Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Partially Observed Dynamic Tensor Response Regression

Jie Zhou, Will Wei Sun, Jingfei Zhang & Lexin Li

To cite this article: Jie Zhou, Will Wei Sun, Jingfei Zhang & Lexin Li (2023) Partially Observed Dynamic Tensor Response Regression, Journal of the American Statistical Association, 118:541, 424-439, DOI: 10.1080/01621459.2021.1938082

To link to this article: https://doi.org/10.1080/01621459.2021.1938082

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 118, NO. 541, 424–439: Theory and Methods
https://doi.org/10.1080/01621459.2021.1938082

Partially Observed Dynamic Tensor Response Regression

Jie Zhou[a], Will Wei Sun [b], Jingfei Zhang[a], and Lexin Li[c]

[a]Department of Management Science, University of Miami Herbert Business School, Miami, FL; [b]Krannert School of Management, Purdue University, West Lafayette, IN; [c]Division of Biostatistics, University of California, Berkeley, Berkeley, CA

ABSTRACT

In modern data science, dynamic tensor data prevail in numerous applications. An important task is to characterize the relationship between dynamic tensor datasets and external covariates. However, the tensor data are often only partially observed, rendering many existing methods inapplicable. In this article, we develop a regression model with a partially observed dynamic tensor as the response and external covariates as the predictor. We introduce the low-rankness, sparsity, and fusion structures on the regression coefficient tensor, and consider a loss function projected over the observed entries. We develop an efficient nonconvex alternating updating algorithm, and derive the finite-sample error bound of the actual estimator from each step of our optimization algorithm. Unobserved entries in the tensor response have imposed serious challenges. As a result, our proposal differs considerably in terms of estimation algorithm, regularity conditions, as well as theoretical properties, compared to the existing tensor completion or tensor response regression solutions. We illustrate the efficacy of our proposed method using simulations and two real applications, including a neuroimaging dementia study and a digital advertising study.

ARTICLE HISTORY
Received March 2020
Accepted May 2021

KEYWORDS
Alzheimer's disease; Digital advertising; Neuroimaging analysis; Nonconvex optimization; Tensor completion; Tensor regression

1. Introduction

In modern data science, dynamic tensor data are becoming ubiquitous in a wide variety of scientific and business applications. The data take the form of a multidimensional array and one mode of the array is time. It is often of keen interest to characterize the relationship between such time-varying tensor datasets and external covariates. One example is a neuroimaging study of Alzheimer's disease (AD) (Thung et al. 2016). Anatomical magnetic resonance imaging (MRI) data are collected for 365 individuals with and without AD every six months over a two-year period. After preprocessing, each image is of dimension $32 \times 32 \times 32$, and stacking these MRI images over time formulates a fourth-way tensor for each subject. An important scientific question is to understand how a patient's structural brain atrophy is associated with clinical and demographic characteristics such as the patient's diagnosis status, age and sex. Another example is a digital advertising study (Bruce, Murthi and Rao 2017). The click-through rate (CTR) of 20 active users reacting to digital advertisements from 2 publishers are recorded for 80 advertisement campaigns on a daily basis over a four-week period. The data for each campaign are formed as a tensor by user by publisher by time. An important business question is to understand how features of an advertisement campaign affect its effectiveness measured by CTR on the target audience. Both questions can be formulated as a supervised tensor learning problem. However, a crucial but often overlooked issue is that the tensor data are often only partially observed in real applications. For instance, in the neuroimaging study, not all individuals have completed all five biannual MRI scans in two years. In the digital advertising study, not all users are exposed to all campaigns nor react to all publishers. Actually, in our digital advertising data, more than 95% of the entire tensor entries are unobserved. In this article, we tackle the problem of supervised tensor learning with partially observed tensor data.

There are several lines of research that are closely related to but also clearly distinctive of the problem we address. The first line studies tensor completion (Jain and Oh 2014; Yuan and Zhang 2016, 2017; Xia and Yuan 2017; Zhang 2019). Tensor completion aims to fill in the unobserved entries of a partially observed tensor, usually by resorting to some tensor low-rankness and sparsity structures. It is unsupervised learning, as it involves no external covariates. While we also handle tensor data with unobserved entries and employ similar low-dimensional structures as tensor completion, our goal is not to complete the tensor. Instead, we target a supervised learning problem, and aim to estimate the relationship between the partially observed tensor datasets and external covariates. Consequently, our model formulation, estimation approach, and theoretical analysis are considerably different from tensor completion. The second line tackles tensor regression where the response is a scalar and the predictor is a tensor (Zhou, Li and Zhu 2013; Wang and Zhu 2017; Hao, Zhang and Cheng 2020; Han, Willett and Zhang 2020). By contrast, we treat tensor as the response and covariates as the predictor. When it comes to theoretical analysis, the two models involve utterly different

CONTACT Will Wei Sun *sunweisurrey8@gmail.com* Krannert School of Management, Purdue University, West Lafayette, IN 47907.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JASA*.

techniques. The third line studies regressions with a tensor-valued response, while imposing different structural assumptions on the resulting tensor regression coefficient (Rabusseau and Kadri 2016; Li and Zhang 2017; Sun and Li 2017; Chen, Raskutti and Yuan 2019; Xu, Hu and Wang 2019). This line of work shares a similar goal as ours; however, none of these existing methods can handle a tensor response with partially observed entries. Moreover, none are able to pool information from the dynamic tensor data collected at adjacent time points. In our experiments, we show that focusing only on the subset of completely observed tensor data, or ignoring the structural smoothness over time would both lead to considerable loss in estimation accuracy. Finally, there have been a number of proposals motivated by similar applications and can handle missing values. Particularly, Li et al. (2013) considered an adaptive voxel-wise approach by modeling each entry of the dynamic tensor separately. We instead adopt a tensor regression approach by jointly modeling all entries of the entire tensor. We later numerically compare our method with Li et al. (2013) and other solutions. Xue and Qu (2020) studied regressions of multi-source data with missing values involving neuroimaging features. However, the images were summarized as a vector instead of a tensor, and were placed on the predictor side. Similarly, Feng et al. (2019) developed a scalar-on-image regression model with missing image scans. By contrast, we place the imaging tensor on the response side.

In this article, we develop a regression model with partially observed dynamic tensor as the response. We assume the coefficient tensor to be both sparse and low-rank, which reduces the dimension of the parameter space, lessens the computational complexity, and improves the interpretability of the model. Furthermore, we impose a fusion structure along the temporal mode of the tensor coefficient, which helps to pool the information from data observed at adjacent time points. All these assumptions are scientifically plausible, and have been widely used in numerous applications (Vounou et al. 2010; Zhou, Li and Zhu 2013; Yin et al. 2015; Rabusseau and Kadri 2016; Bi, Qu and Shen 2018; Tang, Bi and Qu 2019; Zhang et al. 2019). To handle the unobserved entries in the tensor response, we consider a loss function projected over the observed entries, which is then optimized under the low-rankness, sparsity and fusion constraints. We develop an efficient nonconvex alternating updating algorithm, and derive the finite-sample error bound of the estimator from each step of our optimization algorithm.

Unobserved entries in the tensor response have introduced serious challenges. The existing algorithms for estimating a sparse low-rank tensor and technical tools for asymptotic analysis are only applicable to either a single partially observed tensor or a fully observed tensor (e.g., Jain and Oh 2014; Sun and Li 2017). As a result, our proposal differs considerably in terms of estimation algorithm, regularity conditions, as well as theoretical properties. For estimation, since the unobserved entries can occur at different locations for different tensors, the loss function projected over the observed entries takes a complex form. The traditional vector-wise updating algorithms (Jain and Oh 2014; Sun and Li 2017) are no longer applicable. Alternatively, we propose a new procedure that updates the low-rank components of the coefficient tensor in an element-wise fashion (see Step 1 of Algorithm 1 and Equation (7) in Section 3). For regularity conditions, we add a μ-mass condition to ensure that sufficient information is contained in the observed entries for tensor coefficient estimation (see Assumption 1). We also place a lower bound on the probability of the observation p, and discuss its relation with the sample size, tensor dimension, sparsity level and mass parameter μ (see Assumptions 2 and 6). Our lower bound is different from that in the tensor completion literature (Jain and Oh 2014; Yuan and Zhang 2016, 2017; Xia and Yuan 2017), which considered only a single tensor; whereas we consider a collection of n tensors. Consequently, our lower bound on p depends on n, and tends to 0 as n tends to infinity. For theoretical properties, we show that the statistical error of our estimator has an interesting connection with the lower bound on p, which does not appear in the tensor response regression for complete data (Sun and Li 2017). This characterizes the loss at the statistical level when modeling with only partially observed tensors. In summary, our proposal is far from an incremental extension from the complete case scenario, and involves a new set of strategies for estimation and theoretical analysis.

We adopt the following notation throughout the article. Let $[d] = \{1, \ldots, d\}$, and let $\circ$ and $\otimes$ denote the outer product and Kronecker product. For a vector $\mathbf{a} \in \mathbb{R}^d$, let $\|\mathbf{a}\|$ and $\|\mathbf{a}\|_0$ denote its Euclidean norm and ℓ_0 norm, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$, let $\|\mathbf{A}\|$ denote its spectral norm. For a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$, let $\mathcal{A}_{i_1,\ldots,i_m}$ be its $(i_1, \ldots, i_m)$th entry, and $\mathcal{A}_{i_1,\ldots,i_{j-1},:,i_{j+1},\ldots,i_m} = (\mathcal{A}_{i_1,\ldots,i_{j-1},1,i_{j+1},\ldots,i_m}, \ldots, \mathcal{A}_{i_1,\ldots,i_{j-1},d_j,i_{j+1},\ldots,i_m})^\top \in \mathbb{R}^{d_j}$. Let $\text{unfold}_m(\mathcal{A})$ denote the mode-m unfolding of $\mathcal{A}$, which arranges the mode-m fibers to be the columns of the resulting matrix; for example, the mold-1 unfolding of a third-order tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is $\text{unfold}_1(\mathcal{A}) = [\mathcal{A}_{:,1,1}, \ldots, \mathcal{A}_{:,d_2,1}, \ldots, \mathcal{A}_{:,d_2,d_3}] \in \mathbb{R}^{d_1 \times (d_2 d_3)}$. Define the tensor spectral norm as $\|\mathcal{A}\| = \sup_{\|\mathbf{a}_1\|=\cdots=\|\mathbf{a}_m\|=1} |\mathcal{A} \times_1 \mathbf{a}_1 \times_2 \cdots \times_m \mathbf{a}_m|$, and the tensor Frobenius norm as $\|\mathcal{A}\|_F = \sqrt{\sum_{i_1,\ldots,i_m} \mathcal{A}^2_{i_1,\ldots,i_m}}$. For $\mathbf{a} \in \mathbb{R}^{d_j}$, define the j-mode tensor product as $\mathcal{A} \times_j \mathbf{a} \in \mathbb{R}^{d_1 \times \cdots \times d_{j-1} \times d_{j+1} \times \cdots \times d_m}$, such that $(\mathcal{A} \times_j \mathbf{a})_{i_1,\ldots,i_{j-1},i_{j+1},\ldots,i_m} = \sum_{i_j=1}^{d_j} \mathcal{A}_{i_1,\ldots,i_m} a_{i_j}$. For $\mathbf{a}_j \in \mathbb{R}^{d_j}, j \in [m]$, define the multilinear combination of the tensor entries as $\mathcal{A} \times_1 \mathbf{a}_1 \times_2 \ldots \times_m \mathbf{a}_m = \sum_{i_1 \in [d_1]} \cdots \sum_{i_m \in [d_m]} a_{1,i_1} \ldots a_{m,i_m} \mathcal{A}_{i_1,\ldots,i_m}$, where a_{j,i_j} is the i_jth entry of $\mathbf{a}_j$. For two sequences a_n, b_n, we say $a_n = \mathcal{O}(b_n)$ if $a_n \le C b_n$ for some positive constant C.

The rest of the article is organized as follows. Section 2 introduces our regression model with a partially observed dynamic tensor response. Section 3 develops the estimation algorithm. Section 4 investigates the theoretical properties. Section 5 presents the simulation results, and Section 6 illustrates with two real-world datasets, a neuroimaging study and a digital advertising study. All technical proofs are relegated to the supplementary materials.

2. Model

Suppose at each time point t, we collect an mth-order tensor $\mathcal{Y}_t$ of dimension $d_1 \times \cdots \times d_m$, $t \in [T]$. We stack the collected tensors $\mathcal{Y}_1, \ldots, \mathcal{Y}_T$ together, and represent it as an $(m+1)$th-order tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_m \times T}$. Correspondingly, the $(m+1)$th

mode of $\mathcal{Y}$ is referred as the temporal mode. Suppose there are totally n subjects in the study. For each subject i, we collect a dynamic tensor represented as $\mathcal{Y}_i$, along with a q-dimensional vector of covariates $\mathbf{x}_i \in \mathbb{R}^q, i \in [n]$. The response tensor $\mathcal{Y}_i$ can be partially observed, and the missing patterns can vary from subject to subject. We consider the following regression model:

$$\mathcal{Y}_i = \mathcal{B}^* \times_{m+2} \mathbf{x}_i + \mathcal{E}_i, \tag{1}$$

where $\mathcal{B}^* \in \mathbb{R}^{d_1 \times \cdots \times d_m \times T \times q}$ is an $(m+2)$th-order coefficient tensor, and $\mathcal{E}_i \in \mathbb{R}^{d_1 \times \cdots \times d_m \times T}$ is an $(m+1)$th-order error tensor independent of $\mathbf{x}_i$. Without loss of generality, we assume the data are centered, and thus drop the intercept term in model (1). The coefficient tensor $\mathcal{B}^*$ captures the relationship between the dynamic tensor response and the predictor, and is the main object of interest in our analysis. For instance, $\mathcal{B}^*_{i_1,\ldots,i_m,:,l} \in \mathbb{R}^T$ describes the effect of the lth covariate on the time-varying pattern of the $(i_1, \ldots, i_m)$th entry of tensor $\mathcal{Y}_t$. Next, we impose three structures on $\mathcal{B}^*$ to facilitate its analysis.

We first assume that $\mathcal{B}^*$ admits a rank-r CP decomposition structure, in that,

$$\mathcal{B}^* = \sum_{k\in[r]} w_k^* \boldsymbol{\beta}^*_{k,1} \circ \cdots \circ \boldsymbol{\beta}^*_{k,m+2}, \tag{2}$$

where $\boldsymbol{\beta}^*_{k,j} \in \mathbb{S}^{d_j}$, $\mathbb{S}^d = \{\mathbf{a} \in \mathbb{R}^d \mid \|\mathbf{a}\| = 1\}$, and $w_k^* > 0$. The CP structure is one of the most common low-rank structures (Kolda and Bader 2009), and is widely used in tensor data analysis (Zhou, Li and Zhu 2013; Anandkumar et al. 2014; Jain and Oh 2014; Yuan and Zhang 2016, 2017; Zhang 2019; Chen, Raskutti and Yuan 2019, among others). We next assume that $\mathcal{B}^*$ is sparse, in that the decomposed components $\boldsymbol{\beta}^*_{k,j}$'s are sparse. That is, $\boldsymbol{\beta}^*_{k,j} \in \mathcal{S}(d_j, s_j)$ for $j \in [m+1], k \in [r]$, where

$$\begin{aligned}\mathcal{S}(d,s) &= \left\{\boldsymbol{\beta} \in \mathbb{R}^d \mid \sum_{l=1}^{d} 1_{(\beta_l \neq 0)} \le s\right\} \\ &= \left\{\boldsymbol{\beta} \in \mathbb{R}^d \mid \|\boldsymbol{\beta}\|_0 \le s\right\}.\end{aligned}$$

This assumption postulates that the covariates $\mathbf{x}$'s effects are concentrated on a subset of entries of $\mathcal{B}^*$, which enables us to identify the most relevant regions in the dynamic tensor that are affected by the covariates. The sparsity assumption is again widely employed in numerous applications including neuroscience and online advertising (Bullmore and Sporns 2009; Vounou et al. 2010; Sun et al. 2017). We further assume a fusion structure on the decomposed components $\boldsymbol{\beta}^*_{k,j}$ of $\mathcal{B}^*$. That is, $\boldsymbol{\beta}^*_{k,j} \in \mathcal{F}(d_j, f_j)$ for $j \in [m+1], k \in [r]$, where

$$\begin{aligned}\mathcal{F}(d,f) &= \left\{\boldsymbol{\beta} \in \mathbb{R}^d \mid \sum_{l=2}^{d} 1_{(|\beta_l - \beta_{l-1}| \neq 0)} \le f\right\} \\ &= \left\{\boldsymbol{\beta} \in \mathbb{R}^d \mid \|\mathbf{D}\boldsymbol{\beta}\|_0 \le f - 1\right\},\end{aligned}$$

and $\mathbf{D} \in \mathbb{R}^{(d-1)\times d}$ with $\mathbf{D}_{i,i} = -1$, $\mathbf{D}_{i,i+1} = 1$ for $i \in [d-1]$, and other entries being zeros. This assumption encourages temporal smoothness and helps pool information from tensors observed at adjacent time points (Madrid-Padilla and Scott 2017; Sun and Li 2019). Putting the sparsity and fusion structures together, we have

$$\boldsymbol{\beta}^*_{k,j} \in \mathcal{S}(d_j, s_j) \cap \mathcal{F}(d_j, f_j), \quad \text{for} j \in [m+1],\ k \in [r]. \tag{3}$$

We briefly comment that, since the dimension q of the covariates $\mathbf{x}$ is relatively small in our motivating examples, we have chosen not to impose any sparsity or fusion structure on the component $\boldsymbol{\beta}^*_{k,m+2} \in \mathbb{R}^q$, which is the last mode of the coefficient tensor $\mathcal{B}^*$. Nevertheless, we can easily incorporate such a structure for $\boldsymbol{\beta}^*_{k,m+2}$, or other structures. The extension is straightforward, and thus is not further pursued.

A major challenge we face is that many entries of the dynamic tensor response $\mathcal{Y}$ are unobserved. Let $\Omega \subseteq [d_1] \times [d_2] \times \cdots \times [d_{m+1}]$ denote the set of indexes for the observed entries, and Ω_i denote the set of indexes for the observed entries in $\mathcal{Y}_i$, $i \in [n]$. We define a projection function $\Pi_\Omega(\cdot)$ that projects the tensor onto the observed set Ω, such that

$$[\Pi_\Omega(\mathcal{Y})]_{i_1,i_2,\ldots,i_{m+1}} = \begin{cases} \mathcal{Y}_{i_1,i_2,\ldots,i_{m+1}} & \text{if}(i_1,\ldots,i_{m+1}) \in \Omega, \\ 0 & \text{otherwise.}\end{cases}$$

We then consider the following constrained optimization problem:

$$\begin{aligned}\min_{\substack{w_k,\boldsymbol{\beta}_{k,j} \\ k\in[r],j\in[m+2]}} \ & \frac{1}{n}\sum_{i=1}^{n} \left\| \Pi_{\Omega_i}\left(\mathcal{Y}_i - \sum_{k\in[r]} w_k (\boldsymbol{\beta}_{k,m+2}^\top \mathbf{x}_i)\boldsymbol{\beta}_{k,1} \circ \cdots \circ \boldsymbol{\beta}_{k,m+1}\right)\right\|_F^2 \\ & \text{subject to } \|\boldsymbol{\beta}_{k,j}\|_2 = 1, j \in [m+2], \|\boldsymbol{\beta}_{k,j}\|_0 \le \tau_{s_j}, \\ & \left\|\mathbf{D}\boldsymbol{\beta}_{k,j}\right\|_0 \le \tau_{f_j}, j \in [m+1], k \in [r].\end{aligned} \tag{4}$$

In this optimization, both sparsity and fusion structures are imposed through ℓ_0 penalties. Such nonconvex penalties have been found effective in high-dimensional sparse models (Shen, Pan and Zhu 2012; Zhu, Shen and Pan 2014) and fused sparse models (Rinaldo 2009; Wang et al. 2016).

3. Estimation

The optimization problem in Equation (4) is highly nontrivial, as it is a non-convex optimization with multiple constraints and a complex loss function due to the unobserved entries. We develop an alternating block updating algorithm to solve Equation (4), and divide our procedure into multiple alternating steps. First, we solve an unconstrained weighted tensor completion problem, by updating $\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m+1}$, given w_k and $\boldsymbol{\beta}_{k,m+2}$, for $k \in [r]$. Since each response tensor is only partially observed and different tensors may have different missing patterns, the commonly used vector-wise updating approach in tensor analysis is no longer applicable. To address this issue, we propose a new element-wise approach to update the decomposed components of the low-rank tensor. Next, we define a series of operators and apply them to the unconstrained estimators obtained from the first step, so to incorporate the sparsity and fusion constraints on $\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m+1}$. Finally, we update w_k and $\boldsymbol{\beta}_{k,m+2}$, both of which have closed-form solutions. We summarize the procedure in Algorithm 1, then discuss each step.

Algorithm 1 Alternating block updating algorithm for (4)

input: the data $\{(\mathbf{x}_i, \mathcal{Y}_i, \Omega_i), i = 1, \ldots, n\}$, the rank r, the sparsity parameter τ_{s_j}, and the fusion parameter $\tau_{f_j}, j \in [m+1]$.
initialization: set $w_k = 1$, and randomly generate unit-norm vectors $\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m+2}$ from a standard normal distribution, $k \in [r]$.
repeat
  for $k = 1$ to r **do**
    for $j = 1$ to $m + 1$ **do**
      Step 1: obtain the unconstrained estimator $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$, given $\hat{w}_k^{(t)}$, $\hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,j-1}^{(t+1)}, \hat{\boldsymbol{\beta}}_{k,j+1}^{(t)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t)}, \hat{\boldsymbol{\beta}}_{k,m+2}^{(t)}$, by solving (5); normalize $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$.
      Step 2: obtain the constrained estimator $\hat{\boldsymbol{\beta}}_{k,j}^{(t+1)}$, by applying the `Truncatefuse` operator to $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$; normalize $\hat{\boldsymbol{\beta}}_{k,j}^{(t+1)}$.
    end for
    Step 3: obtain $\hat{w}_k^{(t+1)}$, given $\hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}, \hat{\boldsymbol{\beta}}_{k,m+2}^{(t)}$, using (8).
    Step 4: obtain $\hat{\boldsymbol{\beta}}_{k,m+2}^{(t+1)}$, given $\hat{w}_k^{(t+1)}, \hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}$, using (9).
  end for
until the stopping criterion is met.
output: $\hat{w}_k, \hat{\boldsymbol{\beta}}_{k,1}, \ldots, \widehat{\boldsymbol{\beta}}_{k,m+2}, k \in [r]$.

In Step 1, we solve an unconstrained weighted tensor completion problem,

$$\min_{\boldsymbol{\beta}_{k,j}} \frac{1}{n} \sum_{i=1}^{n} \left\{\alpha_{i,k}^{(t)}\right\}^2 \left\| \Pi_{\Omega_i} \left(\mathcal{R}_{i,k}^{(t+1)} - \hat{w}_k^{(t)} \hat{\boldsymbol{\beta}}_{k,1}^{(t+1)} \circ \cdots \circ \times \hat{\boldsymbol{\beta}}_{k,j-1}^{(t+1)} \circ \boldsymbol{\beta}_{k,j} \circ \hat{\boldsymbol{\beta}}_{k,j+1}^{(t)} \circ \cdots \circ \hat{\boldsymbol{\beta}}_{k,m+1}^{(t)} \right) \right\|_F^2, \quad (5)$$

where $\alpha_{i,k}^{(t)} = \boldsymbol{\beta}_{k,m+2}^{(t)\top} \mathbf{x}_i$, and $\mathcal{R}_{i,k}^{(t+1)}$ is a residual term defined as,

$$\mathcal{R}_{i,k}^{(t+1)} = \left(\mathcal{Y}_i - \sum_{k'<k} \hat{w}_{k'}^{(t+1)} \alpha_{i,k'}^{(t+1)} \boldsymbol{\beta}_{k',1}^{(t+1)} \circ \ldots \circ \boldsymbol{\beta}_{k',m+1}^{(t+1)} - \sum_{k'>k} \hat{w}_{k'}^{(t)} \alpha_{i,k'}^{(t)} \boldsymbol{\beta}_{k',1}^{(t)} \circ \ldots \circ \boldsymbol{\beta}_{k',m+1}^{(t)} \right) / \alpha_{i,k}^{(t)}, \quad (6)$$

for $i \in [n], k \in [r]$. The optimization problem in Equation (5) has a closed-form solution. To simplify the presentation, we give this explicit expression when $m = 2$. For the case of $m \geq 3$, the calculation is similar except involving more terms. Specifically, the lth entry of $\tilde{\boldsymbol{\beta}}_{k,3}^{(t+1)}$ is

$$\tilde{\boldsymbol{\beta}}_{k,3,l}^{(t+1)} = \frac{\sum_{i=1}^{n} \left\{\alpha_{i,k}^{(t)}\right\}^2 \sum_{l_1,l_2} \delta_{i,l_1,l_2,l} \mathcal{R}_{i,k,l_1,l_2,l}^{(t+1)} \hat{\boldsymbol{\beta}}_{k,1,l_1}^{(t+1)} \hat{\boldsymbol{\beta}}_{k,2,l_2}^{(t+1)}}{\sum_{i=1}^{n} \left\{\alpha_{i,k}^{(t)}\right\}^2 \sum_{l_1,l_2} \hat{w}_k^{(t)} \delta_{i,l_1,l_2,l} \left\{\hat{\boldsymbol{\beta}}_{k,1,l_1}^{(t+1)}\right\}^2 \left\{\hat{\boldsymbol{\beta}}_{k,2,l_2}^{(t+1)}\right\}^2}, \quad (7)$$

where $\delta_{i,l_1,l_2,l} = 1$ if $(l_1, l_2, l) \in \Omega_i$, and $\delta_{i,l_1,l_2,l} = 0$ otherwise. Here $\mathcal{R}_{i,k,l_1,l_2,l}^{(t+1)}$ refers to the (l_1, l_2, l)th entry of $\mathcal{R}_{i,k}^{(t+1)}$. The expressions for $\tilde{\boldsymbol{\beta}}_{k,1}^{(t+1)}$ and $\tilde{\boldsymbol{\beta}}_{k,2}^{(t+1)}$ can be derived similarly. We remark that, Equation (7) is the key difference between our estimation method and those for a single partially observed tensor (Jain and Oh 2014), or a completely observed tensor (Sun and Li 2017). Particularly, the observed entry indicator $\delta_{i,l_1,l_2,l}$ appears in both the numerator and denominator, and $\delta_{i,l_1,l_2,l}$ is different across different entries of $\tilde{\boldsymbol{\beta}}_{k,3}^{(t+1)}$. Therefore, $\tilde{\boldsymbol{\beta}}_{k,3}^{(t+1)}$ needs to be updated in an element-wise fashion, as $\delta_{i,l_1,l_2,l}$ could not be cancelled. After obtaining Equation (7), we normalize $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$ to ensure a unit norm.

In Step 2, we apply the sparsity and fusion constraints to $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$ obtained in the first step. Toward that goal, we define a truncation operator $\texttt{Truncate}(\mathbf{a}, \tau_s)$, and a fusion operator $\texttt{Fuse}(\mathbf{a}, \tau_f)$, for a vector $\mathbf{a} \in \mathbb{R}^d$ and two integer-valued tuning parameters τ_s and τ_f, as,

$$[\texttt{Truncate}(\mathbf{a}, \tau_s)]_j = \begin{cases} a_j \text{ if} j \in \text{supp}(\mathbf{a}, \tau_s) \\ 0 \text{ otherwise} \end{cases};$$

$$[\texttt{Fuse}(\mathbf{a}, \tau_f)]_j = \sum_{i=1}^{\tau_f} 1_{j \in \mathcal{C}_i} \frac{1}{|\mathcal{C}_i|} \sum_{l \in \mathcal{C}_i} a_l,$$

where $\text{supp}(\mathbf{a}, \tau_s)$ refers to the indexes of τ_s entries with the largest absolute values in $\mathbf{a}$, and $\{\mathcal{C}_i\}_{i=1}^{\tau_f}$ are the fusion groups. This truncation operator ensures that the total number of nonzero entries in $\mathbf{a}$ is bounded by τ_s, and is commonly employed in non-convex sparse optimizations (Yuan and Zhang 2013; Sun et al. 2017). The fusion groups $\{\mathcal{C}_i\}_{i=1}^{\tau_f}$ are calculated as follows. First, the truncation operator is applied to $\mathbf{Da} \in \mathbb{R}^{d-1}$. The resulting $\texttt{Truncate}(\mathbf{Da}, \tau_f - 1)$ has at most $(\tau_f - 1)$ nonzero entries. Then the elements a_j and a_{j+1} are put into the same group if $[\texttt{Truncate}(\mathbf{Da}, \tau_f - 1)]_j = 0$. This procedure in effect groups the elements in $\mathbf{a}$ into τ_f distinct groups, which we denote as $\{\mathcal{C}_i\}_{i=1}^{\tau_f}$. Elements in each of the τ_f groups are then averaged to obtain the final result. Combining the two operators, we obtain the $\texttt{Truncatefuse}(\mathbf{a}, \tau_s, \tau_f)$ operator as,

$$\texttt{Truncatefuse}(\mathbf{a}, \tau_s, \tau_f) = \texttt{Truncate}\{\texttt{Fuse}(\mathbf{a}, \tau_f), \tau_s\},$$

where $\tau_s \leq d$ is the sparsity parameter, and $\tau_f \leq d$ is the fusion parameter. For example, consider $\mathbf{a} = (0.1, 0.2, 0.4, 0.5, 0.6)^\top$, $\tau_s = 3$ and $\tau_f = 2$. Correspondingly, $\mathbf{Da} = (0.1, 0.2, 0.1, 0.1)^\top$. We then have $\texttt{Truncate}(\mathbf{Da}, \tau_f - 1) = (0, 0.2, 0, 0)^\top$. This in effect suggests that a_1, a_2 belong to one group, and a_3, a_4, a_5 belong to the other group. We then average the values of $\mathbf{a}$ in each group, and obtain $\texttt{Fuse}(\mathbf{a}, \tau_f) = (0.15, 0.15, 0.5, 0.5, 0.5)^\top$. Lastly, $\texttt{Truncatefuse}(\mathbf{a}, \tau_s, \tau_f) = \texttt{Truncate}\{\texttt{Fuse}(\mathbf{a}, \tau_f), \tau_s\} = \texttt{Truncate}\{(0.15, 0.15, 0.5, 0.5, 0.5)^\top, 3\} = (0, 0, 0.5, 0.5, 0.5)^\top$. We apply the `Truncatefuse` operator to the unconstrained estimator $\tilde{\boldsymbol{\beta}}_{k,j}^{(t+1)}$ obtained from the first step, with the sparsity parameter τ_{s_j} and the fusion parameter τ_{f_j}, and normalize the result to ensure a unit norm.

In Step 3, we update $\hat{w}_k^{(t+1)}$, given $\hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}, \hat{\boldsymbol{\beta}}_{k,m+2}^{(t)}$, which has a closed-form solution,

$$\hat{w}_k^{(t+1)} = \frac{\mathcal{R}^{(t+1)} \times_1 \hat{\boldsymbol{\beta}}_{k,1}^{(t+1)} \times_2 \cdots \times_{m+1} \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}}{\sum_{i=1}^{n} \left\{\alpha_{i,k}^{(t)}\right\}^2 \left\| \Pi_{\Omega_i} \left(\hat{\boldsymbol{\beta}}_{k,1}^{(t+1)} \circ \cdots \circ \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)} \right) \right\|_F^2}, \quad (8)$$

where $\mathcal{R}^{(t+1)} = \sum_{i=1}^{n} \left\{\alpha_{i,k}^{(t)}\right\}^2 \Pi_{\Omega_i}\left(\mathcal{R}_{i,k}^{(t+1)}\right)$, and $\mathcal{R}_{i,k}^{(t+1)}$ is as defined in Equation (6) by replacing $\hat{\boldsymbol{\beta}}_{k,1}^{(t)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t)}$ with $\hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}$.

In Step 4, we update $\hat{\boldsymbol{\beta}}_{k,m+2}^{(t+1)}$, given $\hat{w}_k^{(t+1)}, \hat{\boldsymbol{\beta}}_{k,1}^{(t+1)}, \ldots, \hat{\boldsymbol{\beta}}_{k,m+1}^{(t+1)}$, which again has a closed-form solution. Write $\tilde{\mathcal{R}}_{i,k}^{(t+1)} = \mathcal{Y}_i - \sum_{k' \neq k, k' \in [r]} w_{k'}^{(t+1)} \alpha_{i,k'}^{(t)} \boldsymbol{\beta}_{k',1}^{(t+1)} \circ \cdots \circ \boldsymbol{\beta}_{k',m+1}^{(t+1)}$, and $\mathcal{A}_k^{(t+1)} = w_k^{(t+1)} \boldsymbol{\beta}_{k,1}^{(t+1)} \circ \cdots \circ \boldsymbol{\beta}_{k,m+1}^{(t+1)}$. Then we have,

$$\hat{\boldsymbol{\beta}}_{k,m+2}^{(t+1)} = \left\{ \frac{1}{n} \sum_{i=1}^{n} \left\| \Pi_{\Omega_i}\left(\mathcal{A}_k^{(t+1)}\right) \right\|_F^2 \mathbf{x}_i \mathbf{x}_i^\top \right\}^{-1} \times n^{-1} \sum_{i=1}^{n} \left\langle \Pi_{\Omega_i}\left(\tilde{\mathcal{R}}_{i,k}^{(t+1)}\right), \Pi_{\Omega_i}\left(\mathcal{A}_k^{(t+1)}\right) \right\rangle \mathbf{x}_i, \quad (9)$$

where $\langle \cdot, \cdot \rangle$ is the tensor inner product.

We make some remarks regarding the convergence of Algorithm 1. First, with a suitable initial value, the iterative estimator from Algorithm 1 converges to a neighborhood that is within the statistical precision of the true parameter at a geometric rate, as we show later in Theorems 1 and 2. These results also provide a theoretical termination condition for Algorithm 1. That is, when the computational error is dominated by the statistical error, we can stop the algorithm. In practice, we iterate the algorithm until the estimates from two consecutive iterations are close, that is, $\max_{j \in [m+2], k \in [r]} \min\left\{ \left\| \hat{\boldsymbol{\beta}}_{k,j}^{(t+1)} - \hat{\boldsymbol{\beta}}_{k,j}^{(t)} \right\|, \left\| \hat{\boldsymbol{\beta}}_{k,j}^{(t+1)} + \hat{\boldsymbol{\beta}}_{k,j}^{(t)} \right\| \right\} \leq 10^{-4}$. Second, with *any* initial value, and if there are no sparsity and fusion constraints, that is, without the `Truncatefuse` step, then Algorithm 1 is guaranteed to converge to a stationary point, because the objective function monotonically decreases at each iteration (Wang and Li 2020). Finally, when imposing the sparsity and fusion constraints, the algorithmic convergence from any initial value becomes very challenging, since both constraints are non-convex. Actually, the general convergence of non-convex optimizations remains an open question. For instance, in the existing non-convex models that employ truncation in optimizations, including sparse PCA (Ma 2013), high-dimensional EM (Wang et al. 2015b), sparse phase retrieval (Cai, Li and Ma 2016), sparse tensor decomposition (Sun et al. 2017), and sparse generalized eigenvalue problem (Tan et al. 2018), the convergence to a stationary point has only been established for a suitable initial value, but not for any initial value. We leave this for future research.

The proposed Algorithm 1 involves a number of tuning parameters, including the rank r, the sparsity parameter τ_{s_j}, and the fusion parameter τ_{f_j}, $j \in [m+1]$. We propose to tune the parameters by minimizing a BIC-type criterion,

$$2 \log\left\{ \frac{1}{n} \sum_{i=1}^{n} \left\| \Pi_{\Omega_i}\left(\mathcal{Y}_i - \hat{\mathcal{B}} \times_{m+2} \mathbf{x}_i\right) \right\|_F^2 \right\} + \frac{\log\left(n \prod_{j=1}^{m+1} d_j\right)}{n \prod_{j=1}^{m+1} d_j} \times \mathrm{df}, \quad (10)$$

where the total degrees of freedom df is the total number of unique nonzero entries of $\boldsymbol{\beta}_{k,j}$. The criterion in Equation (10) naturally balances the model fitting and model complexity. Similar BIC-type criterions have been used in tensor data analysis (Zhou, Li and Zhu 2013; Wang et al. 2015a; Sun and Li 2017). To further speed up the computation, we tune the three sets of parameters r, τ_{s_j} and τ_{f_j} sequentially. That is, among the set of values for r, τ_{s_j}, τ_{f_j}, we first tune r while fixing τ_{s_j}, τ_{f_j} at their maximum values. Then, given the selected r, we tune τ_{s_j}, while fixing τ_{f_j} at its maximum value. Finally, given the selected r and τ_{s_j}, we tune τ_{f_j}. In practice, we find such a sequential procedure yields good numerical performance.

4. Theory

We next derive the nonasymptotic error bound of the actual estimator obtained from Algorithm 1. We first develop the theory for the case of rank $r = 1$, because this case has clearly captured the roles of various parameters, including the sample size, tensor dimension, and proportion of the observed entries, on both the computational and statistical errors. We then generalize to the case of rank $r > 1$. Due to the involvement of the unobserved entries, our theoretical analysis is highly nontrivial, and is considerably different from Sun and Li (2017, 2019). We discuss in detail the effect of missing entries on both the regularity conditions and the theoretical properties.

We first introduce the definition of the sub-Gaussian distribution.

Definition 1 (sub-Gaussian). The random variable ξ is said to follow a sub-Gaussian distribution with a variance proxy σ^2, if $\mathbb{E}(\xi) = 0$, and for all $t \in \mathbb{R}$, $\mathbb{E}(\exp\{t\xi\}) \leq \exp\{t^2\sigma^2/2\}$.

Next we introduce some basic model assumptions common for both $r = 1$ and $r > 1$. Let s_j denote the number of nonzero entries in $\boldsymbol{\beta}_{k,j}^*$, $j \in [m+1]$, and $s = \max_j\{s_j\}$.

Assumption 1. Assume the following conditions hold.

(i) The predictor $\mathbf{x}_i$ satisfies that $\|\mathbf{x}_i\| \leq c_1$, $n^{-1}\sum_{i=1}^n \|\mathbf{x}_i\mathbf{x}_i^\top\|_2 \leq c_2$, $i \in [n]$, and $1/c_0 < \lambda_{\min} \leq \lambda_{\max} < c_0$, where $\lambda_{\min}, \lambda_{\max}$ are the minimum and maximum eigenvalues of the sample covariance matrix $\Sigma = n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$, respectively, and c_0, c_1, c_2 are some positive constants.

(ii) The true tensor coefficient $\mathcal{B}^*$ in (1) satisfies the CP decomposition (2) with sparsity and fusion constraints (3), and the decomposition is unique up to a permutation. Moreover, $\|\mathcal{B}^*\| \leq c_3 w_{\max}^*$ where $w_{\max}^* = \max_k\{w_k^*\}$, $w_{\min}^* = \min_k\{w_k^*\}$, and c_3 is some positive constant. Furthermore, $w_{\max}^* = \mathcal{O}(w_{\min}^*)$.

(iii) The decomposed component $\boldsymbol{\beta}_{k,j}^*$ is a μ-mass unit vector, in that $\max_{l \in d_j} |\boldsymbol{\beta}_{k,j,l}^*| \leq \mu/\sqrt{s}$.

(iv) The entries in the error tensor $\mathcal{E}_i$ are iid sub-Gaussian with a variance proxy σ^2.

(v) The entries of the dynamic tensor response $\mathcal{Y}_i$ are observed independently with an equal probability $p \in (0, 1]$.

We make some remarks about these conditions. Assumption 1(i) is placed on the design matrix, which is mild and can be easily verified when $\mathbf{x}_i$ is of a fixed dimension. Assumption 1(ii)

is about the key structures we impose on the coefficient tensor $\mathcal{B}^*$. It also ensures the identifiability of the decomposition of $\mathcal{B}^*$, which is always imposed in CP decomposition based tensor analysis (Zhou, Li and Zhu 2013; Sun and Li 2017; Chen, Raskutti and Yuan 2019). Assumption 1(iii) is to ensure that the mass of the tensor would not concentrate on only a few entries. In that extreme case, randomly observed entries of the tensor response may not contain enough information to recover $\mathcal{B}^*$. Note that, since $\boldsymbol{\beta}^*_{k,j}$ is a vector of unit length, a relatively small μ implies that the nonzero entries in $\boldsymbol{\beta}^*_{k,j}$ would be more uniformly distributed. This condition has been commonly imposed in the tensor completion literature for the same purpose (Jain and Oh 2014). Assumption 1(iv) assumes the error terms follow a sub-Gaussian distribution. This assumption is again fairly common in the theoretical analysis of tensor models (Cai et al. 2019; Xia, Yuan and Zhang 2020). Finally, Assumption 1(v) specifies the mechanism of how each entry of the tensor response is observed, which is assumed to be independent of each other and have an equal observation probability. We recognize that this is a relatively simple mechanism. It may not always hold in real applications, as the actual observation pattern of the tensor data can depend on multiple factors, and may not be independent for different entries. We impose this condition for our theoretical analysis, even though our estimation algorithm does not require it. In the tensor completion literature, this mechanism has been commonly assumed (Jain and Oh 2014; Yuan and Zhang 2016, 2017; Xia and Yuan 2017). We have chosen to impose this assumption because the theory of supervised tensor learning even for this simple mechanism remains unclear, and is far from trivial. We feel a rigorous theoretical analysis for this mechanism it deserves a full investigation. We leave the study under a more general observation mechanism for future research.

4.1. Theory With $r = 1$

To ease the notation and simplify the presentation, we focus primarily on the case with a third-order tensor response, that is, $m = 2$. This however does not lose generality, as all our results can be extended to the case of $m > 2$ in a straightforward fashion. Let $d = \max\{d_1, \ldots, d_{m+1}\}$. Next, we introduce some additional regularity conditions.

Assumption 2. Assume the observation probability p satisfies that,

$$p \geq \frac{c_4\{\log(d)\}^4\mu^3}{n\, s^{1.5}}.$$

where $c_4 > 0$ is some constant.

Due to Assumption 1(v), the observation probability p also reflects the proportion of the observed entries of the tensor response. Assumption 2 places a lower bound on this proportion to ensure a good recovery of the tensor coefficient. This bound depends on the sample size n, true sparsity parameter s, maximum dimension d, and mass parameter μ. We discuss these dependencies in detail. First, compared to the lower bound conditions on p in the tensor completion literature where a single tensor is considered (Jain and Oh 2014; Yuan and Zhang 2016, 2017; Xia and Yuan 2017; Cai et al. 2019), our lower bound is different, as it depends on the number of tensor samples n, and it tends to 0 as n tends to infinity. When $n = 1$, our lower bound is comparable to that in Jain and Oh (2014), Cai et al. (2019), with s replaced by d, as they did not consider any sparsity. Second, the lower bound on p increases as s decreases, that is, as the data become more sparse. This is because, when the sparsity is involved, both our tensor regression problem and the tensor completion problem become more difficult. Intuitively, when the sparsity increases, the nonzero elements may concentrate on only a few tensor entries. As a result, a larger proportion of the tensor entries needs to be observed to ensure that a sufficient number of nonzero elements can be observed for tensor estimation or completion. We also note that this condition on the lower bound on p is different from the sample complexity condition on n that we will introduce in Assumption 5. The latter suggests that the required sample size n decreases as s decreases. Third, when there is no sparsity, Jain and Oh (2014), Cai et al. (2019) showed that the lower bound on p is of the order $(\log d)^4/(d^{3/2})$, which decreases as d increases. In our setting with the sparsity, however, the lower bound on p increases as d increases. Finally, the lower bound on p increases as the mass parameter μ increases. This is because when μ increases, the mass of the tensor may become more likely to concentrate on a few entries, and thus the entries need to be observed with a larger probability to ensure the estimation accuracy.

Assumption 3. Assume the sparsity and fusion parameters satisfy that $\tau_{s_j} \geq s_j, \tau_{s_j} = \mathcal{O}(s_j)$, and $\tau_{f_j} \geq f_j$. Moreover, define the minimal gap, $\Delta^* = \min_{1<s\leq d_j, \boldsymbol{\beta}^*_{1,j,s} \neq \boldsymbol{\beta}^*_{1,j,s-1}, j\in[3]} |\boldsymbol{\beta}^*_{1,j,s} - \boldsymbol{\beta}^*_{1,j,s-1}|$. Assume that, for the positive constant C_1 as defined in Theorem 1, we have

$$\Delta^* > \frac{C_1\sigma}{w_1^*}\sqrt{\frac{s\log(d)}{np}}.$$

The condition for the sparsity parameter ensures that the truly nonzero elements would not be shrunk to zero. Similar conditions have been imposed in truncated sparse models (Yuan and Zhang 2013; Wang et al. 2015b; Sun et al. 2017; Tan et al. 2018). The conditions for the fusion parameter and the minimum gap ensure that the fused estimator would not incorrectly merge two distinct groups of entries in the true parameter. Such conditions are common in sparse and fused regression models (Tibshirani et al. 2005; Rinaldo 2009).

Assumption 4. Define the initialization error $\epsilon = \max\{|\widehat{w}_1^{(0)} - w_1^*|/w_1^*, \max_j \|\widehat{\boldsymbol{\beta}}^{(0)}_{1,j} - \boldsymbol{\beta}^*_{1,j}\|_2\}$. Assume that

$$\epsilon < \min\left\{\frac{\lambda^3_{\min}}{24\sqrt{10}\, c_2\, \lambda^2_{\max}}, \frac{1}{6}\right\},$$

where c_2 is the same constant as in Assumption 1.

This assumption is placed on the initialization error of Algorithm 1, and requires that the initial values are reasonably close to the true parameters. Particularly, the condition on ϵ requires the initial error to be smaller than some constant, which is a relatively mild condition, since $\boldsymbol{\beta}^*_{1,j}$'s are unit vectors. Such constant initialization condition is commonly employed in the

tensor literature (Sun and Li 2017; Han, Willett and Zhang 2020; Xia, Yuan and Zhang 2020). In Section 4.3, we further propose an initialization procedure, and show both theoretically and empirically that such a procedure can produce initial values that satisfy Assumption 4.

Assumption 5. Assume the sample size n satisfies that

$$n \geq \max\left\{ \frac{c_5\, \sigma^2\, s^2\, \log(d)}{w_1^{*2} p}, \frac{c_6 \sigma s \log(d) \log(\sqrt{s^3/p})}{w_1^* p} \right\}$$

where c_5 and c_6 are some positive constants.

There are two terms in this lower bound, both of which are due to the error tensor $\mathcal{E}_i$ in the model and the missing entries in the response tensor. In addition, the first term is needed to ensure the μ-mass condition is satisfied. When the observational probability p satisfies the lower bound requirement in Assumption 2, the required sample size decreases as s decreases, since in this case the number of free parameters decreases. When the strength of signal w_1^* increases or the noise level σ decreases, the required sample size also decreases.

We now state the main theory for the estimator of Algorithm 1 when $r = 1$.

Theorem 1. Suppose Assumptions 1–5 hold. When the tensor rank $r = 1$, the estimator from the tth iteration of Algorithm 1 satisfies that, with high probability,

$$\max\left\{ |\widehat{w}_1^{(t)} - w_1^*|/w_1^*,\ \max_j \|\widehat{\boldsymbol{\beta}}_{1,j}^{(t)} - \boldsymbol{\beta}_{1,j}^*\|_2 \right\} \leq \underbrace{\kappa^t \epsilon}_{\text{computational error}} + \underbrace{\frac{1}{1-\kappa} \frac{C_1 \sigma}{w_1^*} \sqrt{\frac{s \log(d)}{np}}}_{\text{statistical error}},$$

where $\kappa = 6\sqrt{10} c_2 \lambda_{\max}^2 \epsilon / \lambda_{\min}^3 + 1/2 < 1$ is the positive contraction coefficient, with ϵ as defined in Assumption 4, and the constant $C_1 = (6\sqrt{10}\tilde{C}\lambda_{\max} + \tilde{C}_2 \lambda_{\min}\sqrt{q})/\lambda_{\min}^2$. Here, c_2 is the same constant as defined in Assumptions 1, $\tilde{C}, \tilde{C}_2$ are some positive constants, and q is fixed under Assumption 1(i).

The nonasymptotic error bound in Theorem 1 can be decomposed as the sum of a computational error and a statistical error. The former is related to the optimization procedure, while the latter is related to the statistical model. The statistical error decreases with a decreasing κ, an increasing signal-to-noise ratio as reflected by σ/w_1^*, an increasing sample size n and an increasing observation probability p. When $p = 1$ and $\sigma = 1$, the statistical error rate in our Theorem 1 actually improves the statistical error rate in the completely observed tensor response regression (Sun and Li 2017), which is of order $1/w_1^* \sqrt{s^3 \log(d)/n}$. This improvement is achieved because we have employed a new proof technique using the covering number argument (Ryota and Taiji 2014) in bounding the sparse spectral of the error tensor, which allows us to obtain a sharper rate in terms of the sparsity parameter s. Moreover, when $n = 1$ and $s = d$, our statistical error rate matches with the rate $\sigma/w_1^* \sqrt{d \log(d)/p}$ in the nonsparse tensor completion (Cai et al. 2019).

One of the key challenges of our theoretical analysis is the complicated form of the element-wise estimator $\tilde{\boldsymbol{\beta}}_{k,3}$ in Equation (7). Consequently, one cannot directly characterize the distance between $\tilde{\boldsymbol{\beta}}_{k,3}/\|\tilde{\boldsymbol{\beta}}_{k,3}\|$ and $\boldsymbol{\beta}_{k,3}^*$ with a simple analytical form. Furthermore, the presence of noise error poses several fundamental challenges. The missing entries in noise tensors make existing proof techniques no longer applicable in our theoretical analysis. As we shall demonstrate later, we need to carefully control the upper bound of error tensor with missing entries.

We also briefly comment that, Theorem 1 provides a theoretical termination condition for Algorithm 1. When the number of iterations t exceeds $O\{\log_{1/\kappa}(\epsilon/\varepsilon^*)\}$, where ε^* is the statistical error term in Theorem 1, then the computational error is to be dominated by the statistical error, and the estimator falls within the statistical precision of the true parameter.

4.2. Theory With $r > 1$

Next, we extend our theory to the general rank $r > 1$. The regularity conditions for the general rank case parallel those for the rank one case. Meanwhile, some modifications are needed, due to the interplay among different decomposed components $\boldsymbol{\beta}_{k,j}$.

Assumption 6. Assume the observation probability p satisfies that

$$p \geq \frac{c_7 \{\log(d)\}^4 \mu^3 r w_{\max}^{*2}}{n\, s^{1.5}\, w_{\min}^{*2}},$$

where $c_7 > 0$ is some constants.

For the general rank case, the lower bound on the observation probability p depends additionally on the rank r and the ratio $w_{\max}^*/w_{\min}^*$. In particular, the lower bound will increase with an increasing rank r, which suggests that more observations are needed if the rank of the coefficient tensor increases. When the sample size $n = 1$, our condition is comparable to that in tensor completion (Jain and Oh 2014), where the latter requires $p \geq c\mu^6 r^5 w_{\max}^{*4}/(d^{1.5} w_{\min}^{*4})$ ignoring the logarithm term, with s replaced by d, as they do not consider any sparsity.

Assumption 7. Assume the sparsity and fusion parameters satisfy that $\tau_{s_j} \geq s_j, \tau_{s_j} = \mathcal{O}(s_j)$, and $\tau_{f_j} \geq f_j$. Moreover, define the minimal gap $\Delta^* = \min_{1<s\leq d_j, \boldsymbol{\beta}_{k,j,s}^* \neq \boldsymbol{\beta}_{k,j,s-1}^*, j\in[3], k\in[r],} |\boldsymbol{\beta}_{k,j,s}^* - \boldsymbol{\beta}_{k,j,s-1}^*|$. Assume that,

$$\Delta^* > \frac{C_2 \sigma w_{\max}^*}{w_{\min}^{*2}} \sqrt{\frac{s \log(d)}{np}},$$

where positive constant C_2 is the same constant as defined in Theorem 2.

This assumption is similar to Assumption 3, and it reduces to Assumption 3 when $r = 1$.

Assumption 8. Define $\epsilon = \max_k \left\{|\hat{w}_k^{(0)} - w_k^*|/w_k^*, \max_j \|\widehat{\boldsymbol{\beta}}_{k,j}^{(0)} - \boldsymbol{\beta}_{k,j}^*\|_2\right\}$. Assume ϵ satisfies,

$$\epsilon < \min\left\{\frac{\lambda_{\min}^3 w_{\min}^{*2}}{24\sqrt{10}c_2\lambda_{\max}^2 w_{\max}^{*2} r}, \frac{\lambda_{\min}^3 w_{\min}^{*3}}{4c_1^2 c_2 \lambda_{\max} w_{\max}^{*3} r^2}, \frac{1}{6}\right\},$$

where c_1, c_2 are the same constants as defined in Assumption 1.

It is seen that the initial error depends on the rank r. The upper bound tightens as r increases, as in such a case, the tensor recovery problem becomes more challenging. It is also noted that, when $r = 1$, this condition is still stronger than that in Assumption 4. This is due to the interplay among different decomposed components in the general rank case.

Assumption 9. Define the incoherence parameter $\xi = \max_{j\in[3],k\neq k'} \left|\langle \boldsymbol{\beta}_{k,j}^*, \boldsymbol{\beta}_{k',j}^*\rangle\right|$. Assume,

$$\xi \le \frac{\lambda_{\min}^3 w_{\min}^{*3}}{4c_1^2 c_2 \lambda_{\max} w_{\max}^{*3} r^2},$$

where c_1, c_2 are the same constants as defined in Assumption 1.

For the general rank case, we need to control the correlations between the decomposed components across different ranks. The incoherence parameter ξ quantifies such correlations. As rank r increases, the upper bound on ξ becomes tighter. Similar conditions have been introduced in Anandkumar et al. (2014), Sun et al. (2017), and Hao, Zhang and Cheng (2020).

Assumption 10. Assume the sample size n satisfies that,

$$n \ge \max\left\{\frac{c_5\,\sigma^2\, s^2\, \log(d)}{w_{\min}^{*2} p}, \frac{c_6 \sigma s \log(d)\log(\sqrt{s^3/p})}{w_{\min}^* p}\right\}$$

where c_5 and c_6 are the same positive constants as defined in Assumption 5.

This assumption is similar to Assumption 5, and it reduces to Assumption 5 when $r = 1$.

We next state the main theory for the estimator of Algorithm 1 when $r > 1$.

Theorem 2. Suppose Assumptions 1 and 6–10 hold. For a general rank r, the estimator from the tth iteration of Algorithm 1 satisfies that, with a high probability,

$$\max\left\{\max_k |\widehat{w}_k^{(t)} - w_k^*|/w_k^*, \max_{k,j} \|\widehat{\boldsymbol{\beta}}_{k,j}^{(t)} - \boldsymbol{\beta}_{k,j}^*\|_2\right\} \le \underbrace{\tilde{\kappa}^t \epsilon}_{\text{computational error}} + \underbrace{\frac{1}{1-\tilde{\kappa}} \frac{C_2 w_{\max}^* \sigma}{w_{\min}^{*2}} \sqrt{\frac{s\log(d)}{np}}}_{\text{statistical error}}.$$

where

$$\tilde{\kappa} = \frac{6\sqrt{10}c_2\lambda_{\max}^2 w_{\max}^{*2} r}{\lambda_{\min}^3 w_{\min}^{*2}}\epsilon + \frac{c_1^2 c_2 \lambda_{\max} w_{\max}^{*3} r^2}{\lambda_{\min}^3 w_{\min}^{*3}}\epsilon + \frac{c_1^2 c_2 \lambda_{\max} w_{\max}^{*3} r^2}{\lambda_{\min}^3 w_{\min}^{*3}}\xi + \frac{1}{4} < 1,$$

is the positive contraction coefficient, and the constants $C_2 = (6\sqrt{10}\tilde{C}\lambda_{\max} + 12\tilde{C}_2\sqrt{q}\lambda_{\min})/\lambda_{\min}^2$. Here c_1, c_2 is the same constant as defined in Assumptions 1, $\tilde{C}, \tilde{C}_2$ are some positive constants, and q is fixed under Assumption 1(i).

The contraction coefficient $\tilde{\kappa}$ is greater than κ in Theorem 1, which indicates that the algorithm has a slower convergence rate for the general rank case. Moreover, $\tilde{\kappa}$ increases with an increasing rank r. This agrees with the expectation that, as the tensor recovery problem becomes more challenging, the algorithm will have a slower convergence rate.

4.3. Initialization

As the optimization problem in Equation (4) is nonconvex, the success of Algorithm 1 replies on good initializations. Motivated by Cai et al. (2019), we next propose a spectral initialization procedure for $r = 1$ and $r > 1$, respectively. Theoretically, we show that the produced initial estimator satisfies the initialization Assumption 4 when $r = 1$. Numerically, we demonstrate that the initialization error decays fast for both $r = 1$ and $r > 1$ cases as the sample size n increases, and thus the constant initialization error bound in the initialization Assumptions 4 and 8 is expected to hold with a sufficiently large n.

We first present the initialization procedure for $r = 1$ in Algorithm 2. Denote $\mathcal{T} = n^{-1}\sum_i \Pi_{\Omega_i}(\mathcal{Y}_i)$. Let $\mathbf{A}_1 = \text{unfold}_3(p^{-1}\mathcal{T}) \in \mathbb{R}^{d_3\times d_1 d_2}$, and $\mathbf{B}_1 = \Pi_{\text{off-diag}}(\mathbf{A}_1\mathbf{A}_1^\top) \in \mathbb{R}^{d_3\times d_3}$, where $\Pi_{\text{off-diag}}(\cdot)$ keeps only the off-diagonal entries of the matrix. Let $\mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^\top$ be the rank-r decomposition of $\mathbf{B}_1$. Next, let $\mathbf{A}_2 = \text{unfold}_1(p^{-1}\mathcal{T}) \in \mathbb{R}^{d_1\times d_2 d_3}$, $\mathbf{B}_2 = \Pi_{\text{off-diag}}(\mathbf{A}_2\mathbf{A}_2^\top) \in \mathbb{R}^{d_1\times d_1}$, and $\mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^\top$ be the rank-r decomposition of $\mathbf{B}_2$. We then feed $\mathbf{U}_1$ and $\mathbf{U}_2$ into Algorithm 2.

Algorithm 2 Spectral initialization algorithm for $r = 1$.

input: the number of restarts L, the estimates $\mathbf{U}_1$, $\mathbf{U}_2$, and the sparsity parameter τ_{s_j}, $j \in [3]$.
for $l = 1$ to L **do**
  generate $\mathbf{g}_1^l \sim \text{Normal}(\mathbf{0}, \mathbf{I}_{d_3})$, and compute $\tilde{\mathbf{g}}_1^l = \mathbf{U}_1\mathbf{U}_1^\top\mathbf{g}_1, \mathbf{M}_1^l = p^{-1}\mathcal{T}\times_3\tilde{\mathbf{g}}_1^l$.
  set $\mathbf{v}_1^l$ and $\mathbf{v}_2^l$ as the first left and right singular vector of $\mathbf{M}_1^l$ corresponding to the largest absolute singular value $|\lambda_1^l|$.
end for
for $l = 1$ to L **do**
  generate $\mathbf{g}_2^l \sim \text{Normal}(\mathbf{0}, \mathbf{I}_{d_1})$, and compute $\tilde{\mathbf{g}}_2^l = \mathbf{U}_2\mathbf{U}_2^\top\mathbf{g}_2, \mathbf{M}_2^l = p^{-1}\mathcal{T}\times_3\tilde{\mathbf{g}}_2^l$.
  set $\mathbf{v}_3^l$ and $\mathbf{v}_4^l$ as the left and right singular vector of $\mathbf{M}_2^l$ corresponding to the largest absolute singular value $|\lambda_2^l|$.
end for
choose $(\mathbf{v}_1, \mathbf{v}_2)$ from $\{(\mathbf{v}_1^l, \mathbf{v}_2^l)\}_{l=1}^L$ with the largest $|\lambda_1^l|$; choose $(\mathbf{v}_3, \mathbf{v}_4)$ similarly.
compute $\widehat{\boldsymbol{\beta}}_{1,j}^{(0)} = \texttt{Norm}(\texttt{Truncate}(\tilde{\mathbf{v}}_j, \tau_{s_j}))$ for $j = 1, 2, 3$, where $(\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_3)$ is obtained from $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_3, \mathbf{v}_4)$, and Norm is the normalization operator.
compute $\hat{w}_1^{(0)}$ and $\widehat{\boldsymbol{\beta}}_{1,4}^{(0)}$ using (11).
output: $\hat{w}_1^{(0)}$, $\widehat{\boldsymbol{\beta}}_{1,1}^{(0)}$, $\widehat{\boldsymbol{\beta}}_{1,2}^{(0)}$, $\widehat{\boldsymbol{\beta}}_{1,3}^{(0)}$ and $\widehat{\boldsymbol{\beta}}_{1,4}^{(0)}$.

When $r = 1$, we have $E(\mathbf{A}_1) = w_1^* \sum_i \frac{1}{n}(\boldsymbol{\beta}_{1,4}^{*\top}\mathbf{x}_i)\boldsymbol{\beta}_{1,3}^*(\boldsymbol{\beta}_{1,1}^* \otimes \boldsymbol{\beta}_{1,2}^*)^\top$, whose column space is the span of $\boldsymbol{\beta}_{1,3}^*$. A natural way to estimate the column space of $E(\mathbf{A}_1)$ is from the principal space of $\mathbf{A}_1\mathbf{A}_1^\top$. Similar to Cai et al. (2019), we exclude the diagonal entries of $\mathbf{A}_1\mathbf{A}_1^\top$ to remove their influence on the principal directions. To retrieve tensor factors from the subspace estimate, we first generate random vectors from normal distribution, that is, $\mathbf{g}_1^l$ in line 3 and $\mathbf{g}_2^l$ in line 7 of Algorithm 2 . Then we project the random vectors $\mathbf{g}_1^l$ and $\mathbf{g}_2^l$ onto $\mathbf{U}_1$ and $\mathbf{U}_2$. This projection step helps mitigate the perturbation incurred by both unobserved values and data noise (Cai et al. 2019). Note that $E(\mathbf{M}_1^l \mid \tilde{\mathbf{g}}_1^l) = w_1^* \sum_i n^{-1}(\boldsymbol{\beta}_{1,4}^{*\top}\mathbf{x}_i)\langle\boldsymbol{\beta}_{1,3}^*, \tilde{\mathbf{g}}_1^l\rangle\boldsymbol{\beta}_{1,1}^*\boldsymbol{\beta}_{1,2}^{*\top}$. Correspondingly, the left leading singular vector corresponding to the largest absolute singular value of $\mathbf{M}_1^l$ is expected to be close to $\boldsymbol{\beta}_{1,1}^*$. Similarly, the right leading singular vector of $\mathbf{M}_1^l$ is expected to be close to $\boldsymbol{\beta}_{1,2}^*$. Following the same argument, we can obtain a good estimate of $\boldsymbol{\beta}_{1,2}^*$ and $\boldsymbol{\beta}_{1,3}^*$ from $\mathbf{M}_2^l$. Then, in line 11 of Algorithm 2, we match the identified singular vector pairs with $(\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_3)$. That is, let $l = \arg\max_{j=3,4}\left\{\max_{k=1,2}\{\langle\mathbf{v}_j, \mathbf{v}_k\rangle\}\right\}$. Set $\tilde{\mathbf{v}}_2 = \mathbf{v}_l$, the remaining one in the pair $(\mathbf{v}_3, \mathbf{v}_4)$ as $\tilde{\mathbf{v}}_3$, and $\tilde{\mathbf{v}}_1 = \text{argmin}_{j=1,2}\{\langle\mathbf{v}_j, \tilde{\mathbf{v}}_2\rangle\}$. Next, given $\hat{\boldsymbol{\beta}}_{1,1}^{(0)}$, $\hat{\boldsymbol{\beta}}_{1,2}^{(0)}$, $\hat{\boldsymbol{\beta}}_{1,3}^{(0)}$, we obtain $\hat{w}_1^{(0)}$, $\hat{\boldsymbol{\beta}}_{1,4}^{(0)}$ by solving the following optimization,

$$\min_{w_1>0, \|\boldsymbol{\beta}_{1,4}\|=1} \frac{1}{n}\sum_{i=1}^n \left\| \Pi_{\Omega_i}\left(\mathcal{Y}_i - w_1(\boldsymbol{\beta}_{1,4}^\top\mathbf{x}_i)\hat{\boldsymbol{\beta}}_{1,1}^{(0)} \circ \hat{\boldsymbol{\beta}}_{1,2}^{(0)} \circ \hat{\boldsymbol{\beta}}_{1,3}^{(0)}\right)\right\|_F^2.$$

Finally, leting $\mathcal{A} = \hat{\boldsymbol{\beta}}_{1,1}^{(0)} \circ \hat{\boldsymbol{\beta}}_{1,2}^{(0)} \circ \hat{\boldsymbol{\beta}}_{1,3}^{(0)}$, we obtain the initial estimates $\hat{\boldsymbol{\beta}}_{1,4}^{(0)}$ and $\hat{w}_1^{(0)}$ as

$$\hat{\boldsymbol{\beta}}_{1,4}^{(0)} = \text{Norm}\left(\left\{\frac{1}{n}\sum_i \left\|\Pi_{\Omega_i}(\mathcal{A})\right\|_F^2 \mathbf{x}_i\mathbf{x}_i^\top\right\}^{-1} \times n^{-1}\sum_i \langle\Pi_{\Omega_i}(\mathcal{Y}_i), \Pi_{\Omega_i}(\mathcal{A})\rangle\mathbf{x}_i\right),$$

$$\hat{w}_1^{(0)} = \frac{\sum_i \hat{\boldsymbol{\beta}}_{1,4}^{(0)\top}\mathbf{x}_i\Pi_{\Omega_i}(\mathcal{Y}_i) \times_1 \hat{\boldsymbol{\beta}}_{1,1}^{(0)} \times_2 \hat{\boldsymbol{\beta}}_{1,2}^{(0)} \times_3 \hat{\boldsymbol{\beta}}_{1,3}^{(0)}}{\sum_i\{\hat{\boldsymbol{\beta}}_{1,4}^{(0)\top}\mathbf{x}_i\}^2 \left\|\Pi_{\Omega_i}(\mathcal{A})\right\|_F^2}. \quad (11)$$

We next present the initialization procedure for $r > 1$ in Algorithm 3. We first apply Algorithm 2 to generate two sets $(\mathbf{v}_1^l, \mathbf{v}_2^l)_{l=1}^L$, $(\mathbf{v}_3^l, \mathbf{v}_4^l)_{l=1}^L$. Since $\hat{\boldsymbol{\beta}}_{k,1}$ and $\hat{\boldsymbol{\beta}}_{k,2}$ are from $(\mathbf{v}_1^l, \mathbf{v}_2^l)$, and $\hat{\boldsymbol{\beta}}_{k,2}, \hat{\boldsymbol{\beta}}_{k,3}$ are from $(\mathbf{v}_3^l, \mathbf{v}_4^l)$, we merge the two and find the triplet $(\tilde{\mathbf{v}}_1^l, \tilde{\mathbf{v}}_2^l, \tilde{\mathbf{v}}_3^l)$. Next, we search for $(\hat{\boldsymbol{\beta}}_{k,1}, \hat{\boldsymbol{\beta}}_{k,2}, \hat{\boldsymbol{\beta}}_{k,3})$ such that $|p^{-1}\mathcal{T} \times_1 \tilde{\mathbf{v}}_1 \times_2 \tilde{\mathbf{v}}_2 \times_3 \tilde{\mathbf{v}}_3|$ is maximized. This is because the selected vectors are expected to be close to true factors when $|p^{-1}\mathcal{T} \times_1 \tilde{\mathbf{v}}_1 \times_2 \tilde{\mathbf{v}}_2 \times_3 \tilde{\mathbf{v}}_3|$ is large (Sun et al. 2017). We also remove all those triplets that are close to $(\hat{\boldsymbol{\beta}}_{k,1}, \hat{\boldsymbol{\beta}}_{k,2}, \hat{\boldsymbol{\beta}}_{k,3})$, since they eventually generate the same decomposition vectors up to the tolerance parameter. We then iteratively refine the selected vectors. In our numerical experiments, we have found that one iteration is often enough, and the algorithm is not sensitive to the tolerance parameter ϵ_{th} because of the used refinement step.

Next, we present a proposition showing that the initial estimator obtained from Algorithm 2 satisfies the initialization Assumption 4 when $r = 1$. The theoretical guarantee for the $r > 1$ case is very challenging, and we leave it for future research.

Algorithm 3 Spectral initialization algorithm for $r > 1$.

1: **input:** the number of restarts L, the estimates $\mathbf{U}_1$, $\mathbf{U}_2$, the tolerance parameter ϵ_{th}, and the sparsity parameter τ_{s_j}, $j \in [3]$.
2: obtain $(\mathbf{v}_1^l, \mathbf{v}_2^l)_{l=1}^L$, $(\mathbf{v}_3^l, \mathbf{v}_4^l)_{l=1}^L$ using Algorithm 2.
3: obtain the triplet $S = \{(\tilde{\mathbf{v}}_1^l, \tilde{\mathbf{v}}_2^l, \tilde{\mathbf{v}}_3^l)\}_{l=1}^L$ from $(\mathbf{v}_1^l, \mathbf{v}_2^l)_{l=1}^L$, $(\mathbf{v}_3^l, \mathbf{v}_4^l)_{l=1}^L$.
4: **for** $k = 1$ to r **do**
5: find $(\hat{\boldsymbol{\beta}}_{k,1}, \hat{\boldsymbol{\beta}}_{k,2}, \hat{\boldsymbol{\beta}}_{k,3}) = \arg\max_{(\tilde{\mathbf{v}}_1^l, \tilde{\mathbf{v}}_2^l, \tilde{\mathbf{v}}_3^l)\in S} |p^{-1}T \times_1 \tilde{\mathbf{v}}_1^l \times_2 \tilde{\mathbf{v}}_2^l \times_3 \tilde{\mathbf{v}}_3^l|$.
6: remove all triplets in $(\tilde{\mathbf{v}}_1^l, \tilde{\mathbf{v}}_2^l, \tilde{\mathbf{v}}_3^l)_{l=1}^L$ with $\max\{|\langle\hat{\boldsymbol{\beta}}_{k,1}, \tilde{\mathbf{v}}_1^l\rangle|, |\langle\hat{\boldsymbol{\beta}}_{k,2}, \tilde{\mathbf{v}}_2^l\rangle|, |\langle\hat{\boldsymbol{\beta}}_{k,3}, \tilde{\mathbf{v}}_3^l\rangle|\} > 1 - \epsilon_{th}$.
7: **end for**
8: set $\hat{w}_k = 1$, and randomly generate unit-norm vectors $\hat{\boldsymbol{\beta}}_{k,4}$, $k \in [r]$ from a standard normal distribution.
9: **repeat**
10: update $\hat{\boldsymbol{\beta}}_{k,1}, \hat{\boldsymbol{\beta}}_{k,2}, \hat{\boldsymbol{\beta}}_{k,3}$ using (7), and set $\hat{\boldsymbol{\beta}}_{k,j} = \texttt{Norm}(\texttt{Truncate}(\hat{\boldsymbol{\beta}}_{k,j}.\tau_{s_j}))$, $j \in [3]$,
11: update $\hat{w}_k$ using (8), and update $\hat{\boldsymbol{\beta}}_{k,4}$ using (9), $k \in [r]$.
12: **until** the stopping criterion is met
13: denote the final update of $\hat{w}_k, \{\hat{\boldsymbol{\beta}}_{k,j}\}_{j=1}^4$ as $\hat{w}_k^{(0)}, \{\hat{\boldsymbol{\beta}}_{k,j}^{(0)}\}_{j=1}^4$, $k \in [r]$, respectively.
14: **output:** $\hat{w}_k^{(0)}$, $\widehat{\boldsymbol{\beta}}_{k,1}^{(0)}$, $\widehat{\boldsymbol{\beta}}_{k,2}^{(0)}$, $\widehat{\boldsymbol{\beta}}_{k,3}^{(0)}$, $\widehat{\boldsymbol{\beta}}_{k,4}^{(0)}$, $k \in [r]$.

Proposition 1. Suppose Assumptions 1, 2, 3, and 5 hold. Furthermore, suppose $L \geq C_1'$ for some large enough C_1', $|\sum_i n^{-1}\boldsymbol{\beta}_{1,4}^{*\top}\mathbf{x}_i| \geq C_2'$ for some constant $C_2' > 0$. Then, the initial estimator produced by Algorithm 2 satisfies that

$$\max\left\{|\widehat{w}_1^{(0)} - w_1^*|/w_1^*, \max_j \|\widehat{\boldsymbol{\beta}}_{1,j}^{(0)} - \boldsymbol{\beta}_{1,j}^*\|_2\right\} = O_p\left\{\sqrt{\frac{\log(d)}{nps^2}} + \frac{\sigma}{w_1^*}\sqrt{\frac{s\log(d)}{np}}\right\}.$$

We make some remarks about Proposition 1. First, this result shows that the error of the initial estimator obtained from Algorithm 2 decays with n, and thus the constant initialization error bound on ϵ in Assumption 4 is guaranteed to hold as n increases. Second, the estimation error in Proposition 1 is slower than the statistical error rate in Theorem 1 when $\sigma/w_1^* \leq c/s^{1.5}$. This suggests that, after obtaining the initial estimator from Algorithm 2, applying the alternating block updating Algorithm 1 could further improve the error rate of the estimator.

Finally, we conduct a simulation to evaluate the empirical performance of the proposed spectral initialization Algorithms 2 and 3. We simulate the coefficient tensor $\mathcal{B}^* \in \mathbb{R}^{30\times20\times10\times5} = \sum_{k=1}^r w_k^*\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^* \circ \boldsymbol{\beta}_{k,4}^*$. We generate the entries of $\boldsymbol{\beta}_{k,j}^*$, $k \in [2], j \in [3]$ from iid standard normal, and set $\boldsymbol{\beta}_{k,4}^*$ as $(1, 1, 1, 1, 1)^\top$. We then normalize each vector to have a unit norm, and set $w_k^* = 20$. We consider two ranks, $r = 1$ and $r = 2$, while we vary the sample size $n = \{20, 40, 60, 80, 100\}$. We then generate the error tensor $\mathcal{E}_i$ with iid standard normal entries, and the response tensor $\mathcal{Y}_i \in \mathbb{R}^{30\times20\times10}$, with each entry missing with probability 0.5. For Algorithms 2 and 3, we

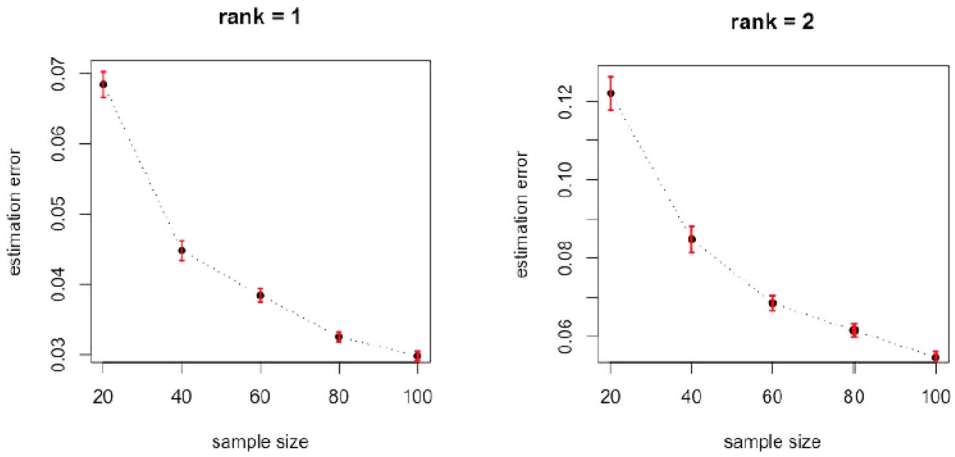


Figure 1. Estimation error of the initial estimator by the spectral initialization algorithms as the sample size increases. The left panel is for $r = 1$, and the right panel is for $r = 2$.

set $L = 30$, $\epsilon_{th} = 0.8$, and τ_{s_j} as d_j. Figure 1 reports the error, $\max_{k,j} \|\widehat{\boldsymbol{\beta}}_{k,j}^{(0)} - \boldsymbol{\beta}_{k,j}^*\|_2$, of the initial estimator based on 100 data replications. It is seen that, as the sample size increases, the estimation error decreases rapidly. This agrees with our finding in Proposition 1, and suggests that the constant initialization error bound in Assumptions 4 and 8 is to hold when n is sufficiently large.

5. Simulations

We carry out simulations to investigate the finite-sample performance of our proposed method. For easy reference, we call our method Partially ObServed dynamic Tensor rEsponse Regression (`POSTER`). We also compare with some alternative solutions. One competing method is the multiscale adaptive generalized estimating equations method (`MAGEE`) proposed by Li et al. (2013), which integrated a voxel-wise approach with generalized estimating equations for adaptive analysis of dynamic tensor imaging data. Another competing method is the sparse tensor response regression method (`STORE`) proposed by Sun and Li (2017), which considered a sparse tensor response regression model but did not incorporate fusion type smoothness constraint and can only handle completely observed data. In our analysis, `STORE` is applied to the complete samples only. Moreover, to examine the effect of using the partially observed samples and incorporating structural smoothness over time, we also consider our method applied to the completely observed samples, or without fusion constraint, which serve as two benchmarks.

We consider two patterns for the unobserved entries, block missing in Section 5.1 and random missing in Section 5.2. Both patterns are common in real data applications. For instance, in our neuroimaging example, individual subjects would miss some scheduled biannual scans, and as a result, the entire tensor images are unobserved, and the missing pattern is more likely a block missing. In our digital advertising example, on the other hand, some users may randomly react to only a subset of advertisements on certain days, and the missing pattern would be closer to a random missing. Finally, in Section 5.3, we consider a model used in Li et al. (2013). The data generation does not comply with our proposed model, and we examine the performance of our method under model misspecification.

To evaluate the estimation accuracy, we report the estimation error of the coefficient tensor $\mathcal{B}^*$ measured by $\|\widehat{\mathcal{B}} - \mathcal{B}^*\|_F$, and the estimation error of the decomposed components $\hat{\boldsymbol{\beta}}_{k,j}$ measured by $\max_{k,j} \min\{\|\hat{\boldsymbol{\beta}}_{k,j} - \boldsymbol{\beta}_{k,j}^*\|, \|\hat{\boldsymbol{\beta}}_{k,j} + \boldsymbol{\beta}_{k,j}^*\|\}$. To evaluate the variable selection accuracy, we compute the true positive rate as the mean of TPR_j, and the false positive rate as the mean of FPR_j, where $\text{TPR}_j = K^{-1}\sum_{k=1}^K \sum_l 1(\boldsymbol{\beta}_{k,j,l}^* \neq 0, \widehat{\boldsymbol{\beta}}_{k,j,l} \neq 0)/\sum_l 1(\boldsymbol{\beta}_{k,j,l}^* \neq 0)$ is the true positive rate of the estimator in mode j, and $\text{FPR}_j = K^{-1}\sum_{k=1}^K \sum_l 1(\boldsymbol{\beta}_{k,j,l}^* = 0, \widehat{\boldsymbol{\beta}}_{k,j,l} \neq 0)/\sum_l 1(\boldsymbol{\beta}_{k,j,l}^* = 0)$ is the false positive rate of the estimator in mode j.

5.1. Block Missing

In the first example, we simulate a fourth-order tensor response $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times T}$, where the fourth mode corresponds to the time dimension, and there are blocks of tensor entries missing along the time mode. More specifically, we generate the coefficient tensor $\mathcal{B}^* \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times T \times q}$ as $\mathcal{B}^* = \sum_{k \in [r]} w_k^* \boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^* \circ \boldsymbol{\beta}_{k,4}^* \circ \boldsymbol{\beta}_{k,5}^*$, where $d_1 = d_2 = d_3 = 32, T = 5, q = 5$, and the true rank $r = 2$. We generate the entries of $\boldsymbol{\beta}_{k,j}^*, j \in [4]$ as iid standard normal. We then apply the `Truncatefuse` operator on $\boldsymbol{\beta}_{k,j}^*, j \in [3]$, with the true sparsity and fusion parameters $(s_j, f_j), j \in [3]$, and apply the `Fuse` operator to $\boldsymbol{\beta}_{k,4}^*$ with the true fusion parameter f_4. We set the true sparsity parameters $s_j = s_0 \times d_j, j \in [3]$ with $s_0 = 0.7$, and set the true fusion parameters $f_j = f_0 \times d_j, j \in [4]$, with $f_0 \in \{0.3, 0.7\}$. A smaller f implies a smaller number of fusion groups in $\boldsymbol{\beta}_{k,j}^*$. We set $\boldsymbol{\beta}_{k,5}^* = (1, \ldots, 1)^\top$, a vector of all ones. We then normalize each vector to have a unit norm. We set the weight $w_k^* \in \{30, 40\}$, with a larger weight indicating a stronger signal. Next, we generate the q-dimensional predictor vector $\mathbf{x}_i$ whose entries are iid Bernoulli with probability 0.5, and the error tensor $\mathcal{E}_i$, whose entries are iid standard normal. Finally, we generate the response tensor $\mathcal{Y}_i$ following model (1). We set the blocks of entries of $\mathcal{Y}_i$ along the fourth mode randomly missing. Among all n subjects, we set the proportion of subjects with missing

Table 1. Simulation example with block missing, for varying missing proportions m_n, m_t, signal strength w_k^*, and fusion setting f_0.

(m_n, m_t)	w_k^*	f_0	method	Error of $\mathcal{B}^*$	Error of $\boldsymbol{\beta}_{k,j}^*$	TPR	FPR
(0.8, 0.4)	30	0.3	STORE	0.586 (0.055)	0.992 (0.109)	0.879 (0.016)	0.369 (0.035)
			MAGEE	1.397 (0.005)	NA	NA	NA
			Complete	0.232 (0.051)	0.366 (0.104)	0.952 (0.017)	0.104 (0.026)
			No-fusion	0.125 (0.003)	0.112 (0.005)	1.000 (0.000)	0.120 (0.000)
			POSTER	0.069 (0.003)	0.068 (0.005)	1.000 (0.000)	0.020 (0.004)
		0.7	STORE	0.574 (0.063)	0.905 (0.113)	0.878 (0.019)	0.343 (0.043)
			MAGEE	1.411 (0.003)	NA	NA	NA
			Complete	0.207 (0.038)	0.259 (0.082)	0.979 (0.008)	0.103 (0.021)
			No-fusion	0.120 (0.003)	0.111 (0.006)	1.000 (0.000)	0.072 (0.000)
			POSTER	0.102 (0.003)	0.098 (0.006)	1.000 (0.000)	0.055 (0.003)
	40	0.3	STORE	0.287 (0.055)	0.402 (0.104)	0.957 (0.013)	0.212 (0.028)
			MAGEE	1.233 (0.002)	NA	NA	NA
			Complete	0.085 (0.022)	0.087 (0.044)	0.995 (0.005)	0.036 (0.011)
			No-fusion	0.115 (0.004)	0.111 (0.005)	1.000 (0.000)	0.120 (0.000)
			POSTER	0.063 (0.004)	0.067 (0.005)	1.000 (0.000)	0.020 (0.004)
		0.7	STORE	0.167 (0.036)	0.160 (0.06)	0.984 (0.009)	0.131 (0.029)
			MAGEE	1.250 (0.002)	NA	NA	NA
			Complete	0.142 (0.030)	0.190 (0.073)	0.984 (0.008)	0.107 (0.026)
			No-fusion	0.107 (0.003)	0.115 (0.005)	1.000 (0.000)	0.093 (0.021)
			POSTER	0.093 (0.004)	0.094 (0.006)	1.000 (0.000)	0.074 (0.019)
(0.8, 0.6)	30	0.3	STORE	0.579 (0.057)	0.975 (0.109)	0.883 (0.016)	0.360 (0.034)
			MAGEE	1.515 (0.004)	NA	NA	NA
			Complete	0.233 (0.051)	0.366 (0.104)	0.952 (0.017)	0.108 (0.026)
			No-fusion	0.155 (0.006)	0.146 (0.008)	1.000 (0.000)	0.120 (0.000)
			POSTER	0.089 (0.006)	0.091 (0.009)	1.000 (0.000)	0.023 (0.005)
		0.7	STORE	0.434 (0.058)	0.729 (0.120)	0.924 (0.015)	0.248 (0.034)
			MAGEE	1.528 (0.004)	NA	NA	NA
			Complete	0.207 (0.038)	0.259 (0.082)	0.979 (0.008)	0.103 (0.021)
			No-fusion	0.151 (0.007)	0.150 (0.009)	1.000 (0.000)	0.072 (0.000)
			POSTER	0.128 (0.008)	0.121 (0.010)	1.000 (0.000)	0.058 (0.002)
	40	0.3	STORE	0.228 (0.045)	0.323 (0.096)	0.971 (0.011)	0.178 (0.021)
			MAGEE	1.310 (0.003)	NA	NA	NA
			Complete	0.090 (0.022)	0.176 (0.073)	0.983 (0.010)	0.054 (0.016)
			No-fusion	0.142 (0.006)	0.142 (0.008)	0.999 (0.001)	0.124 (0.003)
			POSTER	0.082 (0.006)	0.089 (0.009)	1.000 (0.000)	0.023 (0.004)
		0.7	STORE	0.228 (0.047)	0.290 (0.090)	0.969 (0.012)	0.146 (0.029)
			MAGEE	1.325 (0.003)	NA	NA	NA
			Complete	0.137 (0.022)	0.205 (0.076)	0.955 (0.016)	0.159 (0.038)
			No-fusion	0.131 (0.005)	0.141 (0.010)	0.999 (0.001)	0.073 (0.002)
			POSTER	0.110 (0.006)	0.122 (0.016)	0.999(0.001)	0.061 (0.003)

NOTES: Reported are the average estimation errors of $\mathcal{B}^*$ and $\boldsymbol{\beta}_{k,j}^*$, and the true and false positive rates of selection based on 30 data replications (the standard errors in the parentheses). Five methods are compared: STORE of Sun and Li (2017), MAGEE of Li et al. (2013), method applied to the complete data only (Complete), our method without the fusion constraint (No-fusion), and our proposed method (POSTER).

values $m_n \in \{0.8, 0.9\}$, and for each subject with missing values, we set the proportion of missing blocks along the time mode as $m_t \in \{0.4, 0.6\}$. For example, $n = 100$, $m_n = 0.8$ and $m_t = 0.4$ means there are 80 subjects out of 100 having partially observed tensors, and for each of those 80 subjects, the tensor observations at 2 out of 5 time points are missing.

Table 1 reports the average criteria based on 30 data replications with $m_n = 0.8$. The results with $m_n = 0.9$ are similar qualitatively and are reported in the Appendix. Since the method MAGEE of Li et al. (2013) does not decompose the coefficient tensor and does not carry out variable selection, the corresponding criteria of $\boldsymbol{\beta}_{k,j}^*$ and selection are reported as NA. From Table 1, it is clearly seen that our proposed method outperforms all other competing methods in terms of both estimation accuracy and variable selection accuracy.

The computational time of our method scales linearly with the sample size and tensor dimension. Consider the simulation setup with $m_n = 0.8, m_t = 0.4, w_k = 30$, and $f_0 = 0.3$ as an example. When we fix $d_1 = 32$ and other parameters, the average computational time of our method was 112.5, 200.3, and 384.2 seconds for the sample size $n = 100, 200$, and 300, respectively. When we fix $n = 100$ and other parameters, the average computational time of our method was 42.5, 82.3, and 101.8 sec for the tensor dimension $d_1 = 10, 20$, and 30, respectively. The reported computational time does not include tuning. All simulations were run on a personal computer with a 3.2 GHz Intel Core i5 processor.

5.2. Random Missing

In the second example, we simulate data similarly as in Section 5.1, but the entries of the response tensor are randomly missing. We set the observation probability $p \in \{0.3, 0.5\}$. For this setting, MAGEE cannot handle a tensor response with randomly missing entries, whereas STORE or our method applied to the complete data cannot handle either, since there is almost no complete $\mathcal{Y}_i$, with the probability of observing a complete $\mathcal{Y}_i$ being $p^{d_1 d_2 d_3 q}$. Therefore, we can only compare our proposed

Table 2. Simulation example with random missing, for varying observation probability p, signal strength w_k^*, and fusion setting f_0.

p	w_k^*	f_0	method	Error of $\mathcal{B}^*$	Error of $\boldsymbol{\beta}_{k,j}^*$	TPR	FPR
0.5	30	0.3	`No-fusion`	0.091 (0.001)	0.059 (0.001)	1.000 (0.000)	0.121 (0.001)
			`POSTER`	0.055 (0.001)	0.037 (0.001)	1.000(0.000)	0.021 (0.004)
		0.7	`No-fusion`	0.088 (0.001)	0.056 (0.001)	1.000 (0.000)	0.099 (0.026)
			`POSTER`	0.079 (0.002)	0.051 (0.001)	1.000 (0.000)	0.079 (0.024)
	40	0.3	`No-fusion`	0.068 (0.001)	0.044 (0.001)	1.000 (0.000)	0.120 (0.000)
			`POSTER`	0.042 (0.001)	0.029 (0.001)	1.000 (0.000)	0.019 (0.003)
		0.7	`No-fusion`	0.066 (0.001)	0.043 (0.001)	1.000 (0.000)	0.072 (0.000)
			`POSTER`	0.059 (0.001)	0.039 (0.001)	1.000 (0.000)	0.056 (0.003)
0.3	30	0.3	`No-fusion`	0.119 (0.002)	0.078 (0.002)	0.998 (0.001)	0.148 (0.023)
			`POSTER`	0.077 (0.002)	0.054 (0.002)	1.000 (0.000)	0.052 (0.016)
		0.7	`No-fusion`	0.113 (0.002)	0.074 (0.002)	0.998 (0.001)	0.104 (0.026)
			`POSTER`	0.103 (0.002)	0.066 (0.002)	0.998 (0.001)	0.086 (0.024)
	40	0.3	`No-fusion`	0.092 (0.020)	0.060 (0.001)	1.000 (0.000)	0.120 (0.000)
			`POSTER`	0.058 (0.001)	0.042 (0.001)	1.000 (0.000)	0.025 (0.005)
		0.7	`No-fusion`	0.084 (0.001)	0.054 (0.001)	0.999 (0.000)	0.074 (0.001)
			`POSTER`	0.075 (0.001)	0.049 (0.001)	1.000 (0.000)	0.054 (0.030)

NOTES: Reported are the average estimation errors of $\mathcal{B}^*$ and of $\boldsymbol{\beta}_{k,j}^*$, and the true and false positive rates of selection based on 30 data replications (the standard errors in the parentheses). Two methods are compared: our method without the fusion constraint (`No-fusion`), and our proposed method (`POSTER`).

method with the variation that imposes no fusion constraint. Table 2 reports the results based on 30 data replications. It is seen that incorporating the fusion structure clearly improves the estimation accuracy. Moreover, Table 2 shows that the estimation error of our method decreases when the signal strength w_k^* increases or when the observation probability p increases. These patterns agree with our theoretical findings.

5.3. Model Misspecification

In the third example, we simulate data from the model in Li et al. (2013). Data generated this way does not comply with our proposed model (1), and we examine the performance of our method under model misspecification. Following Li et al. (2013), we simulate a third-order tensor response $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times d_2 \times T}$, where the first two modes correspond to imaging space and the third mode corresponds to the time dimension, with $d_1 = d_2 = 88$, $T = 3$, and the sample size $n = 80$. At voxel (j, k) the response of subject i at time point l is simulated according to

$$\mathcal{Y}_{i,j,k,l} = \mathbf{x}_{i,l}^\top \boldsymbol{\beta}_{j,k}^* + \epsilon_{i,j,k,l}, \quad i \in [n],\ l \in [3].$$

The predictor vector $\mathbf{x}_{i,l} = (1, x_{i,l,2}, x_{i,l,3})^\top$, and we consider two settings of generating $\mathbf{x}_{i,l}$. The first setting is that $x_{i,l,2}$ is time-dependent and is generated from a uniform distribution on $[l-1, l]$ for $l = 1, 2, 3$, and $x_{i,l,3}$ is time independent and is generated from a Bernoulli distribution with probability 0.5. The second setting is that both $x_{i,l,2}$ and $x_{i,l,3}$ are time independent and are generated from a Bernoulli distribution with probability 0.5. The error term $\boldsymbol{\epsilon}_{i,j,k} = (\epsilon_{i,j,k,1}, \epsilon_{i,j,k,2}, \epsilon_{i,j,k,3})^\top$ is generated from a multivariate normal $N(0, \Sigma)$, where the diagonal entries of Σ are 1 and $\mathrm{Corr}(\epsilon_{i,j,k,l_1}, \epsilon_{i,j,k,l_2}) = 0.7^{|l_1 - l_2|}$, $l_1, l_2 = 1, 2, 3$. The coefficient $\boldsymbol{\beta}_{j,k}^* = (0, \beta_{j,k,2}^*, \beta_{j,k,3}^*)^\top$, and the coefficient image is divided into six different regions with two different shapes. Following Li et al. (2013), we set $(\beta_{j,k,2}^*, \beta_{j,k,3}^*)$ to $(0, 0)$, $(0.05, 0.9)$, $(0.1, 0.8)$, $(0.2, 0.6)$, $(0.3, 0.4)$ and $(0.4, 0.2)$ in those six regions. Among the 80 subjects, the first half have their 88×88 images observed only at the first two time points.

Figure 2 presents the true and estimated image of $\beta_{j,k,2}^*$, along with the estimation error of the coefficient tensor $\mathcal{B}^*$. The standard error shown in parenthesis is calculated based on 20 replications. The results for $\beta_{j,k,3}^*$ are similar and hence are omitted. It is seen that our method is able to capture all six important regions in both settings of covariates, even if the model is misspecified. When the covariates are time dependent, our method is comparable to Li et al. (2013). When the covariates are time independent, our estimator is more accurate compared to the method of Li et al. (2013).

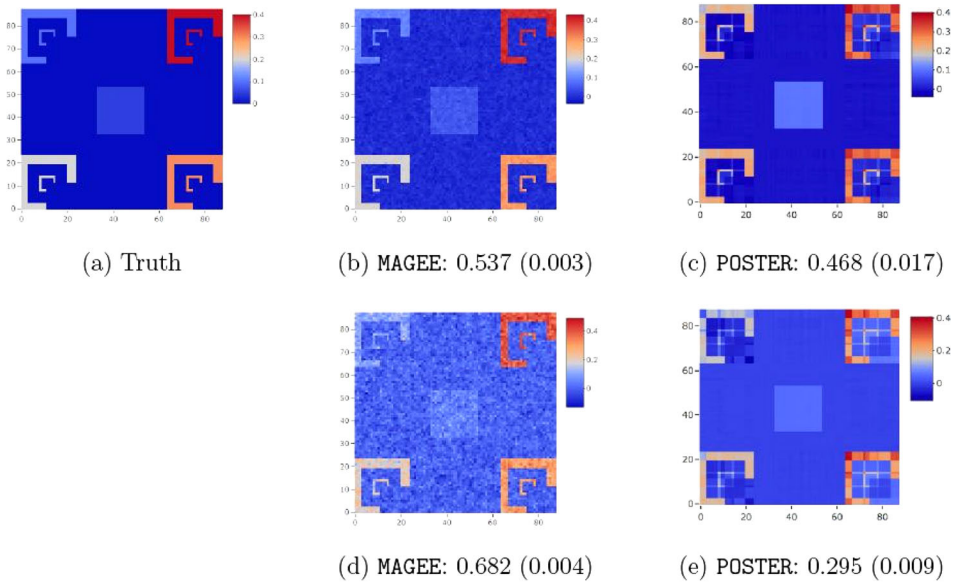
6. Applications

We illustrate the proposed method with two real data applications. The first is a neuroimaging study, where about 50% of subjects have at least one imaging scan missing. The second is a digital advertising study, where about 95% of tensor entries are missing.

6.1. Neuroimaging Application

The first example is a neuroimaging study of dementia. Dementia is a broad category of brain disorders with symptoms associated with decline in memory and daily functioning (Sosa-Ortiz, Acosta-Castillo and Prince 2012). It is of keen scientific interest to understand how brain structures change and differ between dementia patients and healthy controls, which in turn would facilitate early disease diagnosis and development of effective treatment.

The data we analyze are from Alzheimer's disease neuroimaging initiative (ADNI, http://adni.loni.usc.edu), where anatomical MRI images were collected from $n = 365$ participates every six months over a two-year period. Each MRI image, after preprocessing and mapping to a common registration space, is summarized in the form of a $32 \times 32 \times 32$ tensor. For each participant, there are at the most five scans, but many subjects missed some scheduled scans, and 178 subjects out of 365 have at least one scan missing. For each subject, we



(a) Truth (b) MAGEE: 0.537 (0.003) (c) POSTER: 0.468 (0.017)

(d) MAGEE: 0.682 (0.004) (e) POSTER: 0.295 (0.009)

Figure 2. True and estimated image of $\beta^*_{j,k,2}$. The top left panel is the true image of $\beta^*_{j,k,2}$ with six regions. The middle panels are the estimated images by MAGEE, and the right panels by our method POSTER. The top panels correspond to the time dependent covariates, and the bottom panels the time independent covariates. The estimation error (with the standard error in the parenthesis) based on 20 data replications is reported for each image.

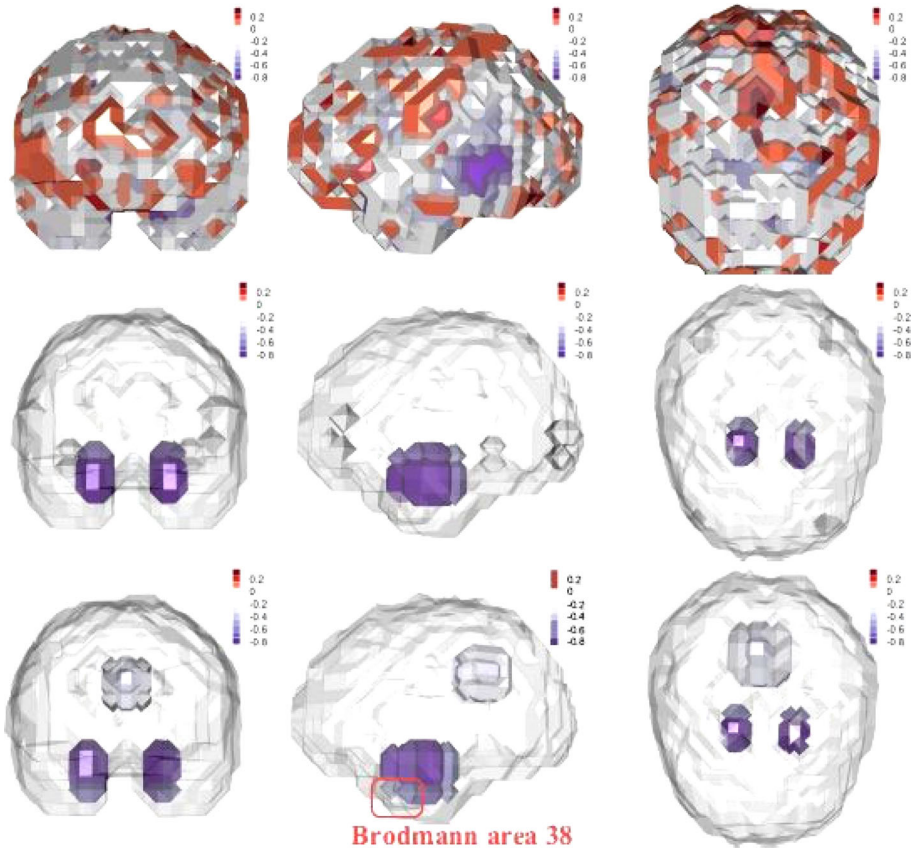


Figure 3. Neuroimaging application example. Shown are the estimated coefficient tensor overlaid on a randomly selected brain image. Top to bottom: MAGEE, STORE, and our method POSTER. Left to right: frontal view, side view, and top view.

stack the MRI brain images collected over time as a fourth-order tensor, which is to serve as the response $\mathcal{Y}_i$. Its dimension is $32 \times 32 \times 32 \times 5$, and there are block missing entries. Among these subjects, 127 have dementia and 238 are healthy controls. In addition, the baseline age and sex of the subjects were collected. As such, the predictor vector $\mathbf{x}_i$ consists of the binary diagnosis status, age and sex. Our goal is to identify brain regions that differ between dementia patients and healthy controls, while controlling for other covariates.

We apply MAGEE, STORE and our POSTER method to this data set. Figure 3 shows the heatmap of the estimated coefficient tensor at the baseline time point obtained by the three

methods. It is seen that the estimate from MAGEE identifies a large number of regions with relatively small signals. Both STORE and POSTER identify several important brain regions, and the parameters in those identified regions are negative, indicating that those regions become less active for patients with dementia. The regions identified by the two methods largely agree with each other, with one exception, that is, Brodmann area 38, which POSTER identifies but STORE does not. The regions identified by both include the hippocampus and the surrounding medial temporal lobe. These findings are consistent with existing neuroscience literature. Hippocampus is found crucial in memory formation, and medial temporal lobe is important for memory storage (Smith and Kosslyn 2007). Hippocampus is commonly recognized as one of the first regions in the brain to suffer damages for patients with dementia (Hampel et al. 2008). There is also clear evidence showing that medial temporal lobe is damaged for dementia patients (Visser et al. 2002). In addition to those two important regions, our method also identifies a small part of the anterior temporal cortex, that is, Brodmann area 38, which is highlighted in Figure 3. This area is involved in language processing, emotion and memory, and is also among the first areas affected by AD, which is the most common type of dementia (Delacourte et al. 1998).

6.2. Digital Advertising Application

The second example is a digital advertising study of CTR for some online advertising campaign. CTR is the number of times a user clicks on a specific advertisement divided by the number of times the advertisement is displayed. It is a crucial measure to evaluate the effectiveness of an advertisement campaign, and plays an important role in digital advertising pricing (Richardson, Dominowska and Ragno 2007).

The data we analyzed are obtained from a major internet company over four weeks in May to June 2016. The CTR of 80 advertisement campaigns were recorded for 20 users by 2 different publishers. Since it is of more interest to understand the user behavior over different days of a week, the data were averaged by days of a week across the four-week period. For each campaign, we stack the CTR data of different users and publishers over seven days of the week as a third-order tensor, which serves as the response $\mathcal{Y}_i$. Its dimension is $20 \times 2 \times 7$, and there are 95% entries missing. Such a missing percentage, however, is not uncommon in online advertising, since a user usually does not see every campaign in every publisher every day. For each campaign, we also observe two covariates. One covariate is the topic of the advertisement campaign, which

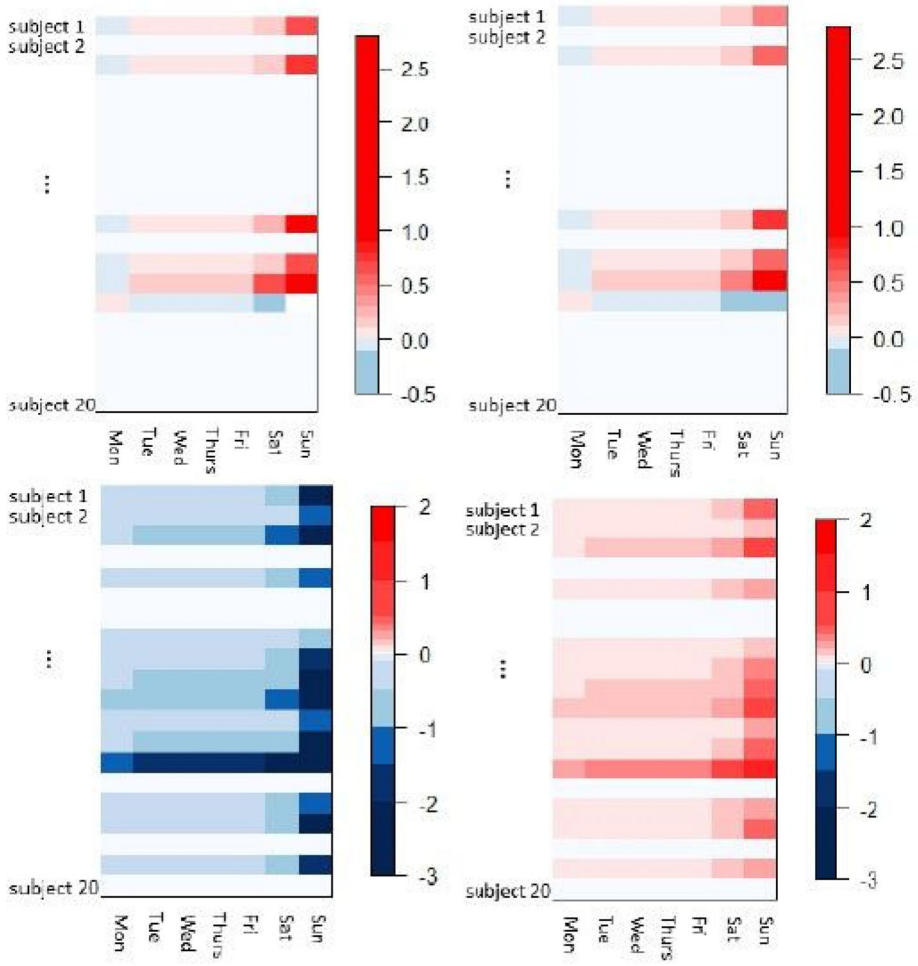


Figure 4. Digital advertising application example. Shown are the estimated coefficient tensor. In each panel, the rows represent users and columns represent days of a week. The top panels are for the topic "online dating," and the bottom panels for "investment." The left panels are slices from the topic mode, and the right panels are slices from the impression mode.

takes three categorical values, "online dating," "investment," or "others." The other covariate is the total number of impressions of the advertisement. The predictor vector $\mathbf{x}_i$ consists of these two covariates. Our goal is to study how the topic and total impression of an advertisement influence its effectiveness measured by CTR.

Due to the large proportion of missing values and nearly random missing patterns, neither `MAGEE` nor `STORE` are applicable to this dataset. We applied our method. For the categorical covariate, topic, we created two dummy variables, one indicating whether the topic was "online dating" or not, and the other indicating whether the topic was "investment" or not. Figure 4 shows the heatmap of the estimated coefficient tensor for one publisher, whereas the result for the other publisher is similar and is thus omitted. The rows of the heatmap represent the users and the columns represent the days of a week. We first consider the topic of "online dating." The top left panel shows that, for this topic, the CTR is higher than other topics during the weekend. The top right panel shows that, if the total impression on "online dating" increases, then the CTR increases more on weekends than weekdays. It is also interesting to see that the topic of "online dating" has a negative impact on the CTR on Mondays. We next consider the topic of "investment." The bottom left panel shows that, for this topic, the CTR is lower than other topics for most users during the weekend. The bottom right panel shows that, if the total impression increases, the CTR increases more on weekends than weekdays. These findings are useful for managerial decisions. Based on the findings about "online dating," one should increase the allocation of "online dating"-related advertisements on weekends, and decrease the allocation on Mondays. On the other hand, the allocation recommendation for "investment"-related advertisements are different. For most users, one should allocate more such advertisements during the early days of a week, and fewer during weekends. For a small group of users who seem to behave differently from the majority, some personalized recommendations regarding "investment" advertisements can also be beneficial.

Supplementary Material

The supplementary materials collect all technical proofs and additional numerical results.

Acknowledgments

The authors thank to the editor Professor Ian McKeague, the associate editor and two anonymous reviewers for their valuable comments and suggestions which led to a much improved article. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research, the National Science Foundation, or the National Institutes of Health.

Funding

Will Wei Sun's research was partially supported by ONR grant N00014-18-1-2759. Jingfei Zhang's research was partially supported by NSF grant DMS-2015190. Lexin Li's research was partially supported by NIH grants R01AG061303, R01AG062542, and R01AG034570.

ORCID

Will Wei Sun http://orcid.org/0000-0002-8412-6430

References

Anandkumar, A., Ge, R., Hsu, D., and Kakade, S. M. (2014), "A Tensor Approach to Learning Mixed Membership Community Models," *Journal of Machine Learning Research*, 15, 2239–2312. [426,431]

Bi, X., Qu, A., and Shen, X. (2018), "Multilayer Tensor Factorization With Applications to Recommender Systems," *Annals of Statistics*, 46, 3308–3333. [425]

Bruce, N. I., Murthi, B., and Rao, R. C. (2017), "A Dynamic Model for Digital Advertising: The Effects of Creative Format, Message Content, and Targeting on Engagement," *Journal of Marketing Research*, 54, 202–218. [424]

Bullmore, E., and Sporns, O. (2009), "Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems," *Nature Reviews Neuroscience*, 10, 186–198. [426]

Cai, C., Li, G., Poor, H., and Chen, Y. (2019), "Nonconvex Low-Rank Tensor Completion From Noisy Data," *NeurIPS*, 32, 1863–1874. [429,430,431,432]

Cai, T. T., Li, X., and Ma, Z. (2016), "Optimal Rates of Convergence for Noisy Sparse Phase Retrieval Via Thresholded Wirtinger Flow," *The Annals of Statistics*, 44, 2221–2251. [428]

Chen, H., Raskutti, G., and Yuan, M. (2019), "Non-Convex Projected Gradient Descent for Generalized Low-Rank Tensor Regression," *Journal of Machine Learning Research*, 20, 1–37. [425,426,429]

Delacourte, A., David, J. P., Sergeant, N., Buée, L., Wattez, A., Vermersch, P., Ghozali, F., Fallet-Bianco, C., Pasquier, F., Lebert, F., Petit, H., Di Menza, C. (1998), "The Biochemical Pathway of Neurofibrillary Degeneration in Aging and Alzheimer's Disease," *American Academy of Neurology*, 52, 1158–1165. [437]

Feng, X., Li, T., Song, X., and Zhu, H. (2019), "Bayesian Scalar on Image Regression With Non-Ignorable Non-Response," *Journal of the American Statistical Association*, 115, 1574-1597. [425]

Hampel, H., Burger, K., Teipel, S. J., Bokde, A. L., Zetterberg, H., Blennow, K. (2008), "Core Candidate Neurochemical and Imaging Biomarkers of Alzheimer's Disease," *Alzheimer's and Dementia*, 4, 38–48. [437]

Han, R., Willett, R., and Zhang, A. (2020), "An Optimal Statistical and Computational Framework for Generalized Tensor Estimation," arXiv:2002.11255. [424,430]

Hao, B., Zhang, A., and Cheng, G. (2020), "Sparse and Low-Rank Tensor Estimation Via Cubic Sketchings," *IEEE Transactions on Information Theory*, 66, 5927–5964. [424,431]

Jain, P., and Oh, S. (2014), "Provable Tensor Factorization With Missing data," *Advances in Neural Information Processing Systems*, 2, 1431–1439. [424,425,426,427,429,430]

Kolda, T. G., and Bader, B. W. (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500. [426]

Li, L., and Zhang, X. (2017), "Parsimonious Tensor Response Regression," *Journal of the American Statistical Association*, 112, 1131–1146. [425]

Li, Y., Gilmore, J. H., Shen, D., Styner, M., Lin, W., and Zhu, H. (2013), "Multiscale Adaptive Generalized Estimating Equations for Longitudinal Neuroimaging Data," *NeuroImage*, 72, 91-105. [425,433,434,435]

Ma, Z. (2013), "Sparse Principal Component Analysis and Iterative Thresholding," *Annals of Statistics*, 41, 772–801. [428]

Madrid-Padilla, O., and Scott, J. (2017), "Tensor Decomposition With Generalized Lasso Penalties," *Journal of Computational and Graphical Statistics*, 26, 537–546. [426]

Rabusseau, G., and Kadri, H. (2016), "Low-Rank Regression With Tensor Responses," in *Advances in Neural Information Processing Systems*, (Vol. 29), eds. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett, Barcelona, Spain: Curran Associates, Inc. [425]

Richardson, M., Dominowska, E., and Ragno, R. (2007), "Predicting Clicks: Estimating the Click-Through Rate for New Ads," in *Proceedings of the 16th International Conference on World Wide Web*. Banff, Alberta, Canada: ACM Press. [437]

Rinaldo, A. (2009), "Properties and Refinements of the Fused Lasso," *The Annals of Statistics*, 37, 2922–2952. [426,429]

Ryota, T., and Taiji, S. (2014), "Spectral Norm of Random Tensors," arXiv:1407.1870. [430]

Shen, X., Pan, W., and Zhu, Y. (2012), "Likelihood-Based Selection and Sharp Parameter Estimation," *Journal of American Statistical Association*, 107, 223–232. [426]

Smith, E. E., and Kosslyn, S. M. (2007), *Cognitive Psychology: Mind and Brian*, Upper Saddle River, NJ: Prentice-Hall, 279–306. [437]

Sosa-Ortiz, A. L., Acosta-Castillo, I., and Prince, M. J. (2012), "Epidemiology of Dementias and Alzheimer's Disease," *Archives of Medical Research*, 43, 600–608. [435]

Sun, W., Lu, J., Liu, H., and Cheng, G. (2017), "Provable Sparse Tensor Decomposition," *Journal of the Royal Statistical Society*, Series B, 79, 899–916. [426,427,428,429,431,432]

Sun, W. W., and Li, L. (2017), "Store: Sparse Tensor Response Regression and Neuroimaging Analysis," *Journal of Machine Learning Research*, 18, 1–37. [425,427,428,429,430,433,434]

——— (2019), "Dynamic Tensor Clustering," *Journal of American Statistical Association*, 114, 1894 – 1907. [426,428]

Tan, K. M., Wang, Z., Liu, H., and Zhang, T. (2018), "Sparse Generalized Eigenvalue Problem: Optimalstatistical Rates Via Truncated Rayleigh Flow," *Journal of the Royal Statistical Society*, Series B, 80, 1057–1086. [428,429]

Tang, X., Bi, X., and Qu, A. (2019), "Individualized Multilayer Tensor Learning With an Application in Imaging Analysis," *Journal of the American Statistical Association*, 115, 836–851. [425]

Thung, K.-H., Wee, C.-Y., Yap, P.-T., and Shen, D. (2016), "Identification of Progressive Mild Cognitive Impairment Patients Using Incomplete Longitudinal MRI Scans," *Brain Structure and Function*, 221, 3979–3995. [424]

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and Smoothness Via the Fused Lasso," *Journal of the Royal Statistical Society*, Series B, 67, 91–108. [429]

Visser, P., Verhey, F. R. J., Hofman, P. A. M., Scheltens, P., Jolles, J. (2002), "Medial Temporal Lobe Atrophy Predicts Alzheimer's Disease in Patients With Minor Cognitive Impairment," *Journal of Neurology, Neurosurgery and Psychiatry*, 72, 491–497. [437]

Vounou, M., Nichols, T. E., Montana, G., Initiative, A. D. N. (2010), "Discovering Genetic Associations With High-Dimensional Neuroimaging Phenotypes: A Sparse Reduced-Rank Regression Approach," *Neuroimage*, 53, 1147–1159. [425,426]

Wang, M., and Li, L. (2020), "Learning From Binary Multiway Data: Probabilistic Tensor Decomposition and its Statistical Optimality," *Journal of Machine Learning Research*, 21, 1–38. [428]

Wang, X., and Zhu, H. (2017), "Generalized Scalar-on-Image Regression Models Via Total Variation," *Journal of the American Statistical Association*, 112, 1156–1168. [424]

Wang, Y., Sharpnack, J., Smola, A., and Tibshirani, R. (2016), "Trend Filtering on Graphs," *Journal of Machine Learning Research*, 17, 1–41. [426]

Wang, Y., Tung, H.-Y., Smola, A., and Anandkumar, A. (2015a), "Fast and Guaranteed Tensor Decomposition Via Sketching," in *Advances in Neural Information Processing Systems* (Vol. 1), eds. C. Cortes, D. D. Lee, M. Sugiyama and R. Garnett, Cambridge, MA: MIT Press. pp. 991–999. [428]

Wang, Z., Gu, Q., Ning, Y., and Liu, H. (2015b), "High Dimensional EM Algorithm: Statistical Optimization and Asymptotic Normality," *NeurIPS*, 28, 2512–2520. [428,429]

Xia, D., and Yuan, M. (2017), "On Polynomial Time Methods for Exact Low Rank Tensor Completion," *Foundations of Computational Mathematics*, 19, 1–49. [424,425,429]

Xia, D., Yuan, M., and Zhang, C. (2020), "Statistically Optimal and Computationally Efficient Low Rank Tensor Completion From Noisy Entries," *Annals of Statistics*, 49(1), 76–99. [429,430]

Xu, Z., Hu, J., and Wang, M. (2019), "Generalized Tensor Regression With Covariates on Multiple Modes," arXiv:1910.09499. [425]

Xue, F., and Qu, A. (2020), "Integrating Multisource Block-Wise Missing Data in Model Selection," *Journal of the American Statistical Association*, 1–14. [425]

Yin, H., Cui, B., Chen, L., Hu, Z., and Zhou, X. (2015), "Dynamic User Modeling in Social Media Systems," *ACM Transactions on Information Systems*, 33, 1–44. [425]

Yuan, M., and Zhang, C. (2016), "On Tensor Completion Via Nuclear Norm Minimization," *Foundations of Computational Mathematics*, 16, 1031–1068. [424,425,426,429]

——— (2017), "Incoherent Tensor Norms and their Applications in Higher Order Tensor Completion," *IEEE Transactions on Information Theory*, 63, 6753–6766. [424,425,426,429]

Yuan, X.-T., and Zhang, T. (2013). "Truncated Power Method for Sparse Eigenvalue Problems," *Journal of Machine Learning Research*, 14, 899–925. [427,429]

Zhang, A. (2019), "Cross: Efficient Low-Rank Tensor Completion," *Annals of Statistics*, 47, 936–964. [424,426]

Zhang, Z., Allen, G. I., Zhu, H., and Dunson, D. (2019), "Tensor Network Factorizations: Relationships Between Brain Structural Connectomes and Traits," *NeuroImage*, 197, 330–343. [425]

Zhou, H., Li, L., and Zhu, H. (2013), "Tensor Regression With Applications in Neuroimaging Data Analysis," *Journal of the American Statistical Association*, 108, 540–552. [424,425,426,428,429]

Zhu, Y., Shen, X., and Pan, W. (2014), "Structural Pursuit Over Multiple Undirected Graphs," *Journal of the American Statistical Association*, 109, 1683–1696. [426]