Online Tensor Max-Norm Regularization via Stochastic Optimization

Anonymous authors
Paper under double-blind review

Abstract

The advent of ubiquitous multidimensional arrays poses unique challenges for low-rank modeling of tensor data due to higher-order relationships, gross noise, and large dimensions of the tensor. In this paper, we consider online low-rank estimation of tensor data where the multidimensional data are revealed sequentially. Induced by the recently proposed tensor-tensor product (t-product), we rigorously deduce the tensor max-norm and formulate the tensor max-norm into an equivalent tensor factorization form, where the factors consist of a tensor basis component and a coefficient one. With this formulation, we develop an online max-norm regularized tensor decomposition (OMRTD) method by optimizing over the basis component and the coefficient tensor alternatively. The algorithm is scalable to the large-scale setting and the sequence of the solutions produced by OMRTD converges to a stationary point of the expected loss function asymptotically. Further, we extend OMRTD for tensor completion. Numerical experiments demonstrate encouraging results for the effectiveness and robustness of our algorithm.

1 Introduction

In the last decade or so, we have witnessed an explosion in data generation due to the development of new, affordable consumer electronics and advances in data storage and communication technologies. Many information processing tasks involve data samples that are naturally structured as multidimensional arrays, also known as tensors. Examples of tensor data include images, videos, hyperspectral images, tomographic images and multichannel electroencephalography data. Low-rank tensor estimation has attracted increasing attention in the research community owing to its successful applications within computer vision (Zhang et al., 2014), data mining (Franz et al., 2009), and signal processing (Sidiropoulos et al., 2017). In this work, specifically, we consider a three-way data tensor $\boldsymbol{\mathcal{Z}}$, and we attempt to learn a low-rank tensor $\boldsymbol{\mathcal{X}}$ that best approximates grossly corrupted observations. This problem typically involves minimizing a weighted combination of the approximation error and a penalty for the tensor rank.

This problem is well studied in the matrix domain, where the goal is to optimize the matrix rank of the prediction matrix. However, this problem is likely to be computationally infeasible. The nuclear norm (Recht et al., 2010) and the max-norm (Srebro et al., 2004) are the two commonly used convex relaxations of the rank function (NP-hard). However, different from matrices, there are many tensor rank definitions because a tensor can be factorized in many ways. Common tensor decompositions include CANDECOMP/PARAFAC (CP) decomposition (Kolda & Bader, 2009), Tucker decomposition (Tucker, 1966), HOSVD decomposition (Lathauwer et al., 2000), tensor-train decomposition (Oseledets, 2011), and t-SVD decomposition (Kilmer et al., 2013; 2021). Several works for tensor robust principal component analysis (RPCA) relying on these decompositions are proposed (Anandkumar et al., 2016; Yang et al., 2020; Lu et al., 2020; Gao et al., 2021). While these approaches have been incredibly successful in many applications, an important shortcoming is that they are not scalable to large-scale tensor data since the memory requirements increase rapidly with the size of data. Moreover, such methods are all implemented in a batch manner, which cannot efficiently capture the dynamics of streaming data.

Motivated by the t-product (Kilmer & Martin, 2011), we rigorously deduce a new tensor max-norm for tensor decomposition. By utilizing the tensor factorization form of the proposed max-norm, we develop an efficient algorithm, termed *online max-norm regularized tensor decomposition* (OMRTD), to solve tensor max-norm regularized problem. OMRTD processes only one sample per time instance, making it scalable to large-scale tensor data. We also extend OMRTD for the tensor completion problem, where low-rank tensor data recovery is carried out in the presence of missing data. Extensive experimental results on the tensor subspace recovery task illustrate that the proposed tensor max-norm always performs better than the tensor nuclear norm in terms of convergence rate and robustness.

The rest of this paper is organized as follows. Section 2 briefly discusses related work and Section 3 introduces some mathematical notations and tensor basics. Section 4 presents the proposed OMRTD method in the presence of complete and missing data. We present experimental results in Section 5 and provide concluding remarks in Section 6.

2 Related Work

Low-rank models find applications in collaborative filtering (Srebro et al., 2004), hyperspectral image restoration (He et al., 2016), and background subtraction (Candès et al., 2011). To handle data contaminated by gross corruption, the Robust Principal Component Analysis (RPCA) (Candès et al., 2011) decomposes the observed matrix into a low-rank component and a sparse component using nuclear norm regularization. The work in Srebro et al. (2004) considered collaborative prediction and learned low-max-norm matrix factorizations by solving a semi-definite program. To establish the connection between the max-norm and the nuclear norm, Srebro & Shraibman (2005) considered collaborative filtering as an example and proved that the max-norm variant enjoys a lower generalization bound than the nuclear norm. In the large-scale setting, there are some efforts that attempted to develop efficient algorithms to solve max-norm regularized or constrained problems (Rennie & Srebro, 2005; Lee et al., 2010). Yet, the applicability of such batch optimization methods is problematic because of their high memory cost. To alleviate this issue, online learning approaches that are based on nuclear norm and max-norm matrix decomposition using stochastic optimization have been proposed in Feng et al. (2013) and Shen et al. (2017), respectively. However, all these approaches are devised for 2-way data, thus limiting their abilities to exploit the intrinsic structure of tensors.

Besides the tensor max-norm used in this paper, there exist several different tensor rank definitions due to the complex multilinear structure of tensors. The CP rank (Kiers, 2000) is defined as the smallest number of rank one tensor decomposition. However, the CP rank and its convex relaxation are NP-hard to compute (Hillar & Lim, 2013). To alleviate this issue, the tractable Tucker rank (Tucker, 1966) is more flexible because it explores the low-rank structure in all modes. The sum-of-nuclear-norms (SNN) is defined as the sum of the nuclear norms of unfolding matrices (Liu et al., 2013), which is served as a convex surrogate for the Tucker rank. The effectiveness of this idea has been well studied in Goldfarb & Qin (2014) and Huang et al. (2014). But it was proved in Romera-Paredes & Pontil (2013) that SNN is not the tightest convex relaxation of the Tucker rank. Recently, based on the tensor-tensor product and tensor singular value decomposition (t-SVD) scheme (Kilmer et al., 2013), a new tensor rank called tensor tubal rank (Kilmer et al., 2013) is defined as the number of nonzero singular tubes of the singular value tensor in t-SVD. Correspondingly, a new tensor nuclear norm is proposed and applied in tensor completion (Zhang & Aeron, 2017), tensor robust PCA (Lu et al., 2020), and tensor data clustering (Zhou et al., 2021).

Nonetheless, all these methods require memorizing all the samples in each iteration and they cannot process samples in a sequential way. To address this concern, several online methods have been developed for streaming tensor data analysis (Yu et al., 2015; Mardani et al., 2015; Zhang et al., 2016; Kasai, 2019; Li et al., 2019; Gilman et al., 2022). Among these works, Mardani et al. (2015) and Kasai (2019) obtain multi-way decompositions of low-rank tensors with missing entries using the CP decomposition. In particular, both Zhang et al. (2016) and Gilman et al. (2022) rely on the t-SVD framework. Note that Zhang et al. (2016) implements online tensor robust PCA using tensor nuclear-norm regularization, while our work adopts our proposed tensor max-norm for tensor decomposition. Indeed, since the tensor max-norm is a more complicated mathematical entity, the development of online methods for the max-norm regularization requires

more attention. Based on the tensor factorization form of the proposed tensor max-norm, we convert the problem into a constrained tensor factorization problem that is amenable to online implementation.

3 Notations and Preliminaries

In this section, we introduce notations and some basic facts about t-SVD that will be used throughout this paper. More related tensor facts can be found in Kilmer et al. (2013; 2021). Throughout this paper, we use lowercase, bold lowercase, bold uppercase, and bold calligraphic letters for scalars, vectors, matrices, and tensors, respectively. For a three-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its (i,j,k)-th entry is denoted as $\boldsymbol{\mathcal{A}}_{i,j,k}$. We use Matlab notation $\boldsymbol{\mathcal{A}}(i,:,:)$, $\boldsymbol{\mathcal{A}}(:,i,:)$ and $\boldsymbol{\mathcal{A}}(:,:,i)$ or $\mathbf{A}^{(i)}$ to denote the i-th horizontal, lateral and frontal slices, respectively. Any lateral slice of size $n_1 \times 1 \times n_3$ is denoted as $\overrightarrow{\boldsymbol{\mathcal{A}}}$. In particular, we also use $\overrightarrow{\boldsymbol{\mathcal{A}}}_i$ to denote the i-th lateral slice of $\boldsymbol{\mathcal{A}}$. The (i,j)-th mode-3 fiber is denoted by $\boldsymbol{\mathcal{A}}(i,j,:)$. The transpose $\boldsymbol{\mathcal{A}}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ is obtained by transposing each frontal slice of $\boldsymbol{\mathcal{A}}$ and then reversing the order of the transposed frontal slices 2 through n_3. We use $\bar{\boldsymbol{\mathcal{A}}} = \texttt{fft}(\boldsymbol{\mathcal{A}}, [\ ], 3)$ to denote the Discrete Fourier transform along mode-3 of $\boldsymbol{\mathcal{A}}$. Similarly, $\boldsymbol{\mathcal{A}}$ can be computed from $\bar{\boldsymbol{\mathcal{A}}}$ using $\boldsymbol{\mathcal{A}} = \texttt{ifft}(\bar{\boldsymbol{\mathcal{A}}}, [\ ], 3)$. The inner product between two tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i,j,k} \boldsymbol{\mathcal{A}}_{i,j,k} \boldsymbol{\mathcal{B}}_{i,j,k}$. The ℓ_1 and Frobenius norms of $\boldsymbol{\mathcal{A}}$ are defined as $\|\boldsymbol{\mathcal{A}}\|_1 = \sum_{i,j,k} |\boldsymbol{\mathcal{A}}_{i,j,k}|$ and $\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\sum_{i,j,k} |\boldsymbol{\mathcal{A}}_{i,j,k}|^2}$, respectively. For a matrix $\mathbf{A}$, its (i,j)-th entry is denoted as $\mathbf{A}_{i,j}$. The i-th row and i-th column of $\mathbf{A}$ are denoted by $\mathbf{a}_{(i)}$ and $\mathbf{a}_i$, respectively. The conjugate transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^H$. The $\ell_{2,\infty}$ norm of $\mathbf{A}$ is defined as the maximum ℓ_2 row norm, i.e., $\|\mathbf{A}\|_{2,\infty} = \max_i \|\mathbf{a}_{(i)}\|_2$.

Besides, for a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define the block vectorizing and its inverse operation as $\texttt{bvec}(\boldsymbol{\mathcal{A}}) = [\mathbf{A}^{(1)}; \mathbf{A}^{(2)}; \cdots; \mathbf{A}^{(n_3)}] \in \mathbb{R}^{n_1 n_3 \times n_2}$ and $\texttt{bvfold}(\texttt{bvec}(\boldsymbol{\mathcal{A}})) = \boldsymbol{\mathcal{A}}$, respectively. We denote $\bar{\mathbf{A}} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$ as a block diagonal matrix with its i-th block on diagonal being the i-th frontal slice of $\bar{\boldsymbol{\mathcal{A}}}$, i.e.,

$$\bar{\mathbf{A}} = \texttt{bdiag}(\bar{\boldsymbol{\mathcal{A}}}) = \begin{bmatrix} \bar{\mathbf{A}}^{(1)} & & \\ & \ddots & \\ & & \bar{\mathbf{A}}^{(n_3)} \end{bmatrix}.$$

Finally, the block circulant matrix $\texttt{bcirc}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ is defined as

$$\texttt{bcirc}(\boldsymbol{\mathcal{A}}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n_3)} & \dots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \dots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \dots & \mathbf{A}^{(1)} \end{bmatrix}.$$

Definition 1 (t-product (Kilmer & Martin, 2011))**.** *The t-product between two tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as*

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = \texttt{bvfold}(\texttt{bcirc}(\boldsymbol{\mathcal{A}}) \cdot \texttt{bvec}(\boldsymbol{\mathcal{B}})). \tag{1}$$

Definition 2 (Identity tensor (Kilmer & Martin, 2011))**.** *The identity tensor $\boldsymbol{\mathcal{I}}_n \in \mathbb{R}^{n \times n \times n_3}$ is a tensor whose first frontal slice is the $n \times n$ identity matrix and all other frontal slices are zeros.*

Definition 3 (Orthogonal tensor (Kilmer & Martin, 2011))**.** *A tensor $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if $\boldsymbol{\mathcal{Q}} * \boldsymbol{\mathcal{Q}}^T = \boldsymbol{\mathcal{Q}}^T * \boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{I}}_n$.*

Definition 4 (t-SVD (Kilmer et al., 2013))**.** *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then it can be factorized as $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T$, where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a tensor whose frontal slices are diagonal matrices.*

Definition 5 (Tensor average and tubal rank (Lu et al., 2020))**.** *For any $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor average rank of $\boldsymbol{\mathcal{A}}$ is defined as*

$$rank_a(\boldsymbol{\mathcal{A}}) = \frac{1}{n_3} rank(\bar{\mathbf{A}}) = \frac{1}{n_3} rank(\texttt{bcirc}(\boldsymbol{\mathcal{A}})).$$

The tensor tubal rank $rank_t(\boldsymbol{\mathcal{A}})$ *is defined as the number of nonzero singular tubes of* $\boldsymbol{\mathcal{S}}$*, i.e.,*

$$rank_t(\boldsymbol{\mathcal{A}}) = \#\{i : \boldsymbol{\mathcal{S}}(i,i,:) \neq \mathbf{0}\},$$

where $\boldsymbol{\mathcal{S}}$ *is from the t-SVD of* $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T$*.*

Definition 6 (Tensor nuclear norm (Lu et al., 2020))**.** *Let* $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T$ *be the t-SVD of* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$*. The tensor nuclear norm of* $\boldsymbol{\mathcal{A}}$ *is defined as* $\|\boldsymbol{\mathcal{A}}\|_* = \langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{I}} \rangle = \sum_{i=1}^{r} \boldsymbol{\mathcal{S}}_{i,i,1}$*, where* $r = rank_t(\boldsymbol{\mathcal{A}})$*.*

It is known that the tensor nuclear norm is the convex envelope of the tensor average rank within the unit ball of the tensor spectral norm (Lu et al., 2020).

4 Method

This section first formulates the MRTD problem that is amenable for online optimization and then elaborates on the proposed OMRTD algorithm, which is solved by the stochastic optimization technique.

4.1 Problem Formulation

The basic problem corresponds to recovering a low tubal rank tensor corrupted by sparse errors. Suppose we are given a third-order tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n_1 \times N \times n_3}$ consisting of N samples that can be decomposed as a low tubal-rank component $\boldsymbol{\mathcal{X}}$ and a sparse noise tensor $\boldsymbol{\mathcal{E}}$, and the samples $\boldsymbol{\mathcal{Z}}(:,i,:)$, $i = 1, \ldots, N$, are observed sequentially. Our goal is to estimate the two components $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{E}}$ on the fly by solving the following convex program:

$$\min_{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}} \frac{1}{2} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2} \|\boldsymbol{\mathcal{X}}\|_{\max}^2 + \lambda_2 \|\boldsymbol{\mathcal{E}}\|_1, \tag{2}$$

where λ_1 and λ_2 are two positive parameters. Here, we define the max-norm of the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times N \times n_3}$ using the following tensor factorization form as:

$$\|\boldsymbol{\mathcal{X}}\|_{\max} \overset{\text{def}}{=} \min_{\substack{\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3} \\ \boldsymbol{\mathcal{R}} \in \mathbb{R}^{N \times r \times n_3}}} \{\|\bar{\mathbf{L}}\|_{2,\infty} \cdot \|\bar{\mathbf{R}}\|_{2,\infty} : \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}^T\}, \tag{3}$$

where r is an upper bound on the tensor tubal rank of $\boldsymbol{\mathcal{X}}$. As shown in Zhang et al. (2016), the tensor nuclear norm can be written as the following tensor factorization form:

$$\|\boldsymbol{\mathcal{X}}\|_* = \min_{\substack{\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3} \\ \boldsymbol{\mathcal{R}} \in \mathbb{R}^{N \times r \times n_3}}} \{\frac{n_3}{2}(\|\boldsymbol{\mathcal{L}}\|_F^2 + \|\boldsymbol{\mathcal{R}}\|_F^2) = \|\bar{\mathbf{L}}\|_F^2 + \|\bar{\mathbf{R}}\|_F^2 : \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}^T\}. \tag{4}$$

Intuitively, the tensor nuclear norm constrains the row norms of $\bar{\mathbf{L}}$ and $\bar{\mathbf{R}}$ on average, while the tensor max-norm constrains the largest ℓ_2 row norm, hence is tighter. Since the t-product in the spatial domain corresponds to matrix multiplication in the Fourier domain, i.e., $\bar{\mathbf{X}} = \bar{\mathbf{L}}\bar{\mathbf{R}}^H$, we have $\|\boldsymbol{\mathcal{X}}\|_{\max} = \|\bar{\mathbf{X}}\|_{\max}$, where $\|\mathbf{A}\|_{\max}$ denotes the max-norm of $\mathbf{A}$ defined in Srebro et al. (2004). Plugging (3) into (2), we have the following equivalent form:

$$\min_{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{E}}} \frac{1}{2} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}^T - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2} \|\bar{\mathbf{L}}\|_{2,\infty}^2 \|\bar{\mathbf{R}}\|_{2,\infty}^2 + \lambda_2 \|\boldsymbol{\mathcal{E}}\|_1. \tag{5}$$

Intuitively, the variable $\boldsymbol{\mathcal{L}}$ corresponds to a basis for the clean data and each horizontal slice of $\boldsymbol{\mathcal{R}}$ corresponds to the coefficients associated with each sample. At first glance, (5) can only be optimized in a batch manner because $\|\bar{\mathbf{R}}\|_{2,\infty}^2$ couples all the samples. Fortunately, we have the following proposition that allows us to optimize the horizontal slices of $\boldsymbol{\mathcal{R}}$ sequentially.

Proposition 1. *Problem* (5) *is equivalent to the following constrained program:*

$$\begin{aligned} \min_{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{E}}} & \frac{1}{2} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}^T - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2} \|\bar{\mathbf{L}}\|_{2,\infty}^2 + \lambda_2 \|\boldsymbol{\mathcal{E}}\|_1 \\ s.t. \quad & \|\bar{\mathbf{R}}\|_{2,\infty}^2 \leq 1. \end{aligned} \tag{6}$$

Moreover, there exists an optimal solution $(\boldsymbol{\mathcal{L}}^\star, \boldsymbol{\mathcal{R}}^\star, \boldsymbol{\mathcal{E}}^\star)$ *attained such that* $\|\bar{\mathbf{R}}^\star\|_{2,\infty} = 1$*.*

Algorithm 1 Online Max-Norm Regularized Tensor Decomposition

Input: Observed samples $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n_1 \times N \times n_3}$, and parameters λ_1 and λ_2.
Initialize: Random basis $\boldsymbol{\mathcal{L}}_0 \in \mathbb{R}^{n_1 \times r \times n_3}$, $\boldsymbol{\mathcal{A}}_0 = \boldsymbol{\mathcal{B}}_0 = \mathbf{0}$.

1: **for** $t = 1, 2, \ldots, N$ **do**
2: Access the t-th sample $\overrightarrow{\boldsymbol{\mathcal{Z}}}_t$.
3: Update $\{\overrightarrow{\boldsymbol{\mathcal{R}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}_t^\star\}$ by solving (10).
4: Update the accumulation tensors:

$$\boldsymbol{\mathcal{A}}_t = \boldsymbol{\mathcal{A}}_{t-1} + \overrightarrow{\boldsymbol{\mathcal{R}}}_t^\star * \overrightarrow{\boldsymbol{\mathcal{R}}}_t^{\star T},$$
$$\boldsymbol{\mathcal{B}}_t = \boldsymbol{\mathcal{B}}_{t-1} + (\overrightarrow{\boldsymbol{\mathcal{Z}}}_t - \overrightarrow{\boldsymbol{\mathcal{E}}}_t^\star) * \overrightarrow{\boldsymbol{\mathcal{R}}}_t^{\star T}.$$

5: Update the basis $\boldsymbol{\mathcal{L}}_t$ by optimizing the surrogate function:

$$\bar{\mathbf{L}}_t = \arg\min_{\bar{\mathbf{L}}} \frac{1}{tn_3}\Big(\frac{1}{2}\mathrm{tr}\big(\bar{\mathbf{L}}^H \bar{\mathbf{L}} \bar{\mathbf{A}}_t\big) - \mathrm{tr}\big(\bar{\mathbf{L}}^H \bar{\mathbf{B}}_t\big)\Big) + \frac{\lambda_1}{2t}\|\bar{\mathbf{L}}\|_{2,\infty}^2.$$

6: **end for**

Output: Optimal basis $\boldsymbol{\mathcal{L}}_N$.

Let $\overrightarrow{\boldsymbol{\mathcal{Z}}}_i$, $\overrightarrow{\boldsymbol{\mathcal{R}}}_i$ and $\overrightarrow{\boldsymbol{\mathcal{E}}}_i$ be the i-th lateral slices of tensors $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{R}}^T$ and $\boldsymbol{\mathcal{E}}$, respectively. We define

$$\widetilde{\ell}(\overrightarrow{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{R}}}, \overrightarrow{\boldsymbol{\mathcal{E}}}) \stackrel{\text{def}}{=} \frac{1}{2}\|\overrightarrow{\boldsymbol{\mathcal{Z}}} - \boldsymbol{\mathcal{L}} * \overrightarrow{\boldsymbol{\mathcal{R}}} - \overrightarrow{\boldsymbol{\mathcal{E}}}\|_F^2 + \lambda_2\|\overrightarrow{\boldsymbol{\mathcal{E}}}\|_1.$$

Equipped with Proposition 1, we can rewrite the original problem (5) as the following, with each sample being separately processed:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{E}}} & \sum_{i=1}^{N} \widetilde{\ell}(\overrightarrow{\boldsymbol{\mathcal{Z}}}_i, \boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{R}}}_i, \overrightarrow{\boldsymbol{\mathcal{E}}}_i) + \frac{\lambda_1}{2}\|\bar{\mathbf{L}}\|_{2,\infty}^2 \\ \text{s.t.} \quad & \forall i = 1, \ldots, N, k = 1, \ldots, n_3, \|\overrightarrow{\bar{\mathbf{R}}}_i^{(k)}\|_F^2 \leq 1, \end{aligned} \tag{7}$$

where $\overrightarrow{\bar{\boldsymbol{\mathcal{R}}}}_i = \texttt{fft}(\overrightarrow{\boldsymbol{\mathcal{R}}}_i, [\ ], 3)$ and $\overrightarrow{\bar{\mathbf{R}}}_i^{(k)}$ denotes the k-th frontal slice of $\overrightarrow{\bar{\boldsymbol{\mathcal{R}}}}_i$. This is indeed equivalent to optimizing the *empirical loss* function:

$$f_N(\boldsymbol{\mathcal{L}}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{Z}}}_i) + \frac{\lambda_1}{2N}\|\bar{\mathbf{L}}\|_{2,\infty}^2, \tag{8}$$

where $\ell(\boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{Z}}}) = \min_{\overrightarrow{\boldsymbol{\mathcal{R}}} \in \mathscr{P}, \overrightarrow{\boldsymbol{\mathcal{E}}}} \widetilde{\ell}(\overrightarrow{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{R}}}, \overrightarrow{\boldsymbol{\mathcal{E}}})$ with $\mathscr{P} = \{\overrightarrow{\boldsymbol{\mathcal{R}}} : \forall k, \|\overrightarrow{\bar{\mathbf{R}}}^{(k)}\|_F^2 \leq 1\}$. We assume that each sample is drawn independently and identically distributed from some unknown distribution. In stochastic optimization, one is usually interested in the minimization of the *expected loss* function

$$f(\boldsymbol{\mathcal{L}}) \stackrel{\text{def}}{=} \mathbb{E}[\ell(\boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{Z}}})] = \lim_{N \to +\infty} f_N(\boldsymbol{\mathcal{L}}). \tag{9}$$

4.2 Online Implementation of OMRTD

We now describe our online algorithm for minimizing the empirical loss function (8). The detailed algorithm is summarized in Algorithm 1. We start with an initial random dictionary $\boldsymbol{\mathcal{L}}_0$. The key idea is that at each iteration t, we first minimize the loss function with respect to $\{\overrightarrow{\boldsymbol{\mathcal{R}}}_t, \overrightarrow{\boldsymbol{\mathcal{E}}}_t\}$ given the previous $\boldsymbol{\mathcal{L}}_{t-1}$ by solving (7), and then further refine the dictionary $\boldsymbol{\mathcal{L}}_t$ by minimizing the cumulative loss.

Algorithm 2 Data Projection (Problem (10))

Input: Observed data sample $\overrightarrow{\boldsymbol{\mathcal{Z}}} \in \mathbb{R}^{n_1 \times 1 \times n_3}$, $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3}$, and parameters λ_2 and ϵ.
Initialize: $\overrightarrow{\boldsymbol{\mathcal{E}}} = \mathbf{0}$.

```
while not converged do
    Compute the potential solution $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}}$ using (11).
    if $\forall k, \|\overrightarrow{\mathbf{R}}^{(k)}_{\text{cand}}\|_F \leq 1$ then
        Set $\overrightarrow{\boldsymbol{\mathcal{R}}} = \overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}}$.
    else
        for $k = 1, 2, \ldots, n_3$ do
            if $\|\overrightarrow{\mathbf{R}}^{(k)}_{\text{cand}}\|_F \leq 1$ then
                Set $\overrightarrow{\mathbf{R}}^{(k)} = \overrightarrow{\mathbf{R}}^{(k)}_{\text{cand}}$.
            else
                Update $\overrightarrow{\mathbf{R}}^{(k)}$ by Algorithm 5.
            end if
        end for
    end if
    $\overrightarrow{\boldsymbol{\mathcal{R}}} = \texttt{ifft}(\overrightarrow{\boldsymbol{\mathcal{R}}}, [\,], 3)$.
    $\overrightarrow{\boldsymbol{\mathcal{E}}} = S_{\lambda_2}[\overrightarrow{\boldsymbol{\mathcal{Z}}} - \boldsymbol{\mathcal{L}} * \overrightarrow{\boldsymbol{\mathcal{R}}}]$.
end while
```

Output: Optimal $\overrightarrow{\boldsymbol{\mathcal{R}}}^{\star}$ and $\overrightarrow{\boldsymbol{\mathcal{E}}}^{\star}$.

Optimize $\overrightarrow{\boldsymbol{\mathcal{R}}}$ and $\overrightarrow{\boldsymbol{\mathcal{E}}}$: Given $\boldsymbol{\mathcal{L}}_{t-1}$ in the previous iteration, we obtain the optimal solution $\{\overrightarrow{\boldsymbol{\mathcal{R}}}^{\star}_t, \overrightarrow{\boldsymbol{\mathcal{E}}}^{\star}_t\}$ by solving

$$\{\overrightarrow{\boldsymbol{\mathcal{R}}}^{\star}_t, \overrightarrow{\boldsymbol{\mathcal{E}}}^{\star}_t\} = \underset{\overrightarrow{\boldsymbol{\mathcal{R}}} \in \mathscr{P}, \overrightarrow{\boldsymbol{\mathcal{E}}}}{\arg\min} \frac{1}{2}\|\overrightarrow{\boldsymbol{\mathcal{Z}}}_t - \boldsymbol{\mathcal{L}}_{t-1} * \overrightarrow{\boldsymbol{\mathcal{R}}} - \overrightarrow{\boldsymbol{\mathcal{E}}}\|_F^2 + \lambda_2 \|\overrightarrow{\boldsymbol{\mathcal{E}}}\|_1. \tag{10}$$

We employ a block coordinate descent algorithm which alternatively updates one variable at a time with the other being fixed until some stopping criteria is satisfied. When $\overrightarrow{\boldsymbol{\mathcal{E}}}$ is fixed, we first compute an initial guess of $\overrightarrow{\boldsymbol{\mathcal{R}}}$ by $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}} = (\boldsymbol{\mathcal{L}}_{t-1}^T * \boldsymbol{\mathcal{L}}_{t-1})^{-1} * \boldsymbol{\mathcal{L}}_{t-1}^T * (\overrightarrow{\boldsymbol{\mathcal{Z}}}_t - \overrightarrow{\boldsymbol{\mathcal{E}}})$ without considering the constraint that $\overrightarrow{\boldsymbol{\mathcal{R}}} \in \mathscr{P}$. When $\boldsymbol{\mathcal{L}}$ is not full tubal rank, $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}}$ is computed by

$$\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}} = (\boldsymbol{\mathcal{L}}_{t-1}^T * \boldsymbol{\mathcal{L}}_{t-1} + \epsilon \boldsymbol{\mathcal{I}}_r)^{-1} * \boldsymbol{\mathcal{L}}_{t-1}^T * (\overrightarrow{\boldsymbol{\mathcal{Z}}}_t - \overrightarrow{\boldsymbol{\mathcal{E}}}), \tag{11}$$

where ϵ is a small positive constant (we set $\epsilon = 0.01$ in our experiments). If this initial guess $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}}$ satisfies the constraint that $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}} \in \mathscr{P}$, we will use $\overrightarrow{\boldsymbol{\mathcal{R}}}_{\text{cand}}$ as the new iterate for $\overrightarrow{\boldsymbol{\mathcal{R}}}$. Otherwise, for each k, if $\|\overrightarrow{\mathbf{R}}^{(k)}_{\text{cand}}\|_F > 1$, we will introduce a positive dual variable η and solve

$$\begin{aligned} &\max_{\eta} \min_{\overrightarrow{\mathbf{R}}^{(k)}} \frac{1}{2}\|\overrightarrow{\mathbf{Z}}^{(k)}_t - \overline{\mathbf{L}}_{t-1}\overrightarrow{\mathbf{R}}^{(k)} - \overrightarrow{\mathbf{E}}^{(k)}\|_F^2 + \frac{\eta}{2}(\|\overrightarrow{\mathbf{R}}^{(k)}\|_F^2 - 1) \\ &\text{s.t.} \quad \eta > 0, \|\overrightarrow{\mathbf{R}}^{(k)}\|_F = 1. \end{aligned} \tag{12}$$

The closed-form solution of (12) is given by

$$\overrightarrow{\mathbf{R}}^{(k)} = (\overline{\mathbf{L}}_{t-1}^H \overline{\mathbf{L}}_{t-1} + \eta \mathbf{I}_r)^{-1} \overline{\mathbf{L}}_{t-1}^H (\overrightarrow{\mathbf{Z}}^{(k)}_t - \overrightarrow{\boldsymbol{\mathcal{E}}}^{(k)}),$$

where $\mathbf{I}_r$ denotes the $r \times r$ identity matrix. According to Proposition 2 of Shen et al. (2017), we understand that for each k, $\|\overrightarrow{\mathbf{R}}^{(k)}\|_F$ is a strictly monotonically decreasing function with respect to η. This allows us to

Algorithm 3 The Update of $\boldsymbol{\mathcal{L}}$

Input: $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3}$ in the previous iteration, accumulation tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, and parameter λ_1.

$\bar{\boldsymbol{\mathcal{L}}} = \texttt{fft}(\boldsymbol{\mathcal{L}}, [\,], 3)$, $\bar{\boldsymbol{\mathcal{A}}} = \texttt{fft}(\boldsymbol{\mathcal{A}}, [\,], 3)$, and $\bar{\boldsymbol{\mathcal{B}}} = \texttt{fft}(\boldsymbol{\mathcal{B}}, [\,], 3)$.
repeat
  Compute the subgradient of $\frac{1}{2}\|\bar{\mathbf{L}}\|_{2,\infty}^2$:
    $\mathbf{U} = \partial(\frac{1}{2}\|\bar{\mathbf{L}}\|_{2,\infty}^2)$.
  for $k = 1, 2, \ldots, n_3$ **do**
    $\widetilde{\mathbf{A}} = \bar{\mathbf{A}}^{(k)}$, $\widetilde{\mathbf{B}} = \bar{\mathbf{B}}^{(k)}$, $\widetilde{\mathbf{U}} = \mathbf{U}_k$.
    for $j = 1, 2, \ldots, r$ **do**
      $\bar{\mathbf{L}}(:, j, k) = \bar{\mathbf{L}}(:, j, k) - \frac{1}{\widetilde{\mathbf{A}}_{j,j}}\Big(\frac{1}{n_3}\big(\bar{\mathbf{L}}^{(k)}\widetilde{\mathbf{a}}_j - \widetilde{\mathbf{b}}_j\big) + \lambda_1 \widetilde{\mathbf{u}}_j\Big)$.
    end for
  end for
until convergence

Output: $\boldsymbol{\mathcal{L}} = \texttt{ifft}(\bar{\boldsymbol{\mathcal{L}}}, [\,], 3)$.

search the optimal $\overrightarrow{\bar{\mathbf{R}}}^{(k)}$ as well as the dual variable η using bisection method. To be concrete, we denote

$$\overrightarrow{\bar{\mathbf{R}}}^{(k)}(\eta) = (\bar{\mathbf{L}}_{t-1}^H \bar{\mathbf{L}}_{t-1} + \eta \mathbf{I}_r)^{-1} \bar{\mathbf{L}}_{t-1}^H (\overrightarrow{\bar{\mathbf{Z}}}_t^{(k)} - \overrightarrow{\bar{\boldsymbol{\mathcal{E}}}}^{(k)})$$

and maintain a lower bound η_1 and an upper bound η_2 such that $\|\overrightarrow{\bar{\mathbf{R}}}^{(k)}(\eta_1)\|_F \geq 1$ and $\|\overrightarrow{\bar{\mathbf{R}}}^{(k)}(\eta_2)\|_F \leq 1$. This ensures that the optimal η lies within the interval $[\eta_1, \eta_2]$ and we can find this value efficiently using bisection method outlined in Algorithm 5 in the Appendix. By comparing $\|\overrightarrow{\bar{\mathbf{R}}}^{(k)}(\eta)\|_F$ at $\eta = (\eta_1 + \eta_2)/2$ with 1, we can either increase η_1 or decrease η_2 until $\|\overrightarrow{\bar{\mathbf{R}}}^{(k)}(\eta)\|_F$ equals one. Note that $\|\overrightarrow{\bar{\mathbf{R}}}_{\text{cand}}^{(k)}\|_F > 1$, which implies that the optimal value for η is greater than ϵ, thus we can easily set $\eta_1 = 0$. Once $\overrightarrow{\boldsymbol{\mathcal{R}}}$ is found, we can update $\overrightarrow{\boldsymbol{\mathcal{E}}}$ using the soft-thresholding operator (Donoho, 1995): $\overrightarrow{\boldsymbol{\mathcal{E}}} = S_{\lambda_2}[\overrightarrow{\boldsymbol{\mathcal{Z}}}_t - \boldsymbol{\mathcal{L}}_{t-1} * \overrightarrow{\boldsymbol{\mathcal{R}}}]$.

Optimize $\boldsymbol{\mathcal{L}}$: When $\{\overrightarrow{\boldsymbol{\mathcal{Z}}}_i, \overrightarrow{\boldsymbol{\mathcal{R}}}_i^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}_i^\star\}_{i=1}^t$ are available, we can update the basis $\boldsymbol{\mathcal{L}}_t$ by optimizing the surrogate function

$$g_t(\boldsymbol{\mathcal{L}}) = \frac{1}{t}\sum_{i=1}^{t} \widetilde{\ell}(\overrightarrow{\boldsymbol{\mathcal{Z}}}_i, \boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{R}}}_i^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}_i^\star) + \frac{\lambda_1}{2t}\|\bar{\mathbf{L}}\|_{2,\infty}^2. \tag{13}$$

It is easy to verify that the minimizer of (14) is given by

$$\boldsymbol{\mathcal{L}}_t = \arg\min_{\boldsymbol{\mathcal{L}}} \frac{1}{t}\|\boldsymbol{\mathcal{Z}}_t - \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}_t^\star - \boldsymbol{\mathcal{E}}_t^\star\|_F^2 + \frac{\lambda_1}{2t}\|\bar{\mathbf{L}}\|_{2,\infty}^2,$$

where $\boldsymbol{\mathcal{Z}}_t = [\overrightarrow{\boldsymbol{\mathcal{Z}}}_1, \ldots, \overrightarrow{\boldsymbol{\mathcal{Z}}}_t] \in \mathbb{R}^{n_1 \times t \times n_3}$, $\boldsymbol{\mathcal{R}}_t^\star = [\overrightarrow{\boldsymbol{\mathcal{R}}}_1^\star, \ldots, \overrightarrow{\boldsymbol{\mathcal{R}}}_t^\star] \in \mathbb{R}^{r \times t \times n_3}$ and $\boldsymbol{\mathcal{E}}_t^\star = [\overrightarrow{\boldsymbol{\mathcal{E}}}_1^\star, \ldots, \overrightarrow{\boldsymbol{\mathcal{E}}}_t^\star] \in \mathbb{R}^{n_1 \times t \times n_3}$. Let $\bar{\mathbf{L}}_t = \texttt{bdiag}(\bar{\boldsymbol{\mathcal{L}}}_t)$, we obtain

$$\begin{aligned}\bar{\mathbf{L}}_t &= \arg\min_{\bar{\mathbf{L}}} \frac{1}{tn_3}\|\bar{\mathbf{Z}}_t - \bar{\mathbf{L}}\bar{\mathbf{R}}_t^\star - \bar{\mathbf{E}}_t^\star\|_F^2 + \frac{\lambda_1}{2t}\|\bar{\mathbf{L}}\|_{2,\infty}^2 \\ &= \arg\min_{\bar{\mathbf{L}}} \frac{1}{tn_3}\Big(\frac{1}{2}\mathrm{tr}\big(\bar{\mathbf{L}}^H\bar{\mathbf{L}}\bar{\mathbf{A}}_t\big) - \mathrm{tr}\big(\bar{\mathbf{L}}^H\bar{\mathbf{B}}_t\big)\Big) + \frac{\lambda_1}{2t}\|\bar{\mathbf{L}}\|_{2,\infty}^2.\end{aligned} \tag{14}$$

Here, $\boldsymbol{\mathcal{A}}_t = \sum_{i=1}^t \overrightarrow{\boldsymbol{\mathcal{R}}}_i^\star * \overrightarrow{\boldsymbol{\mathcal{R}}}_i^{\star T} \in \mathbb{R}^{r \times r \times n_3}$, $\boldsymbol{\mathcal{B}}_t = \sum_{i=1}^t (\overrightarrow{\boldsymbol{\mathcal{Z}}}_i - \overrightarrow{\boldsymbol{\mathcal{E}}}_i^\star) * \overrightarrow{\boldsymbol{\mathcal{R}}}_i^{\star T} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathrm{tr}(\cdot)$ denotes the trace operation. The subgradient of the squared $\ell_{2,\infty}$ norm can be computed as in Shen et al. (2017). To be specific, let $\boldsymbol{\Theta}$ denote the set of row indices corresponding to the rows with maximum ℓ_2 row norm of $\bar{\mathbf{L}}$. Define $\mathbf{Q} \in \mathbb{R}^{n_1 n_3 \times n_1 n_3}$ to be a positive semi-definite diagonal matrix with $\mathbf{Q}_{i,i} \neq 0$ if and only if $i \in \boldsymbol{\Theta}$ and all other entries are zeros such that $\sum_{i=1}^{n_1 n_3} \mathbf{Q}_{i,i} = 1$. The subgradient of $\frac{1}{2}\|\bar{\mathbf{L}}\|_{2,\infty}^2$ can be written as

$\mathbf{U} = \partial(\frac{1}{2}\|\bar{\mathbf{L}}\|_{2,\infty}^2) = \mathbf{Q}\bar{\mathbf{L}}$. We use block coordinate descent (Bertsekas, 1999) to update the columns of $\bar{\mathbf{L}}$ sequentially, see more details in Algorithm 3. Note that $\mathbf{U}$ is a diagonal matrix, we can write $\mathbf{U}$ as

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & & \\ & \ddots & \\ & & \mathbf{U}_{n_3} \end{bmatrix}.$$

In this manner, the subgradient of the squared $\ell_{2,\infty}$ norm of $\bar{\mathbf{L}}$ w.r.t. $\bar{\mathbf{L}}^{(k)}$ is $\mathbf{U}_k$. By assuming that the objective function of (14) is strongly convex w.r.t. $\bar{\mathbf{L}}$, it is guaranteed that the solution of this block coordinate descent scheme always converges to the global optimum.

4.3 Complexity and Convergence Analysis

Here, we provide further analysis on complexity and convergence of our method.

Computational Complexity. In each iteration, the computational burden is dominated by the cost for solving (10). The computational complexity of (11) involves computing the inverse of n_3 $r \times r$ matrices, matrix multiplications, and the (inverse) Fast Fourier Transform, totally $\mathcal{O}(n_1 r^2 n_3 + n_1 n_3 \log n_3)$. For the basis update, obtaining a subgradient of the squared $\ell_{2,\infty}$ norm of $\bar{\mathbf{L}}$ is $\mathcal{O}(n_1 r n_3)$ and one-pass update for the columns in $\bar{\mathbf{L}}$ in Algorithm 3 requires $\mathcal{O}(n_1 r^2 n_3)$.

Memory Cost. OMRTD requires $\mathcal{O}(n_1 r n_3)$ to load $\boldsymbol{\mathcal{L}}_{t-1}$ and $\overrightarrow{\boldsymbol{\mathcal{Z}}}_t$ to obtain $\{\overrightarrow{\boldsymbol{\mathcal{R}}}_t, \overrightarrow{\boldsymbol{\mathcal{E}}}_t\}$. To store the accumulation tensor $\boldsymbol{\mathcal{A}}_t$, we need $\mathcal{O}(r^2 n_3)$ memory while that for $\boldsymbol{\mathcal{B}}_t$ is $\mathcal{O}(n_1 r n_3)$. Finally, we find that only $\boldsymbol{\mathcal{A}}_t$ and $\boldsymbol{\mathcal{B}}_t$ are needed for the computation of the new iterate $\boldsymbol{\mathcal{L}}_t$. Hence, the memory cost of OMRTD is $\mathcal{O}(n_1 r n_3)$, i.e., independent of N, making our algorithm appealing for large-scale streaming tensor data.

Convergence. Three assumptions are made throughout our analysis: 1) The observed samples are generated independent identically distributed from some distribution and there exist two positive constants α_0, α_1, such that the conditions $\alpha_0 \leq \|\overrightarrow{\boldsymbol{\mathcal{Z}}}_t\|_F \leq \alpha_1$ holds almost surely for all $t \geq 1$; 2) The surrogate function $g_t(\boldsymbol{\mathcal{L}})$ in (13) is strongly convex. Particularly, we assume that the smallest singular value of the matrix $\bar{\mathbf{A}}_t$ is not smaller than some positive constant β_1; 3) The minimizer for $\ell(\boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{Z}}})$ is unique. Notice that $\widetilde{\ell}(\overrightarrow{\boldsymbol{\mathcal{Z}}}, \boldsymbol{\mathcal{L}}, \overrightarrow{\boldsymbol{\mathcal{R}}}, \overrightarrow{\boldsymbol{\mathcal{E}}})$ is strongly convex w.r.t. $\overrightarrow{\boldsymbol{\mathcal{E}}}$ and convex w.r.t. $\overrightarrow{\boldsymbol{\mathcal{R}}}$. We can enforce this assumption by adding a term $\frac{\epsilon}{2}\|\overrightarrow{\boldsymbol{\mathcal{R}}}\|_F^2$ to the objective function, where ϵ is a small positive constant. Following proof techniques in Shen et al. (2017) and using the expression $\boldsymbol{\mathcal{L}} * \overrightarrow{\boldsymbol{\mathcal{R}}} = \texttt{bvfold}(\texttt{bcirc}(\boldsymbol{\mathcal{L}}) \cdot \texttt{bvec}(\overrightarrow{\boldsymbol{\mathcal{R}}}))$, we derive several theoretical results: 1) The positive stochastic process $g_t(\boldsymbol{\mathcal{L}}_t)$ converges almost surely; 2) The numerical convergence rate of $\boldsymbol{\mathcal{L}}_t$ is nonasymptotic, i.e., $\|\boldsymbol{\mathcal{L}}_{t+1} - \boldsymbol{\mathcal{L}}_t\|_F = \mathcal{O}(1/t)$; 3) The empirical loss function, $f_t(\boldsymbol{\mathcal{L}}_t)$ defined in (8) converges almost surely to the same limit of its surrogate $g_t(\boldsymbol{\mathcal{L}}_t)$. Consequently, $f(\boldsymbol{\mathcal{L}}_t)$ converges almost surely; 4) $\boldsymbol{\mathcal{L}}_t$ converges to a stationary point of the expected loss function (9) when t tends to infinity.

4.4 Extension to Tensor Completion

In this subsection, we study OMRTD for the case of data having missing entries. To be specific, let $\boldsymbol{\mathcal{W}} = [0,1] \in \mathbb{R}^{n_1 \times N \times n_3}$ be a tensor such that $\boldsymbol{\mathcal{W}}_{i,j,k} = 1$ if $\boldsymbol{\mathcal{Z}}_{i,j,k}$ is observed and $\boldsymbol{\mathcal{W}}_{i,j,k} = 0$ otherwise. The locations of the observed entries can be indexed by a set $\boldsymbol{\Omega} = \{(i,j,k) : \boldsymbol{\mathcal{W}}_{i,j,k} = 1\}$. We reformulate the problem (2) for tensor completion by solving the following problem:

$$\min_{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}} \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{\mathcal{X}}\|_{\max}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}\|_1 \quad \text{s.t.} \quad P_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{M}}) = P_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{Z}}), \tag{15}$$

where $P_{\boldsymbol{\Omega}}(\cdot)$ is the orthogonal projector onto the span of tensors vanishing outside $\boldsymbol{\Omega}$ so that the (i,j,k)-th entry of $P_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{M}})$ is equal to $\boldsymbol{\mathcal{M}}_{i,j,k}$ if $(i,j,k) \in \boldsymbol{\Omega}$ and zero otherwise. We again minimize (15) by alternating minimization strategy. At the t-th iteration, when $\boldsymbol{\mathcal{L}}_{t-1}$ is given, the update for $\{\overrightarrow{\boldsymbol{\mathcal{M}}}_t, \overrightarrow{\boldsymbol{\mathcal{R}}}_t, \overrightarrow{\boldsymbol{\mathcal{E}}}_t\}$ corresponds

Algorithm 4 Updating tensor columns of $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{E}}$

Input: Partially observed data sample $P_{\boldsymbol{\Phi}}(\overrightarrow{\boldsymbol{\mathcal{Z}}}) \in \mathbb{R}^{n_1 \times 1 \times n_3}$, $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3}$, and parameter λ_2.
Initialize: $\overrightarrow{\boldsymbol{\mathcal{D}}} = P_{\boldsymbol{\Phi}}(\overrightarrow{\boldsymbol{\mathcal{Z}}})$, $\overrightarrow{\boldsymbol{\mathcal{R}}} = \overrightarrow{\boldsymbol{\mathcal{E}}} = \overrightarrow{\boldsymbol{\mathcal{J}}} = \mathbf{0}$, and $\mu = 0.1$.

1: **repeat**
2: $\quad \overrightarrow{\boldsymbol{\mathcal{M}}} = \frac{\boldsymbol{\mathcal{L}} * \overrightarrow{\boldsymbol{\mathcal{R}}} + \overrightarrow{\boldsymbol{\mathcal{E}}} + \mu \overrightarrow{\boldsymbol{\mathcal{D}}} - \overrightarrow{\boldsymbol{\mathcal{J}}}}{\mu+1}$.
3: $\quad$ Update $\{\overrightarrow{\boldsymbol{\mathcal{R}}}^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}^\star\}$ using Algorithm 2.
4: $\quad \overrightarrow{\boldsymbol{\mathcal{D}}} = \overrightarrow{\boldsymbol{\mathcal{M}}} + \frac{\overrightarrow{\boldsymbol{\mathcal{J}}}}{\mu}$, $P_{\boldsymbol{\Phi}}(\overrightarrow{\boldsymbol{\mathcal{D}}}) = P_{\boldsymbol{\Phi}}(\overrightarrow{\boldsymbol{\mathcal{Z}}})$.
5: $\quad \overrightarrow{\boldsymbol{\mathcal{J}}} = \overrightarrow{\boldsymbol{\mathcal{J}}} + \mu(\overrightarrow{\boldsymbol{\mathcal{M}}} - \overrightarrow{\boldsymbol{\mathcal{D}}})$.
6: **until** convergence

Output: Optimal $\overrightarrow{\boldsymbol{\mathcal{M}}}^\star$, $\overrightarrow{\boldsymbol{\mathcal{R}}}^\star$ and $\overrightarrow{\boldsymbol{\mathcal{E}}}^\star$.

to solving

$$\{\overrightarrow{\boldsymbol{\mathcal{M}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{R}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}_t^\star\} = \underset{\overrightarrow{\boldsymbol{\mathcal{M}}}, \overrightarrow{\boldsymbol{\mathcal{R}}} \in \mathscr{P}, \overrightarrow{\boldsymbol{\mathcal{E}}}}{\arg\min} \frac{1}{2}\|\overrightarrow{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{L}}_{t-1} * \overrightarrow{\boldsymbol{\mathcal{R}}} - \overrightarrow{\boldsymbol{\mathcal{E}}}\|_F^2 + \lambda_2 \|\overrightarrow{\boldsymbol{\mathcal{E}}}\|_1$$
$$\text{s.t. } P_{\boldsymbol{\Omega}_t}(\overrightarrow{\boldsymbol{\mathcal{M}}}) = P_{\boldsymbol{\Omega}_t}(\overrightarrow{\boldsymbol{\mathcal{Z}}}_t), \tag{16}$$

where $\boldsymbol{\Omega}_t = \{(i,t,k) | (i,t,k) \in \boldsymbol{\Omega}\}$. To solve (16) in an efficient manner, we resort to "variable splitting" of $\overrightarrow{\boldsymbol{\mathcal{M}}}$, which transforms (16) into the following:

$$\{\overrightarrow{\boldsymbol{\mathcal{M}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{D}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{R}}}_t^\star, \overrightarrow{\boldsymbol{\mathcal{E}}}_t^\star\} = \underset{\overrightarrow{\boldsymbol{\mathcal{M}}}, \overrightarrow{\boldsymbol{\mathcal{D}}}, \overrightarrow{\boldsymbol{\mathcal{R}}} \in \mathscr{P}, \overrightarrow{\boldsymbol{\mathcal{E}}}}{\arg\min} \frac{1}{2}\|\overrightarrow{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{L}}_{t-1} * \overrightarrow{\boldsymbol{\mathcal{R}}} - \overrightarrow{\boldsymbol{\mathcal{E}}}\|_F^2 + \lambda_2 \|\overrightarrow{\boldsymbol{\mathcal{E}}}\|_1$$
$$\text{s.t. } \overrightarrow{\boldsymbol{\mathcal{M}}} = \overrightarrow{\boldsymbol{\mathcal{D}}}, P_{\boldsymbol{\Omega}_t}(\overrightarrow{\boldsymbol{\mathcal{D}}}) = P_{\boldsymbol{\Omega}_t}(\overrightarrow{\boldsymbol{\mathcal{Z}}}_t). \tag{17}$$

This problem can now be solved by using the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). Specifically, the augmented Lagrangian function of (17) is

$$\widehat{\ell}(\overrightarrow{\boldsymbol{\mathcal{M}}}, \overrightarrow{\boldsymbol{\mathcal{D}}}, \overrightarrow{\boldsymbol{\mathcal{R}}}, \overrightarrow{\boldsymbol{\mathcal{E}}}) = \frac{1}{2}\|\overrightarrow{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{L}}_{t-1} * \overrightarrow{\boldsymbol{\mathcal{R}}} - \overrightarrow{\boldsymbol{\mathcal{E}}}\|_F^2 + \lambda_2 \|\overrightarrow{\boldsymbol{\mathcal{E}}}\|_1 + \langle \overrightarrow{\boldsymbol{\mathcal{J}}}, \overrightarrow{\boldsymbol{\mathcal{M}}} - \overrightarrow{\boldsymbol{\mathcal{D}}} \rangle + \frac{\mu}{2}\|\overrightarrow{\boldsymbol{\mathcal{M}}} - \overrightarrow{\boldsymbol{\mathcal{D}}}\|_F^2, \tag{18}$$

where $\overrightarrow{\boldsymbol{\mathcal{J}}}$ is the Lagrange multiplier and $\mu > 0$ is a penalty parameter. The implementation of the ADMM algorithm is outlined in Algorithm 4. Finally, we define $\boldsymbol{\mathcal{B}}_t = \sum_{i=1}^t (\overrightarrow{\boldsymbol{\mathcal{M}}}_i^\star - \overrightarrow{\boldsymbol{\mathcal{E}}}_i^\star) * \overrightarrow{\boldsymbol{\mathcal{R}}}_i^{\star^T}$ and the update of $\boldsymbol{\mathcal{L}}_t$ is exactly the same as in OMRTD. We dub this approach *robust OMRTD* (rOMRTD) in our experiments.

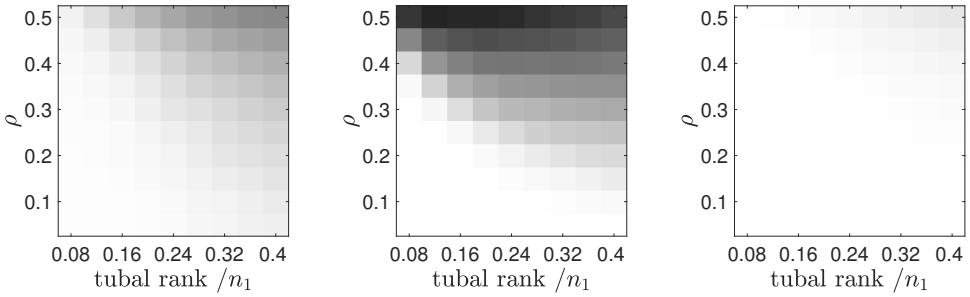
5 Experiments

In this section, we present several experimental results on synthetic data. All experiments are conducted on a PC with an AMD Ryzen 9 5950X 3.40GHz CPU and 64GB RAM with Matlab R2023b. We set $\lambda_1 = \lambda_2 = 1/\sqrt{n_1}$ for OMRTD/rOMRTD, and we follow the default parameter settings for the baselines.

Data generation We generate the clean data tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{V}}^T$, where the entries of $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{N \times r \times n_3}$ are drawn i.i.d. from $\mathcal{N}(0,1)$ distribution. Here, we set $n_1 = 50$ and $n_3 = 20$. The observed data tensor $\boldsymbol{\mathcal{Z}}$ is generated by $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}$, where $\boldsymbol{\mathcal{E}}$ is a sparse tensor with a fraction of ρ non-zero entries. The elements in $\boldsymbol{\mathcal{E}}$ are from a uniform distribution over the interval of $[-10, 10]$.

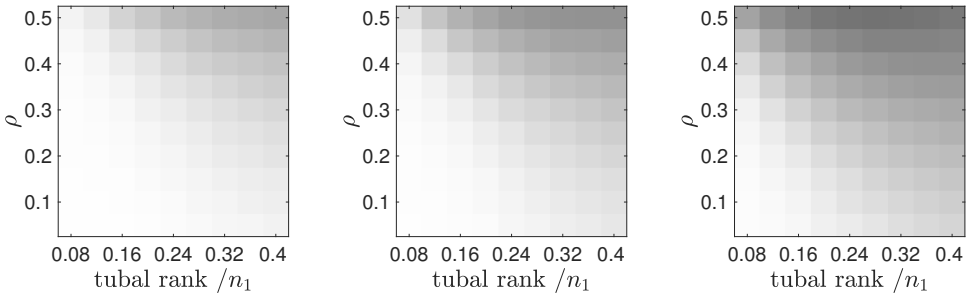
Evaluation metric We evaluate the fitness of the recovered tensor subspace $\boldsymbol{\mathcal{L}}$ (with each frontal slice being normalized) and the ground truth $\boldsymbol{\mathcal{U}}$ based on the idea of Expressed Variance (EV) (Xu et al., 2010):

$$\text{EV}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{L}}) = \frac{\|\boldsymbol{\mathcal{U}}^T * \boldsymbol{\mathcal{L}}\|_F^2}{\|\boldsymbol{\mathcal{U}}\|_F^2} = \frac{\text{tr}\big((\boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{U}}^T * \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{L}}^T)(:,:,1)\big)}{\text{tr}\big((\boldsymbol{\mathcal{U}}^T * \boldsymbol{\mathcal{U}})(:,:,1)\big)}.$$

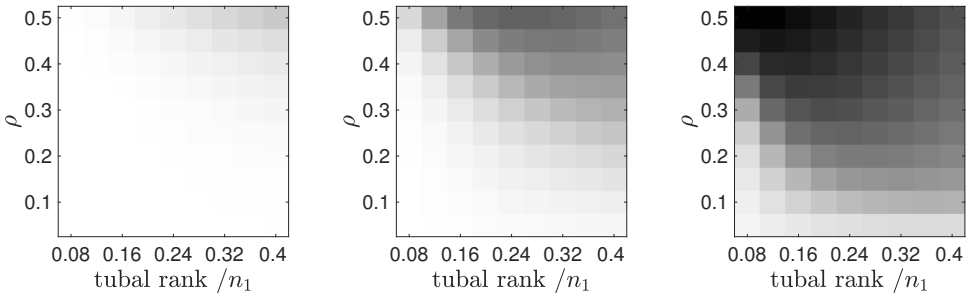


(a) OMRTD (b) OTRPCA (c) TRPCA

Figure 1: Performance of tensor subspace recovery using complete data under different intrinsic dimensions and corruptions. Brighter cells represent better performance.



(a) rOMRTD ($\tau = 10\%$) (b) rOMRTD ($\tau = 30\%$) (c) rOMRTD ($\tau = 50\%$)



(d) TNN+TRPCA ($\tau = 10\%$) (e) TNN+TRPCA ($\tau = 30\%$) (f) TNN+TRPCA ($\tau = 50\%$)

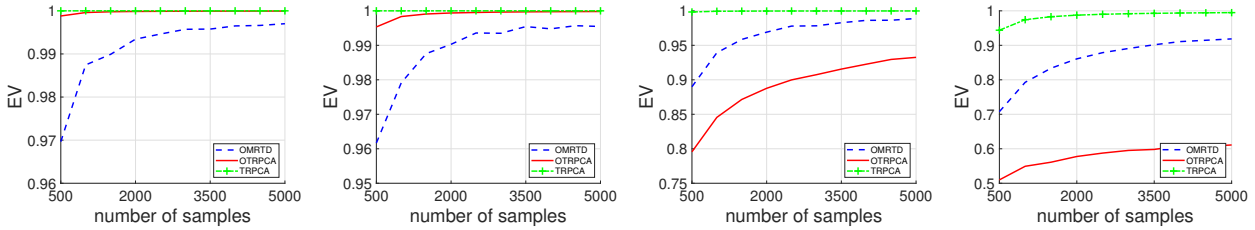
Figure 2: Performance of tensor subspace recovery using missing data under different intrinsic dimensions and corruptions. Brighter cells represent better performance.

The value of EV ranges between 0 and 1 and a higher value indicates better recovery. The Monte Carlo simulations are repeated 10 times and we report the averaged EV of these 10 random trials.

Baselines For the complete data experiments, we compare the performance of OMRTD with OTRPCA (Zhang et al., 2016) and TRPCA (Lu et al., 2020). For the case of data having missing entries, we choose an approach that first performs tensor completion based on the tensor nuclear norm (TNN) (Zhang & Aeron, 2017; Lu et al., 2018) and then conducts TRPCA (Lu et al., 2020) on the recovered data (TNN+TRPCA) as the baseline.

5.1 Robustness

We first study the robustness of OMRTD in terms of EV value, and compare it to the tensor nuclear norm based OTRPCA and the batch algorithm TRPCA. In this set of experiments, the total number of samples $N = 2000$. We vary the true tubal rank from $0.08n_1$ to $0.4n_1$, with a step size $0.04n_1$, and the corruption fraction ρ ranges from 0.05 to 0.5, with a step size 0.05. The results are represented in Figure 1. Since TRPCA accesses all the data in each iteration, it always achieves the best performance. We observe that



(a) $\rho = 0.05$ (b) $\rho = 0.1$ (c) $\rho = 0.3$ (d) $\rho = 0.5$

Figure 3: EV value against the number of samples under different corruption fractions.

both OMRTD and OTRPCA perform comparably in easy settings (i.e., few corruption and low tubal rank). However, in tough cases, OMRTD outperforms OTRPCA. For example, when the true tubal rank is 12 and $\rho = 0.5$, the EV values of OMRTD and OTRPCA are 0.8274 and 0.5840, respectively. In order to further investigate this phenomenon, we plot the EV curve against the fraction of corruption for some given tubal ranks in Figure 5 in the Appendix. Notably, when manipulating a low tubal rank tensor, OTRPCA exhibits similar performance compared to OMRTD under a low level of noise, e.g., the true tubal rank is no more than 8 and ρ is no more than 0.25. However, as the true tubal rank gets larger or the fraction of corruption increases, OTRPCA degrades faster than OMRTD. This is possibly because the proposed tensor max-norm is a tighter approximation to the tensor tubal rank.

Next, we study the effectiveness of rOMRTD in the missing data scenario, where we set the percentage of missing entries τ to be 10%, 30% and 50%. Figure 2 indicates that TNN+TRPCA performs better than rOMRTD when $\tau = 10\%$. When the number of missing entries increases, the EV values of TNN+TRPCA drop rapidly, especially when the fraction of corruption becomes large. Finally, rOMRTD outperforms TNN+TRPCA for almost all different tubal ranks and ρ's when $\tau = 50\%$. The detailed plot of the EV curve against the fraction of corruption under some specific tubal ranks for $\tau = 30\%$ is shown in Figure 6 in the Appendix.

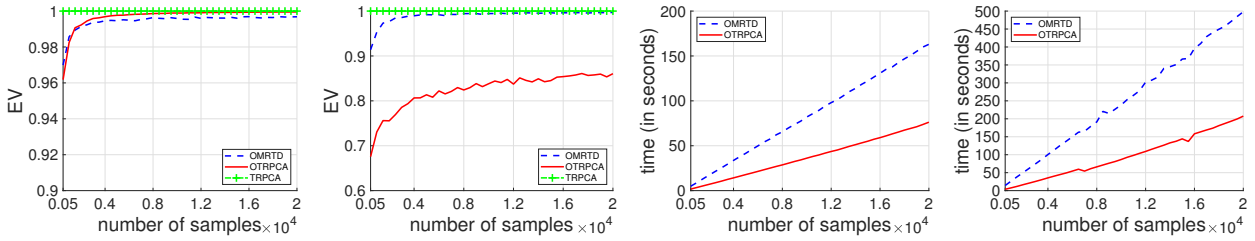
5.2 Convergence Rate

We now examine the convergence of OMRTD in terms of the EV curve as a function of the number of samples. We first fix the true tubal rank to be $0.2n_1 = 10$. The results are depicted in Figure 3. As expected, TRPCA achieves the best performance since it is a batch method and it requires to access all the data during optimization. OMRTD is comparable to OTRPCA when the corruption level is low (see Figure 3a and Figure 3b) and the gap between the EV values for these two methods is below 0.04. When data are grossly corrupted, OMRTD converges faster than OTRPCA (see Figure 3c and Figure 3d), which again suggests that tensor max-norm might be a better fit than the tensor nuclear norm when the signal to noise ratio is low.

We then compare the convergence rate of OMRTD and OTRPCA under different n_1's in Figure 4a and Figure 4b. The tubal rank of data is set to be $0.1n_1$ and the error corruption ρ is fixed to be 0.3. We observe that when $n_1 = 50$, OMRTD is generally slightly worse than OTRPCA and the gap between the EV values is below 0.004 for the same number of samples. However, when $n_1 = 100$, OMRTD significantly outperforms OTRPCA. It attains the EV value of 0.95 only with accessing 1000 samples, whereas OTRPCA cannot obtain the same accuracy even using 20000 samples.

5.3 Computational Complexity

As we discussed, when we solve the dual problem to optimize $\overrightarrow{\mathcal{R}}$, the initial solution $\overrightarrow{\mathcal{R}}_{\text{cand}}$ may violate the constraint. Thus, OMRTD is inferior to OTRPCA in terms of computation. We plot the running time with respect to the number of samples for $n_1 \in \{50, 100\}$ in Figure 4c and Figure 4d, which show that OTRPCA is about 2.4 times faster than OMRTD. When $n_1 = 50$, OMRTD and OTRPCA take 76 seconds to achieve the EV values of 0.9963 (with 9500 samples) and 0.9995 (with 20000 samples), respectively. The caveat here is that when $n_1 = 100$, OMRTD and OTRPCA take 207 seconds to achieve the EV values of around



(a) n1 = 50 (b) n1 = 100 (c) n1 = 50 (d) n1 = 100

Figure 4: (a)-(b) EV value as a function of the number of samples for $n_1 \in \{50, 100\}$. The intrinsic dimension is $0.1n_1$ and the corruption fraction $\rho = 0.3$. (c)-(d) Running time as a function of the number of samples for $n_1 \in \{50, 100\}$.

0.9945 (with a little bit more than 8000 samples) and 0.86 (with 20000 samples), respectively. We can even expect that the gap between the EV values of these two methods will get even larger as n_1 increases. It is reasonable to conclude that the advantage of OMRTD over OTRPCA in terms of convergence rate significantly outweighs the increase in computational complexity when n_1 is moderately large.

6 Conclusion

In this paper, we have developed a tensor max-norm based low-rank tensor model and developed an online algorithm for the max-norm regularized tensor decomposition (OMRTD) problem. The main idea of OMRTD is to reformulate the objective function of max-norm regularized tensor decomposition as a constrained problem using the tensor factorization form of the max-norm, which can be solved by stochastic optimization. We further extended the proposed method to the missing data scenario. Comprehensive simulations demonstrate the effectiveness of OMRTD and suggest that the tensor max-norm might be a tighter relaxation of the tensor average rank compared to the tensor nuclear norm.

References

Anima Anandkumar, Prateek Jain, Yang Shi, and U. N. Niranjan. Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 268–276, 2016.

Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust PCA via stochastic optimization. In *Advances in Neural Information Processing Systems*, pp. 404–412, 2013.

Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. TripleRank: Ranking semantic web data by tensor decomposition. In *Proceedings of the International Semantic Web Conference*, pp. 213–228, 2009.

Quanxue Gao, Pu Zhang, Wei Xia, Deyan Xie, Xinbo Gao, and Dacheng Tao. Enhanced tensor RPCA and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):2133–2140, 2021.

Kyle Gilman, Davoud Ataee Tarzanagh, and Laura Balzano. Grassmannian optimization for online tensor completion and tracking with the t-SVD. *IEEE Transactions on Signal Processing*, 70:2152–2167, 2022.

Donald Goldfarb and Zhiwei (Tony) Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 35(1):225–253, 2014.

Wei He, Hongyan Zhang, Liangpei Zhang, and Huanfeng Shen. Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):178–188, 2016.

Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6): 1–39, 2013.

Bo Huang, Cun Mu, Donald Goldfarb, and John Wright. Provable low-rank tensor recovery. *Optimization-Online*, 4252:2, 2014.

Hiroyuki Kasai. Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations. *Neurocomputing*, 347:177–190, 2019.

Henk A. L. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122, 2000.

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

Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

Misha E. Kilmer, Lior Horesh, Haim Avron, and Elizabeth Newman. Tensor-tensor algebra for optimal representation and compression of multiway data. *Proceedings of the National Academy of Sciences*, 118 (28), 2021.

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

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Jason Lee, Benjamin Recht, Ruslan Salakhutdinov, Nathan Srebro, and Joel A. Tropp. Practical large-scale optimization for max-norm regularization. In *Advances in Neural Information Processing Systems*, pp. 1297–1305, 2010.

Ping Li, Jiashi Feng, Xiaojie Jin, Luming Zhang, Xianghua Xu, and Shuicheng Yan. Online robust low-rank tensor modeling for streaming data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 30(4):1061–1075, 2019.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Exact low tubal rank tensor recovery from Gaussian measurements. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2504–2510, 2018.

Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2020.

Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing*, 63(10):2663–2677, 2015.

I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine learning*, pp. 713–719, 2005.

Bernardino Romera-Paredes and Massimiliano Pontil. A new convex relaxation for tensor completion. In *Advances in Neural Information Processing Systems*, pp. 2967–2975, 2013.

Jie Shen, Huan Xu, and Ping Li. Online optimization for max-norm regularization. *Machine Learning*, 106: 419–457, 2017.

Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proceedings of the Annual Conference on Learning Theory*, pp. 545–560, 2005.

Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 1329–1336, 2004.

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Huan Xu, Constantine Caramanis, and Shie Mannor. Principal component analysis with contaminated data: The high dimensional case. In *Proceedings of the Annual Conference on Learning Theory*, pp. 490–502, 2010.

Jing-Hua Yang, Xi-Le Zhao, Teng-Yu Ji, Tian-Hui Ma, and Ting-Zhu Huang. Low-rank tensor train for tensor robust principal component analysis. *Applied Mathematics and Computation*, 367:124783, 2020.

Rose Yu, Dehua Cheng, and Yan Liu. Accelerated online low-rank tensor learning for multivariate spatio-temporal streams. In *Proceedings of the International Conference on Machine learning*, pp. 238–247, 2015.

Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2017.

Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 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*, pp. 3842–3849, 2014.

Zemin Zhang, Dehong Liu, Shuchin Aeron, and Anthony Vetro. An online tensor robust PCA algorithm for sequential 2D data. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2434–2438, 2016.

Pan Zhou, Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Tensor low-rank representation for data recovery and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5): 1718–1732, 2021.

A Appendix

A.1 Proof of Proposition 1

Proof. Let us denote $\kappa = \|\bar{\mathbf{R}}\|_{2,\infty}$, which is positive as long as $\boldsymbol{\mathcal{R}}$ is not a zero tensor. Otherwise the recovered low tubal-rank component $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}^T$ is a zero tensor. Now we construct two variables $\widetilde{\boldsymbol{\mathcal{L}}} = \kappa \boldsymbol{\mathcal{L}}$

Algorithm 5 Bisection Method for Solving Problem (12)

Input: $\mathbf{L} \in \mathbb{C}^{n_1 \times r}$, $\mathbf{z} \in \mathbb{C}^{n_1}$, $\mathbf{e} \in \mathbb{C}^{n_1}$.

Initialize: Set $\eta_1 = 0$ and η_2 to be large enough such that $\|\overrightarrow{\overline{\mathbf{R}}}^{(k)}(\eta_2)\|_F \leq 1$.

```
repeat
    Compute the middle point: η = ½(η_1 + η_2).
    if ‖R⃗̄^(k)(η)‖_F < 1 then
        Update η_2 = η.
    else
        Update η_1 = η.
    end if
until ‖R⃗̄^(k)‖_F = 1
```

Output: Optimal $\overrightarrow{\overline{\mathbf{R}}}^{(k)}$ and η.

and $\widetilde{\boldsymbol{\mathcal{R}}} = \frac{1}{\kappa}\boldsymbol{\mathcal{R}}$, and replace $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{R}}$ in (5) with $\frac{1}{\kappa}\widetilde{\boldsymbol{\mathcal{L}}}$ and $\kappa\widetilde{\boldsymbol{\mathcal{R}}}$, respectively. Our problem can be written as

$$\min_{\widetilde{\boldsymbol{\mathcal{L}}},\widetilde{\boldsymbol{\mathcal{R}}},\boldsymbol{\mathcal{E}}} \frac{1}{2}\|\boldsymbol{\mathcal{Z}} - (\frac{1}{\kappa}\widetilde{\boldsymbol{\mathcal{L}}}) * (\kappa\widetilde{\boldsymbol{\mathcal{R}}})^T - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2}\|\frac{1}{\kappa}\overline{\widetilde{\mathbf{L}}}\|_{2,\infty}^2\|\kappa\overline{\widetilde{\mathbf{R}}}\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}\|_1,$$

which is equivalent to solving

$$\min_{\widetilde{\boldsymbol{\mathcal{L}}},\widetilde{\boldsymbol{\mathcal{R}}},\boldsymbol{\mathcal{E}}} \frac{1}{2}\|\boldsymbol{\mathcal{Z}} - \widetilde{\boldsymbol{\mathcal{L}}} * \widetilde{\boldsymbol{\mathcal{R}}}^T - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2}\|\overline{\widetilde{\mathbf{L}}}\|_{2,\infty}^2\|\overline{\widetilde{\mathbf{R}}}\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}\|_1.$$

Since $\widetilde{\boldsymbol{\mathcal{R}}} = \frac{1}{\kappa}\boldsymbol{\mathcal{R}}$, we have $\|\overline{\widetilde{\mathbf{R}}}\|_{2,\infty} = \frac{1}{\kappa}\|\overline{\mathbf{R}}\|_{2,\infty} = 1$. Therefore, we can reformulate our MRTD problem as a constrained program:

$$\min_{\widetilde{\boldsymbol{\mathcal{L}}},\widetilde{\boldsymbol{\mathcal{R}}},\boldsymbol{\mathcal{E}}} \frac{1}{2}\|\boldsymbol{\mathcal{Z}} - \widetilde{\boldsymbol{\mathcal{L}}} * \widetilde{\boldsymbol{\mathcal{R}}}^T - \boldsymbol{\mathcal{E}}\|_F^2 + \frac{\lambda_1}{2}\|\overline{\widetilde{\mathbf{L}}}\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}\|_1 \quad \text{s.t.} \quad \|\overline{\widetilde{\mathbf{R}}}\|_{2,\infty}^2 = 1.$$

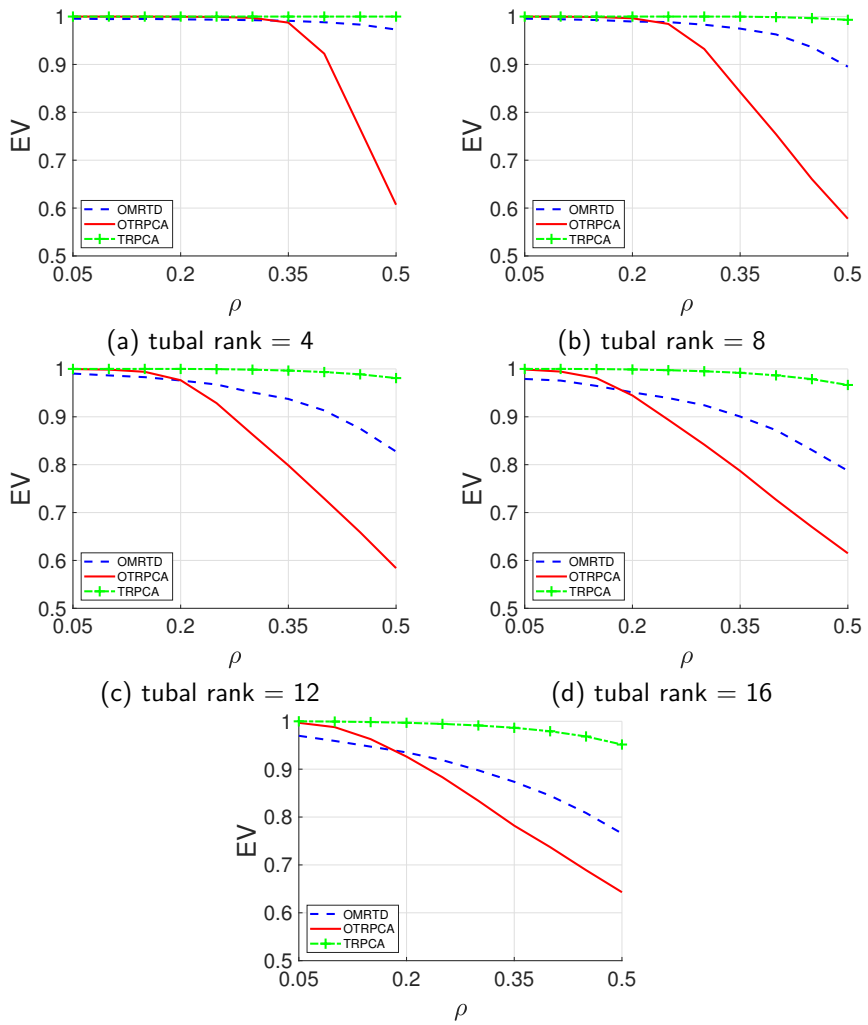
To see why the above problem is equivalent to (6), we only need to show that any optimal solution $(\boldsymbol{\mathcal{L}}^\star, \boldsymbol{\mathcal{R}}^\star, \boldsymbol{\mathcal{E}}^\star)$ to (6) must satisfy $\|\overline{\mathbf{R}}^\star\|_{2,\infty} = 1$. Again, suppose that $\kappa = \|\overline{\mathbf{R}}^\star\|_{2,\infty} < 1$. Let $\boldsymbol{\mathcal{L}}' = \kappa\boldsymbol{\mathcal{L}}^\star$ and $\boldsymbol{\mathcal{R}}' = \frac{1}{\kappa}\boldsymbol{\mathcal{R}}^\star$. It is clear that $(\boldsymbol{\mathcal{L}}', \boldsymbol{\mathcal{R}}', \boldsymbol{\mathcal{E}}^\star)$ are still feasible. However, the objective value now becomes

$$\begin{aligned}
&\frac{1}{2}\|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{L}}' * \boldsymbol{\mathcal{R}}'^T - \boldsymbol{\mathcal{E}}^\star\|_F^2 + \frac{\lambda_1}{2}\|\overline{\mathbf{L}}'\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}^\star\|_1 \\
=&\frac{1}{2}\|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{L}}^\star * \boldsymbol{\mathcal{R}}^{\star^T} - \boldsymbol{\mathcal{E}}^\star\|_F^2 + \frac{\lambda_1}{2}\kappa^2\|\overline{\mathbf{L}}^\star\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}^\star\|_1 \\
<&\frac{1}{2}\|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{L}}^\star * \boldsymbol{\mathcal{R}}^{\star^T} - \boldsymbol{\mathcal{E}}^\star\|_F^2 + \frac{\lambda_1}{2}\|\overline{\mathbf{L}}^\star\|_{2,\infty}^2 + \lambda_2\|\boldsymbol{\mathcal{E}}^\star\|_1,
\end{aligned}$$

which contradicts the assumption that $(\boldsymbol{\mathcal{L}}^\star, \boldsymbol{\mathcal{R}}^\star, \boldsymbol{\mathcal{E}}^\star)$ is optimal. Thus we complete the proof. □

A.2 Algorithm Details

A.3 Supplementary Experimental Results



(a) tubal rank = 4

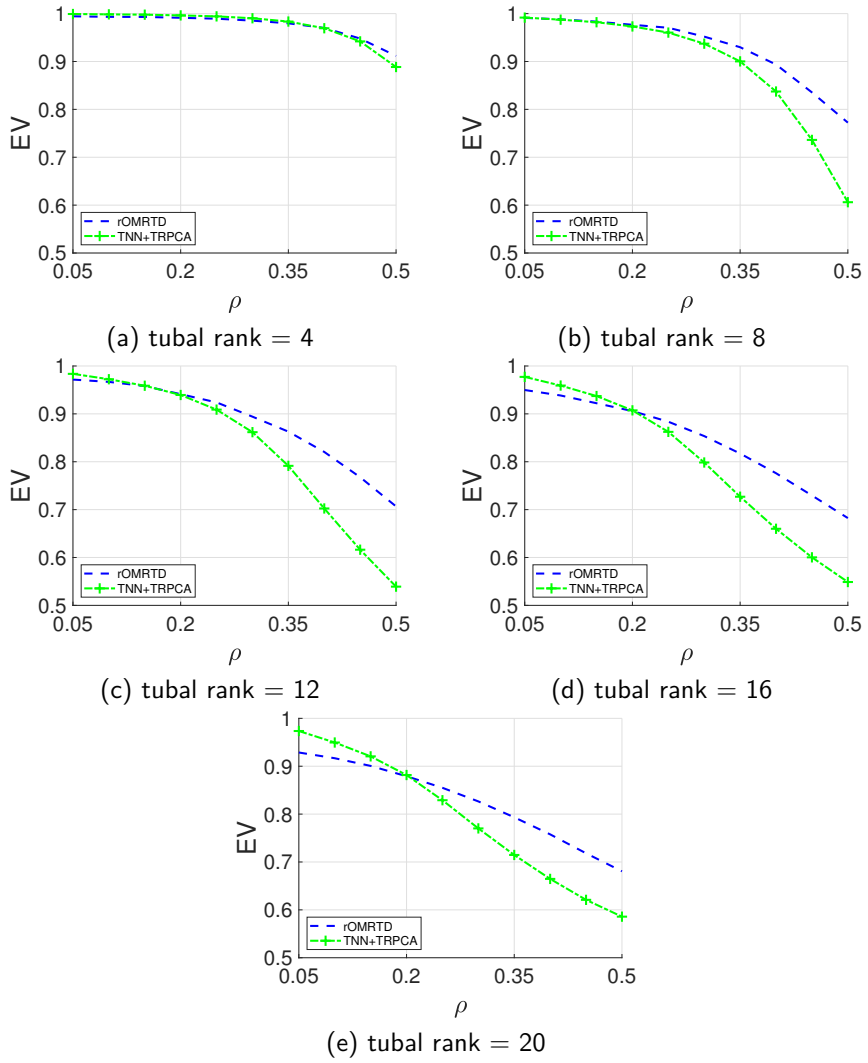
(b) tubal rank = 8

(c) tubal rank = 12

(d) tubal rank = 16

(e) tubal rank = 20

Figure 5: EV value as a function of corruption fraction for different intrinsic dimensions of complete data.



(a) tubal rank = 4

(b) tubal rank = 8

(c) tubal rank = 12

(d) tubal rank = 16

(e) tubal rank = 20

Figure 6: EV value as a function of corruption fraction for different intrinsic dimensions of missing data when $\tau = 30\%$.