Open Access Library Journal
2022, Volume 9, e9412
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Tensor Robust Principal Component Analysis via Hybrid Truncation Norm

Yujie Luan[1], Wei Jiang[1,2]

[1]School of Mathematics, Liaoning Normal University, Dalian, China
[2]College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China
Email: lyj13154220661@163.com

How to cite this paper: Luan, Y.J. and Jiang, W. (2022) Tensor Robust Principal Component Analysis via Hybrid Truncation Norm. *Open Access Library Journal*, **9**: e9412.
https://doi.org/10.4236/oalib.1109412

Received: October 1, 2022
Accepted: October 21, 2022
Published: October 24, 2022

Abstract

This paper mainly studies the problem of tensor robust principal component analysis (TRPCA), in order to accurately recover the low rank and sparse components from the observed signals. Most of the existing robust principal component analysis (RPCA) methods are based on nuclear norm minimizationss. These methods minimize all singular values at the same time, so they can not approach the rank well in practice. In this paper, the idea of truncated nuclear norm regularization is extended to RPCA. At the same time, in order to improve the stability of the model, the tensor truncated Frobenius norm is newly defined. Truncated nuclear norm and truncated Frobenius norm are considered at the same time called hybrid truncated model of tensor. This method minimizes min $(m,n)-r$ singular values. In addition, this paper also gives an effective method to determine the contraction operator, and develops an effective iterative algorithm based on alternating direction to solve this optimization problem. Experimental results show the effectiveness and accuracy of this method.

Subject Areas

Machine Learning

Keywords

Tensor Robust Principal Component Analysis (TRPCA), Hybrid Truncation Norm , Alternating Direction Multiplier Method (ADMM)

1. Introduction

Tensor is a multidimensional extension of matrix and an important data format, which can express the internal structure of more complex high-order data. In fact, tensor is also the natural form of high-dimensional and multi-way real-world data. For example, in the field of image processing [1], a color image is a 3-order tensor of *height* × *weight* × *channel* and a multispectral image is a 3-order tensor of *height* × *weight* × *channel*. Therefore, tensor analysis has important practical significance and application value in the fields of machine learning [2], computer vision [3], data mining [4], and so on. The tensor, we interest, is frequently low-rank, or approximately so [5] and hence has a much lower-dimensional structure. This has stimulated the problem of low rank tensor estimation and recovery, which has attracted great attention in many different fields. The classical principal component analysis (PCA) [6] is the most widely used method for data analysis and dimensionality reduction. For the data slightly damaged by small noise, it has the characteristics of high computational efficiency and powerful function. However, a major problem with principal component analysis is that it is easily affected by seriously damaged or bizarre observations, which are ubiquitous in real-world data. So far, many principal component analysis models [7] [8] have been proposed, but almost all of them have high computational cost.

RPCA is the first polynomial time algorithm with strong performance guarantee. Suppose that we are given an observed $\mathbf{X} \in \mathbb{R}^{m \times n}$, which can be decomposed as $\mathbf{X} = \mathbf{L}_0 + \mathbf{E}_0$, where $\mathbf{L}_0$ is low-rank and $\mathbf{E}_0$ is sparse. [6] shows that if the singular vector of $\mathbf{L}_0$ satisfies some incoherent conditions, such as $\mathbf{L}_0$ is low rank and $\mathbf{E}_0$ is sparse enough, $\mathbf{L}_0$ and $\mathbf{E}_0$ can be recovered accurately with high probability by solving the convex problem (1):

$$\min_{\mathbf{L},\mathbf{E}} \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 \tag{1}$$

where $\|\mathbf{L}\|_*$ denotes the nuclear norm (sum of the singular values of $\mathbf{L}$), and $\|\mathbf{E}\|_1$ denotes the ℓ_1-norm (sum of the absolute values of all the entries of $\mathbf{E}$). The parameter λ is suggested to be set as $1/\sqrt{\max(n_1, n_2)}$ which works well in practice. In terms of algorithm, (1) can be solved by efficient algorithm, and the cost will not be much higher than PCA. This method and its generalization have been successfully applied to the fields of background modeling [9], subspace clustering [10], video compression sensing [11] and so on.

A major disadvantage of RPCA is that it can only process two-dimensional (matrix) data. However, real data is usually multi-dimensional in nature-the information is stored in multi-way arrays known as tensors [5]. To use RPCA, we must first convert the high-dimensional data into a matrix. However, such preprocessing usually leads to information loss, resulting in performance degradation. In order to alleviate this problem, a common method is to use the multidimensional structure of tensor data to deal with it.

In this paper, tensor robust principal component analysis (TRPCA) is studied in order to accurately recover the low rank tensor damaged by sparse noise. Suppose that we are given an observed $\mathcal{X}$, which can be decomposed as $\mathcal{X} = \mathcal{L}_0 + \mathcal{E}_0$, where $\mathcal{L}_0$ is low-rank and $\mathcal{E}_0$ is sparse, and both components are of arbitrary magnitudes. Note that we do not know the locations of the nonzero elements of $\mathcal{E}_0$, not even how many there are. Now we consider a similar problem to RPCA. This is the problem of tensor RPCA studied in this work.

It is not easy to generalize RPCA to tensor. The main problem of low rank tensor estimation is the definition of tensor rank [12]. At present, several definitions of tensor rank and its convex relaxation have been proposed, but each rank has its limitations. The CANDECOMP/PARAFAC (CP) rank of tensor [5] is defined as the minimum number of tensor rank 1 decomposition, which is NP-hard to calculate. Because of its unclear convex envelope, the recovery of low CP rank tensor is challenging [13]. A robust tensor CP decomposition problem is studied. Although the recovery is guaranteed, the algorithm is nonconvex. To avoid this problem, Tucker *et al.* presented the tractable Tucker rank [14], and its convex relaxation has also been widely used. For a k-way tensor $\mathcal{X}$, The Tucker rank is a vector defined as $rank_{tc}(\mathcal{X}) := \left(rank\left(\mathbf{X}^{\{1\}}\right), rank\left(\mathbf{X}^{\{2\}}\right), \cdots, rank\left(\mathbf{X}^{\{k\}}\right)\right)$. Motivated by the fact that the nuclear norm is the convex envelope of the matrix rank within the unit ball of the spectral norm, the Sum of Nuclear Norms (SNN) [15], defined as $\sum_i \left\|\mathbf{X}^{(i)}\right\|_*$, is used as a convex surrogate of $\sum_i rank\left(\mathbf{X}^{\{i\}}\right)$. [16] gives a TRPCA model based on SNN:

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} \sum_{i=1}^{k} \lambda_i \left\|\mathbf{L}^{\{i\}}\right\|_* + \|\mathbf{E}\|_1 \\ &\text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}. \end{aligned} \tag{2}$$

[17] fully studied the effectiveness of this method. However, SNN is still not the tightest convex relaxation of Tucker rank. In other words, the model (2) is basically suboptimal. Recently, with the introduction of tensor product (T-product) [18] and tensor singular value decomposition (T-SVD), Kilmer *et al.* [15] proposed the definitions of tensor multi rank and tubal rank. Then a new tensor nuclear norm appears and is applied to tensor completion [19] and tensor robust principal component analysis [20]. In [21], the author realized that tensor product is based on convolution like operation, which can be realized by using discrete Fourier transform (DFT). Soon after, a more general definition of tensor product was proposed, which was based on arbitrary reversible linear transformation. Lu *et al.* [22] accurately recover the low rank and sparse components by solving a weighted combined convex programming of tensor nuclear norm and tensor ℓ_1-norm that is:

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \|\mathcal{E}\|_1 \\ &\text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}. \end{aligned} \tag{3}$$

However, due to the nuclear norm considers each singular value of the tensor, and each singular value has different influence on the results. Therefore, [23] proposed the partial sum of tensor tube nuclear norm (PSTNN), and established a minimization model based on PSTNN to solve the tensor RPCA problem. Although the truncated nuclear norm can be used as a convex alternative to the rank function, the model lacks strong stability. The high stability of the model means that the elements of the output data set do not change significantly. [24] proposed a hybrid model of truncated nuclear norm and truncated Frobenius norm for matrix completion. In this model, nuclear norm controls low rank attribute and Frobenius norm controls stability.

Inspired by [24], we propose a new tensor robust principal component analysis method, trying to establish a more stable and ideal model. The main contributions are summarized as follows.

1) We propose two new regularization terms, tensor truncated nuclear norm (T-TNN) and tensor truncated Frobenius norm (T-TFN). Respectively, T-TNN is defined as the sum of the last min $(m,n)-r$ singular values of all frontal slices of $\bar{\boldsymbol{\mathcal{A}}}$. T-TFN is defined as the square root of the square sum of the last min $(m,n)-r$ singular values of all frontal slices of $\bar{\boldsymbol{\mathcal{A}}}$. Based on these two new definitions, a tensor hybrid truncated norm (T-HTN) regularization model is proposed, which uses the combination of T-TNN and T-TFN to achieve tensor TRPCA. The model not only improves the stability, but also effectively improves the accuracy of recovery accuracy.

2) A simple two-step iterative algorithm is designed to solve the proposed T-HTN model. Then the convergence of the method is deduced to ensure the feasibility of the algorithm. In order to reduce the cost of computation, we allow the quadratic penalty parameters to change adaptively according to some update rules.

3) A large number of experiments are carried out on synthetic data and real images. The experimental results show the advantages of this method in effectiveness and stability.

The rest of this paper is organized as follows. Section II introduces some notations, and presents some new tensor norm induced by the transform-based T-product. In Section III, we propose the T-HTN regularization model and give the optimization method to solve it. The Section IV gives the experimental results of synthetic and real data. The Section V draws a conclusion.

2. Preliminaries

2.1. Notations

In this section, we give some notations used in this paper. We denote scalars by lowercase letters, e.g., a, vectors by boldface lowercase letters, e.g., $\boldsymbol{a}$, matrices by boldface capital letters, e.g., $\mathbf{A}$, and tensors by boldface Euler script letters, e.g., $\boldsymbol{\mathcal{A}}$. Respectively, the field of real number and complex number are denoted as $\mathbb{R}$ and $\mathbb{C}$. For a 3-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, we denote its (i,j,k)-th entry as $\boldsymbol{\mathcal{A}}_{ijk}$ or a_{ijk} and use the MATLAB notation $\boldsymbol{\mathcal{A}}(i,:,:)$, $\boldsymbol{\mathcal{A}}(:,i,:)$, $\boldsymbol{\mathcal{A}}(:,:,i)$ to denote separately the i-th horizontal, lateral and frontal slice [25]. Ordinarily, the frontal slice $\boldsymbol{\mathcal{A}}(:,:,i)$ is denoted as $\mathbf{A}^{(i)}$. The (i,j)-th tube of $\boldsymbol{\mathcal{A}}$ is denoted by $\boldsymbol{\mathcal{A}}(i,j,:)$, which is a vector of length n_3. For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\mathbb{C}^{n_1 \times n_2}$, we represent their inner product as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \operatorname{Tr}(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{B})$, where $\operatorname{Tr}(\cdot)$ stands for the trace, and $\boldsymbol{A}^{\mathrm{T}}$ is the conjugate transpose of $\boldsymbol{A}$. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}_n$. Moreover, for any two tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ in $\mathbb{C}^{n_1 \times n_2 \times n_3}$, $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i=1}^{n_3} \langle \boldsymbol{A}^{(i)}, \boldsymbol{B}^{(i)} \rangle$. Some norms of tensor are also used. The Frobenius norm of a tensor is defined as $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk}|\mathcal{A}_{ijk}|^2}$ and the ℓ_1 norm is defined as $\|\mathcal{A}\|_1 = \sum_{ijk}|\mathcal{A}_{ijk}|$. The above norms are also defined in this way in a vector or matrix.

Let $L:\mathbb{R}^{n_1\times n_2\times n_3}\to\mathbb{C}^{n_1\times n_2\times n_3}$ be invertible linear transform on tensor space. L^{-1} is the inverse transform of L. $\bar{\mathcal{A}}$ is obtained via linear transform L on each tube fiber of $\mathcal{A}$, *i.e.*, $\bar{\mathcal{A}} = L(\mathcal{A})$, and $\mathcal{A} = L^{-1}(\bar{\mathcal{A}})$.

We define a block diagonal matrix based on frontal slice as

$$\bar{\boldsymbol{A}} = \text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\boldsymbol{A}}^{(1)} & & & \\ & \bar{\boldsymbol{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\boldsymbol{A}}^{(n_3)} \end{bmatrix}, \tag{4}$$

where the $\text{bidiag}(\cdot)$ is an operator, maps the tensor $\bar{\mathcal{A}}$ to the block diagonal matrix $\bar{\mathbf{A}}$ size $n_1n_3\times n_2n_3$. On the contrary, $\text{unbdiag}(\cdot)$ maps the block diagonal matrix into a tensor:

$$\text{unbdiag}\left(\text{bdiag}(\bar{\mathcal{A}})\right) = \bar{\mathcal{A}}. \tag{5}$$

where $\text{unbdiag}(\cdot)$ denotes the inverse operator of $\text{bdiag}(\cdot)$.

The block circulant matrix $\text{bcirc}(\mathcal{A})\in\mathbb{R}^{n_1\times n_2\times n_3}$ of $\mathcal{A}$ is defined as

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n_3)} & \cdots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \cdots & \mathbf{A}^{(1)} \end{bmatrix}$$

Lemma 1 *Given any real vector* $\boldsymbol{v}\in\mathbb{R}^n$, *the associated* $\boldsymbol{v}$ *satisfies* $\boldsymbol{v}_1\in\mathbb{R}^n$ *and*

$$\text{conj}(\bar{\boldsymbol{v}}_i) = \bar{\boldsymbol{v}}_{n-i+2}, i = 2,\cdots,\left\lfloor\frac{n+1}{2}\right\rfloor. \tag{6}$$

Conversely, for any given complex $\bar{v}\in\mathbb{C}^n$, [6] is established.

By using Lemma 1, we have

$$\begin{cases} \bar{\mathbf{A}}^{(1)}\in\mathbb{R}^{n_1\times n_2} \\ \text{conj}\left(\bar{\mathbf{A}}^{(i)}\right) = \bar{\mathbf{A}}^{(n_3-i+2)}, i = 2,\cdots,\left\lfloor\frac{n_3+1}{2}\right\rfloor \end{cases} \tag{7}$$

2.2. Preliminaries

In this part, we will give relevant definitions and theorems.

Now, a pair of folding operators are defined as follows. For a tensor $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, we define

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n_3)} \end{bmatrix}, \quad \text{fold}\left(\text{unfold}(\mathcal{A})\right) = \mathcal{A}.$$

where the operator maps the tensor $\mathcal{A}$ to a matrix of size $n_1 n_3 \times n_2$ and fold is its inverse operator.

Definition 1 (***T-product***) [26] *Consider that L is an arbitrary invertible linear transform. The T-product between tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *and* $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ *based on L is defined as*

$$\mathcal{C} = \mathcal{A} *_L \mathcal{B} = L^{-1}\left(\operatorname{unbdiag}\left(\operatorname{bdiag}\left(\bar{\mathcal{A}}\right) \times \operatorname{bdiag}\left(\bar{\mathcal{B}}\right)\right)\right) \tag{8}$$

where L^{-1} is the inverse transform of *L*, and × denotes the standard matrix product.

T-product is similar as matrix product, except that cyclic convolution is used to replace the multiplication between elements. When $n_3 = 1$, the T-product simplifies to the standard matrix product.

Definition 2 (***Tensor transpose***) [26] *Let L be an arbitrary invertible linear transform. For a tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, *the tensor transpose of* $\mathcal{A}$ *is defined as* $\mathcal{A}^{\mathrm{T}}$, *satisfies* $L\left(\mathcal{A}^{\mathrm{T}}\right)^{(i)} = \left(L\left(\mathcal{A}\right)^{(i)}\right)^{\mathrm{T}}, i = 1, \cdots, n_3$.

Definition 3 (***Identity tensor***) *Suppose that L is any invertible linear transform. Tensor* $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ *is an identity tensor if the first frontal slice of* $L(\mathcal{I})$ *is a* $n \times n$ *sized identity matrix and whose other frontal slices being all zeros.*

It is clear that $\mathcal{A} *_L \mathcal{I} = \mathcal{A}$ and $\mathcal{I} *_L \mathcal{A} = \mathcal{A}$ given the appropriate dimensions. The tensor $\bar{\mathcal{I}} = fft\left(\mathcal{I}, [\,], 3\right)$ is a tensor that each frontal slice being the identity matrix.

Definition 4 (***Orthogonal tensor***) *Consider that L is any invertible linear transform. Tensor* $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ *is orthogonal if it satisfies* $\mathcal{Q}^{\mathrm{T}} *_L \mathcal{Q} = \mathcal{Q} *_L \mathcal{Q}^{\mathrm{T}} = \mathcal{I}$.

Definition 5 (***F-diagonal tensor***) *Tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *is said to be an F-diagonal tensor if each frontal slice of* $L(\mathcal{A})$ *is a diagonal matrix via an arbitrary invertible linear transform L.*

Definition 6 (***Inverse tensor***) *A frontal square tensor* $\mathcal{A}$ *of size* $n \times n \times n_3$ *has an inverse tensor* $\mathcal{B} = \mathcal{A}^{-1}$ *provided.*

$$\mathcal{A} *_L \mathcal{B} = \mathcal{I} \text{ and } \mathcal{B} *_L \mathcal{A} = \mathcal{I}. \tag{9}$$

Theorem 1 (***T-SVD***) *Suppose that L is an arbitrary invertible linear transform and* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. *Then, T-SVD of* $\mathcal{A}$ *is given by*

$$\mathcal{A} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^{\mathrm{T}}, \tag{10}$$

where $\mathcal{U} *_L \mathcal{U}^{\mathrm{T}} = \mathcal{I}$, $\mathcal{V} *_L \mathcal{V}^{\mathrm{T}} = \mathcal{I}$, and $\mathcal{S} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ is an F-diagonal tensor.

The specific process will be presented in algorithm 1

Definition 7 (***Tensor multi-rank and tubal rank***) *For* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, *let L be an arbitrary invertible linear transform satisfying* $\mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{L}\mathbf{L}^{\mathrm{T}} = \ell \boldsymbol{I}_{n_3}$ *with* $\ell > 0$ *being a constant. The tensor multi-rank is a vector denoted as* $rank_m\left(\mathcal{A}\right) = \left(rank\left(\bar{\mathbf{A}}\right)^{(1)}, \cdots, rank\left(\bar{\mathbf{A}}\right)^{(n_3)}\right)^{\mathrm{T}}$. *The tensor tubal rank is defined as the number of nonzero singular tubes of* $\mathcal{S}$, *i.e.,*

$$rank_t\left(\mathcal{A}\right) = \#\left\{i, \mathcal{S}\left(i,i,:\right) \neq 0\right\} = \#\left\{i, \mathcal{S}\left(i,i,1\right) \neq 0\right\}. \tag{11}$$

Definition 8 (*TNN*) *Assume that* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *with corresponding T-SVD* $\mathcal{A} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^{\mathrm{T}}$; *its tensor nuclear norm is defined as the sum of the singular values of all frontal slices of* $\bar{\mathcal{A}}$, *i.e.*

$$\left\|\mathcal{A}\right\|_* = \sum_{i=1}^{r} \mathcal{S}\left(i,i,1\right) = \left\langle \mathcal{S}, \mathcal{I} \right\rangle = \frac{1}{n_3}\left\|\bar{\mathbf{A}}\right\|_*, \tag{12}$$

Theorem 2 (*Tensor Singular Value Thresholding*) *For any* $\tau > 0$ *and* $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, *the solution of the problem*

$$\min_{\mathcal{X}} \tau \left\|\mathcal{X}\right\|_* + \frac{1}{2}\left\|\mathcal{X} - \mathcal{Y}\right\|_F^2, \tag{13}$$

is given by $\mathcal{D}_\tau\left(\mathcal{Y}\right)$, which is defined as

$$\mathcal{D}_\tau\left(\mathcal{Y}\right) = \mathcal{U} *_L \mathcal{S}_\tau *_L \mathcal{V}^{\mathrm{T}}, \tag{14}$$

where $\mathcal{S}_\tau = L^{-1}\left(\left(\bar{\mathcal{S}} - \tau\right)_+\right)$ with $x_+ = \max\left(x, 0\right)$.

Definition 9 (*Tensor Frobenius norm* (*TFN*)) *For any* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, *the tensor Frobenius norm of* $\mathcal{A}$ *is defined as*

$$\left\|\mathcal{A}\right\|_F = \sqrt{\left\langle \mathcal{A}, \mathcal{A} \right\rangle} = \frac{1}{\sqrt{n_3}}\left\|\bar{\mathbf{A}}\right\|_F. \tag{15}$$

```
Algorithm 1 T-SVD
Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.
Output: T-SVD components $\mathcal{U}, \mathcal{S}, \mathcal{V}$.
    1. Compute $\bar{\mathcal{A}} = fft(\mathcal{A}, [], 3)$
    2. Compute each frontal slice of $\bar{\mathcal{U}}, \bar{\mathcal{S}}, \bar{\mathcal{V}}$ by
        for i=1, $\cdots$, $\lceil \frac{n_3+1}{2} \rceil$ do
        $[\bar{\mathbf{U}}^{(i)}, \bar{\mathbf{S}}^{(i)}, \bar{\mathbf{V}}^{(i)}] = SVD(\bar{\mathbf{A}}^{(i)})$;
        end for.
        for $i = \lceil \frac{n_3+1}{2} \rceil + 1, \cdots n_3$ do
        $\bar{\mathbf{U}}^{(i)} = \mathrm{conj}\left(\bar{\mathbf{U}}^{(n_3-i+2)}\right)$ ;
        $\bar{\mathbf{S}}^{(i)} = \bar{\mathbf{S}}^{(n_3-i+2)}$ ;
        $\bar{\mathbf{V}}^{(i)} = \mathrm{conj}\left(\bar{\mathbf{V}}^{(n_3-i+2)}\right)$ ;
        end for.
    3. Compute $\bar{\mathcal{U}} = fft(\mathcal{U}, [], 3), \bar{\mathcal{S}} = fft(\mathcal{S}, [], 3)\, and\, \bar{\mathcal{V}} = fft(\mathcal{V}, [], 3)$.
```

2.3. Background

In this section, we briefly review the early work related to RPCA, and then discuss some recent developments of TRPCA and its application in computer vision.

RPCA tries to separate a low rank matrix $\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}$ and a sparse matrix $\mathbf{E} \in \mathbb{R}^{n_1 \times n_2}$ from their sum $\mathbf{M} = \mathbf{L} + \mathbf{E}$. As shown in [2], when the underlying tensor $\mathbf{L}$ satisfies the matrix incoherence conditions, the solution of the following problem:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{E}} \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1\\&\text{s.t. } \mathbf{L}+\mathbf{E}=\mathbf{M}\end{aligned} \tag{16}$$

can highly recover $\mathbf{L}$ and $\mathbf{E}$ with high probability with parameter $\lambda = 1/\sqrt{\max(n_1,n_2)}$. In order to solve (16), various methods have been proposed [27]. Among them, Alternating Direction Method of Multipliers (ADMM, or also known as inexact augmented Lagrange multiplier) are widely used. In addition, Zhou *et al.* and Agarwal *et al.* [1] proved that even under small noise measurements, convex approximation by nuclear norm can still achieve bounded and stable results.

Unlike the matrix case, it is very difficult to define tensor rank. Many versions of tensor rank are proposed [5] [28]. Using the matricization, converting the tensor into matrix, Lu *et al.* [14] introduced the tensor nuclear norm as $\|\mathcal{A}\|_* = \frac{1}{n_3}\|\bar{\mathbf{A}}\|_*$, where $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$. Then, TRPCA is then given by

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \|\mathcal{E}\|_1\\&\text{s.t. } \mathcal{X}=\mathcal{L}+\mathcal{E}.\end{aligned}$$

But the rank function may not be well approximated by nuclear norm. To overcome this problem, the T-TNN method was proposed in [29], which proposes a new definition of tensor nuclear norm, extends the truncated nuclear norm from the matrix case to the tensor case:

$$\|\mathcal{A}\|_* = Tr(\mathcal{S}) = Tr\left(\bar{\mathcal{S}}^{(1)}\right) = \left\|\bar{\mathbf{A}}^{(1)}\right\|_*$$

Afterwards, Jiang *et al.* [2] propose the partial sum of the tubal nuclear norm (PSTNN) of a tensor.

$$\|\mathcal{A}\|_{\text{PSTNN}} = \sum_{i=1}^{n_3}\left\|\hat{\mathcal{A}}^{(i)}\right\|_{P=N}$$

The tensor RPCA model using PSTNN was formulated as

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{\text{PSTNN}} + \|\mathcal{E}\|_1\\&\text{s.t. } \mathcal{O}=\mathcal{L}+\mathcal{E}.\end{aligned}$$

The PSTNN is a surrogate of the tensor tubal multi-rank. Though PSTNN is a better approximation to the rank function, the stability issue is still challenging.

3. Tensor Robust Principal Component Analysis Based on Mixed Truncated Norm (T-HTN)

In this section, we mainly introduce the tensor truncated nuclear norm(T-TNN) and tensor truncated Frobenius norm (T-TFN) newly defined in this paper, and propose a tensor hybrid truncated norm(T-HTN) regularization model to improve effectiveness and stability. In addition, the iterative strategy is given.

3.1. T-HTN Model

As mentioned earlier, T-TNN is a better approximation of the rank function, which only considers the last min $(m,n)-r$ singular value [30]. Since Frobenius norm can improve stability, we consider using Frobenius norm term to enhance stability, but the traditional Frobenius norm still needs to consider all singular values, which will make our T-TNN ineffective, which urges us to only consider the last min $(m,n)-r$ singular values for enhancing stability in the definition of Frobenius norm. Based on these analyses, we will construct a more reasonable regularization term called "tensor truncated Frobenius norm (T-TFN)" to replace the original Frobenius norm and match it with T-TNN. Firstly, we give the T-TNN and T-TFN defined in this paper:

Definition 10 (***Tensor truncated nuclear norm*** (***T-TNN***)) *For* $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, *the tensor truncated nuclear norm* $\|\mathcal{A}\|_{r,*}$ *is defined as the sum of the last min* $(m,n)-r$ *singular values of all frontal slices of* $\bar{\mathcal{A}}$, *i.e.*,

$$\begin{aligned}\|\mathcal{A}\|_{r,*} &= \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=r+1}^{\min(n_1,n_2)}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)\\ &= \|\mathcal{A}\|_* - \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)\end{aligned} \tag{17}$$

where $\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)$ denotes the *j*-th largest singular value of $\bar{\mathbf{A}}^{(i)}\in\mathbb{R}^{n_1\times n_2}$.

Definition 11 (***Tensor truncated Frobenius norm*** (***T-TFN***)) *Given a tensor* $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, *the tensor truncated Frobenius norm* (*T-TFN*) *nuclear norm* $\|\mathcal{A}\|_{r,F}$ *is defined as the square root of the square sum of the last min* $(m,n)-r$ *singular values of all frontal slices of* $\bar{\mathcal{A}}$, *i.e.*,

$$\|\mathcal{A}\|_{r,F} = \frac{1}{\sqrt{n_3}}\sum_{i=1}^{n_3}\sqrt{\sum_{j=1}^{\min(n_1,n_2)}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)^2 - \sum_{j=1}^{r}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)^2}. \tag{18}$$

Therefore, $\|\mathcal{A}\|_{r_r F}^2$ can be defined as

$$\begin{aligned}\|\mathcal{A}\|_{r,F}^2 &= \frac{1}{n_3}\sum_{i=1}^{n_j}\sum_{j=r+1}^{\min(n_1,n_2)}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)^2\\ &= \|\mathcal{A}\|_F^2 - \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r}\left(\sigma_j\left(\bar{\mathbf{A}}^{(i)}\right)\right)^2\end{aligned} \tag{19}$$

Evidently, T-TFN could perfectly correspond to T-TNN as they both only concern the influence of the last min $(m,n)-r$ singular values. Therefore, the two can be combined to establish a more effective hybrid truncated tensor robust principal component model (T-HTN):

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}} : \|\mathcal{L}\|_{r,*} + \gamma\|\mathcal{L}\|_{r,F}^2 + \lambda\|\mathcal{E}\|_1\\ &\text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}.\end{aligned} \tag{20}$$

Since $\|\mathcal{L}\|_{r,*} + \gamma\|\mathcal{L}\|_{r,F}^2$ is nonconvex, it is not easy to solve the model (19). To this end, we establish the following theorems.

Theorem 3 *For any* $\mathcal{L}\in\mathbb{R}^{n_1\times n_2\times n_3}$ *with corresponding T-SVD* $\mathcal{L} = \mathcal{A} *_L \mathcal{S} *_L \mathcal{B}^{\mathrm{T}}$, *the following formula holds*:

$$\|\mathcal{L}\|_{r,*} = \|\mathcal{L}\|_* - \max_{\substack{\mathcal{A} *_L \mathcal{A}^{\mathrm{T}} = \mathcal{I}\\ \mathcal{B}^{\mathrm{T}} *_L \mathcal{B} = \mathcal{I}}} Tr\left(\mathcal{A} *_L \mathcal{S} *_L \mathcal{B}^{\mathrm{T}}\right) \tag{21}$$

Theorem 4 *For any* $\mathcal{L}\in\mathbb{R}^{n_1\times n_2\times n_3}$ *with corresponding T-SVD* $\mathcal{L}=\mathcal{A}*_L\mathcal{S}*_L\mathcal{B}^{\mathrm{T}}$, *the following formula holds:*

$$\|\mathcal{L}\|_{r,F}^2=\|\mathcal{L}\|_F^2-\max_{\mathcal{A}*_L\mathcal{A}^{\mathrm{T}}=\mathcal{I}}\|\mathcal{A}*_L\mathcal{L}\|_F^2 \tag{22}$$

Through the above theorems, Equation (19) can be rewritten as

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}}:\|\mathcal{L}\|_*-\max_{\substack{\mathcal{A}*_L\mathcal{A}^{\mathrm{T}}=\mathcal{I}\\ \mathcal{B}^{\mathrm{T}}*_L\mathcal{B}=\mathcal{I}}}Tr\left(\mathcal{A}*_L\mathcal{L}*_L\mathcal{B}^{\mathrm{T}}\right)\\ &\qquad+\gamma\left(\|\mathcal{L}\|_F^2-\max_{\mathcal{A}^{\mathrm{T}}\mathcal{A}=\mathcal{I}}\|\mathcal{A}*_L\mathcal{L}\|_F^2\right)+\lambda\|\mathcal{E}\|_1\\ &\text{s.t. }\mathcal{X}=\mathcal{L}+\mathcal{E}\end{aligned} \tag{23}$$

3.2. Solving Scheme

In order to solve the optimization problem (22), we will exploit an efficient iterative approach, which is divided into two steps. Firstly, we define $\mathcal{L}_0=\mathcal{O}$ as the initial value of $\mathcal{L}$. At the *l*-th iteration, fix $\mathcal{L}_l$ and perform the T-SVD with $\mathcal{L}_l=\mathcal{U}*_L\mathcal{Q}*_L\mathcal{V}^{\mathrm{T}}$, where $\mathcal{U}\in\mathbb{R}^{n_1\times r\times n_3}$, $\mathcal{V}\in\mathbb{R}^{n_2\times r\times n_3}$, $\mathcal{Q}\in\mathbb{R}^{n_1\times n_2\times n_3}$, we set $\mathcal{A}_l=\mathcal{U}(:,1:r,:)^{\mathrm{T}}\in\mathbb{R}^{r\times n_1\times n_3}$, $\mathcal{B}_l=\mathcal{V}(:,1:r,:)^{\mathrm{T}}\in\mathbb{R}^{r\times n_2\times n_3}$. Then fixing $\mathcal{A}_l$ and $\mathcal{B}_l$, we can update $\mathcal{L}_{l+1}$ and $\mathcal{E}_{l+1}$ by solving the next convex optimization problem:

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}}:\|\mathcal{L}\|_*-Tr\left(\mathcal{A}_l*_L\mathcal{L}*_L\mathcal{B}_l^{\mathrm{T}}\right)+\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l*_L\mathcal{L}\|_F^2\right)+\lambda\|\mathcal{E}\|_1\\ &\text{s.t. }\mathcal{X}=\mathcal{L}+\mathcal{E}\end{aligned} \tag{24}$$

In short, we implement these two steps and alternate between them until the iterative scheme meets the tolerance error. Algorithm 2 outlines the complete procedure for (22). Now the remaining key issue is how to solve the model (23), which will be discussed in the next subsection.

```
Algorithm 2 T-HTN
Input: Observation data $\mathcal{X}$ ,parament $\lambda > 0$, and tolerance
    $\varepsilon_0 > 0$.
Initialize: $\mathcal{L}_0 = 0$
repeat
    1. Given $\mathcal{L}_l$,calculate the SVD of $\mathcal{L}_l$
    2. Compute each frontal slice of $\mathcal{L}_l = \mathcal{U} *_L \mathcal{Q} *_L \mathcal{V}^T$
    3. Take $\mathcal{A}_l$ and $\mathcal{B}_l$ as
        $\mathcal{A}_l = \mathcal{U}(:, 1 : r, :)^T \in \mathbb{R}^{r\times n_1\times n_3}$
        $\mathcal{B}_l = \mathcal{V}(:, 1 : r, :)^T \in \mathbb{R}^{r\times n_2\times n_3}$.
    4. Solve the convex optimization problem:(23)
    Until $\|\mathcal{L}_{l+1} - \mathcal{L}_l\|_F \le \varepsilon_0$; .
Output: $\mathcal{L}_{l+1}$
```

3.3. Optimization

In this section, an efficient iterative optimization scheme is created to optimize (23). According to the augmented Lagrangian multiplier method, the commonly used strategy is to approximately minimize the augmented Lagrangian function by adopting an alternating scheme, which we use to solve the model (23). To make the variables separable, we introduce the auxiliary variable $\mathcal{Z}$, thus (23) is equivalent to the following form:

$$\begin{aligned}&\min_{\mathcal{L},\mathcal{E}}:\|\mathcal{L}\|_*-Tr\left(\mathcal{A}_l *_L \mathcal{L} *_L \mathcal{B}_l^{\mathrm{T}}\right)+\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right)+\lambda\|\mathcal{E}\|_1\\ &\text{s.t. } \mathcal{L}=\mathcal{Z} \quad \mathcal{X}=\mathcal{L}+\mathcal{E}\end{aligned} \tag{25}$$

The augmented Lagrange function of (24) can be written as:

$$\begin{aligned}&\mathscr{L}(\mathcal{L},\mathcal{Z},\mathcal{E},\mathcal{Y},\mathcal{W})\\ &=\|\mathcal{L}\|_*-Tr\left(\mathcal{A}_l *_L \mathcal{Z} *_L \mathcal{B}_l^{\mathrm{T}}\right)+\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right)+\lambda\|\mathcal{E}\|_1\\ &\quad+\langle\mathcal{W},\mathcal{L}-\mathcal{Z}\rangle+\frac{\alpha}{2}\|\mathcal{L}-\mathcal{Z}\|_F^2+\langle\mathcal{Y},\mathcal{X}-\mathcal{L}-\mathcal{E}\rangle+\frac{\beta}{2}\|\mathcal{X}-\mathcal{L}-\mathcal{E}\|_F^2\end{aligned} \tag{26}$$

where $\mathcal{Y},\mathcal{W}$ is the Lagrange multiplier matrix and $\alpha,\beta>0$ is the penalty parameter. So, we can adopt the alternating iteration strategy, *i.e.*, fix some variables and solve the remaining one.

Step 1: Keep $\mathcal{L}_k,\mathcal{Z}_k,\mathcal{E},\mathcal{Y}_k,\mathcal{W}_k$ unchanged and update $\mathcal{E}_{k+1}$ through $\mathscr{L}(\mathcal{L}_k,\mathcal{Z}_k,\mathcal{E},\mathcal{Y}_k,\mathcal{W}_k)$

$$\begin{aligned}\mathcal{E}_{k+1}&=\arg\min_{\mathcal{E}}\mathscr{L}(\mathcal{L}_k,\mathcal{Z}_k,\mathcal{E},\mathcal{Y}_k,\mathcal{W}_k)\\ &=\arg\min_{\mathcal{E}}\lambda\|\mathcal{E}\|_1+\langle\mathcal{Y}_k,\mathcal{X}_k-\mathcal{L}_k-\mathcal{E}\rangle+\frac{\beta}{2}\|\mathcal{X}_k-\mathcal{L}_k-\mathcal{E}\|_F^2\\ &=\arg\min_{\mathcal{E}}\lambda\|\mathcal{E}\|_1+\frac{\beta}{2}\left\|\mathcal{E}-\left(\mathcal{X}_k-\mathcal{L}_k+\beta^{-1}\mathcal{Y}_k\right)\right\|_F^2\\ &=\mathcal{S}_{\lambda,\beta^{-1}}\left(\mathcal{X}_k-\mathcal{L}_k+\beta^{-1}\mathcal{Y}_k\right)\end{aligned} \tag{27}$$

Step 2: Keep $\mathcal{L},\mathcal{Z}_k,\mathcal{E}_{k+1},\mathcal{Y}_k,\mathcal{W}_k$ unchanged and update $\mathcal{L}_{k+1}$ through $\mathscr{L}(\mathcal{L},\mathcal{Z}_k,\mathcal{E}_{k+1},\mathcal{Y}_k,\mathcal{W}_k)$

$$\begin{aligned}\mathcal{L}_{k+1}&=\arg\min_{\mathcal{L}}\mathscr{L}(\mathcal{L},\mathcal{Z}_k,\mathcal{E}_{k+1},\mathcal{Y}_k,\mathcal{W}_k)\\ &=\arg\min_{\mathcal{L}}\|\mathcal{L}\|_*+\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right)+\langle\mathcal{W}_k,\mathcal{L}-\mathcal{Z}_k\rangle\\ &\quad+\frac{\alpha}{2}\|\mathcal{L}-\mathcal{Z}_k\|_F^2+\langle\mathcal{Y}_k,\mathcal{X}_k-\mathcal{L}-\mathcal{E}_{k+1}\rangle+\frac{\beta}{2}\|\mathcal{X}_k-\mathcal{L}-\mathcal{E}_{k+l}\|_F^2\\ &=\arg\min_{\mathcal{L}}\|\mathcal{L}\|_*+\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right)+\frac{\alpha}{2}\left\|\mathcal{L}-\mathcal{Z}_k+\alpha^{-1}\mathcal{W}_k\right\|_F^2\\ &\quad+\frac{\beta}{2}\left\|\mathcal{L}-\mathcal{X}_k+\mathcal{E}_{k+1}-\beta^{-1}\mathcal{Y}_k\right\|_F^2\end{aligned} \tag{28}$$

Obviously, Equation (27) is a quadratic function about $\mathcal{L}$. Let

$$\begin{aligned}G(\mathcal{L})=\gamma\left(\|\mathcal{L}\|_F^2-\|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right)+\frac{\alpha}{2}\left\|\mathcal{L}-\mathcal{Z}_k+\alpha^{-1}\mathcal{W}_k\right\|_F^2\\ +\frac{\beta}{2}\left\|\mathcal{L}-\mathcal{X}_k+\mathcal{E}_{k+1}-\beta^{-1}\mathcal{Y}_k\right\|_F^2\end{aligned}$$

be a function with respect to $\mathcal{L}$, then $G(\mathcal{L})$ can be approximated by the linearization at $\mathcal{L}_k$:

$$G(\mathcal{L})=G(\mathcal{L}_k)+\nabla G(\mathcal{L}_k)(\mathcal{L}-\mathcal{L}_k)+\frac{\tau}{2}\|\mathcal{L}-\mathcal{L}_k\|_F^2.$$

where τ is a parameter, and

$$\nabla G(\mathcal{L}_k) = \left[(2\gamma + \alpha + \beta)\mathcal{I} - 2\gamma \mathcal{A}_l *_L \mathcal{A}_l^{\mathrm{T}}\right] *_L \mathcal{L} \\ + \mathcal{W}_k - \mathcal{Y}_k + \beta(\mathcal{E}_{k+1} - \mathcal{X}_k) - \alpha \mathcal{Z}_k$$

denotes the gradient of the function $G(\cdot)$ at $\mathcal{L}_k$. Thus, (27) with respect to $\mathcal{L}$ can be reformulated as:

$$\mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_* + \frac{\tau}{2}\left\|\mathcal{L} - \mathcal{L}_k + \tau^{-1}\nabla G(\mathcal{L}_k)\right\|_F^2 \tag{29}$$

By Theorem 1 and Theorem 2, this could be solved as

$$\mathcal{L}_{k+1} = \mathcal{S}_{\tau^{-1}} - \left(\mathcal{L}_k - \frac{\nabla G(\mathcal{L}_k)}{\tau}\right) \tag{30}$$

Step 3: Keep $\mathcal{L}_{k+1}, \mathcal{Z}, \mathcal{E}_{k+1}, \mathcal{Y}_k, \mathcal{W}_k$ unchanged and update $\mathcal{Z}_{k+1}$ through $\mathscr{L}(\mathcal{L}_{k+1}, \mathcal{Z}, \mathcal{E}_{k+1}, \mathcal{Y}_k, \mathcal{W}_k)$

$$\begin{aligned} \mathcal{Z}_{k+1} &= \arg\min_{\mathcal{Z}} \mathscr{L}(\mathcal{L}_{k+1}, \mathcal{Z}, \mathcal{E}_{k+1}, \mathcal{Y}_k, \mathcal{W}_k) \\ &= \arg\min_{\mathcal{Z}} \langle \mathcal{W}_k, \mathcal{L}_{k+1} - \mathcal{Z}\rangle + \frac{\alpha}{2}\|\mathcal{L}_{k+1} - \mathcal{Z}\|_F^2 - Tr\left(\mathcal{A}_l *_L \mathcal{Z} *_L \mathcal{B}^{\mathrm{T}}\right) \end{aligned} \tag{31}$$

Obviously, the objective function of (30) is a quadratic function. Through a simple derivation operation, we get

$$-\mathcal{A}_l^{\mathrm{T}} *_L \mathcal{B}_l - \mathcal{W}_k - \alpha\mathcal{L}_{k+1} + \alpha\mathcal{Z} = 0$$

This means that

$$\mathcal{Z}_{k+1} = \mathcal{L}_{k+1} + \alpha^{-1}\left(\mathcal{A}_l^{\mathrm{T}} *_L \mathcal{B}_l + \mathcal{W}_k\right) \tag{32}$$

Step 4: Update Lagrange multipliers $\mathcal{Y}_{k+1}, \mathcal{W}_{k+1}$:

$$\begin{aligned} \mathcal{Y}_{k+1} &= \mathcal{Y}_k + \beta(\mathcal{X}_{k+1} - \mathcal{L}_{k+l} - \mathcal{E}_{k+1}) \\ \mathcal{W}_{k+1} &= \mathcal{W}_k + \alpha(\mathcal{L}_{k+1} - \mathcal{Z}_{k+l}) \end{aligned} \tag{33}$$

To sum up, algorithm 3 gives a complete program.

```
Algorithm 3 solve by ADMM
Initialize: $\mathcal{L}_0 = \mathcal{Y}_0 = \mathcal{W}_0 = 0, \rho = 1.1, \alpha_0 = \beta_0 = 1e - 3, \varepsilon = 1e - 8.$
While not converged do
    1. Update $\mathcal{E}_{k+1}$ by
       $\mathcal{E}_{k+1} = \mathcal{S}_{\lambda,\beta^{-1}}(\mathcal{X}_k - \mathcal{L}_k + \beta^{-1}\mathcal{Y}_k);$
    2. Update $\mathcal{L}_{k+1}$ by
       $\mathcal{L}_{k+1} = \mathcal{S}_{\tau^{-1}} - (\mathcal{L}_k - \frac{\nabla G(\mathcal{L}_k)}{\tau});$
    3. Update $\mathcal{Z}_{k+1}$ by
       $\mathcal{Z}_{k+1} + \alpha^{-1}(\mathcal{A}_l^T *_L \mathcal{B}_l + \mathcal{W}_k);$
    4.$\mathcal{Y}_{k+1} = \mathcal{Y}_k + \beta(\mathcal{X}_{k+1} - \mathcal{L}_{k+l} - \mathcal{E}_{k+1});$
    5.$\mathcal{W}_{k+1} = \mathcal{W}_k + \alpha(\mathcal{L}_{k+1} - \mathcal{Z}_{k+l});$
    6. $k = k + 1;$
    7. Check the convergence conditions:
       $\|\mathcal{L}_{k+1} - \mathcal{L}_k\|_\infty \le \varepsilon, \|\mathcal{E}_{k+1} - \mathcal{E}_k\|_\infty \le \varepsilon$
       $\|\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\|_\infty \le \varepsilon$
end while
```

3.4. Adaptive Parameter

Previous studies have shown that the computational cost could increase significantly, since the algorithm may converge slowly when the fixed penalty parameter α and β are chosen too small or large [28] [29], and we need to constantly try different values to pick a good value of α and β. It is not easy to choose the optimal α and β. Accordingly, a dynamical α and β may be required to accelerate convergence in real applications. With the aid of the way in [31], the following adaptive update criterion is used:

$$\alpha_{k+1} = \min\left(\alpha_{\max}, \rho_1 \alpha_k\right) \tag{34}$$

$$\beta_{k+1} = \min\left(\beta_{\max}, \rho_2 \beta_k\right) \tag{35}$$

where $\alpha_{\max}$ and $\beta_{\max}$ is an upper bound of α_k and β_k, and ρ is defined as:

$$\rho_1 = \begin{cases} \rho_0, & \text{if } \dfrac{\alpha_k \max\left(\left\|\mathcal{L}^{k+1} - \mathcal{L}^k\right\|_F, \left\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\right\|_F\right)}{\|\mathcal{Y}\|_F} < k \\ 1, & \text{otherwise} \end{cases} \tag{36}$$

$$\rho_2 = \begin{cases} \rho_0, & \text{if } \dfrac{\beta_k \max\left(\left\|\mathcal{L}^{k+1} - \mathcal{L}^k\right\|_F, \left\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\right\|_F\right)}{\|\mathcal{Y}\|_F} < k \\ 1, & \text{otherwise} \end{cases} \tag{37}$$

where $\rho_1 > 1$ and $\rho_2 > 1$ are constant and $k > 0$ is a threshold chosen in advance.

3.5. Convergence Analysis

This section discusses the convergence of the proposed algorithm.

Firstly, the optimal value of Equation (24) is expressed as follows

$$\begin{aligned} p^* = \inf \Big\{ & \|\mathcal{L}\|_* - Tr\left(\mathcal{A}_l *_L \mathcal{Z} *_L \mathcal{B}_l^{\mathrm{T}}\right) + \gamma\left(\|\mathcal{L}\|_F^2 - \|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right) \\ & + \lambda \|\mathcal{E}\|_1 : \mathcal{L} = \mathcal{Z} \quad \mathcal{X} = \mathcal{L} + \mathcal{Z} \Big\} \end{aligned} \tag{38}$$

The augmented Lagrange function is written as

$$\begin{aligned} & \mathscr{L}_0\left(\mathcal{L}, \mathcal{Z}, \mathcal{E}, \mathcal{Y}, \mathcal{W}\right) \\ & = \|\mathcal{L}\|_* - Tr\left(\mathcal{A}_l *_L \mathcal{Z} *_L \mathcal{B}^{\mathrm{T}}\right) + \gamma\left(\|\mathcal{L}\|_F^2 - \|\mathcal{A}_l *_L \mathcal{L}\|_F^2\right) + \lambda \|\mathcal{E}\|_1 \\ & \quad + \langle \mathcal{W}, \mathcal{L} - \mathcal{Z} \rangle + \langle \mathcal{Y}, \mathcal{X} - \mathcal{L} - \mathcal{E} \rangle \end{aligned} \tag{39}$$

In addition, in order to further analyze the convergence of the algorithm, the following three lemmas are established

Lemma 2 *Let* $\left(\mathcal{L}^*, \mathcal{Z}^*, \mathcal{E}^*, \mathcal{X}^*, \mathcal{Y}^*, \mathcal{W}^*\right)$ *be the optimal solution of the augmented Lagrange function* (38) *and satisfy* $\mathcal{X}^* = \mathcal{L}^* + \mathcal{E}^*$, *then*:

$$p^* - p_{k+1} \le \left\langle \mathcal{W}^*, \mathcal{L}_{k+1} - \mathcal{Z}_{k+1} \right\rangle + \left\langle \mathcal{Y}^*, \mathcal{X}_{k+1} - \mathcal{L}_{k+1} - \mathcal{E}_{k+1} \right\rangle \tag{40}$$

where $\left(\mathcal{L}_{k+1}, \mathcal{Z}_{k+1}, \mathcal{X}_{k+1}, \mathcal{E}_{k+1}\right)$ is the solution of the $k+1$-th iteration in the algorithm 3, p_{k+1} is the k-th iterative solution of (38).

$$p^* = \left\|\mathcal{L}_{k+1}\right\|_* - Tr\left(\mathcal{A}_l *_L \mathcal{Z}_{k+1} *_L \mathcal{B}_l^{\mathrm{T}}\right) + \gamma\left(\left\|\mathcal{L}_{k+1}\right\|_F^2 - \left\|\mathcal{A}_l *_L \mathcal{L}_{k+1}\right\|_F^2\right) + \lambda\left\|\mathcal{E}_{k+1}\right\|_1 .$$

Lemma 3 *Let* $\mathcal{R}_{k+1} = \mathcal{L}_{k+1} - \mathcal{Z}_{k+1}$, $\mathcal{W}_{k+1} = \mathcal{W}_k + \alpha\mathcal{R}_{k+1}$, $\mathcal{Q}_{k+1} = \mathcal{X}_{k+1} - \mathcal{L}_{k+1} - \mathcal{E}_{k+1}$, $\mathcal{Y}_{k+1} = \mathcal{Y}_k + \beta\mathcal{Q}_{k+1}$ *then the following inequality holds*:

$$\begin{aligned} p_{k+1} - p^* \le {} & \left\langle \mathcal{W}_{k+1}, -\mathcal{R}_{k+1}\right\rangle + \left\langle \mathcal{Y}_{k+1}, -\mathcal{Q}_{k+1}\right\rangle \\ & -\alpha\left\langle \mathcal{Z}_k - \mathcal{Z}_{k+1}, -\mathcal{R}_{k+1} + \mathcal{Z}^* - \mathcal{Z}_{k+1}\right\rangle \\ & +\beta\left\langle \mathcal{E}_k - \mathcal{E}_{k+1}, -\mathcal{Q}_{k+1} + \mathcal{L}^* - \mathcal{X}_{k+1} - \mathcal{E}_{k+1}\right\rangle \end{aligned} \tag{41}$$

Lemma 4 *Let* $\left(\mathcal{L}^*, \mathcal{Z}^*, \mathcal{E}^*, \mathcal{X}^*, \mathcal{Y}^*, \mathcal{W}^*\right)$ *be the optimal solution of the augmented Lagrange function* (38) *and* $\mathcal{R}_{k+1} = \mathcal{L}_{k+1} - \mathcal{Z}_{k+1}$, $\mathcal{Q}_{k+1} = \mathcal{X}_{k+1} - \mathcal{L}_{k+1} - \mathcal{E}_{k+1}$, *define*:

$$t_k = \frac{1}{\alpha}\left\|\mathcal{W}_k - \mathcal{W}^*\right\|_F^2 + \alpha\left\|\mathcal{Z}^* - \mathcal{Z}_{k+1}\right\|_F^2 + \frac{1}{\beta}\left\|\mathcal{Y}_k - \mathcal{Y}^*\right\|_F^2 + \beta\left\|\mathcal{L}^* - \mathcal{L}_{k+1}\right\|_F^2 \tag{42}$$

Then t_k is decreasing in each iteration and satisfies the following relationship:

$$t_k - t_{k+1} \ge \alpha\left\|\mathcal{R}_{k+1} - \left(\mathcal{Z}_k - \mathcal{Z}_{k+1}\right)\right\|_F^2 + \beta\left\|\mathcal{Q}_{k+1} - \left(\mathcal{E}_k - \mathcal{E}_{k+1}\right)\right\|_F^2 \tag{43}$$

According to the above results, the following convergence theorems are obtained:

Theorem 5 *Based on the conditions in lemmas* 2, 3 *and* 4, *the iterative solution* $p_k \to p^*$, *when* $k \to \infty$.

4. Experiments

In this section, we will conduct numerical experiments to confirm our main results. We studied the ability of T-HTN to recover various tube rank tensors from various sparse noises, and applied T-HTN to image denoising.

4.1. Synthetic Data Experiments

In this section, we compare the accuracy and speed of T-HTN and TNN based TRPCA [22] on synthetic datasets. Given tensor size $n_1 \times n_2 \times n_3$ and tubal rank $r \ll \min\{n_1, n_2\}$, we first generate a tensor $\mathcal{L}_0 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ by $\mathcal{L}_0 = \mathcal{A} * \mathcal{B}$,where the elements of tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$, $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$ are sampling from independent identically distributed (*i.e.*) standard Gaussian distribution. Then, we from $\mathcal{L}^*$ by $\mathcal{L}^* = \sqrt{n_1 n_2 n_3}\mathcal{L}_0 / \left\|\mathcal{L}_0\right\|_F$. Next, the support of $\mathcal{E}^*$ is uniformly sampled at random. For any $(i, j, k) \in supp\left(\mathcal{E}^*\right)$, we get $\mathcal{E}_{ijk}^* = \mathcal{B}_{ijk}$, where $\mathcal{B}$ is a tensor with independent Bernoulli ±1 entries. Thus, we get the observation tensor $\mathcal{M} = \mathcal{L}^* + \mathcal{E}^*$. In the experiment part, we set the parameters $\lambda = 1/\sqrt{n_3 \max\{n_1, n_2\}}$. We use the relative square error (RSE) to evaluate the estimated $\mathcal{L}^*$ of the underlying tensor $\hat{\mathcal{L}}$, that is $RSE\left(\hat{\mathcal{L}}, \mathcal{L}^*\right) := \frac{\left\|\hat{\mathcal{L}} - \mathcal{L}^*\right\|_F}{\left\|\mathcal{L}^*\right\|_F}$.

4.1.1. Effectiveness and Efficiency of T-HTN

Firstly, we prove that T-HTN can accurately recover the underlying tensor $\mathcal{L}^*$. Based on experience, we test the recovery performance of tensors of different sizes by setting $n = n_1 = n_2 \in \{150, 300, 350\}$ and $n_3 = 3$, with $\left(r_t\left(\mathcal{L}^*\right), \left\|\mathcal{E}^*\right\|_0\right) = \left(0.1n, 0.1n^2 n_3\right)$. Then a more difficult setting $\left(r_t\left(\mathcal{L}^*\right), \left\|\mathcal{E}^*\right\|_0\right) = \left(0.1n, 0.3n^2 n_3\right)$ is tested. The results are shown in Table 1. and Table 2. It is not difficult to see that T-HTN and TRPCA have the same good recovery performance, because both can accurately recover the underlying tensor, but T-HTN has obvious advantages in time.

To test the influence of the distribution of outliers on T-HTN performance, we obtain the outlier tensor element $\mathcal{E}^*$ from *i.i.d* Gaussian distribution $\mathcal{N}(0,1)$, or uniform distribution $\mathcal{U}[0,1]$. The corresponding results are given in Table 3 and Table 4. The performance of T-HTN and TRPCA is not significantly affected by the outlier's distribution. This phenomenon can be explained by Theorem 4.1 in [22], that is when the outlier tensor $\mathcal{E}^*$ is uniformly distributed, TRPCA can accurately restore the underlying tensor $\mathcal{L}^*$ under some mild conditions. In [22], only one hypothesis is made for the random location distribution, but no hypothesis is made for the size or symbol of non-zero terms. Because the T-HTN proposed in this paper can well simulate TRPCA with low tube rank tensors, T-HTN is also robust to outlier.

Table 1. Comparison with TRPCA [22] in both accuracy and speed for different tensor sizes when $\left(r_t\left(\mathcal{L}^*\right), \left\|\mathcal{E}^*\right\|_0\right) = \left(0.1n, 0.1n^2 n_3\right)$.

d	$r_t\left(\mathcal{L}^*\right)$	$\left\|\mathcal{E}^*\right\|_0$	Algorithm	$r_t\left(\hat{\mathcal{L}}\right)$	$\frac{\left\|\hat{\mathcal{L}}-\mathcal{L}^*\right\|_F}{\left\|\mathcal{L}^*\right\|_F}$	$\frac{\left\|\hat{\mathcal{E}}-\mathcal{E}^*\right\|_F}{\left\|\mathcal{E}^*\right\|_F}$	Time
150	15	3e3	TRPCA	15	7.25e-7	5.79e-7	64.77
150	15	3e3	T-HTN	15	2.46e-7	2.4e-7	**3.41**
300	30	2.7e4	TRPCA	30	7.04e-7	5.28e-7	125.39
300	30	2.7e4	T-HTN	30	2.54e-7	2.48e-7	**8.25**
350	35	3.68e4	TRPCA	35	7.16e-7	5.14e-7	174.58
350	35	3.68e4	T-HTN	35	6.09e-8	8.13e-8	**10.30**

Table 2. Comparison with TRPCA [22] in both accuracy and speed for different tensor sizes when $\left(r_t\left(\mathcal{L}^*\right), \left\|\mathcal{E}^*\right\|_0\right) = \left(0.1n, 0.3n^2 n_3\right)$.

d	$r_t\left(\mathcal{L}^*\right)$	$\left\|\mathcal{E}^*\right\|_0$	Algorithm	$r_t\left(\hat{\mathcal{L}}\right)$	$\frac{\left\|\hat{\mathcal{L}}-\mathcal{L}^*\right\|_F}{\left\|\mathcal{L}^*\right\|_F}$	$\frac{\left\|\hat{\mathcal{E}}-\mathcal{E}^*\right\|_F}{\left\|\mathcal{E}^*\right\|_F}$	Time
150	15	3e3	TRPCA	15	1.29e-6	8.25e-7	69.36
150	15	3e3	T-HTN	15	7.38e-7	5.62e-7	**6.72**
300	30	2.7e4	TRPCA	30	1.13e-6	7.73e-7	127.3
300	30	2.7e4	T-HTN	30	7.15e-7	5.73e-7	**14.66**
350	35	3.68e4	TRPCA	35	1.08e-6	7.21e-7	175.19
350	35	3.68e4	T-HTN	35	7.24e-7	5.44e-7	**28.94**

Table 3. Comparison with TRPCA [22] in both accuracy and speed for different tensor sizes when the outlier tensor element $\mathcal{E}^*$ from i.i.d Gaussian distribution $\mathcal{N}(0,1)$.

d	$r_t(\mathcal{L}^*)$	$\|\mathcal{E}^*\|_0$	Algorithm	$r_t(\hat{\mathcal{L}})$	$\frac{\|\hat{\mathcal{L}}-\mathcal{L}^*\|_F}{\|\mathcal{L}^*\|_F}$	$\frac{\|\hat{\mathcal{E}}-\mathcal{E}^*\|_F}{\|\mathcal{E}^*\|_F}$	Time
150	15	3e3	TRPCA	15	4.26e-7	7.48e-7	70.42
150	15	3e3	T-HTN	15	5.29e-7	8.29e-8	**7.20**
300	30	2.7e4	TRPCA	30	4.15e-7	6.83e-7	129.33
300	30	2.7e4	T-HTN	30	5.08e-7	9.23e-8	**15.84**
350	35	3.68e4	TRPCA	35	9.83e-8	6.72e-7	174.96
350	35	3.68e4	T-HTN	35	8.76e-8	8.03e-8	**28.99**

Table 4. Comparison with TRPCA [22] in both accuracy and speed for different tensor sizes when the outlier tensor element $\mathcal{E}^*$ from i.i.d Gaussian distribution $\mathcal{U}(0,1)$.

d	$r_t(\mathcal{L}^*)$	$\|\mathcal{E}^*\|_0$	Algorithm	$r_t(\hat{\mathcal{L}})$	$\frac{\|\hat{\mathcal{L}}-\mathcal{L}^*\|_F}{\|\mathcal{L}^*\|_F}$	$\frac{\|\hat{\mathcal{E}}-\mathcal{E}^*\|_F}{\|\mathcal{E}^*\|_F}$	Time
150	15	3e3	TRPCA	15	4.08e-7	7.98e-7	78.36
150	15	3e3	T-HTN	15	5.17e-7	8.64e-8	**8.45**
300	30	2.7e4	TRPCA	30	6.82e-7	7.04e-7	134.57
300	30	2.7e4	T-HTN	30	4.93e-7	9.420e-8	**19.06**
350	35	3.68e4	TRPCA	35	9.62e-8	7.26e-7	192.39
350	35	3.68e4	T-HTN	35	3.79e-7	6.34e-6	**30.78**

In order to further verify the efficiency of the proposed T-HTN, we consider a special case: fixing the tubal rank of the underlying tensor $\mathcal{L}^*$ and changing the size of the underlying tensor $\mathcal{L}^*$. Explicitly, we fix $r_t(\mathcal{L}^*)=10$, and vary $n \in \{100,150,300,400,500\}$ with $n_3=3$. We tested each data 10 times and calculated the average time. Figure 1 shows the relationship between average time and tensor size. Obviously, both can accurately recover the underlying tensor, but in terms of running time, TRPCA expands super-linearly and T-HTN expands approximately linear scaling.

4.1.2. Effects of the Number of Truncated Singular Values

The performance of T-HTN largely depends on the number of truncated elements of $\mathcal{L}^*$ in (23), that is, the number of singular values to be retained. Here, we discuss the influence of different truncation degrees of Frobenius norm and nuclear norm on the accuracy and speed of T-HTN. Specifically, we consider the tensor with the size of $300\times300\times3$ and set up four different truncation methods $r^*=r_t(\mathcal{L}^*)$ and sparsity $s^*=\|\mathcal{E}^*\|_0$ as $(r^*,s^*)\in\{(10,4.05e4),(10,8.1e4),(15,4.05e4),(15,8.1e4)\}$, where the elements outliers follow *i.i.d.* $\mathcal{N}(0,1)$. By varying the initialized $r\in\{5,10,15,\cdots,50\}$. Firstly, we give the effect of the number of truncated singular values on the recovery performance of the underlying tensor $\mathcal{L}^*$.The results are shown in Figure 2. There is a phrase transition point r_{pt}. Once the reserved singular valuer is greater than it, the RSE of $\hat{\mathcal{L}}$ will decrease rapidly. Then, we give the influence of the number of truncated singular values on the estimation performance of outlier tensor $\mathcal{E}^*$. Figure 3 shows the RSE and l_0-norm of $\hat{\mathcal{E}}$ finally solved. Finally, we show the effect of the number of truncated singular values on the running time of T-HTN in Figure 4. Obviously, the more singular values remain, the longer the running time, and the more complex the underling tensor becomes, which is also consistent with our intuition.

4.2. Real Data Experiments

In this section, we evaluate the efficiency of T-HTN proposed in this paper on real data sets and compare it with TRPCA. Specifically, we do tensor restoration experiments on color image data. The purpose is to restore the original image from its corrupted observation. For the estimation $\hat{\mathcal{L}}$ of the underlying tensor $\mathcal{L}^*$, we choose the relative square error(RSE), the peak signal-to-noise ratio (PSNR), and the structured similarity index(SSIM) as the evaluation index, which is respectively defined as,

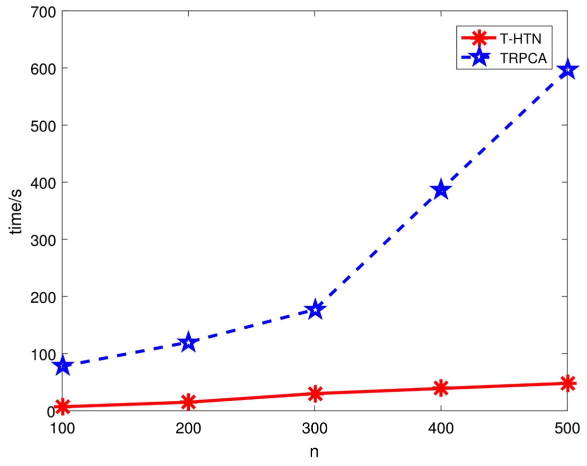

Figure 1. Computation time of TRPCA and proposed T-HTN.

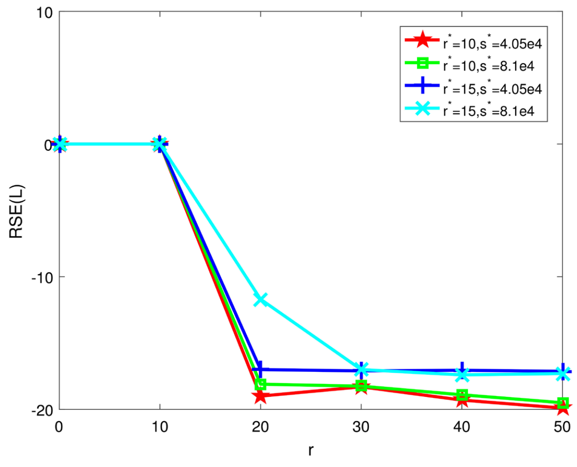

Figure 2. Effects of initialized tubal rank r on the recovery performance of the underlying tensor L.

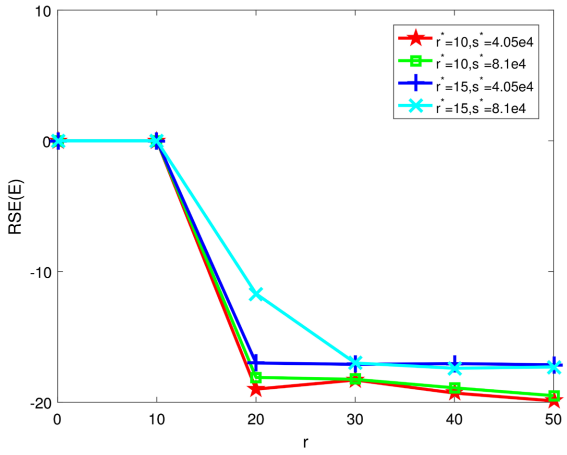

Figure 3. Effects of initialized tubal rank r on the recovery performance of the underlying tensor E.

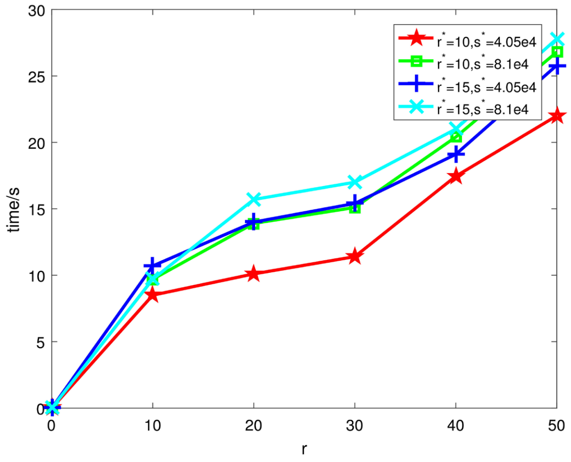

Figure 4. Effects of initialized tubal rank r on the running time of T-HTN.

$$\text{PSNR} = 10\log_{10}\left(\frac{n_1 n_2 n_3 \left\|\mathcal{L}^*\right\|_\infty^2}{\left\|\hat{\mathcal{L}} - \mathcal{L}^*\right\|_F^2}\right),$$

and

$$\text{SSIM} = \frac{\left(2\mu_{\mathcal{L}^*}\mu_{\hat{\mathcal{L}}} + c_1\right)\left(\sigma_{\mathcal{L}\hat{\mathcal{L}}^*} + c_2\right)}{\left(\mu_{\mathcal{L}^*}^2 + \mu_{\hat{\mathcal{L}}}^2 + c_1\right)\left(\sigma + \sigma + c_2\right)}.$$

where $\mu_{\mathcal{L}^*}$ and $\mu_{\hat{\mathcal{L}}}$ are the average values of original tensor and recovered tensor represents the covariance of $\mathcal{L}^*$ and $\hat{\mathcal{L}}$, $\sigma_{\mathcal{L}^*}$ and $\sigma_{\hat{\mathcal{L}}}$ respectively denotes the standard deviation of $\mathcal{L}^*$ and $\hat{\mathcal{L}}$. Natural scenes images follow natural statistics [32]. As shown by Hu *et al.* [33], information of image scenes is dominated by the top 20 singular values, which is low-rank. In **Figure 5**, our experimental data are from 10 common color images. The size of the selected images is $300 \times 300 \times 3$. When these color images are converted into matrices, they can be regarded as low rank structures. The same is true for tensor data.

Table 5. PSNR values of different truncated singular values.

	$r_0 = 2$	$r_0 = 4$	$r_0 = 6$	$r_0 = 8$	$r_0 = 10$	$r_0 = 12$	$r_0 = 14$	$r_0 = 16$	$r_0 = 18$	$r_0 = 20$
Img 1	27.35	27.47	26.58	28.92	**29.42**	29.31	29.28	29.39	29.39	29.85
Img 2	41.75	**43.72**	43.68	42.56	42.89	41.53	41.72	42.66	41.07	42.79
Img 3	27.62	27.83	27.56	28.43	28.05	29.11	**29.17**	29.03	28.59	28.72
Img 4	31.25	31.74	30.89	**32.64**	32.56	32.29	31.88	32.03	31.67	32.52
Img 5	25.32	25.98	27.03	26.52	**27.44**	27.43	27.29	26.94	26.97	26.85
Img 6	31.57	31.26	30.89	32.55	**34.29**	34.17	34.26	34.15	34.07	39.08
Img 7	42.59	41.32	**43.72**	43.60	42.28	42.25	41.08	41.73	40.54	40.72
Img 8	31.60	32.17	31.56	**32.25**	32.19	32.07	31.26	32.02	30.63	31.24
Img 9	33.27	34.52	34.73	**35.07**	34.92	35.05	34.27	34.46	34.29	33.72
Img 10	27.56	28.73	28.25	**28.89**	28.86	28.55	27.69	28.48	27.65	27.32

Figure 5. Tested color images.

Number of Optimal Truncated Singular Values

First, we apply our algorithm to a picture data. We need to preset the number of truncated singular values. Therefore, we designed an experiment in this part, set the number of reserved singular values to $r_0 \in \{2, 4, \cdots, 18, 20\}$, test all possible values, and select the best value. We test 10 images and selected the number of optimal truncated singular values. The detailed results are shown in Table 5. In the worst case of our approach, it is sufficient to test r_0 from 1 to 20. Even if the best r_0 is not selected, its effectiveness will not be lost.

4.3. Conclusions

This paper introduces a new regularization term T-TFN, integrates T-TFN and T-TNN, and proposes a new model T-HTN for TRPCA. Because of the emergence of T-TFN, this new T-HTN regularization method can improve the effectiveness and stability of the model. Then, an effective iterative framework is designed to solve the new optimization model. In addition, its convergence is also derived in the mathematical field. The experimental results on synthetic data, real visual images and recommendation systems, especially the use of statistical analysis methods, illustrate the advantages of HTN over other methods. In other words, HTN method is more effective and stable.

There is still room for further research in this field, for example, how to prove the stability of the proposed method mathematically, and how to select the proportion of missing elements to ensure some theoretical analysis of effective performance.

This work is supported by Innovative talents support plan of colleges and universities in Liaoning Province (2021).

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Lu, C., Feng, J., Chen, Y., *et al.* (2017) Tensor Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Tensors via Convex Optimization. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, *CVPR* 2016, Las Vegas, 27-30 June 2016, 5249-5257. https://doi.org/10.1109/CVPR.2016.567

[2] De Lathauwer, L. and Vandewalle, J. (2004) Dimensionality Reduction in Higher-Order Signal Processing and Rank-(R1, R2, …, RN) Reduction in Multilinear Algebra. *Linear Algebra and Its Applications*, **391**, 31-55. https://doi.org/10.1016/j.laa.2004.01.016

[3] Vasilescu, M. and Terzopoulos, D. (2003) Multilinear Subspace Analysis of Image Ensembles. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, 18-20 June 2003, II-93.

[4] Cyganek, B. (2015) Visual Pattern Recognition Framework Based on the Best Rank Tensor Decomposition. In: Tavares, J.M.R.S. and Jorge, R.N., Eds., *Developments in Medical Image Processing and Computational Vision*, Springer International Publishing, Berlin, 89-103.

[5] Kolda, T.G. and Ba Der, B.W. (2009) Tensor Decompositions and Applications. *SIAM Review*, **51**, 455-500. https://doi.org/10.1137/07070111X

[6] Candès, E.J., Li, X.D., *et al.* (2009) Robust Principal Component Analysis? *Journal of the ACM*, **58**, 1-37.

[7] Wright, J., Ganesh, A., Rao, S., *et al.* (2009) Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices via Convex Optimization. 23*rd Annual Conference on Neural Information Processing Systems* 2009, Vancouver, 7-10 December 2009.

[8] Bao, B.K., Liu, G., Xu, C., *et al.* (2012) Inductive Robust Principal Component Analysis. *IEEE Transactions on Image Processing*, **21**, 3794-3800. https://doi.org/10.1109/TIP.2012.2192742

[9] Javed, S., Mahmood, A., Bouwmans, T., *et al.* (2017) Motion-Aware Graph Regularized RPCA for Background Modeling of Complex Scenes. 2016 *IEEE* 23*rd International Conference on Pattern Recognition* (*ICPR*), Cancun, 4-8 December 2016, 120-125. https://doi.org/10.1109/ICPR.2016.7899619

[10] Liu, G., Lin, Z., Yan, S., *et al.* (2010) Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 171-184.

[11] Golbabaee, M. and Vandergheynst, P. (2012) Compressed Sensing of Simultaneous Low-Rank and Joint-Sparse Matrices. https://doi.org/10.1109/ICASSP.2012.6288484

[12] Hillar, C.J. and Lim, L.H. (2009) Most Tensor Problems Are NP-Hard. *Journal of the ACM*, **60**, Article No. 45. https://doi.org/10.1145/2512329

[13] Gandy, S., Recht, B. and Yamada, I. (2011) Tensor Completion and Low-n-Rank

Tensor Recovery via Convex Optimization. *Inverse Problems*, **27**, Article ID: 025010. https://doi.org/10.1088/0266-5611/27/2/025010

[14] Tucker, L. (1966) Some Mathematical Notes on Three-Mode Factor Analysis. *Psychometrika*, **31**, 279-311. https://doi.org/10.1007/BF02289464

[15] Kilmer, M.E., Braman, K., Hao, N., *et al.* (2013) Third-Order Tensors as Operators on Matrices: A Theoretical and Computational Framework with Applications in Imaging. *SIAM Journal on Matrix Analysis and Applications*, **34**, 148-172. https://doi.org/10.1137/110837711

[16] Bo, H., Mu, C., Goldfarb, D., *et al.* (2015) Provable Models for Robust Low-Rank Tensor Completion. *Pacific Journal of Optimization*, **11**, 339-364.

[17] Anandkumar, A., Jain, P., Shi, Y., *et al.* (2015) Tensor vs Matrix Methods: Robust Tensor Decomposition under Block Sparse Perturbations.

[18] Kilmer, M.E., *et al.* (2011) Factorization Strategies for Third-Order Tensors. *Linear Algebra and Its Applications*, **435**, 641-658. https://doi.org/10.1016/j.laa.2010.09.020

[19] Wang, A., Zhong, J. and Li, X. (2018) Coherent Low-Tubal-Rank Tensor Completion. 2017 4*th IAPR Asian Conference on Pattern Recognition* (*ACPR*), Nanjing, 26-29 November 2017, 518-523. https://doi.org/10.1109/ACPR.2017.66

[20] Lu, C., Feng, J., Chen, Y., *et al.* (2020) Tensor Robust Principal Component Analysis with a New Tensor Nuclear Norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 925-938. https://doi.org/10.1109/TPAMI.2019.2891760

[21] Li, T. and Ma, J. (2019) Non-Convex Penalty for Tensor Completion and Robust PCA.

[22] Lu, C., Feng, J., Chen, Y., *et al.* (2020) Tensor Robust Principal Component Analysis with a New Tensor Nuclear Norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 925-938. https://doi.org/10.1109/TPAMI.2019.2891760

[23] Jiang, T.X., Huang, T.Z., Zhao, X.L., *et al.* (2017) Multi-Dimensional Imaging Data Recovery via Minimizing the Partial Sum of Tubal Nuclear Norm. *Journal of Computational and Applied Mathematics*, **372**, Article ID: 112680.

[24] Ye, H., Li, H., *et al.* (2019) A Hybrid Truncated Norm Regularization Method for Matrix Completion. *IEEE Transactions on Image Processing*, **28**, 5171-5186.

[25] Kilmer, M.E., Martin, C.D. and Perrone, L. (2008) A Third-Order Generalization of the Matrix SVD as a Product of Third-Order Tensors.

[26] Lu, C., Peng, X. and Wei, Y. (2019) Low-Rank Tensor Completion with a New Tensor Nuclear Norm Induced by Invertible Linear Transforms. 2019 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 5989-5997. https://doi.org/10.1109/CVPR.2019.00615

[27] Lu, C., Feng, J., Lin, Z. and Yan, S. (2018) Exact Low Tubal Rank Tensor Recovery from Gaussian Measurements. *Proceedings of the* 27*th International Joint Conference on Artificial Intelligence*, Stockholm, 13-19 July 2018, 2504-2510. https://doi.org/10.24963/ijcai.2018/347

[28] He, B.S., Yang, H. and Wang, S.L. (2000) Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities. *Journal of Optimization Theory and Applications*, **106**, 337-356. https://doi.org/10.1023/A:1004603514434

[29] Yang, J. and Yuan, X. (2012) Linearized Augmented Lagrangian and Alternating Direction Methods for Nuclear Norm Minimization. *Mathematics of Computation*, **82**, 301-329. https://doi.org/10.1090/S0025-5718-2012-02598-1

[30] Xue, S.K., Qiu, W.Y., Liu, F., *et al.* (2017) Low-Rank Tensor Completion by Trun-

cated Nuclear Norm Regularization. 2018 24*th International Conference on Pattern Recognition* (*ICPR*), Beijing, 20-24 August 2018, 2600-2605. https://doi.org/10.1109/ICPR.2018.8546008

[31] Lin, Z., Liu, R. and Li, H. (2015) Linearized Alternating Direction Method with Parallel Splitting and Adaptive Penalty for Separable Convex Programs in Machine Learning. *JMLR*: *Workshop and Conference Proceedings*, Vol. 29, 116-132. https://doi.org/10.1007/s10994-014-5469-5

[32] Fan, H., Kuang, G. and Qiao, L. (2017) Fast Tensor Principal Component Analysis via Proximal Alternating Direction Method with Vectorized Technique. *Applied Mathematics*, **8**, 77-86. https://doi.org/10.4236/am.2017.81007

[33] Hu, Y., *et al.* (2013) Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 2117-2130. https://doi.org/10.1109/TPAMI.2012.271