IISE Transactions

Taylor & Francis
Taylor & Francis Group

IISE Transactions

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uiie21

Robust tensor-on-tensor regression for multidimensional data modeling

Hung Yi Lee, Mostafa Reisi Gahrooei, Hongcheng Liu & Massimo Pacella

To cite this article: Hung Yi Lee, Mostafa Reisi Gahrooei, Hongcheng Liu & Massimo Pacella (2024) Robust tensor-on-tensor regression for multidimensional data modeling, IISE Transactions, 56:1, 43-53, DOI: 10.1080/24725854.2023.2183440

To link to this article: https://doi.org/10.1080/24725854.2023.2183440

IISE TRANSACTIONS
2024, VOL. 56, NO. 1, 43–53
https://doi.org/10.1080/24725854.2023.2183440

Robust tensor-on-tensor regression for multidimensional data modeling

Hung Yi Lee[a], Mostafa Reisi Gahrooei[a], Hongcheng Liu[a], and Massimo Pacella[b]

[a]Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA; [b]Department of Innovation Engineering, University of Salento, Lecce, Italy

ABSTRACT

In recent years, high-dimensional data, such as waveform signals and images have become ubiquitous. This type of data is often represented by multiway arrays or tensors. Several statistical models, including tensor regression, have been developed for such tensor data. However, these models are sensitive to the presence of arbitrary outliers within the tensors. To address the issue, this article proposes a Robust Tensor-On-Tensor (RTOT) regression approach, which has the capability of modeling high-dimensional data when the data is corrupted by outliers. Through several simulations and case studies, we evaluate the performance of the proposed method. The results reveal the advantage of the RTOT over some benchmarks in the literature in terms of estimation error. A Python implementation is available at https://github.com/Reisi-Lab/RTOT.git.

ARTICLE HISTORY
Received 24 November 2021
Accepted 7 February 2023

KEYWORDS
Alternating direction method of multipliers; PARAFAC/CANDECOMP; Tensor decomposition; Robust principal component analysis

1. Introduction

In recent years, multi-dimensional arrays, or so-called *tensors*, have played an important role in the analysis of many real-world applications, such as manufacturing (Fang *et al.*, 2019; Wahba *et al.*, 2019; Yan *et al.*, 2019; Gahrooei *et al.*, 2021), healthcare (Zhou *et al.*, 2013; Zhao *et al.*, 2019; Zhou and Kan, 2021), and agriculture (Kanning *et al.*, 2018; Li *et al.*, 2020). The popularity of tensors is mainly due to their capability to preserve structural information of high-dimensional data compared with traditional vector forms. That is, unlike vectors that break the spatial and temporal structure of high-dimensional data (e.g., multichannel profiles and images), tensors preserve these structures. Therefore, the extension of classic data analytics methods from a vector to tensors results in more accurate estimations when structured high-dimensional data is available. For example, in semiconductor manufacturing, a large number of correlated temporal sensing data (profiles) may be represented by tensors to estimate the yield or a quality characteristic of a wafer (Gahrooei *et al.*, 2021; Wang *et al.*, 2021). As another example, in prognostics, a set of thermal images collected over time from a rotary machine can be represented by tensors for the prediction of the remaining lifetime of a machine (Fang *et al.*, 2019). Also, Electroencephalography (EEG) signals produce multichannel data that may be represented by tensors (Naskovska *et al.*, 2017). Many tensor data analytical methods have been developed in the past few years, including tensor regression models, which are the focus of this article.

Tensor regression takes many different variations depending on the form of the inputs or output. Scalar-on-tensor regression models (Zhao *et al.*, 2012; Fang *et al.*, 2019) estimate a scalar response given a tensor input. Tensor-on-scalar techniques (Yan *et al.*, 2019) take a set of scalar inputs to estimate a tensor response. And, finally, tensor-on-tensor models estimate a tensor response from a single or multiple tensor inputs (Xue *et al.*, 2017; Lock, 2018; Liu *et al.*, 2020; Gahrooei *et al.*, 2021). These techniques have been used in different applications, including prediction of neurological disorders (Zhou *et al.*, 2013), prediction and control of a manufacturing turning process (Yan *et al.*, 2019), and estimation of the overlay errors in semiconductor manufacturing (Gahrooei *et al.*, 2021). The main challenge in developing these techniques is dealing with a large number of parameters (due to the high dimensionality of data) that may result in severe overfitting while capturing the structural attributes of the data. For this purpose, tensor regression models include a low rankness constraint, for example, by introducing a low-rank decomposition on the tensor of parameters (Zhou *et al.*, 2013). Among these decomposition methods, the PARAFAC/CANDECOMP (CP) decomposition (Harshman, 1970; Kolda and Bader, 2009), in which the original tensor is represented as a linear sum of rank-1 tensors, is commonly used.

In addition to the challenges caused by high dimensionality of data, the potential presence of outliers (i.e., gross corruption of observations (Candès *et al.*, 2011)) in tensors is another obstacle that needs to be overcome. These gross outliers are common in many modern applications, such as road traffic data and manufacturing processes, where some measurements may be corrupted (Candes *et al.*, 2011; Kaur and Datta, 2019; Hu and Work, 2020; Hullait *et al.*, 2021). For example, according to Hullait *et al.* (2021) around 1-5% of data collected from jet engines during the pass-off test is contaminated by outliers. Similarly, Hu and Work (2020)

CONTACT Mostafa Reisi Gahrooei mreisigahrooei@ufl.edu
Supplemental data for this article can be accessed online at https://doi.org/10.1080/24725854.2023.2183440.

Table 1. Summary of the literature and the relevance of this work.

	Tensor Decomposition	*Tensor Regression*
Not Robust	Decomposes a tensor into a low-rank tensor and a tensor of identically distributed noise (Kolda and Bader, 2009)	Estimates an output tensor $\mathcal{Y}$ given an input tensor (or tensors) $\mathcal{X}$ by assuming a linear model without considering contamination within data, e.g., (Zhao *et al.*, 2012; Lock, 2018; Gahrooei *et al.*, 2021; Wang *et al.*, 2021).
Robust	Decomposes a tensor into a low-rank tensor ($\mathcal{L}$), a sparse tensor of contamination ($\mathcal{S}$) and noise. The sparse tensor $\mathcal{S}$ captures the contamination beyond overall noise in the data (Hu and Work, 2020; Li *et al.*, 2019; Xue *et al.*, 2017; Goldfarb and Qin, 2014; Candès *et al.*, 2011; Huang and Ding, 2008)	This work: Trains a model under the scenario that output tensors in the training data may contain contaminated observations beyond general noise.

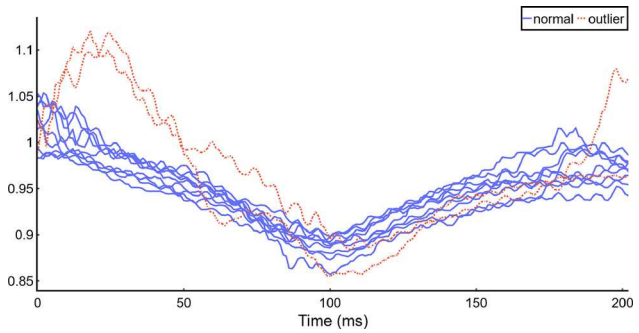

Figure 1. Example of normal and abnormal λ curves. The goal is to construct a model that predicts λ curves given a set of other curves acquired by sensors of an engine.

reported about 1.2% contamination in traffic data. To identify and isolate corrupted observations within a tensor, robust tensor recovery and decomposition techniques have been previously developed that decompose a tensor, $\mathcal{X}$, into a summation of a low-rank and sparse tensors (i.e., $\mathcal{X} = \mathcal{L} + \mathcal{S}$) by imposing low-rankness and sparsity penalties such as nuclear-norm and L_1 regularization on the decomposition components, $\mathcal{L}$ and $\mathcal{S}$ (Huang and Ding, 2008; Goldfarb and Qin, 2014; Xue *et al.*, 2017; Hu and Work, 2020) Nevertheless, these recovery techniques identify the corruptions based on the spatio-temporal structure of a tensor and are not applicable to a tensor regression setting, as they do not consider the input-output correlations when performing decomposition. More specifically, in a tensor-on-tensor regression setting, robust tensor recovery approaches may erroneously consider observations in the output tensor as outliers, even though they are explainable by the input tensor. Table 1 summarizes the literature and the position of this article in relevance to the existing literature.

Let us consider the problem of estimating the λ-curve (an indicator of vehicle exhaust emission) of a vehicle engine from several sensor readings. This problem can be formulated as a tensor-on-tensor regression (Gahrooei *et al.*, 2021). Figure 1 illustrates several examples of λ-curves (each curve corresponds to an output of a sampled engine), in which the solid curves represent the normal ones and dashed curves show the ones with outliers. One approach to constructing a model to estimate the λ-curve is to omit corrupted curves and construct the model only based on presumably normal ones. Nevertheless, this approach eliminates potential useful information that otherwise could have improved the model performance. Alternatively, one can study λ-curves and decompose a curve as a summation of normal and outlier parts ($\mathcal{L}$ and $\mathcal{S}$), then, uses the normal part of the curves for model constructions. However, this decomposition is unsupervised and ignores the input–output relations. That is, the decomposition of the λ-curve may depend on the other sensor readings (curves) that are the predictors of the output λ-curve. These challenges motivate us to develop a robust tensor-on-tensor regression approach that performs modeling and outlier isolation simultaneously. This proposed approach learns the model by automatically separating the outliers.

Existing tensor regression approaches (Zhao *et al.*, 2012; Lock, 2018; Gahrooei *et al.*, 2021; Wang *et al.*, 2021) produce biased models when the training data contains samples in which the output tensors are grossly corrupted (some entries of the tensor are contaminated), as they assume the elements of error tensor are identically distributed. In the situations where the data contains outliers (similar to the λ-curve example), or large arbitrary noise, this assumption is not valid, resulting in biased predictions. Our approach decomposes the error terms into two terms, one that captures the sparse corruptions with arbitrary distributions and the other captures the noise. This approach significantly improves the flexibility of the tensor models and allows for more accurate predictions. In addition, the proposed approach allows for the use of corrupted or faulty historical data, which increases the training data size and improves the model generalizability.

In this article, we develop a regression model to predict a tensor of arbitrary dimensions $P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}$ from another tensor of arbitrary dimensions $P_1 \times P_2 \times \cdots \times P_L$, particularly when the output tensor (responses) is corrupted by gross outliers. We assume the input tensors are not contaminated by gross outliers without loss of generality. This assumption is reasonable because the output should not be explainable by the gross corruptions within the input, and therefore, the input corruptions should be removed via existing pre-processing schemes. To further explain this point, please note that the input tensor is an independent random variable (tensor variable) and the corrupted entries within the input tensor (of a sample) can be identified by investigating the correlation structure of that tensor. On the other hand, the output tensor depends on the input one. Therefore, the corrupted elements within an output tensor should be identified by investigating both the correlation structure within the tensor and its relationship with the input. To further justify why outlier detection should be done

simultaneously with the regression when the output tensor is corrupted, let us consider the following scenario: Let $\mathcal{X}$ be an input tensor and $\mathcal{B}_1$ and $\mathcal{B}_2$ be a dense and a highly sparse (but structured) tensors, respectively. For example, $\mathcal{B}_2$ is a tensor with a few blocks of non-zero values. Now, assume $\mathcal{Y} = \langle \mathcal{X}, \mathcal{B}_1 \rangle_L + \langle \mathcal{X}, \mathcal{B}_2 \rangle_L + \mathcal{S} + \mathcal{E}$, where $\langle \mathcal{X}, \mathcal{B} \rangle_L$ is contract tensor product of $\mathcal{X}$ and $\mathcal{B}$ (see Section 2) and $\mathcal{S}$ is a sparse tensor of corruptions. If we apply existing algorithms for separating outliers within the output tensor, $\langle \mathcal{X}, \mathcal{B}_2 \rangle_L + \mathcal{S}$ is mostly extracted as outliers, because $\mathcal{B}_2$ is highly sparse and existing techniques for outlier detection in tensors rely on global correlation structures. This will reduce the predictive power of the subsequent regression model.

To estimate the parameters of a tensor-on-tensor regression model where the noisy output tensor data is contaminated by gross outliers, we combine the Alternating Direction Method of Multipliers (ADMM) with the Block Coordinate Descent algorithm (BCD) algorithms. When solving the problem, we take advantage of the *contract tensor product* from Lock (2018), tensor nuclear norm, and CP decomposition. To evaluate the performance of the proposed method, we provide a simulation study containing 27 different scenarios and two case studies. The case study estimates the lambda curve (an indication of the polluting performance of a vehicle engine) based on a collection of other operational sensor measurements taken on the engine (Gahrooei *et al.*, 2019).

The rest of this article is organized as follows: Section 2 introduces some notations and multi-linear algebra used in the article. Section 3 reviews the tensor-on-tensor regression. Section 4 discusses the formulation of robust tensor-on-tensor and the optimization algorithm for robust parameter estimation. Section 5 provides the simulation results using synthetic data. Section 6 reports the results of case studies. Finally, Section 7 summarizes the article.

2. Notations and preliminaries of multilinear algebra

In this section, we introduce notations and basic tensor algebra used in this article. Throughout the article, we denote a scalar by a lower case letter, for example, a; a vector or a matrix by boldface lower and upper case letter, for example, $\mathbf{a}$ and $\mathbf{A}$, respectively; and a tensor by a calligraphic letter, for example, $\mathcal{A}$. Given a tensor $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2, \ldots \times P_L \times P_{L+1} \cdots \times P_{L+M}}$, we denote the mode-$j$ matricization of $\mathcal{B}$ by $\mathcal{B}^{(j)} \in \mathbb{R}^{P_j \times P_{-j}}$, where $P_{-j} = P_1 \times P_2 \times \cdots \times P_{j-1} \times P_{j+1} \times \cdots \times P_{L+M}$. This matricization is obtained by augmenting the jth mode fibers, where tensor fibers are defined by fixing all but one index of a tensor. We also denote the operator $vec(\cdot)$ as vectorization operator which unfolds the input tensor into its corresponding column vector. The Frobenius norm of a tensor $\mathcal{X}$ is the square root of the sum of the squares of all its elements, denoted as $||\mathcal{X}||_F$, which can be calculated as $||\mathcal{X}||_F = ||\mathcal{X}^{(1)}||_F$. The nuclear norm of a tensor $\mathcal{X}$ is denoted by $||\mathcal{X}||_*$ and is computed as the weighted sum the nuclear norm of the tensor matricizations along all modes: $||\mathcal{X}||_* = \sum_{i=1}^{m} ||\mathcal{X}^{(i)}||_*$, where m is the order of the tensor. The nuclear norm of a matrix is computed as the sum of the singular values of that matrix. Following Lock (2018), the *contract tensor product* is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle_{K,(p_1,p_2,\ldots,p_L,q_1,q_2,\ldots,q_M)} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_K=1}^{I_K} \mathcal{X}_{p_1,p_2,\ldots,p_L,i_1,i_2,\ldots,i_K} \mathcal{Y}_{i_1,i_2,\ldots,i_K,q_1,q_2,\ldots,q_M},$$

where $\mathcal{X} \in \mathbb{R}^{P_1 \times P_2 \times \cdots \times P_L \times I_1 \times I_2 \times \cdots \times I_K}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K \times Q_1 \times Q_2 \times \cdots \times Q_M}$. Note that for $\mathbf{X} \in \mathbb{R}^{P \times I}$ and $\mathbf{Y} \in \mathbb{R}^{I \times Q}$,

$$\langle \mathbf{X}, \mathbf{Y} \rangle_1 = \mathbf{XY}.$$

3. Tensor-on-tensor regression

In this section, we review the Tensor-On-Tensor (TOT) regression proposed by Lock (2018). As we mentioned earlier, tensor regression takes many forms depending on its inputs/outputs. In this article, we particularly focus on a regression problem whose input and output are tensors and hence, named TOT regression. Let $\mathcal{X} \in \mathbb{R}^{P_1 \times P_2 \times \cdots \times P_L}$ and $\mathcal{Y} \in \mathbb{R}^{P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}}$. Then, the TOT considers the following linear model:

$$\mathcal{Y} = \langle \mathcal{X}, \mathcal{B} \rangle_L + \mathcal{E}. \tag{R1}$$

Here, $\mathcal{B}$ is the tensor of model parameters and $\mathcal{E}$ is the tensor of model errors. Like most of the high-order models, TOT also suffers from the curse of dimensionality; i.e., the model is prone to overfitting if no constraints are considered on the tensor of model parameters. One main constraint considered is that the tensor of parameters is low-rank, and therefore, can be decomposed into the product of several low-dimension matrices. Two commonly used techniques in tensor decomposition are Tucker decomposition (Tucker, 1966) and CP decomposition (Kolda and Bader, 2009). The former decomposes the input tensor as a multiplication of a core tensor, whose dimensionality is much smaller than the original one, and a series of factor matrices, whereas the latter (CP decomposition) is a special case of the former where the core tensor is diagonal. These decomposition techniques reduce the dimensionality of the tensor and hence, alleviate the burden of overfitting. For example, Lock (2018) considers a low-rank decomposition of the parameter tensor as follows:

$$\mathcal{B} = \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M} = \sum_{\gamma=1}^{R} u_1^{\gamma} \circ u_2^{\gamma} \circ \cdots \circ u_{L+M}^{\gamma}, \tag{1}$$

$$\mathcal{B}^* = \arg\min_{\mathcal{B}} ||\mathcal{Y} - \langle \mathcal{X}, \mathcal{B} \rangle_L||_F^2 + \lambda ||\mathcal{B}||_F^2, \tag{2}$$

where $\mathbf{U}_{L+M} \in \mathbb{R}^{P_j \times R}$ whose γ^{th} column is u_j^{γ}, and $u_1 \circ u_2 \cdots \circ u_{L+M}$ denote the outer product of vectors and is defined as $(u_1 \circ u_2 \cdots \circ u_{L+M})_{i_1 i_2 \cdots i_{L+M}} = (u_1)_{i_1} (u_2)_{i_2} \cdots (u_{L+M})_{i_{L+M}}$. Together with the tensor decomposition, a low-rankness penalty based on the nuclear norm (a convex surrogate of the rank of a tensor) is also commonly used to further regularize the rank of the tensor (Shang *et al.*, 2014;

Dian *et al.*, 2019; Liu *et al.*, 2020). The nuclear norm is tightly related to the tensor decomposition techniques, as it uses them for rank estimation. In this article, we employ the CP decomposition to reduce the dimension of the model parameter tensor and impose the nuclear norm to regularize the rank of it so that the low rankness resolves the overfitting.

4. Robust TOT Regression

In this section, we introduce the Robust TOT (RTOT) regression framework to construct a robust model based on training data with contaminated output tensors. Given a set of training data $\{(\mathcal{X}_i, \mathcal{Y}_i)|\ \mathcal{X}_i \in \mathbb{R}^{P_1 \times P_2 \times \cdots \times P_L}$ and $\mathcal{Y}_i \in \mathbb{R}^{P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}}\}_{i=1}^N$ in which $\{\mathcal{Y}_i\}_{i=1}^N$ are contaminated by outliers, the goal of RTOT is to estimate the relationship between the input tensor and the response, while extracting and detecting the outliers within the output using the following linear form:

$$\mathcal{Y}_i = \langle \mathcal{X}_i, \mathcal{B} \rangle_L + \mathcal{S}_i + \mathcal{E}_i, \tag{M1}$$

where $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2 \times \cdots \times P_{L+M}}$ is the tensor of parameters, $\mathcal{S}_i \in \mathbb{R}^{P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}}$ is a sparse tensor representing the outliers, and $\mathcal{E}_i$ is the tensor of errors. A more compact formulation of model (M1) can be obtained by folding the tensors into ones with one extra mode, containing all samples. That is, we construct the output tensor $\mathcal{Y} \in \mathbb{R}^{N \times P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}}$ to be estimated by input tensor $\mathcal{X} \in \mathbb{R}^{N \times P_1 \times P_2 \times \cdots \times P_L}$ via a linear model as follows:

$$\mathcal{Y} = \langle \mathcal{X}, \mathcal{B} \rangle_L + \mathcal{S} + \mathcal{E}, \tag{M2}$$

where $\mathcal{S} \in \mathbb{R}^{N \times P_{L+1} \times P_{L+2} \times \cdots \times P_{L+M}}$ is a sparse tensor representing the outliers and $\mathcal{E}$ is the dense tensor of errors. Tensor $\mathcal{S}$ allows for separating corruptions that do not follow the i.i.d distribution of overall noise and therefore eliminates the influence of these corruptions on parameter estimation. Model (M2) is similar to the TOT model in (2) considered in Lock (2018) and Liu *et al.*, (2020). The difference is that, we introduce the outliers $\mathcal{S}$ to the model, which makes it more robust and allows for the detection of outliers. Similar to TOT, due to the large number of parameters to be estimated in tensor $\mathcal{B}$, the above model results in overfitting without imposing a constraint on $\mathcal{B}$. To avoid such overfitting, we impose nuclear norm regularization on the tensor of parameters, which is presented later on. In addition, including $\mathcal{S}$ significantly increases model flexibility (and therefore causes overfitting) if no assumption is made on $\mathcal{S}$ and $\mathcal{E}$. We assume $\mathcal{S}$ and $\mathcal{E}$ are sparse and dense tensors, respectively.

In addition, (M2) has a form that is close to the tensor recovery problem (Goldfarb and Qin, 2014; Lu *et al.*, 2016; Xue *et al.*, 2017; Li *et al.*, 2019) that are tensor extensions of robust principle component analysis that recovers a corrupted low-rank matrix (Zhou *et al.*, 2010; Candès *et al.*, 2011; Wong and Lee, 2017). The tensor recovery problems consider a model defined as,

$$\mathcal{Y} = \mathcal{L} + \mathcal{S}, \tag{M3}$$

where $\mathcal{L}$ and $\mathcal{S}$ are low-rank and sparse components of $\mathcal{Y}$ respectively. Nevertheless, our proposed approach has several main differences with these tensor recovery problems: First, these problems are unsupervised and aim to recover a corrupted low-rank tensor (i.e., $\mathcal{L}$ is a recovered version of $\mathcal{Y}$). In contrast, our approach is a supervised approach that creates a predictive model based on potentially corrupted data for estimating an output tensor given a new input tensor $\mathcal{X}$. Second, unlike the recovery methods that assume $\mathcal{Y}$ is a corrupted low-rank tensor, our proposed approach has no assumption on low-rankness of $\mathcal{Y}$ or $\mathcal{X}$ and only assumes that the tensor of model parameters is low-rank. Finally, the tensor recovery methods do not consider a noisy setting where $\mathcal{Y}$ is noisy data. Our framework considers more realistic and general case that the output tensors are noisy.

The goal of the proposed approach is to find an estimator $\hat{\mathcal{B}}$ of $\mathcal{B}$ and an estimator $\hat{\mathcal{S}}$ of $\mathcal{S}$ in a way that the error is minimized, while the tensor of parameters remains low-rank and the tensor of outliers remains sparse. For this purpose, we solve the following optimization problem:

$$\min_{\mathcal{B},\mathcal{S}} \frac{\mu_1}{2} ||\mathcal{Y} - \langle \mathcal{X}, \mathcal{B} \rangle_L - \mathcal{S}||_F^2 + \mu_2 ||\mathcal{S}||_1 + \sum_{i=1}^{L+M} ||\mathcal{B}^{(i)}||_*, \tag{P1}$$

where the first term minimizes the prediction error, the second term regularizes the sparsity of $\mathcal{S}$, and the last term ensures that the tensor of parameters is low-rank. The objective function (P1) is equivalent to the following constrained problem:

$$\begin{aligned} &\min_{\mathcal{B},\mathcal{S}}\ \mu_2 ||\mathcal{S}||_1 + \sum_{i=1}^{L+M} ||\mathcal{B}^{(i)}||_*, \\ &s.t. ||\mathcal{Y} - \langle \mathcal{X}, \mathcal{B} \rangle_L - \mathcal{S}||_F^2 < \delta^2, \end{aligned} \tag{3}$$

where δ^2 is proportional to the variance of elementwise noise ($\mathcal{E}$) in the model and is related to μ_1 in (P1). This constraint allows for inexact reconstruction of noisy and corrupted tensor $\mathcal{Y}$ into dense and sparse tensors $\langle \mathcal{X}, \mathcal{B} \rangle_L$ and $\mathcal{S}$. For example, if $\delta \to \infty$ then $\mathcal{S}$ and $\mathcal{B}$ tend to tensors of zeros, which results in a model with no prediction power. In contrast, if $\delta \to 0$, then the problem requires exact estimation of $\mathcal{Y}$, which either results in a non-sparse estimation of $\mathcal{S}$ or an estimation of $\mathcal{B}$ that overfits to the training data depending on the choice of μ_2.

To emphasize on the importance of the two penalty terms in (P1), we note that, unlike TOT, model M2 does not have a unique solution in the following sense: There exists $(\mathcal{B}_1, \mathcal{S}_1) \neq (\mathcal{B}_2, \mathcal{S}_2)$ such that $\langle \mathcal{X}, \mathcal{B}_1 \rangle + \mathcal{S}_1 = \langle \mathcal{X}, \mathcal{B}_2 \rangle + \mathcal{S}_2$. To see this, assume $\mathcal{B}_1$ and $\mathcal{B}_2$ are different in exactly one element say $(p_1, p_2, ..., p_{L+m})$. Then, $\langle \mathcal{X}, \mathcal{B}_1 \rangle$ differs $\langle \mathcal{X}, \mathcal{B}_2 \rangle$ only at that element. However, this difference can be adjusted by choosing right values of $\mathcal{S}_1(p_1, p_2, ..., p_{L+m})$ and $\mathcal{S}_2(p_1, p_2, ..., p_{L+m})$. The sparsity and low-rankness penalties in (P1) alleviates this issue by choosing the most parsimonious model (characterized by $\mu_2 ||\mathcal{S}||_1 + \sum_{i=1}^{L+M} ||\mathcal{B}^{(i)}||_*$) that produce accurate predictions (characterized by $||\mathcal{Y} - \langle \mathcal{X}, \mathcal{B} \rangle_L - \mathcal{S}||_F^2$) of $\mathcal{Y}$.

In order to solve (P1), we propose an ADMM approach due to its capability in solving objective functions that are decomposable into differentiable and non-differentiable terms. Although, ADMM provides a framework to approach this problem, it does not directly solve the problem. Instead, it translates the problem into other optimization problems that are solved either numerically or analytically. Particularly, we combine the ADMM with the BCD algorithm to estimate the factor matrices and derive closed–form solutions for estimating $\mathcal{S}$. In order to let (P1) conform with ADMM, we first consider CP decomposition of $\mathcal{B}$ that approximates the original tensor as a sum of low-rank components:

$$\mathcal{B} = \sum_{\gamma=1}^{R} u_1^{\gamma} \circ u_2^{\gamma} \circ \cdots \circ u_{L+M}^{\gamma} = \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}, \quad (4)$$

where $u_i^{\gamma} \in \mathbb{R}^{P_i}$ $(i = 1, 2, ..., L+M)$ is a column vector and is the γ^{th} column of $\mathbf{U}_i \in \mathbb{R}^{P_i \times R}$. Second, we introduce auxiliary variables $\mathbf{J}_i$ such that $\mathbf{U}_i = \mathbf{J}_i$ for $i = 1, 2, ..., L+M$. The introduction of these auxiliary variables is to make the objective function separable. The resulting optimization is, therefore, as follows:

$$\begin{aligned} &\min \ \frac{\mu_1}{2}||\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \mathbf{U}_{L+M} \rangle_L - \mathcal{S}||_F^2 + \mu_2||\mathcal{S}||_1 + \sum_{i=1}^{L+M} ||\mathbf{J}_i||_* \\ &s.t. \quad \mathbf{U}_i = \mathbf{J}_i \quad \text{for } i = 1, 2, ..., L+M \end{aligned} \quad \text{(P2)}$$

where μ_1 and μ_2 are positive constants. Finally, the *contract tensor product* can be rewritten into a matrix product (Lock, 2018) with respect to U_i. In other words, $vec(\langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \mathbf{U}_{L+M} \rangle_L) = \mathbf{C}_i vec(\mathbf{U}_i)$, where $\mathbf{C}_i$ is defined as

$$\begin{aligned} &\mathbf{C}_i := [\mathbf{C}_i^1 | \mathbf{C}_i^2 | \cdots | \mathbf{C}_i^R] \quad \text{and} \\ &\mathbf{C}_i^{\gamma} = \langle \mathcal{X}, u_1^{\gamma} \circ u_2^{\gamma} \circ \cdots \circ u_{i-1}^{\gamma} \circ u_{i+1}^{\gamma} \circ u_{L+M}^{\gamma} \rangle_{L-1}^{(i)} \quad \text{for } i = 1, 2, ..., L+M, \end{aligned}$$

and $vec()$ is an operator that vectorizes a tensor. As a result, the corresponding augmented Lagrangian form of (P2) is written as,

$$\begin{aligned} \mathcal{L}(\mathbf{U}_i, \mathbf{J}_i, \mathbf{Z}_i, \mathcal{S}) := &\sum_{i=1}^{L+M} \frac{\mu_1}{2} ||vec(\mathcal{Y}) - \mathbf{C}_i vec(\mathbf{U}_i) - vec(\mathcal{S})||_2^2 \\ &+ \mu_2||\mathcal{S}||_1 + \sum_{i=1}^{L+M} ||\mathbf{J}_i||_* + \frac{\mu_3}{2}||\mathbf{U}_i - \mathbf{J}_i||_F^2 \\ &+ \langle \mathbf{U}_i - \mathbf{J}_i, \mathbf{Z}_i \rangle, \end{aligned} \quad (5)$$

where the last three terms are due to the equality constraints in (P2), $\mathbf{Z}_i$ is the corresponding dual variable, and μ_3 is a positive constant. Now, we are ready to employ ADMM to derive the updating equations that solves (5).

To do so, let us first derive the updating rule for each of the variables in (5) as shown below:

$\mathbf{J}_i$ is updated as follows:

$$\begin{aligned} \mathbf{J}_i &= \arg\min_{\mathbf{J}_i} \ \frac{1}{\mu_3}||\mathbf{J}_i||_* + \frac{1}{2}\left\| \mathbf{J}_i - \left(\mathbf{U}_i + \frac{1}{\mu_3}\mathbf{Z}_i \right) \right\|_F^2 \\ &= \Phi_{\mu_3}\left(\mathbf{U}_i + \frac{1}{\mu_3}\mathbf{Z}_i \right) \end{aligned} \quad (6)$$

where the operator $\Phi_\tau(X)$ is the singular value shrinkage operator followed the definition in Cai *et al.* (2010); $\mathbf{U}_i$ is updated as follows:

$$\begin{aligned} \mathbf{U}_i = \arg\min_{\mathbf{U}_i} &\frac{\mu_1}{2}||vec(\mathcal{Y}) - \mathbf{C}_i vec(\mathbf{U}_i) - vec(\mathcal{S})||_2^2 + \frac{\mu_3}{2}||\mathbf{U}_i \\ &- \mathbf{J}_i||_F^2 + \langle \mathbf{Z}_i, \mathbf{U}_i - \mathbf{J}_i \rangle \end{aligned} \quad (7)$$

which can be solved by a first-order method since the problem is convex; $\mathcal{S}$ is updated as follows:

$$\begin{aligned} \mathcal{S} = \arg\min_{\mathcal{S}} &\frac{1}{2}||\mathcal{S} - (\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M} \rangle_L)||_F^2 \\ &+ \frac{\mu_2}{\mu_1}||\mathcal{S}||_1 \\ = &\text{prox}_{\frac{\mu_2}{\mu_1}}(\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M} \rangle_L) \end{aligned} \quad (8)$$

where $\text{prox}_\tau(\cdot)$ is a proximity operator with l_1-norm regularization defined as follow:

$$\text{prox}_\tau(t) = \text{sign}(t)\max\{|t| - \tau, 0\}. \quad (9)$$

Algorithm 1 summarizes the pseudocode of the proposed ADMM algorithm. Note that the stopping criteria of Algorithm 1 are determined by primal and dual residuals r^k and s^k, respectively, at iteration k. The former is defined as $r^k = \sqrt{\sum_{i=1}^{L+M} ||\mathbf{U}_i^k - \mathbf{J}_i^k||_F^2}$ and the latter is defined as $s^k = \sqrt{\sum_{i=1}^{L+M} ||\mathbf{J}_i^k - \mathbf{J}_i^{k-1}||_F^2}$. We stop the algorithm when both r^k and $s^k < \epsilon$. Also, the algorithm's input $\hat{R}$ is defined as the number of columns of $\mathbf{U}_i$ for $i = 1, 2, ..., L+M$. For example, if $\hat{R} = 3$, then, $\mathbf{U}_i \in \mathbb{R}^{P_i \times 3}$ for $i = 1, 2, ..., L+M$.

Algorithm 1: ADMM Solver for RTOT

Input: μ_1, μ_2, μ_3, $\hat{R}$, $\epsilon = 10^{-6}$
initialize: $\mathbf{U}_i = \mathbf{J}_i = \text{rand}(P_i, \hat{R})$, $\mathbf{Z}_i = 0$ for $i = 1, 2, ..., L+M$, $\mathcal{S} = 0$;
while $r^k > \epsilon$ *or* $s^k > \epsilon$ **do**
  for $i = 1, 2, ..., L+M$ **do**
    Update $\mathbf{J}_i$ and $\mathbf{U}_i$ via (6) and (15);
  end
  Update $\mathcal{S}$ via (8);
  $\mathbf{Z}_i = \mathbf{Z}_i + \mu_3(\mathbf{U}_i - \mathbf{J}_i)$
end
Output: $\mathbf{U}_i$ for $i = 1, 2, ..., L+M$ and $\mathcal{S}$

4.1. Selection of tuning parameters

The proposed method requires input parameters μ_1, μ_2, μ_3 and $\hat{R}$. Due to the proximity operator, we can either fix μ_1

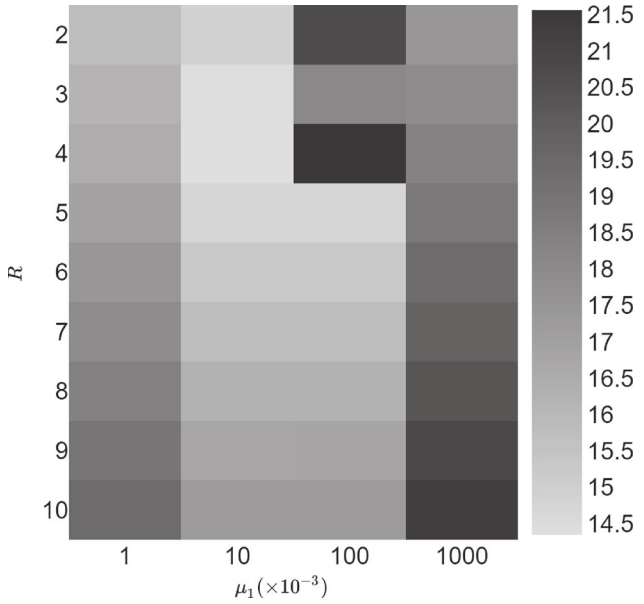

Figure 2. An example of heat map of AIC with respect to the hyper-parameters of RTOT.

or μ_2 and perturb the others. μ_3 is the learning rate and has a default value 1×10^{-10} and will dynamically change as the algorithm proceeds. To find the best combination of them, we conduct a grid search and select the one that has the lowest modified Akaike Information Criterion (AIC) defined as follows (Cavanaugh and Neath, 2019; Roy and Michailidis, 2022):

$$\text{AIC} = \log\ ||\mathcal{Y} - \langle \mathcal{X}, \hat{\mathcal{B}} \rangle_L - \hat{\mathcal{S}}||_F^2 + c_1 \log\ ||\hat{\mathcal{S}}||_0 + c_2 \cdot \hat{R},$$

where c_1 and c_2 are positive constant ($c_1 = 2$ and $c_2 = 0.5$ in the implementation), and $||\hat{\mathcal{S}}||_0$ denotes the number of non-zero elements within the tensor $\hat{\mathcal{S}}$. The modified AIC considers both underfitting (the first term) and overfitting (the second term) simultaneously. The number of the fitting parameter in typical AIC is replaced with $\hat{R}$ due to its proportionality to the complexity of the model. For all studies in the next section, we perform the grid search over $\hat{R} \in \{2, 3, 4, ..., 10\}$ and $\mu_1 \in \{0.005, 0.007, 0.009, ..., 0.015\}$ (this range is chosen empirically), and we let $\mu_2 = 0.0015$. Figure 2 shows an example of AIC for different values of hyper-parameters on a synthetic data set. In this example, $R = 4$ and $\mu_1 = 0.009$ minimize the AIC criterion and are selected for final model construction. Note that, larger ranks (e.g., $R = 6, ..., 10$) cause overfitting (model with excessive flexibility) for most values of μ_1 and generate larger values of AIC. Similarly, when $R = 2$ the model underfits data (the error term is large) which produces larger value of AIC. Furthermore, when μ_1 is large the model highly penalizes the fitting error and produce dense $\mathcal{S}$. On the other hand, when μ_1 is small the model underfits to data and produces highly sparse $\mathcal{S}$.

5. Performance evaluation using simulation

In this section, we synthesize multiple sets of data in different scenarios to evaluate the performance of the proposed method designated as RTOT in comparison with three benchmarks. The first benchmark is the TOT regression proposed by Lock (2018) designated as TOT and the second one is a combination of TOT and *robust principal component analysis* designated as RPCA where we first apply the principal component analysis to $\mathcal{Y}$ to isolate the outlier $\hat{\mathcal{S}}$ and acquire its low-rank representation $\hat{\mathcal{Y}}$. Then, we apply TOT to find $\hat{\mathcal{B}}$ such that $\hat{\mathcal{Y}} = \langle \mathcal{X}, \hat{\mathcal{B}} \rangle_L$. Note the main difference between RTOT and RPCA is that the former seeks $\hat{\mathcal{S}}$ and $\hat{\mathcal{B}}$ simultaneously whereas the latter has to perform these two procedures sequentially. The third benchmark is a convolutional neural network designated as CNN which has two convolution layers of size $4 \times 4 \times 64$ and $4 \times 4 \times 32$ and two deconvolution layers of size $4 \times 4 \times 32$ and $4 \times 4 \times 64$ aside from the input and output layers that have the same size of $\mathcal{X}$ and $\mathcal{Y}$ respectively. To measure the performance of these approaches, we consider the *Relative Prediction Error* (RPE) defined as (Lock, 2018):

$$\text{RPE} := \frac{||\mathcal{Y} - \langle \mathcal{X}, \hat{\mathcal{B}} \rangle_L||_F}{||\mathcal{Y}||_F}. \tag{10}$$

All the methods in this work are implemented in Python 3.7 with the implementation of TOT referring to the R package provided by Lock (2018) and all experiments are conducted under a machine equipped with Intel(R)Core(TM) i9-9880H CPU and 32GB RAM.

5.1. Data simulation and test environment setting

To test the performance of the proposed method and the benchmarks, we implement a fully crossed factorial simulation with the following conditions: $R \in \bar{R} := \{5, 7, 9\}$ and density of the outlier $D \in \bar{D} := \{0, 0.03, 0.05, 0.1, 0.15\}$.

For each of the 15 scenarios ($|\bar{R}| \times |\bar{D}|$), we generate two data sets (designated by data sets 1 and 2), with 500 and 200 samples, respectively. In the first data set, the output tensors $\mathcal{Y}$ are generated as follows:

$$\mathcal{Y} = \langle \mathcal{X}, \mathcal{B} \rangle_L + \mathcal{E}, \tag{11}$$

where $\mathcal{B} = \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}$ and $\mathbf{U}_i \in \mathbb{R}^{Pi \times R}$ for $i = 1, 2, ..., L + M$ whose elements simulated from a standard normal distribution, i.e., $N(0, 1)$. The input tensors $\mathcal{X}$ are generated as follows: First, we generate a basis matrix defined as $\mathbf{U}_i^x \in \mathbb{R}^{P_i \times 5}$ for $i = 1, 2, 3, ..., L$ whose elements are drawn independently from $N(0, 1)$. Next, we uniformly generate random weights $w_i \sim U(0, 1)$ for $i = 1, 2, 3, ..., L$ and set $\mathcal{X} = w_1 \cdot \mathbf{U}_1^x \circ w_2 \cdot \mathbf{U}_2^x \circ \cdots \circ w_L \cdot \mathbf{U}_L^x$. Finally, we generate $\mathcal{Y}$ as (11) with the elements of $\mathcal{E}$ simulated from $N(0, 1)$.

In the second data set, $\mathcal{X}$ and $\mathcal{Y}$ has the relation defined as:

$$\mathcal{Y} = \langle \mathcal{X}, \mathcal{B}_1 \rangle_L + \langle \mathcal{X}, \mathcal{B}_2 \rangle_L + \mathcal{E}, \tag{12}$$

where we generate $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{X}$ in the same way as in the first data set except that $\mathcal{B}_2$ is highly sparse whose elements are all zero except the values of the first 50 elements of $vec(\mathcal{B}_2)$ that are generated from $N(0, 1)$. Also, $\mathcal{X}$ is generated using Fourier basis functions defined as

Table 2. Comparison between the proposed method (RTOT) and the benchmarks in terms of RPE (standard deviation) in data set 1 settings.

R	*Method*	*D* = 0	*D* = 0.03	*D* = 0.05	*D* = 0.1	*D* = 0.15
5	RPCA	0.0881(0.0056)	0.0892(0.0047)	0.0880(0.0051)	0.0867(0.0055)	0.0880(0.0058)
5	RTOT	**0.0871(0.0056)**	**0.0884(0.0047)**	**0.0873(0.0052)**	**0.0859(0.0052)**	**0.0875(0.0065)**
5	TOT	**0.0871(0.0056)**	0.1047(0.0077)	0.1188(0.0144)	0.1382(0.0119)	0.1708(0.0232)
5	CNN	0.0989(0.0349)	0.1250(0.0105)	0.1440(0.0122)	0.1797(0.0148)	0.2234(0.0239)
7	RPCA	0.0902(0.0058)	0.0910(0.0064)	0.0898(0.0067)	0.0905(0.004)	0.0888(0.0058)
7	RTOT	**0.0892(0.006)**	**0.0900(0.0066)**	**0.0887(0.007)**	**0.0896(0.004)**	**0.0884(0.0054)**
7	TOT	0.0893(0.006)	0.1043(0.0078)	0.1279(0.0348)	0.1572(0.0187)	0.1720(0.0213)
7	CNN	0.0924(0.0061)	0.1237(0.0143)	0.1489(0.0239)	0.1917(0.0165)	0.2190(0.0167)
9	RPCA	0.088(0.0067)	0.0902(0.0074)	0.0895(0.0063)	0.0894(0.0071)	0.0895(0.0067)
9	RTOT	**0.0867(0.0067)**	**0.0892(0.0075)**	**0.0886(0.0066)**	**0.0888(0.0073)**	**0.0889(0.0066)**
9	TOT	**0.0868(0.0067)**	0.104(0.0095)	0.1245(0.0125)	0.1522(0.0234)	0.1767(0.0241)
9	CNN	0.0899(0.0067)	0.1344(0.0576)	0.1474(0.0287)	0.1846(0.0167)	0.2253(0.0221)

Table 3. Comparison between the proposed method (RTOT) and the benchmarks in terms of RPE (standard deviation) in data set 2 settings.

R	*Method*	*D* = 0	*D* = 0.03	*D* = 0.05	*D* = 0.1	*D* = 0.15
5	RPCA	0.1184(0.0416)	0.1255(0.0376)	0.1121(0.0282)	0.1116(0.0492)	0.1150(0.0345)
	RTOT	**0.07200(0.0058)**	**0.0734(0.0082)**	**0.0734(0.007)**	**0.0732(0.0068)**	**0.0778(0.0053)**
	TOT	0.0760(0.0087)	0.1200(0.0305)	0.1655(0.043)	0.2108(0.0518)	0.2953(0.0693)
	CNN	0.8496(0.0507)	0.8402(0.0564)	0.882(0.0482)	0.9129(0.0705)	0.9493(0.0364)
7	RPCA	0.1040(0.0352)	0.1192(0.0301)	0.1151(0.0294)	0.1200(0.0421)	0.1063(0.0399)
	RTOT	**0.0691(0.0059)**	**0.075(0.0068)**	**0.0718(0.0048)**	**0.0737(0.0042)**	**0.0792(0.0061)**
	TOT	**0.0691(0.0059)**	0.1133(0.0264)	0.1644(0.0466)	0.2071(0.0517)	0.2732(0.0627)
	CNN	0.8504(0.0355)	0.8441(0.0402)	0.8665(0.047)	0.9053(0.0462)	0.9603(0.0481)
9	RPCA	0.1051(0.0277)	0.1090(0.0352)	0.1243(0.0391)	0.1169(0.0404)	0.1148(0.0436)
	RTOT	**0.0700(0.0066)**	**0.0714(0.0063)**	**0.0742(0.0059)**	**0.0754(0.006)**	**0.0787(0.0069)**
	TOT	0.0706(0.0065)	0.0886(0.0158)	0.1400(0.0366)	0.2209(0.0479)	0.2726(0.0742)
	CNN	0.8359(0.0504)	0.8631(0.0547)	0.8747(0.037)	0.8926(0.0444)	0.9444(0.0549)

$$u_j^r = \begin{cases} \left[\cos\left(c\pi r x_1\right), \ldots, \cos\left(c\pi r x_{P_j}\right)\right]^\top, & \text{if } r \text{ is odd} \\ \left[\sin\left(c\pi r x_1\right), \ldots, \sin\left(c\pi r x_{P_j}\right)\right]^\top, & \text{if } r \text{ is even} \end{cases} \tag{13}$$

where $c > 0$, $r = 1, 2, \ldots, 5$ and $x_j = \frac{j}{P_j}$ for $j = 1, 2, 3, \ldots, L$. The dimension of $\mathcal{B}$ in the first data set is $(15, 20, 5, 10)$ whereas that in the second data set is $(35, 29, 7, 29)$ for both $\mathcal{B}_1$ and $\mathcal{B}_2$.

After generating all the samples in both data sets 1 and 2, we randomly split the data into training (400 samples and 120 samples for data sets 1 and 2, respectively) and testing (100 samples and 80 samples for data sets 1 and 2, respectively) sets. We, then, add outliers to the training sets by randomly choosing D percent of the samples and picking a starting index i_0 from $vec(\mathcal{Y})$. Starting from i_0 till $i_0 + l$, we reset each element to a value that follows $U(.8, 2)$ (the range of the distribution is chosen empirically so that it is significant enough for the corresponding element to be considered as the outlier). After that, we reshape $vec(\mathcal{Y})$ back to its original shape. Throughout the simulation, we let $L = 2$, $M = 2$, $l = 5$ for data set 1 and $l \in \{10, 11, 12, \ldots, 30\}$ for data set 2 (we randomly pick one element from the range). We replicate the simulations 20 times for each of the 15 scenarios to acquire the mean and standard deviation (std) of RPE.

5.2. Simulation results

In this section, we report the simulation results obtained over the 15 scenarios described in the previous section. As is depicted in Table 2 and Table 3, the proposed method outperforms benchmarks in almost all scenarios. As is reported, the averaged RPE of TOT and CNN increases significantly as D increases. On the other hand, RPCA and RTOT maintain their prediction performance close to scenarios where data contains no outliers. This result is expected since TOT and CNN are not designed for the data containing outliers whereas the other two have their mechanisms to isolate outliers from the training data, and hence, are more accurate. In addition, the proposed method has comparable results to TOT even when data contains no outliers.

Table 2 reports the performance (in terms of RPE and its standard deviation) of all methods when applied to data set 1. When $D > 0$ the proposed method outperforms all other benchmarks. For example, when $R = 5$ and $D = 0.0.03$, RTOT achieves $RPE = 0.0884$ compared with 0.0892, 0.1047, and 0.1250 obtained by RPCA, TOT, and CNN, respectively. Table 3 reports the performance of all methods when applied to dataset 2. As is reported, RTOT has the best performance among all other approaches when $D > 0$. For example, when $R = 7$ and $D = 0.1$, RTOT achieves $RPE = 0.0737$ compared with 0.1200, 0.2071, and 0.9053 obtained by RPCA, TOT, and CNN, respectively. Please note that data set 2, in which we amplify part of $\mathcal{Y}$ by adding an extra term $\langle \mathcal{X}, \mathcal{B}_2 \rangle_L$, appeared to be a more challenging case for RPCA (compared with data set 1). More specifically, the performance of RPCA is always significantly inferior to RTOT (the average RPE in data set 2 of RPCA is higher than that of RTOT in all scenarios), which reveals the potential issue of using the RPCA. As mentioned earlier, RPCA performs the outlier elimination and prediction in a sequential manner, which may lose some of the information from the input during the elimination process, and hence, lowering the overall prediction performance. Furthermore,

Table 4. Performance evaluation of outlier detection of RTOT for both scenarios (data sets 1 and 2) in terms of ACC, FNR, and FPR.. The true positives are outlier elements within $\mathcal{S}$.

Data Set 1					*Data Set 2*				
D	*R*	*ACC*	*FNR*	*FPR*	*D*	*R*	*ACC*	*FNR*	*FPR*
0.03	5	0.9998	0.0000	0.0002	0.03	5	0.9998	0.0642	0.0000
0.03	7	0.9999	0.0000	0.0001	0.03	7	0.9999	0.0279	0.0000
0.03	9	0.9998	0.0000	0.0002	0.03	9	0.9999	0.049	0.0000
0.05	5	0.9997	0.0000	0.0003	0.05	5	0.9998	0.0342	0.0000
0.05	7	0.9997	0.0000	0.0003	0.05	7	0.9999	0.019	0.0000
0.05	9	0.9997	0.0000	0.0003	0.05	9	0.9999	0.0211	0.0000
0.1	5	0.9993	0.0000	0.0007	0.1	5	0.9996	0.0374	0.0000
0.1	7	0.9994	0.0000	0.0006	0.1	7	0.9998	0.0235	0.0000
0.1	9	0.9993	0.0000	0.0007	0.1	9	0.9994	0.056	0.0000
0.15	5	0.999	0.0000	0.001	0.15	5	0.9996	0.028	0.0000
0.15	7	0.9991	0.0000	0.0009	0.15	7	0.9995	0.0373	0.0000
0.15	9	0.9991	0.0000	0.0009	0.15	9	0.9994	0.0471	0.0000

Table 5. Averaged computational time (in seconds) required for one replication with different choices of *R*. RTOT-D refers to RTOT with dynamic learning rate and RTOT-S refers to RTOT when learning rate is set to a fixed value (static).

Method	$R = 7$	$R = 10$	$R = 20$	$R = 30$
RPCA	1.1560	1.5901	3.7869	6.0684
RTOT-D	0.4793	0.6488	2.2690	3.8052
RTOT-S	1.5891	2.8569	7.1529	16.3950
TOT	0.6400	1.1102	3.2528	5.5942
CNN	0.1702	0.1562	0.1606	0.1566

we evaluate the performance of the proposed method in terms of localizing the outliers during the training of the model. More specifically, we compare the estimated sparse tensor $\hat{\mathcal{S}}$ to the original tensor $\mathcal{S}$ in both simulation scenarios. Table 4 reports the ACCuracy (ACC), False Negative Rate (FNR), and False Positive Rate (FPR) of outlier detection. As is reported, under all simulated settings, the proposed method can isolate outliers with high accuracy ($ACC > 0.99$).

Table 5 reports the computational time of the proposed method and benchmarks. In this table, we evaluate the computational time of the RTOT when the value of learning rate μ_3 is set to a fixed value (ROTO-S) and when it is changed dynamically (RTOT-D) as discussed in Section 4.1. As it is reported in the table, the RTOT with dynamic learning rate (RTOT-D) outperforms TOT and RPCA with the latter being the worse one as it requires performing RPCA before applying TOT to estimate the model parameters. The RTOT with fixed learning rate, however, is slower than other methods as it requires many iterations to enforce the constraints in (P2). Please note that the TOT algorithm uses a BCD algorithm to solve its problem and does not have a learning rate. Note that CNN takes full advantage of GPU, and hence, it achieves the lowest computational time. The required computational time of RTOT can be further reduced via parallelization, which is a subject of future work.

6. Performance evaluation using case study

We evaluate the performance of the proposed robust approach using two real data sets. The first data set is obtained from vehicle engine sensors and the second data is related to EEG signals and functional Magnetic Resonance Imaging (fMRI) scans obtained from the OpenNeuro website (https://openneuro.org/datasets/ds000116/versions/00003).

6.1. Case study I: Vehicle engine lambda sensor prediction

The NOx Storage Catalyst (NSC) is a catalyst system designed to treat the exhaust gas produced by vehicles. The NSC process consists of two alternating phases: (i) absorption: where NOx molecules are trapped/absorbed by zeolites–coated converter support; (ii) regeneration: where the stored NOx is reduced by a catalyst when the absorber is saturated. It is well-known that, during the regeneration phase, optimal combustion is required to ensure an ideal conversion rate of the catalytic converter. The NSC only operates efficiently at stoichiometric conditions, which requires the combustion process in a rich air-to-fuel status. The relative air/fuel ratio normalized by stoichiometry (designated by the Greek letter λ), which is measured runtime by a sensor upstream of the NSC, is used as an indicator to show if the regeneration phase is running correctly. A good sign of λ should maintain a value within a set-point interval of 0.92–0.95. When the λ signal falls below a threshold (0.8–0.9) which is called λ-undershoot, it would worsen the performance of NSC. Thus, developing a model that could estimate the λ signal based on other operation signals collected by onboard sensors (such as inner torque, rotational speed, and quantity of fuel injected) could further improve the engine operation condition, as well as the calibration of the engine control unit.

In this case study, we apply our proposed method and three other benchmarks to estimate the λ-upstream curve given a set of five input curves. We evaluate the performance of the methods by computing the RPE over several replications. The data set has 285 samples with each containing five input signals and one lambda signal corresponding to the output. All the data points are recorded within a 2-second interval with 203 measurements in total. To comply with our model, we reshape the output into $285 \times 29 \times 7$ (285×203 without reshaping) and the input into $285 \times 29 \times 35$, and then randomly partition the data into training (200 samples) and testing (85 samples) sets. The number of replications of the training/testing process is set to 20 and at each iteration, we acquire RPE from all four methods. Figure 3 shows

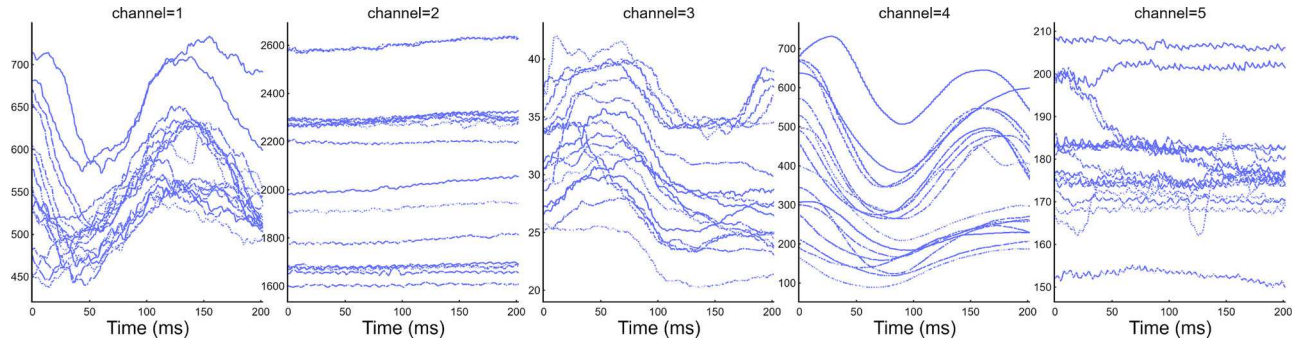

Figure 3. Example of the input signals.

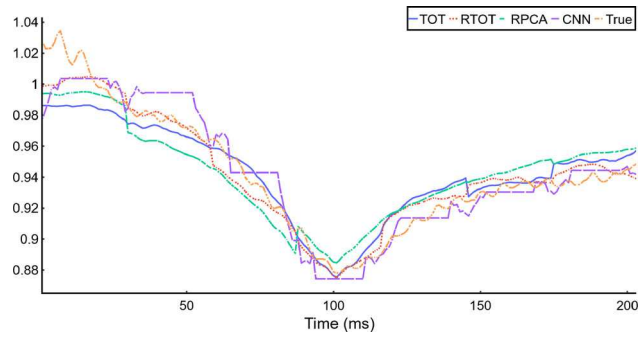

Figure 4. Example of a λ curve prediction by the proposed method and benchmarks.

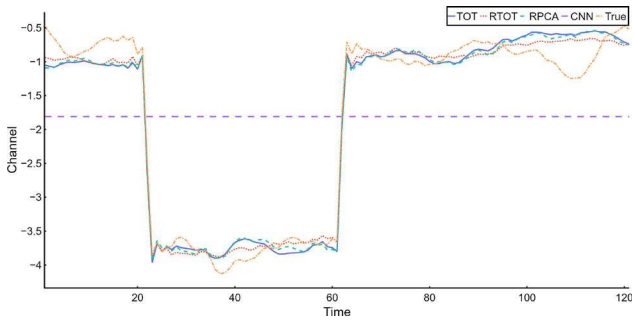

Figure 5. Example of EEG prediction by the proposed method and benchmarks.

Table 6. RPE of the predicted original lambda curve by proposed and benchmark methods.

RPCA	*RTOT*	*TOT*	*CNN*
0.0517(0.0012)	**0.0514(0.0009)**	0.0527(0.0012)	0.0599(0.0051)

Table 7. RPE of the predicted EGG by proposed and benchmark methods.

RPCA	*RTOT*	*TOT*	*CNN*
0.3228(0.0046)	**0.3155(0.0063)**	0.3236(0.0052)	0.9999(0.0000)

examples of the five input signals and Figure 4 shows an example of a λ curve together with the predicted curves obtained by RTOT, RPCA, TOT, and CNN. As is illustrated, while all methods produce reasonable predictions, the curve predicted by RTOT shows fewer deviations from the true curve (for example, TOT, RPCA, and CNN have a relatively large bias of around 0.5 sec compared with RTOT). Table 6 reports the average of RPE along with standard deviation, which indicates the superior performance of RTOT. As is reported, the proposed method has the lowest RPE (0.0514) compared with the other approaches.

6.2. Case study II: EEG prediction from fMRI

The integration of EEG and fMRI, in which EEG has high temporal resolution and low spatial resolution, and fMRI has high spatial resolution and low temporal resolution has been used as a tool to study brain activity. The advantage of this simultaneous EEG-fMRI framework is that it guarantees the same signal source and also provides a way to analyze the connection between the variations of the brain signals and brain activities.

In this case study, we intend to predict the EEG, which is a $16 \times 3 \times 121$ tensor from the fMRI, which is a $16 \times 10 \times 8$ tensor. The first dimension of these tensors represents the number of subjects in the study. Note that the original shape of EEG is $16 \times 37 \times 121$; however, we pick the three most relevant channels among 37 channels (the second dimension) that are closest to the active regions of the brain during the experiment. We reshape the fMRI from 16×80 to $16 \times 10 \times 8$ to comply with our approach. To evaluate the performance of the proposed method (RTOT), we compare the prediction results with three benchmarks TOT, RPCA, and CNN and replicate the training/testing process 20 times to calculate the average RPE and its variance. The 14 training samples are randomly selected from the data with the remaining two serving as the test sample. As reported in Table 7, in which each cell reports the mean of the RPE and its variance (number in parenthesis), RTOT demonstrates the lowest RPE compared with the other benchmarks. Figure 5 depicts a prediction result of one of the EEG channels. In this figure most of the methods predict the curve well within the time interval ranging from 20 to 60 except CNN. The inferior performance of CNN is mainly attributed to the high dimensionality and small sample size of the data. Although all other methods show a comparable result in the other two intervals ($[0, 20]$ and $[60, 120]$), underestimation and overestimation can be observed in the predictions of RTOT, TOT and RPCA.

The estimated model parameters can serve as a new data that contains the interrelation between the EEG and fMRI data. One can use CP or Tucker decomposition on this tensor of parameters to extract features for the purpose of decision making such as classification.

7. Conclusion

This article proposes a robust TOT approach to model processes with high-dimensional inputs and contaminated outputs. The approach extends the TOT regression and borrows techniques from tensor recovery. To estimate the parameters, a least square loss function is applied, and to avoid overfitting, a nuclear norm regularization that penalizes the rank of the tensor is used. To solve the resulting minimization problem, we propose an ADMM algorithm that first transforms the problem into its corresponding Lagrangian form and then decomposes the entire problem into several sub-problems with closed-form solutions.

To evaluate the performance of the proposed method, we provide a simulation study that has 15 different scenarios and two case studies with the data coming from real-world applications. In all simulation and case studies, We compared the proposed approach to three benchmark methods, namely TOT, RPCA, which is a combination of RPCA and TOT, and CNN. As it is reported in the results, not only can RTOT maintain a decent accuracy when data has been corrupted by outliers but also demonstrate high efficiency during the training phase. In addition, we show a comparable result to TOT even when the data has no outliers. Future work may extend the proposed framework to situations where data contains missing values.

Funding

Dr. Reisi is partially supported by the National Science Foundation (NSF CMMI 2027024) and UF foundation. Dr. Hung Yi Lee and Dr. Hongcheng Liu are partially supported by the National Science Foundation (NSF CMMI 2016571)

Notes on contributors

Hung Yi Lee received his PhD in Industrial and Systems Engineering from the University of Florida in 2022 and his MS degree in Industrial Engineering from National Cheng Kung University, Taiwan, in 2013. Lee's research interest lies in operations research, stochastic optimization, and machine learning.

Mostafa Reisi Gahrooei is an assistant professor with the Department of Industrial and Systems Engineering, the University of Florida, Gainesville, FL, USA. His research focuses on modeling, monitoring, and control of complex systems with multimodal, functional, and high-dimensional data. Dr. Reisi Gahrooei is a member of the Institute for Operations Research and the Management Sciences (INFORMS) and the Institute of Industrial and Systems Engineers (IISE).

Hongcheng Liu is an assistant professor of industrial and systems engineering at the University of Florida. His research interests lie in algorithms, operations research, stochastic optimization, and high-dimensional machine and statistical learning. He is also interested in the applications in radiotherapy treatment planning, medical decision making, and transportation modeling.

Massimo Pacella is an associate professor in the Department of Engineering for Innovation at the University of Salento, Italy. He received an MSc in computer engineering from the University of Lecce, Italy, and a PhD in manufacturing and production systems from the Polytechnic of Milan, Italy. He was awarded a Fulbright Fellowship, and he was a research scholar at the Department of Industrial and Operations Engineering, University of Michigan, USA. His major research interests are in functional data processing and profile monitoring, including applied and methodological aspects of machine learning and statistical modeling, integrated with engineering principles. The primary applications of his research are in manufacturing and automotive. He is a member of the Italian Association for Manufacturing Technology (AITeM).

ORCID

Mostafa Reisi Gahrooei http://orcid.org/0000-0002-7633-9575

References

Cai, J.-F., Candès, E.J. and Shen, Z. (2010) A singular value thresholding algorithm for matrix completion. *SIAM Journal of Optimization*, **20**(4), 1956–1982.

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

Cavanaugh, J.E. and Neath, A.A. (2019) The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *Wiley Interdisciplinary Reviews: Computational Statistics*, **11**(3), e1460.

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

Fang, X., Paynabar, K. and Gebraeel, N. (2019) Image-based prognostics using penalized tensor regression. *Technometrics*, **61**(3), 369–384.

Gahrooei, M.R., Paynabar, K., Pacella, M. and Shi, J. (2019) Process modeling and prediction with large number of high-dimensional variables using functional regression. *IEEE Transactions on Automation Science and Engineering*, **17**(2), 684–696.

Gahrooei, M.R., Yan, H., Paynabar, K. and Shi, J. (2021) Multiple tensor-on-tensor regression: An approach for modeling processes with heterogeneous sources of data. *Technometrics*, **63**(2), 147–159.

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

Harshman, R.A. (1970) Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, **16**, 1–84.

Hu, Y. and Work, D.B. (2020) Robust tensor recovery with fiber outliers for traffic events. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **15**(1), 1–27.

Huang, H. and Ding, C. (2008) Robust tensor factorization using r1 norm. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Piscataway, NJ, pp. 1–8.

Hullait, H., Leslie, D.S., Pavlidis, N.G. and King, S. (2021) Robust function-on-function regression. *Technometrics*, **63**(3), 396–409.

Kanning, M., Kühling, I., Trautz, D. and Jarmer, T. (2018) High-resolution UAV-based hyperspectral imagery for lai and chlorophyll estimations from wheat for yield prediction. *Remote Sensing*, **10**(12), 2000.

Kaur, A. and Datta, A. (2019) Detecting and ranking outliers in high-dimensional data. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, **11**(1), 75–87.

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

Li, B., Xu, X., Zhang, L., Han, J., Bian, C., Li, G., Liu, J. and Jin, L. (2020) Above-ground biomass estimation and yield prediction in potato by using UAV-based RGB and hyperspectral imaging. *ISPRS Journal of Photogrammetry and Remote Sensing*, **162**, 161–172.

Li, P., Feng, J., Jin, X., Zhang, L., Xu, X. and Yan, S. (2019) Online robust low-rank tensor modeling for streaming data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, **30**(4), 1061–1075.

Liu, Y., Liu, J. and Zhu, C. (2020) Low-rank tensor train coefficient array estimation for tensor-on-tensor regression. *IEEE Transactions on Neural Networks and Learning Systems*, **31**(12), 5402–5411.

Lock, E.F. (2018) Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics*, **27**(3), 638–647.

Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z. and Yan, S. (2016) Tensor robust principal component analysis: Exact recovery of corrupted

low-rank tensors via convex optimization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Piscataway, NJ, pp. 5249–5257.

Naskovska, K., Korobkov, A.A., Haardt, M. and Haueisen, J. (2017) Analysis of the photic driving effect via joint EEG and MEG data processing based on the coupled CP decomposition, in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE Press, Piscataway, NJ, pp. 1285–1289.

Roy, S. and Michailidis, G. (2022) Regularized high dimension low tubal-rank tensor regression. *Electronic Journal of Statistics*, **16**(1), 2683–2723.

Shang, F., Liu, Y. and Cheng, J. (2014) Generalized higher-order tensor decomposition via parallel ADMM, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI-14*, Quebec City, QC, Canada, pp. 1279–1285.

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

Wahba, A., Wang, L.-C., Zhang, Z. and Sumikawa, N. (2019) Wafer pattern recognition using Tucker decomposition, in *2019 IEEE 37th VLSI Test Symposium (VTS)*, IEEE Press, Piscataway, NJ, pp. 1–6.

Wang, F., Gahrooei, M.R., Zhong, Z., Tang, T. and Shi, J. (2021) An augmented regression model for tensors with missing values. *IEEE Transactions on Automation Science and Engineering*, **19**(4), 2968–2984.

Wong, R.K. and Lee, T.C. (2017) Matrix completion with noisy entries and outliers. *The Journal of Machine Learning Research*, **18**(1), 5404–5428.

Xue, N., Papamakarios, G., Bahri, M., Panagakis, Y. and Zafeiriou, S. (2017) Robust low-rank tensor modelling using Tucker and CP decomposition, in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, Greece, pp. 1185–1189.

Yan, H., Paynabar, K. and Pacella, M. (2019) Structured point cloud data analysis via regularized tensor regression for process modeling and optimization. *Technometrics*, **61**(3), 385–395.

Zhao, Q., Caiafa, C.F., Mandic, D.P., Chao, Z.C., Nagasaka, Y., Fujii, N., Zhang, L. and Cichocki, A. (2012) Higher order partial least squares (HOPLS): A generalized multilinear regression method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(7), 1660–1673.

Zhao, Y., Yan, H., Holte, S.E., Kerani, R.P. and Mei, Y. (2019) Rapid detection of hot-spot by tensor decomposition with application to weekly gonorrhea data, in *International Workshop on Intelligent Statistical Quality Control*, Springer, Hong Kong, pp. 265–286.

Zhou, H. and Kan, C. (2021) Tensor-based ECG anomaly detection toward cardiac monitoring in the internet of health things. *Sensors*, **21**(12), 4173.

Zhou, H., Li, L. and Zhu, H. (2013) Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, **108**(502), 540–552.

Zhou, Z., Li, X., Wright, J., Candes, E. and Ma, Y. (2010) Stable principal component pursuit, in *2010 IEEE International Symposium on Information Theory*, IEEE Press, Piscataway, NJ, pp. 1518–1522.

A Appendix

A.1. Updater deduction

Based on (5), we totally have four variables, $\mathbf{U}_i$, $\mathbf{J}_i$, $\mathbf{Z}_i$ and $\mathcal{S}$ need to be updated, and below details how we update these variables.

- $\mathbf{J}_i = \arg\min_{\mathbf{J}_i} \frac{1}{\mu_3}\|\mathbf{J}_i\|_* + \frac{1}{2}\|\mathbf{J}_i - (\mathbf{U}_i + \frac{1}{\mu_3}\mathbf{Z}_i)\|_F^2$ in which, according to (Cai *et al.*, 2010), can be solved via the singular value shrinkage operator defined as

$$\Phi_\tau(\mathbf{X}) := \mathbf{U}\mathbf{D}_\tau\mathbf{V}^\top \tag{14}$$

where $\mathbf{D}_\tau = \mathrm{diag}\{(\sigma_1-\tau)_+, (\sigma_2-\tau)_+, (\sigma_n-\tau)_+, \ldots, (\sigma_r-\tau)_+\}$ given that $U \in \mathrm{R}^{m\times r}$ and $V^\top \in \mathrm{R}^{r\times n}$, and σ_i is ith the singular value of the matrix $\mathbf{X}$. Therefore, applying (14), we have that

$$\begin{aligned}\mathbf{J}_i &= \arg\min_{\mathbf{J}_i} \frac{1}{\mu_3}\|\mathbf{J}_i\|_* + \frac{1}{2}\left\|\mathbf{J}_i - \left(\mathbf{U}_i + \frac{1}{\mu_3}\mathbf{Z}_i\right)\right\|_F^2 \\ &= \Phi_{\mu_3}\left(\mathbf{U}_i + \frac{1}{\mu_3}\mathbf{Z}_i\right).\end{aligned}$$

- $\mathbf{U}_i = \arg\min_{\mathbf{U}_i} \frac{\mu_1}{2}\|vec(\mathcal{Y}) - \mathbf{C}_i vec(\mathbf{U}_i) - vec(\mathcal{S})\|_2^2 + \frac{\mu_3}{2}\|\mathbf{U}_i - \mathbf{J}_i\|_F^2 + \langle \mathbf{Z}_i, \mathbf{U}_i - \mathbf{J}_i\rangle$ which is a convex problem and can be solved using first-order method as follows:

$$\begin{aligned}&-\mu_1\mathbf{C}_i^\top(vec(\mathcal{Y}) - \mathbf{C}_i vec(\mathbf{U}_i) - vec(\mathcal{S})) + \mu_3(vec(\mathbf{U}_i) \\ &\quad - vec(\mathbf{J}_i)) + vec(\mathbf{Z}_i) = 0 \\ &\Rightarrow \mu_1\mathbf{C}_i^\top\mathbf{C}_i vec(\mathbf{U}_i) + \mu_3\mathbf{U}_i = \mu_1\mathbf{C}_i(vec(\mathcal{Y}) \\ &\quad - vec(\mathcal{S})) + \mu_3 \cdot vec(\mathbf{J}_i) - vec(\mathbf{Z}_i) \\ &\Rightarrow \mathbf{U}_i = (\mu_1\mathbf{C}_i^\top\mathbf{C}_i + \mu_3\mathbf{I})^{-}1(\mu_1\mathbf{C}_i(vec(\mathcal{Y}) \\ &\quad - vec(\mathcal{S})) + \mu_3 \cdot vec(\mathbf{J}_i) - vec(\mathbf{Z}_i)).\end{aligned} \tag{15}$$

- $\mathcal{S} = \arg\min_{\mathcal{S}} \frac{1}{2}\|\mathcal{S} - (\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}\rangle_L)\|_F^2 + \frac{\mu_2}{\mu_1}\|\mathcal{S}\|_1$ which can be transformed into a Lasso problem as follow:

$$\begin{aligned}&\min_{\mathcal{S}} \frac{1}{2}\|\mathcal{S} - (\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}\rangle_L)\|_F^2 + \frac{\mu_2}{\mu_1}\|\mathcal{S}\|_1 \\ &= \frac{1}{2}\|vec(\mathcal{S}) - (vec(\mathcal{Y}) - vec(\langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}\rangle_L))\|^2 \\ &\quad + \frac{\mu_2}{\mu_1}\|vec(\mathcal{S})\|_1\end{aligned}$$

and can be solved via the proximity operator $\mathrm{prox}_\tau(t) := \mathrm{sign}(t) \cdot \max(|t| - \tau, 0)$. Therefore, we have

$$\begin{aligned}\mathcal{S} &= \arg\min_{\mathcal{S}} \frac{1}{2}\|\mathcal{S} - (\mathcal{Y} - \langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}\rangle_L)\|_F^2 + \frac{\mu_2}{\mu_1}\|\mathcal{S}\|_1 \\ &= \mathrm{prox}_{\frac{\mu_2}{\mu_1}}\left(vec(\mathcal{Y}) - vec(\langle \mathcal{X}, \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{U}_{L+M}\rangle_L)\right)\end{aligned} \tag{16}$$

- $\mathbf{Z}_i = \mathbf{Z}_i + \mu_3(\mathbf{U}_i - \mathbf{J}_i)$ which is the cumulative sum of the dual infeasibility.