\rho > 1$, $\mu_0 = 1e-3$, $\epsilon = 1e-8$, $\kappa > 0$.

while *not converged* **do**

1. Update $\mathbf{Or}^*$ by
If $\kappa = 1$ or $k \mod \kappa = 1$, update $\mathbf{Or}^*$ by $\mathbf{C}^*(\mathcal{M}^{(k)})$, where $\mathcal{M}^{(k)} = \mathcal{P} - \mathcal{S}^{(k)} - \frac{\mathcal{Q}^{(k)}}{\mu_k}$;

2.Update $\mathcal{L}^{(k+1)}$ by
$\mathcal{L}^{(k+1)} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_* + \frac{\mu_k}{2}\|\mathcal{L} - (\mathcal{M}^{(k)})^{\mathbf{Or}^*}\|_F^2$;

3. Update $\mathcal{S}^{(k+1)}$ by
$\mathcal{S}^{(k+1)} = \arg\min_{\mathcal{S}} \lambda\|\mathcal{S}^{\mathbf{Or}^*}\|_1 + \frac{\mu_k}{2}\|\mathcal{L}^{(k+1)} + \mathcal{S}^{\mathbf{Or}^*} - \mathcal{P}^{\mathbf{Or}^*} + (\frac{\mathcal{Q}^{(k)}}{\mu_k})^{\mathbf{Or}^*}\|_F^2$;

4.$(\mathcal{Q}^{(k+1)})^{\mathbf{Or}^*}$=$(\mathcal{Q}^{(k)})^{\mathbf{Or}^*} + \mu(\mathcal{L}^{(k+1)} + (\mathcal{S}^{(k+1)})^{\mathbf{Or}^*} - \mathcal{P}^{\mathbf{Or}^*})$;

5.Update μ_{k+1} by $\mu_{k+1} = \min(\rho\mu_k, \mu_{\max})$;

6. Check the convergence conditions
$\|\mathcal{L}^{(k+1)} - \mathcal{L}^{(k)}\|_\infty \leq \epsilon$,
$\|(\mathcal{S}^{(k+1)})^{\mathbf{Or}^*} - (\mathcal{S}^{(k)})^{\mathbf{Or}^*}\|_\infty \leq \epsilon$,
$\|\mathcal{L}^{(k+1)} + (\mathcal{S}^{(k+1)})^{\mathbf{Or}^*} - \mathcal{P}^{\mathbf{Or}^*}\|_\infty \leq \epsilon$;

end

by more than 3 dB on the case of small noise level. This demonstrates the superiority of our TRPCA-SPV in tensor recovery. For case of TRPCA-SPV v.s. TRPCA, TRPCA-SPV can attain much better results compared to TRPCA. The gap between MPSNR results by TRPCA-SPV and TRPCA even achieve 5dB in the case of $\delta = 5$ and $c = 0.05 : 0.1 : 0.25$. This illustrate the huge affecting of SPV on TRPCA, and TRPCA-SPV can eliminate it well.

Image classification

In this part, image classification is conducted on two datasets including ORL database[5] and CMU PIE database [6].

Each image with size of $n_1 \times n_2$ is contaminated by the mixed noise, in which δ is set as $\delta = 0 : 5 : 30$ and c is set as $c = 0 : 0.05 : 0.3$. For each noise level, all five algorithms are used to recover the low rank tensor structure from the noised images. The performance of the algorithms is evaluated by classification accuracy via k nearest neighbor (kNN), where $k = 1$ in the experiments. For each dataset, 90% of samples are randomly selected as training set, and the rest are taken as testing set. The experiments are repeated 10 times, and the average values of accuracy of all methods are reported in Fig. 3-4. For RPCA and TRPCA, the parameter λ is set to $\lambda = 1/\sqrt{\max(n_1 n_2, n_3)}$ and $\lambda = 1/\sqrt{\max(n_1, n_2) n_3}$ respectively as suggested in (Lu et al. 2019), in which n_3 is the number of samples. For TRPCA-SPV, the parameter λ is set to $\lambda = 1/\sqrt{\max(n_1, n_2) n_3}$ as well. For Liu, we find that it does not perform well when λ_i's are set to the values suggested in theory (Huang et al. 2015). We empirically set it as $70 \times [0.2, 0.3, 0.5]$. For SNN, we empirically set $\lambda = [\frac{70}{3}, \frac{70}{3}, \frac{70}{3}]$. All results are presented in Fig. 3-4. The cell with more dark red corresponds to higher classification accuracy.

[5] https://cam-orl.co.uk/facedatabase.html

[6] https://www.ri.cmu.edu/project/pie-database/

From Fig. 3-4, there are some observations as following: In general, TRPCA-SPV achieves more stable and better performance compared to other methods (RPCA, SNN, Liu and TRPCA). In addition, TRPCA-SPV can attain better results compared to TRPCA, because TRPCA-SPV exploits the low rank structure within the tensor data more exactly.

Sensitivity analysis of parameters

In this part, an experiment is conducted with two datasets (including ORL database and Pavia University), in which each image in datasets contaminated by the mixed noise with $\delta = 15$ and $c = 0.15$, to investigate the influence of the parameter κ.

The experiments for each parameter κ are repeated 10 times, the results obtained by the different methods are shown in Fig. 5 (a) and (b), from which we have the following observations: (1) In general, the results by TRPCA-SPV are robust against to the parameter κ. (2) For all cases of TRPCA-SPV, the results by TRPCA-SPV are much better than TRPCA.

In addition, from Fig. 5 (c) and (d), the curve by TRPCA-SPV is shock depend on the parameter κ at the begin, and tend to stability with more iterations of the algorithm, in which

$$\begin{aligned}\text{Error} = &\max(\|\mathcal{L}^{(k+1)} - \mathcal{L}^{(k)}\|_\infty, \|\mathcal{S}^{(k+1)} - \mathcal{S}^{(k)}\|_\infty, \\ &\|\mathcal{L}^{(k+1)} + (\mathcal{S}^{(k+1)})^{\mathbf{Or}^*} - \mathcal{P}^{\mathbf{Or}^*}\|_\infty). \qquad (10)\end{aligned}$$

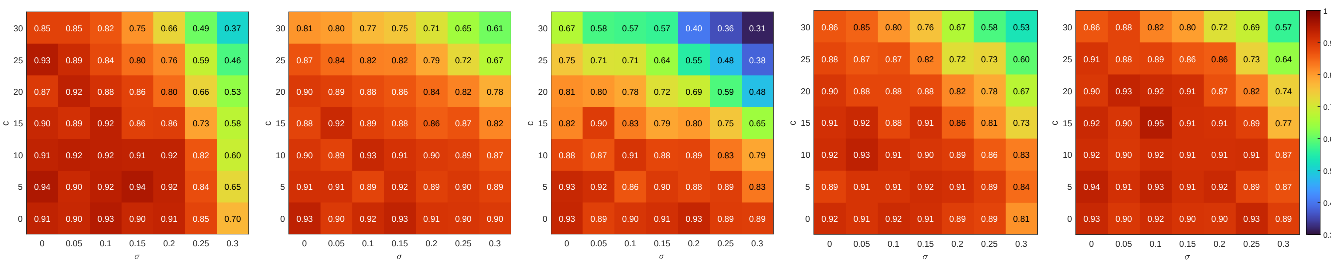
Conclusion and Future Work

This paper focuses on solving a new problem (SPV in tensor recovery) which has not been explored so far. We aim to accurately recover a low rank tensor from a high-dimensional tensor data with chaos tensor slices sequence. The example given in Figure 1 shows a huge gap between results by tensor recovery using tensor with different slices sequence. To deal with this issue, TRSPV is proposed. Furthermore, we discuss SPV of several key tensor recovery problems in theoretically. To this end, we first study the row (or column) permutations invariance of a key low rank matrix recovery problem (Principal Component Analysis). Then, SPI of several key tensor recovery problems are discussed in theoretically, and we get the following results: (1) Tensor recovery based on the weighted sum of the nuclear norm of the unfolding matrices has SPI. (2) For $n_3 \leq 3$, tensor recovery based on tensor nuclear norm has SPI. For the case of $n_3 > 3$, experimental results shows the effectiveness of the proposed algorithm, and eliminate SPV in tensor recovery well.

Although, tensor recovery based on t-product usually get a significant performance compared with other tensor recovery methods. But it can not be applied in higher order tensor recovery in straightway. Consider the real data such as color video are in higher order tensor form. It is interesting to develop an effective higher order tensor algorithm using the idea of tensor-tensor product.

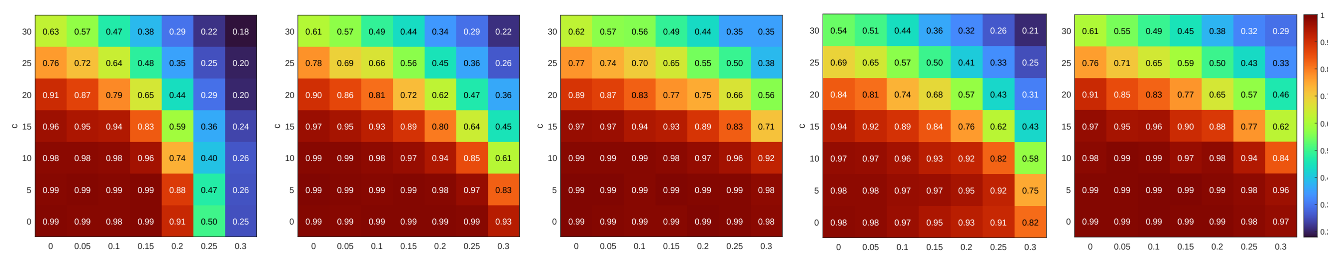
		Botswana					Pavia University				
δ	c	RPCA	SNN	Liu	TRPCA	TRPCA-SPV	RPCA	SNN	Liu	TRPCA	TRPCA-SPV
5	5%	29.90	34.52	36.82	32.06	**38.44**	27.56	29.82	32.03	30.65	**36.60**
	15%	29.04	33.02	35.34	30.06	**37.11**	26.90	29.21	31.60	28.07	**35.39**
	25%	27.73	30.81	32.92	28.78	**34.98**	25.55	27.96	30.53	26.03	**33.48**
15	5%	28.11	30.91	32.42	31.11	**34.21**	25.58	27.19	28.07	30.22	**31.51**
	15%	27.32	29.47	30.92	28.99	**32.34**	24.77	26.43	28.35	27.17	**30.38**
	25%	25.78	27.23	28.48	27.18	**29.67**	23.20	24.96	26.99	24.67	**27.76**
25	5%	26.84	29.17	30.37	29.34	**31.65**	23.63	25.12	26.94	28.50	**29.02**
	15%	26.05	27.55	28.67	26.83	**29.77**	22.74	24.30	26.30	25.21	**27.49**
	25%	24.29	25.14	26.06	24.18	**26.79**	21.11	22.76	24.77	22.49	**24.81**

Table 2: MPSNR results by different methods on Botswana and Pavia University.



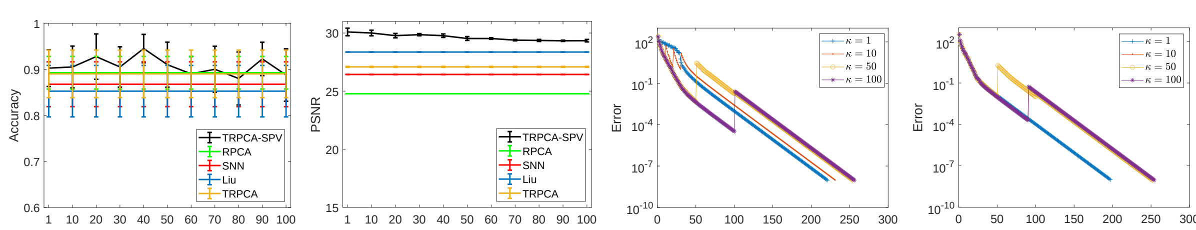
(a) (b) (c) (d) (e)

Figure 3: Classification accuracies of the 5 algorithms on ORL database: (a) RPCA (b) SNN (c) Liu (d) TRPCA (e) TRPCA-SPV



(a) (b) (c) (d) (e)

Figure 4: Classification accuracy result on CMU PIE database: (a) RPCA (b) SNN (c) Liu (d) TRPCA (e) TRPCA-SPV



(a) (b) (c) (d)

Figure 5: Sensitivity analysis of parameter κ for TRPCA-SPV on (a) ORL database and (b) Pavia University; Convergence analysis for TRPCA-SPV with different κ on (c) ORL database and (d) Pavia University.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China [grant nos. 61922064, U2033210, 62101387], in part by the Zhejiang Provincial Natural Science Foundation [grant nos. LR17F030001, LQ19F020005], and in part by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery [grant number RGPIN-2020-05525].

References

Bertsekas, D. P. 1997. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334.

Bondy, J. A.; Murty, U. S. R.; et al. 1976. *Graph theory with applications*, volume 290. Macmillan London.

Cai, C.; Li, G.; Poor, H. V.; and Chen, Y. 2021. Nonconvex low-rank tensor completion from noisy data. *Operations Research*.

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37.

Candes, E. J.; and Plan, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936.

Candès, E. J.; and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mmathematics*, 9(6): 717–772.

Chandrasekaran, V.; Sanghavi, S.; Parrilo, P. A.; and Willsky, A. S. 2009. Sparse and low-rank matrix decompositions. *IFAC Proceedings Volumes*, 42(10): 1493–1498.

Dian, R.; Li, S.; and Fang, L. 2019. Learning a low tensor-train rank representation for hyperspectral image super-resolution. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9): 2672–2683.

Eckart, C.; and Young, G. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218.

Gandy, S.; Recht, B.; and Yamada, I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2): 025010.

Hu, W.; Tao, D.; Zhang, W.; Xie, Y.; and Yang, Y. 2016. The twist tensor nuclear norm for video completion. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12): 2961–2973.

Huang, B.; Mu, C.; Goldfarb, D.; and Wright, J. 2015. Provable models for robust low-rank tensor completion. *Pacific Journal of Optimization*, 11(2): 339–364.

Kilmer, M. E.; and Martin, C. D. 2011. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3): 641–658.

Kolda, T. G.; and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM Review*, 51(3): 455–500.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2012. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 208–220.

Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; and Yan, S. 2019. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4): 925–938.

Lu, C.; Peng, X.; and Wei, Y. 2019. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5996–6004.

Luo, Y.; Tao, D.; Ramamohanarao, K.; Xu, C.; and Wen, Y. 2015. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE transactions on Knowledge and Data Engineering*, 27(11): 3111–3124.

Tan, H.; Cheng, B.; Wang, W.; Zhang, Y.-J.; and Ran, B. 2014. Tensor completion via a multi-linear low-n-rank factorization model. *Neurocomputing*, 133: 161–169.

Wei, W.; Zhang, L.; Jiao, Y.; Tian, C.; Wang, C.; and Zhang, Y. 2018. Intracluster structured low-rank matrix analysis method for hyperspectral denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2): 866–880.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics & Intelligent Laboratory Systems*, 2(1): 37–52.

Wright, J.; Ganesh, A.; Rao, S. R.; Peng, Y.; and Ma, Y. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Neural Information Processing Systems*, volume 58, 289–298.

Xu, H.; Caramanis, C.; and Sanghavi, S. 2012. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5): 3047–3064.

Yang, J.-H.; Zhao, X.-L.; Ji, T.-Y.; Ma, T.-H.; and Huang, T.-Z. 2020. Low-rank tensor train for tensor robust principal component analysis. *Applied Mathematics and Computation*, 367: 124783.

Zhang, F.; Wang, J.; Wang, W.; and Xu, C. 2021. Low-tubal-rank plus sparse tensor recovery with prior subspace information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3492–3507.

Zhang, X.; Wang, D.; Zhou, Z.; and Ma, Y. 2019. Robust low-rank tensor recovery with rectification and alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1): 238–255.

Zhang, X.; Zhou, Z.; Wang, D.; and Ma, Y. 2014a. Hybrid singular value thresholding for tensor completion. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Zhang, X.-D. 2017. *Matrix analysis and applications*. Cambridge University Press.

Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; and Kilmer, M. 2014b. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3842–3849.

Zheng, Y.-B.; Huang, T.-Z.; Ji, T.-Y.; Zhao, X.-L.; Jiang, T.-X.; and Ma, T.-H. 2019. Low-rank tensor completion via smooth matrix factorization. *Applied Mathematical Modelling*, 70: 677–695.

Zhou, P.; Lu, C.; Lin, Z.; and Zhang, C. 2017. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3): 1152–1163.

Zhou, Z.; Li, X.; Wright, J.; Candes, E.; and Ma, Y. 2010. Stable principal component pursuit. In *2010 IEEE International Symposium on Information Theory*, 1518–1522.