Technometrics

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: https://www.tandfonline.com/loi/utch20

Tensor Envelope Partial Least-Squares Regression

Xin Zhang & Lexin Li

To cite this article: Xin Zhang & Lexin Li (2017) Tensor Envelope Partial Least-Squares Regression, Technometrics, 59:4, 426-436, DOI: 10.1080/00401706.2016.1272495

To link to this article: https://doi.org/10.1080/00401706.2016.1272495

TECHNOMETRICS
2017, VOL. 59, NO. 4, 426–436
http://dx.doi.org/10.1080/00401706.2016.1272495

Taylor & Francis
Taylor & Francis Group

Tensor Envelope Partial Least-Squares Regression

Xin Zhang[a] and Lexin Li[b]

[a]Department of Statistics, Florida State University, Tallahassee, FL; [b]Division of Biostatistics, University of California, Berkeley, CA

ABSTRACT

Partial least squares (PLS) is a prominent solution for dimension reduction and high-dimensional regressions. Recent prevalence of multidimensional tensor data has led to several tensor versions of the PLS algorithms. However, none offers a population model and interpretation, and statistical properties of the associated parameters remain intractable. In this article, we first propose a new tensor partial least-squares algorithm, then establish the corresponding population interpretation. This population investigation allows us to gain new insight on how the PLS achieves effective dimension reduction, to build connection with the notion of sufficient dimension reduction, and to obtain the asymptotic consistency of the PLS estimator. We compare our method, both analytically and numerically, with some alternative solutions. We also illustrate the efficacy of the new method on simulations and two neuroimaging data analyses. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received December 2015
Revised November 2016

KEYWORDS
Dimension reduction;
Multidimensional array;
Neuroimaging analysis;
Partial least squares;
Reduced rank regression;
Sparsity principle

1. Introduction

Data taking the form of multidimensional array, a.k.a. *tensor*, are becoming ubiquitous in numerous scientific and engineering applications. For instance, in the neuroimaging field, a variety of imaging modalities produce multidimensional tensors. Examples include electroencephalography (EEG, 2-way tensor, i.e., matrix), anatomical magnetic resonance image (MRI, 3-way tensor), functional magnetic resonance images (fMRI, 4-way tensor), among others. Such multidimensional data are, inherently, of both extremely high dimensionality and complex structure. For instance, a $30 \times 36 \times 30$ MRI image turns into a 32,400-dimensional vector, whereas individual image voxels exhibit strong spatial correlations. Analyzing such data, given usually a limited sample size, poses new challenges to statistical modeling.

Partial least squares (PLS) has been a prominent solution for dimension reduction and high-dimensional regressions. Originating and prospering in the field of chemometrics, early developments of PLS concentrated on effective algorithms and empirical predictions, and tended not to address the population PLS models. See Martens and Næs (1989) for a review of classical PLS in the chemometrics community. Helland (1988) defined the first population model for PLS with a scalar response and a vector of predictors. Since then a number of works have developed statistical views and models for PLS (Helland 1992; Frank and Friedman 1993; Næs and Helland 1993; Helland 2001; Naik and Tsai 2005; Li, Cook, and Tsai 2007; Cook, Helland, and Su 2013). More recently, inspired by the prevalence of tensor data, there have been proposals of PLS for a scalar/vector response, and a tensor predictor (Bro 1996; Eliseyev and Aksenova 2013; Zhao et al. 2013). However, none of the existing tensor PLS algorithms offers any insight on a population model, and statistical properties of the associated parameters remain intractable.

In this article, we first propose a new *tensor partial least-squares* algorithm. We then establish the corresponding population model, which is closely associated with the nascent *envelope* concept proposed by Cook, Li, and Chiaromonte (2010). Establishing a population interpretation for the tensor PLS algorithm leads to several appealing outcomes. First, it offers new insights on how PLS achieves effective dimension reduction. We find that PLS uses, implicitly, a generalized sparsity principle, in that only part of the predictors information associates with the response and the rest is irrelevant. Second, it shows that PLS also connects with the notion of sufficient dimension reduction (Cook 1998; Li and Wang 2007), where no regression information would be lost after reduction. Third, it allows one to establish statistical properties of the PLS model parameters. Specifically, we obtain the $\sqrt{n}$-consistency of the PLS estimator, which to our knowledge is the first such result for tensor PLS.

Our proposal is related to but also distinct from several lines of research on dimension reduction and regression with tensor variables. First, the key of our proposal is dimension reduction, and in that regard, it relates to other dimension reduction solutions, notably principal components analysis (PCA) and independent components analysis (ICA). See Calhoun, Liu, and Adalı (2009), Guo, Ahn, and Zhu (2015), and Ahn et al. (2015) for some review and recent applications of PCA and ICA in neuroimaging analysis. Whereas PCA and ICA are both unsupervised dimension reduction solutions, our proposed PLS is a supervised one, and it achieves dimension reduction and prediction of response variables simultaneously. There have also

CONTACT Lexin Li *lexinli@berkeley.edu*
Supplementary materials for this article are available online. Please go to *http://www.tandfonline.com/r/TECH*.

been some recent development in supervised sufficient dimension reduction for tensor variables (Zhong, Xing, and Suslick 2015; Ding and Cook 2015). However, their focus is on reducing the tensor into a few composite variables, whereas our PLS solution not only constructs the reduced dimensional latent variables, but also provides a direct estimate of the original coefficients of interest and a prediction of responses. Second, our method shares a similar aim as a number of recent proposals on regressions that associate an image with clinical variables (Reiss and Ogden 2010; Goldsmith, Huang, and Crainiceanu 2014; Wang et al. 2014; Zhou and Li 2014; Zhu, Fan, and Kong 2014; Zhao and Leng 2014; Sun et al. 2015, among others). In particular, our tensor PLS model is closely related to tensor predictor regression of Zhou, Li, and Zhu (2013), and we will compare the two solutions, both analytically and numerically, in this article. Finally, our work is built upon the nascent envelope concept of Cook, Li, and Chiaromonte (2010), and extends Cook, Helland, and Su (2013) from regression with a vector predictor to a tensor predictor. Such an extension, however, is far from trivial. We are essentially aiming at a totally different regression problem, that is, tensor predictor regression, than the classical multivariate linear regression as in Cook, Helland, and Su (2013). Again, we will carefully compare the two methods in Section 4.

Throughout the article, we employ the following tensor notations and operations, following Kolda and Bader (2009). A multidimensional array $\boldsymbol{A} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ is called an *m-way tensor*. A *fiber* is the higher order analog of matrix row and column, and is defined by fixing every index of the tensor but one. Matrix column is a mode-1 fiber, and row is a mode-2 fiber. The $\text{vec}(\boldsymbol{A})$ *operator* stacks the entries of a tensor into a column vector, so that the entry $a_{i_1 \ldots i_m}$ of $\boldsymbol{A}$ maps to the jth entry of $\text{vec}(\boldsymbol{A})$, in which $j = 1 + \sum_{k=1}^{m} (i_k - 1) \prod_{k'=1}^{k-1} p_{k'}$. The *mode-k matricization*, $\boldsymbol{A}_{(k)}$, maps a tensor $\boldsymbol{A}$ into a matrix, denoted by $\boldsymbol{A}_{(k)} \in \mathbb{R}^{p_k \times (\prod_{j \neq k} p_j)}$, so that the $(i_1, \ldots, i_m)$ element of $\boldsymbol{A}$ maps to the (i_k, j) element of the matrix $\boldsymbol{A}_{(k)}$, where $j = 1 + \sum_{k' \neq k} (i_{k'} - 1) \prod_{k'' < k', k'' \neq k} p_{k''}$. The *k-mode product* $\boldsymbol{A} \times_k \boldsymbol{D}$ of a tensor $\boldsymbol{A}$ and a matrix $\boldsymbol{D} \in \mathbb{R}^{s \times p_k}$ results in an m-way tensor in $\mathbb{R}^{p_1 \times \cdots \times p_{k-1} \times s \times p_{k+1} \times \cdots \times p_m}$, where each element is the product of mode-k fiber of $\boldsymbol{A}$ multiplied by $\boldsymbol{D}$. When $\boldsymbol{D}$ reduces to a $1 \times p_k$ row vector, $\boldsymbol{A} \times_k \boldsymbol{D}$ reduces to an $(m-1)$-way tensor in $\mathbb{R}^{p_1 \times \cdots \times p_{k-1} \times p_{k+1} \times \cdots \times p_m}$. The *Tucker decomposition* of tensor $\boldsymbol{A}$, often represented by a shorthand, $[\![\boldsymbol{D}; \boldsymbol{G}_1, \ldots, \boldsymbol{G}_m]\!]$, is defined as $\boldsymbol{A} = \boldsymbol{D} \times_1 \boldsymbol{G}_1 \times_2 \cdots \times_m \boldsymbol{G}_m$, where $\boldsymbol{D} \in \mathbb{R}^{\tilde{p}_1 \times \cdots \times \tilde{p}_m}$ is the core tensor, and $\boldsymbol{G}_k \in \mathbb{R}^{p_k \times \tilde{p}_k}$, $k = 1, \ldots, m$, are the factor matrices. $\|\cdot\|$ denotes the Euclidean norm, and $\|\cdot\|_F$ denotes the Frobenius norm.

The rest of the article is organized as follows. Section 2 reviews some existing PLS algorithms, then presents our new tensor PLS algorithm. Section 3 establishes the population interpretation for the proposed algorithm, along with its asymptotic properties, through a newly developed concept, tensor envelope. Section 4 compares our tensor PLS solution with some alternative solutions (Cook, Helland, and Su 2013; Zhou, Li, and Zhu 2013). Section 5 presents simulations and Section 6 the real-data analysis. Section 7 concludes the article with a discussion. All technical proofs are relegated to the supplement.

2. Tensor PLS Algorithm

2.1 Vector PLS Algorithms

We begin with a quick review of two dominant PLS algorithms for a vector predictor in the literature, the nonlinear iterative partial least squares, or simply, the *Wold algorithm* (Wold 1966), and the statistically inspired modification of PLS, or the *SIMPLS algorithm* (de Jong 1993). We present their population version in Algorithms 1 and 2, respectively. We have formulated the algorithm exposition in a way that follows the traditional description of those algorithms, and at the same time makes their comparison straightforward. We denote the r-dimensional response vector as $\boldsymbol{Y}$, and the p-dimensional predictor vector as $\boldsymbol{X}$. Without loss of generality, we assume the predictor $\boldsymbol{X}$ and the response $\boldsymbol{Y}$ are centered. Their sample observations will be denoted as $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i = 1, \ldots, n$. The sample version of the PLS algorithms can be obtained by plugging in the corresponding sample estimators.

Algorithm 1 The Wold algorithm

[0] Initialize $\boldsymbol{E}_0 = \boldsymbol{X} \in \mathbb{R}^p$.
for $s = 1, \ldots, d$ **do**
[1] Find $\boldsymbol{w}_s \in \mathbb{R}^p$ that maximizes $\|\text{cov}(\boldsymbol{Y}, \boldsymbol{w}^\mathsf{T} \boldsymbol{E}_{s-1})\|$, subject to $\boldsymbol{w}^\mathsf{T} \boldsymbol{w} = 1$
[2] Define a random variable $t_s = \boldsymbol{w}_s^\mathsf{T} \boldsymbol{X} \in \mathbb{R}$
[3] Compute a vector $\boldsymbol{v}_s = \text{cov}(\boldsymbol{X}, t_s)/\text{var}(t_s) \in \mathbb{R}^p$
[4] Deflate the predictor vector $\boldsymbol{E}_s = \boldsymbol{E}_{s-1} - t_s \boldsymbol{v}_s$
end for
[5] Reduce $\boldsymbol{X} \in \mathbb{R}^p$ to $\boldsymbol{T} = \boldsymbol{w}^\mathsf{T} \boldsymbol{X} = (t_1, \ldots, t_d)^\mathsf{T} \in \mathbb{R}^d$, where $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d)$
[6] Regress $\boldsymbol{Y} \in \mathbb{R}^r$ on $\boldsymbol{T} \in \mathbb{R}^d$

Algorithm 2 The SIMPLS algorithm

[0] Initialize $\boldsymbol{C}_0 = \text{cov}(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}^{p \times r}$
for $s = 1, \ldots, d$ **do**
[1] Find $\boldsymbol{w}_s \in \mathbb{R}^p$ that maximizes $\boldsymbol{w}^\mathsf{T} \boldsymbol{C}_{s-1} \boldsymbol{C}_{s-1}^\mathsf{T} \boldsymbol{w}$, subject to $\boldsymbol{w}^\mathsf{T} \boldsymbol{w} = 1$
[2] Define a random variable $t_s = \boldsymbol{w}_s^\mathsf{T} \boldsymbol{X} \in \mathbb{R}$
[3] Compute a vector $\boldsymbol{v}_s = \text{cov}(\boldsymbol{X}, t_s)/\text{var}(t_s) \in \mathbb{R}^p$
[4] Deflate the cross covariance $\boldsymbol{C}_s = \boldsymbol{Q}_s \boldsymbol{C}_0$, where $\boldsymbol{Q}_s \in \mathbb{R}^{p \times p}$ is the projection matrix onto the orthogonal subspace of $\text{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s)$
end for
[5] Reduce $\boldsymbol{X} \in \mathbb{R}^p$ to $\boldsymbol{T} = \boldsymbol{w}^\mathsf{T} \boldsymbol{X} = (t_1, \ldots, t_d)^\mathsf{T} \in \mathbb{R}^d$, where $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d)$
[6] Regress $\boldsymbol{Y} \in \mathbb{R}^r$ on $\boldsymbol{T} \in \mathbb{R}^d$

From the algorithm exposition, we see that, PLS essentially aims to reduce the predictor $\boldsymbol{X} \in \mathbb{R}^p$ to a lower dimensional vector $\boldsymbol{T} = \boldsymbol{W}^\mathsf{T} \boldsymbol{X} \in \mathbb{R}^d$. Since $d \leq p$, and often $d \ll p$, substantial dimension reduction is achieved. It corresponds to a latent factor model, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{T} + \boldsymbol{E}$, and $\boldsymbol{Y} = \boldsymbol{V}\boldsymbol{T} + \boldsymbol{F}$, where $\boldsymbol{T} \in \mathbb{R}^d$ denotes the score vector, $\boldsymbol{U} \in \mathbb{R}^{p \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times d}$ are the loading matrices, and $\boldsymbol{E}$ and $\boldsymbol{F}$ are the error terms. The core of PLS is to find the latent vector $\boldsymbol{T}$ in the form of linear combinations $\boldsymbol{T} = \boldsymbol{W}^\mathsf{T} \boldsymbol{X}$ of the predictor vector $\boldsymbol{X}$, then regress $\boldsymbol{Y}$ on $\boldsymbol{T}$. The algorithm

seeks the column $\boldsymbol{w}_s$ of the factor coefficient matrix $\boldsymbol{W}$ one at a time.

It is interesting to compare the two PLS algorithms. We see that both share the same iterative procedure of estimating $\boldsymbol{W}$. However, the two differ in terms of the objective function for $\boldsymbol{w}_s$ and the object for "deflation." Specifically, at each iterative step, the Wold algorithm finds $\boldsymbol{w}_s$ that maximizes a function of the covariance between the response vector and the deflated predictor vector $\boldsymbol{E}_{s-1}$. It then updates the deflated vector $\boldsymbol{E}_s$ by subtracting the linear term of regressing $\boldsymbol{X}$ on $t_s = \boldsymbol{w}_s^\top \boldsymbol{X}$ from the current deflated vector $\boldsymbol{E}_{s-1}$. Note that the deflated variable $\boldsymbol{E}_s$ is a nonlinear function of $\boldsymbol{X}$. By contrast, the SIMPLS algorithm finds $\boldsymbol{w}_s$ that maximizes a function of the deflated cross-covariance term $\boldsymbol{C}_{s-1}$, which equals the cross-covariance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ initially, and is deflated as $\boldsymbol{C}_s = \boldsymbol{Q}_s \boldsymbol{C}_0$ subsequently. The deflation is through a linear projection $\boldsymbol{Q}_s$ in the original scale: $\boldsymbol{Q}_s \boldsymbol{C}_0 = \boldsymbol{Q}_s \mathrm{cov}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{cov}(\boldsymbol{Q}_s \boldsymbol{X}, \boldsymbol{Y})$. Such differences lead to different interpretation of the estimators resulting from the two algorithms. There is a population interpretation for the SIMPLS estimator (Helland 1988; Cook, Helland, and Su 2013), but no such interpretation for the Wold estimator in general.

It is also noteworthy that, in Algorithm 2,

$$\begin{aligned} \boldsymbol{v}_s &= \mathrm{cov}(\boldsymbol{X}, t_s)/\mathrm{var}(t_s) \\ &= \mathrm{cov}(\boldsymbol{X}, \boldsymbol{w}_s^\top \boldsymbol{X})/\mathrm{var}(t_s) = \boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{w}_s / \mathrm{var}(t_s), \end{aligned}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \mathrm{cov}(\boldsymbol{X})$ is the covariance of $\boldsymbol{X}$. The projection matrix $\boldsymbol{Q}_s$ for deflation can thus be viewed as the projection matrix onto the orthogonal subspace of $\mathrm{span}(\boldsymbol{\Sigma}_{\boldsymbol{X}} \boldsymbol{W}_s) = \mathrm{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s)$. This equivalent formulation of $\boldsymbol{Q}_s$ is shown to give exactly the same solution as the SIMPLS algorithm (Cook, Helland, and Su 2013). It will later benefit the extension of PLS from a vector predictor to a tensor and its population interpretation.

2.2 Existing Tensor PLS Algorithms

Motivated by rapid increase of applications involving tensor data, there have emerged a number of proposals that extend PLS from a vector $\boldsymbol{X} \in \mathbb{R}^p$ to a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ (Bro 1996; Eliseyev and Aksenova 2013; Zhao et al. 2013). We give a sketch of the algorithm proposed by Bro (1996) in Algorithm 3, formulated in a way that one can easily see the connection between the vector PLS and the tensor PLS. Comparing Algorithm 3 to the Wold PLS Algorithm 1, we see that the two algorithms share the same structure. The only modification is to replace $\boldsymbol{w}^\top \boldsymbol{E}_{s-1}$ in Step 1, and $\boldsymbol{w}_s^\top \boldsymbol{X}$ in Step 2 of Algorithm 1, with $\boldsymbol{E}_{s-1} \times_1 \boldsymbol{w}_{1,s}^\top \times_2 \cdots \times_m \boldsymbol{w}_{m,s}^\top$ in Step 1, and $\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{w}_{1,s}^\top \times_2 \cdots \times_m \boldsymbol{w}_{m,s}^\top$ in Step 2 of Algorithm 3, respectively, where $\times_k$ denotes the k-mode product. This modification is natural when $\boldsymbol{\mathcal{X}}$ and accordingly $\boldsymbol{E}_{s-1}$ become a tensor, and the product between two vectors is replaced by the k-mode tensor product. There are also some variants of Algorithm 3. For instance, Zhao et al. (2013) replaced the vectors $\boldsymbol{w}_{k,s}$, $k = 1, \ldots, m$, with matrices $\boldsymbol{W}_{k,s} \in \mathbb{R}^{p_k \times u_k}$, for some pre-specified dimensions u_k, $k = 1, \ldots, m$.

Algorithm 3 An existing tensor PLS algorithm

[0] Initialize $\boldsymbol{E}_0 = \boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$.
for $s = 1, \ldots, d$ **do**
[1] Find $\boldsymbol{w}_{k,s} \in \mathbb{R}^{p_k}$, for all $k = 1, \ldots, m$, that maximize

$$\|\mathrm{cov}(\boldsymbol{Y}, \boldsymbol{E}_{s-1} \times_1 \boldsymbol{w}_{1,s}^\top \times_2 \cdots \times_m \boldsymbol{w}_{m,s}^\top)\|$$

[2] Define a random variable $t_s = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{w}_{1,s}^\top \times_2 \ldots \times_m \boldsymbol{w}_{m,s}^\top \in \mathbb{R}$
[3] Compute a tensor $\boldsymbol{v}_s = \mathrm{cov}(\boldsymbol{\mathcal{X}}, t_s)/\mathrm{var}(t_s) \in \mathbb{R}^{p_1 \times \cdots \times p_m}$
[4] Deflate the predictor $\boldsymbol{E}_s = \boldsymbol{E}_{s-1} - t_s \boldsymbol{v}_s$
end for
[5] Reduce $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ to a vector $\boldsymbol{T} = (t_1, \ldots, t_d)^\top \in \mathbb{R}^d$
[6] Regress $\boldsymbol{Y} \in \mathbb{R}^r$ on $\boldsymbol{T} \in \mathbb{R}^d$

2.3 A New Tensor PLS Algorithm

Similar to their vector counterpart of the Wold PLS, the existing tensor PLS algorithms do not have a population interpretation of their estimators. This motivates us to develop a PLS algorithm for tensor predictor that has a clear population model and interpretation. The new algorithm is given in Algorithm 4, while its population interpretation and statistical properties are established in Section 3.

Algorithm 4 A new tensor PLS algorithm.

[0] Initialize the cross covariance tensor $\boldsymbol{\mathcal{C}}_0 = \boldsymbol{\mathcal{C}} \equiv \mathrm{cov}(\boldsymbol{\mathcal{X}}, \boldsymbol{Y}) \in \mathbb{R}^{p_1 \times \cdots \times p_m \times r}$, and the covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{Y}} \equiv \mathrm{cov}(\boldsymbol{Y})$ and $\boldsymbol{\Sigma}_{\boldsymbol{X}} \equiv \mathrm{cov}\{\mathrm{vec}(\boldsymbol{\mathcal{X}})\} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$.
for $k = 1, \ldots, m$ **do**
[1] Standardize the mode-k cross covariance matrix:

$$\tilde{\boldsymbol{C}}_{0k} = \boldsymbol{C}_{0(k)} \left(\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1/2} \otimes \boldsymbol{\Sigma}_m^{-1/2} \otimes \cdots \otimes \boldsymbol{\Sigma}_{k+1}^{-1/2} \otimes \boldsymbol{\Sigma}_{k-1}^{-1/2} \otimes \cdots \otimes \boldsymbol{\Sigma}_1^{-1/2} \right) \in \mathbb{R}^{p_k \times (r \prod_{j \neq k} p_j)}$$

for $s = 1, \ldots, d_k$ **do**
[2] Find $\boldsymbol{w}_{ks} \in p_k$ that maximizes $\boldsymbol{w}^\top \tilde{\boldsymbol{C}}_{(s-1)k} \tilde{\boldsymbol{C}}_{(s-1)k} \boldsymbol{w}^\top$, subject to $\boldsymbol{w}^\top \boldsymbol{w} = 1$
[3] Deflate the cross covariance $\tilde{\boldsymbol{C}}_{sk} = \boldsymbol{Q}_{sk} \tilde{\boldsymbol{C}}_{0k}$, where $\boldsymbol{Q}_{sk} \in \mathbb{R}^{p_k \times p_k}$ is the projection matrix onto the orthogonal subspace of $\mathrm{span}(\boldsymbol{\Sigma}_k \boldsymbol{W}_{sk})$, and $\boldsymbol{W}_{sk} = (\boldsymbol{w}_{1k}, \ldots, \boldsymbol{w}_{sk}) \in \mathbb{R}^{p_k \times s}$
end for
end for
[4] Reduce $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ to a tensor $\boldsymbol{\mathcal{T}} = [\![\boldsymbol{\mathcal{X}}; \boldsymbol{W}_1^\top, \ldots, \boldsymbol{W}_m^\top]\!] \in \mathbb{R}^{d_1 \times \cdots \times d_m}$, where $\boldsymbol{W}_k = (\boldsymbol{w}_{1k}, \ldots, \boldsymbol{w}_{d_k k}) \in \mathbb{R}^{p_d \times d_k}$
[5] Regress $\boldsymbol{Y} \in \mathbb{R}^r$ on $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$

As Algorithm 4 reveals, the new tensor PLS solution essentially reduces a tensor predictor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ to a latent tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$, where $d_k \leq p_k$. As such, substantial dimension reduction is achieved. The latent tensor takes the form $\boldsymbol{\mathcal{T}} = [\![\boldsymbol{\mathcal{X}}; \boldsymbol{W}_1^\top, \ldots, \boldsymbol{W}_m^\top]\!]$, which is a Tucker decomposition of the original tensor $\boldsymbol{\mathcal{X}}$, and $\{\boldsymbol{W}_k\}_{k=1}^m$ are the factor matrices. This decomposition is a generalization of the linear combination $\boldsymbol{W}^\top \boldsymbol{X}$ when $\boldsymbol{X}$ is a vector. The new algorithm then estimates

each factor matrix $\boldsymbol{W}_k$, $k = 1, \ldots, m$, sequentially and independently of other factor matrices, and estimates its columns $\boldsymbol{w}_{sk}$, $s = 1, \ldots, d_k$, $k = 1, \ldots, m$, one at a time.

We make a few more remarks about the new algorithm. First, in the initialization step, the covariance of the tensor $\boldsymbol{X}$ is assumed to have a *separable Kronecker covariance structure*, that is, $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \text{cov}\{\text{vec}(\boldsymbol{X})\} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$. This separable structure assumption is important to help reduce the number of free parameters in tensor regression. It is also fairly commonly adopted in the tensor literature (e.g., Hoff 2011; Fosdick and Hoff 2014). Given the iid data, each individual matrix $\boldsymbol{\Sigma}_k$ can be estimated using

$$\widehat{\boldsymbol{\Sigma}}_k = \left(n \prod_{j \neq k} p_j \right)^{-1} \sum_{i=1}^{n} \boldsymbol{X}_{i(k)} \boldsymbol{X}_{i(k)}^{\mathsf{T}}, \tag{1}$$

where $\boldsymbol{X}_{i(k)}$ is the mode-k matricization of $\boldsymbol{X}_i$, $i = 1, \ldots, n$. This estimator is invertible as long as $n \prod_{j \neq k} p_j > p_k$. In the neuroimaging data, the values of $p_1, \ldots, p_m$ are usually comparable, and thus it almost always holds that $n \prod_{j \neq k} p_j > p_k$ even for a very small n. This estimator is also $\sqrt{n}$-consistent when $\text{vec}(\boldsymbol{X})$ has finite fourth moments (Li and Zhang 2016, Lemma 1). Furthermore, if the tensor normality of $\boldsymbol{X}$ holds, one may also employ the maximum likelihood estimator of $\boldsymbol{\Sigma}_k$ (Manceur and Dutilleul 2013). That is, for $k = 1, \ldots, m$, we start with (1) as initial estimators, then we set in turn,

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n \prod_{j \neq k} p_j} \sum_{i=1}^{n} \boldsymbol{X}_{i(k)} \left\{ (\widehat{\boldsymbol{\Sigma}}_m)^{-1} \otimes \cdots \otimes (\widehat{\boldsymbol{\Sigma}}_{k+1})^{-1} \otimes (\widehat{\boldsymbol{\Sigma}}_{k-1})^{-1} \otimes \cdots \otimes (\widehat{\boldsymbol{\Sigma}}_1)^{-1} \right\} \boldsymbol{X}_{i(k)}^{\mathsf{T}}, \tag{2}$$

where we iteratively update each $\boldsymbol{\Sigma}_k$ given the rest. Again, all such iteratively updated $\widehat{\boldsymbol{\Sigma}}_k$'s are invertible, and the resulting maximum likelihood estimators are $\sqrt{n}$-consistent. Second, in Step 1 of Algorithm 4, the mode-k matricization $\boldsymbol{C}_{0(k)}$ of $\boldsymbol{C}_0$ is standardized to $\tilde{\boldsymbol{C}}_{0k}$. This step is to adjust for potentially different scales in different modes of $\boldsymbol{X}$. However, the standardization step can be omitted, as the theoretical properties of the algorithm do not change with or without this step. Third, the loop of Steps 2 and 3 is essentially a SIMPLS algorithm for a vector case. Here, the projection matrix $\boldsymbol{Q}_{sk}$ in the deflation step takes the form of projection onto the complementary space of $\text{span}(\boldsymbol{\Sigma}_k \boldsymbol{W}_{sk})$. This definition of the projection matrix works for both vector and tensor predictors. Finally, after PLS identifies a lower dimensional latent tensor $\boldsymbol{T}$, regression is carried out for a vector $\boldsymbol{Y}$ on a tensor $\boldsymbol{T}$, and its estimation is discussed in detail in the next section.

3. Tensor PLS Population Model

In this section, we establish a population interpretation for the tensor PLS estimator of Algorithm 4. We achieve this through three steps: first, we consider a population regression model and its parameter estimation; then we introduce some sparsity assumption to this model; and finally we show that tensor PLS actually gives an estimator of the regression parameter of this population model under the sparsity assumption.

3.1 A Population Model

Consider the following tensor linear model

$$Y_k = \langle \boldsymbol{B}_{::k}, \boldsymbol{X} \rangle + \varepsilon_k, \quad k = 1, \ldots, r,$$

where $\boldsymbol{B}_{::k} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ denotes the sub-tensor of the regression coefficient tensor $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_m \times r}$, $\langle \cdot, \cdot \rangle$ denotes the inner product, and $\varepsilon_k \in \mathbb{R}$ is the kth error term corresponding to the kth response variable Y_k, and is independent of $\boldsymbol{X}$, $k = 1, \ldots, m$. The above model can be written in a more compact form,

$$\boldsymbol{Y} = \boldsymbol{B}_{(m+1)} \text{vec}(\boldsymbol{X}) + \boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{B}_{(m+1)} \in \mathbb{R}^{r \times \prod_{k=1}^{m} p_k}$ is the mode-$(m+1)$ matricization of $\boldsymbol{B}$, $\boldsymbol{Y} = (Y_1, \ldots, Y_r)^{\mathsf{T}} \in \mathbb{R}^r$, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_r)^{\mathsf{T}} \in \mathbb{R}^r$. Furthermore, we assume the separable Kronecker covariance structure that $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \text{cov}\{\text{vec}(\boldsymbol{X})\} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$, with each $\boldsymbol{\Sigma}_k > 0$. Next, we consider three different estimators of the regression coefficient tensor $\boldsymbol{B}$.

The first is obtained by simply vectorizing the tensor predictor $\boldsymbol{X}$, then adopting the usual ordinary least squares (OLS). However, this estimator involves inversion of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \text{cov}\{\text{vec}(\boldsymbol{X})\}$. Its dimension, $\prod_{k=1}^{m} p_k \times \prod_{k=1}^{m} p_k$, is extremely large. For that reason, we no longer consider this estimator in this article.

The second is also an OLS estimator of $\boldsymbol{B}$, but under the separable covariance structure such that $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$. Toward that end, we have the following result.

Lemma 3.1. Under the tensor predictor linear model (3) and the separable Kronecker covariance structure $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1 > 0$, the following is true

$$\boldsymbol{B} = [\![\boldsymbol{C}; \boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_m^{-1}, \boldsymbol{I}_r]\!],$$

where $\boldsymbol{C} = \text{cov}(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}^{p_1 \times \cdots \times p_m \times r}$ is the cross covariance tensor between $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Consequently, we have the modified OLS estimator of $\boldsymbol{B}$ under the separable covariance,

$$\widehat{\boldsymbol{B}}_{\text{OLS}} = [\![\widehat{\boldsymbol{C}}; \widehat{\boldsymbol{\Sigma}}_1^{-1}, \ldots, \widehat{\boldsymbol{\Sigma}}_m^{-1}, \boldsymbol{I}_r]\!], \tag{4}$$

where $\widehat{\boldsymbol{C}} = \widehat{\text{cov}}(\boldsymbol{X}, \boldsymbol{Y})$ is the usual sample cross-covariance estimator, and $\widehat{\boldsymbol{\Sigma}}_k$ is a sample estimator of $\boldsymbol{\Sigma}_k$, such as (1) and (2) described in Section 2.3.

The third is an estimator based on the tensor PLS Algorithm 4. That is, we first reduce the predictor tensor $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ to a latent tensor $\boldsymbol{T} \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ through factor matrices $\{\boldsymbol{W}_k\}_{k=1}^{m}$, then fit a tensor predictor linear regression of $\boldsymbol{Y}$ on $\boldsymbol{T}$. We call this a PLS estimator of $\boldsymbol{B}$ in model (3), and the next lemma summarizes this estimator.

Lemma 3.2. Under the tensor predictor linear model (3), the partial least-squares estimator of the regression coefficient tensor $\boldsymbol{B}$ is of the form,

$$\widehat{\boldsymbol{B}}_{\text{PLS}} = [\![\widehat{\boldsymbol{\Psi}}; \widehat{\boldsymbol{W}}_1, \ldots, \widehat{\boldsymbol{W}}_m, \boldsymbol{I}_r]\!], \tag{5}$$

where $\widehat{\boldsymbol{\Psi}} = [\![\widehat{\boldsymbol{C}}_{\boldsymbol{T}}; (\widehat{\boldsymbol{W}}_1^{\mathsf{T}} \widehat{\boldsymbol{\Sigma}}_1 \widehat{\boldsymbol{W}}_1)^{-1}, \ldots, (\widehat{\boldsymbol{W}}_m^{\mathsf{T}} \widehat{\boldsymbol{\Sigma}}_m \widehat{\boldsymbol{W}}_m)^{-1}, \boldsymbol{I}_r]\!]$ is the OLS estimator for $\boldsymbol{\Psi}$ while respecting the separable covariance structure, $\widehat{\boldsymbol{C}}_{\boldsymbol{T}} = \widehat{\text{cov}}(\boldsymbol{T}, \boldsymbol{Y})$, and $\widehat{\boldsymbol{W}}_k$ is the sample factor matrix estimator obtained by the tensor PLS Algorithm 4, $k = 1, \ldots, m$.

In Section 3.3, we show that, under appropriate conditions, why $\widehat{\boldsymbol{B}}_{\text{PLS}}$ is a legitimate estimator of $\boldsymbol{B}$ in (3). In Section 5, we also numerically compare $\widehat{\boldsymbol{B}}_{\text{PLS}}$ in (5) with $\widehat{\boldsymbol{B}}_{\text{OLS}}$ in (4). Intuitively, thanks to substantial dimension reduction from $\boldsymbol{X}$ to $\boldsymbol{T}$, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ is expected to improve over $\widehat{\boldsymbol{B}}_{\text{OLS}}$. The next proposition summarizes the further relation between $\widehat{\boldsymbol{B}}_{\text{OLS}}$ and $\widehat{\boldsymbol{B}}_{\text{PLS}}$. That is, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ is a Tucker decomposition of $\widehat{\boldsymbol{B}}_{\text{OLS}}$ with projections onto the m latent subspaces of tensor PLS.

Proposition 3.1. For the latent tensor $\boldsymbol{T}_i = [\![\boldsymbol{X}_i; \widehat{\boldsymbol{W}}_1^{\mathsf{T}}, \ldots, \widehat{\boldsymbol{W}}_m^{\mathsf{T}}]\!]$, $i = 1, \ldots, n$ from the sample PLS estimation in Algorithm 4, the following holds true,

$$\begin{aligned}\widehat{\boldsymbol{B}}_{\text{PLS}} &= \widehat{\boldsymbol{B}}_{\text{OLS}} \times_1 \boldsymbol{P}_{\widehat{\boldsymbol{W}}_1(\widehat{\boldsymbol{\Sigma}}_1)} \times_2 \cdots \times_m \boldsymbol{P}_{\widehat{\boldsymbol{W}}_m(\widehat{\boldsymbol{\Sigma}}_m)} \\ &= [\![\widehat{\boldsymbol{B}}_{\text{OLS}}; \boldsymbol{P}_{\widehat{\boldsymbol{W}}_1(\widehat{\boldsymbol{\Sigma}}_1)}, \ldots, \boldsymbol{P}_{\widehat{\boldsymbol{W}}_m(\widehat{\boldsymbol{\Sigma}}_m)}, \boldsymbol{I}_r]\!],\end{aligned}$$

where $\boldsymbol{P}_{\widehat{\boldsymbol{W}}_k(\widehat{\boldsymbol{\Sigma}}_k)} = \widehat{\boldsymbol{W}}_k(\widehat{\boldsymbol{W}}_k^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{W}}_k)^{-1}\widehat{\boldsymbol{W}}_k^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}_k$ is the projection onto $\text{span}(\widehat{\boldsymbol{W}}_k)$ with inner product matrix $\widehat{\boldsymbol{\Sigma}}_k$, $k = 1, \ldots, m$.

We also briefly comment on the existence and the identifiability issue of the two estimators $\widehat{\boldsymbol{B}}_{\text{OLS}}$ and $\widehat{\boldsymbol{B}}_{\text{PLS}}$. As long as $n \prod_{j \neq k} p_j > p_k$, which is a very mild condition and almost always holds in the neuroimaging context, both the OLS and PLS estimators exist. Moreover, we note that $\boldsymbol{\Sigma}_k$ in the separable covariance structure is not identifiable, since one can replace $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Sigma}_j$ with another pair $\lambda\boldsymbol{\Sigma}_k$ and $\lambda^{-1}\boldsymbol{\Sigma}_j$ for an arbitrary $\lambda > 0$ without affecting $\boldsymbol{\Sigma}_{\boldsymbol{X}}$. On the other hand, the OLS estimator is uniquely defined, because $\widehat{\boldsymbol{B}}_{\text{OLS}} = (\widehat{\boldsymbol{\Sigma}}_m^{-1} \otimes \cdots \otimes \widehat{\boldsymbol{\Sigma}}_1^{-1})\widehat{\boldsymbol{C}}_{(m+1)}$ from the proof of Lemma 3.1. Then, the PLS estimator is also uniquely defined, because by Proposition 3.1, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ can be formulated as a function of $\widehat{\boldsymbol{B}}_{\text{OLS}}$ and the projection matrix $\boldsymbol{P}_{\widehat{\boldsymbol{W}}_k(\widehat{\boldsymbol{\Sigma}}_k)} = \widehat{\boldsymbol{W}}_k(\widehat{\boldsymbol{W}}_k^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{W}}_k)^{-1}\widehat{\boldsymbol{W}}_k^{\mathsf{T}}\widehat{\boldsymbol{\Sigma}}_k$, and the latter is not affected by scale change in $\widehat{\boldsymbol{\Sigma}}_k$.

3.2 Tensor Envelope

We next introduce some sparsity assumption to model (3), which would shed useful insight on how PLS achieves effective dimension reduction. It is closely connected with a nascent concept of *envelope* (Cook, Li, and Chiaromonte 2010). We thus first develop the envelope notion for our tensor regression model (3), then use this new concept to establish the population interpretation of the tensor PLS estimator in the next section.

The envelope notion of Cook, Li, and Chiaromonte (2010) is built upon a key assumption that some aspects of the predictors are irrelevant to the response and do not intertwine with the rest of the predictors. We adopt this principle in our tensor model (3). More specifically, suppose there exist a series of subspaces, $\mathcal{S}_k \subseteq \mathbb{R}^{p_k}$, $k = 1, \ldots, m$, such that,

$$\boldsymbol{X} \times_k \boldsymbol{Q}_k \perp\!\!\!\perp \boldsymbol{X} \times_k \boldsymbol{P}_k, \qquad \boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X} \times_k \boldsymbol{Q}_k \mid \boldsymbol{X} \times_k \boldsymbol{P}_k, \tag{6}$$

where $\boldsymbol{P}_k \in \mathbb{R}^{p_k \times p_k}$ is the projection matrix onto $\mathcal{S}_k$, $\boldsymbol{Q}_k = \boldsymbol{I}_{p_k} - \boldsymbol{P}_k \in \mathbb{R}^{p_k \times p_k}$ is the projection onto the complement space of $\mathcal{S}_k$, and $\times_k$ is the k-mode product. Here, $\perp\!\!\!\perp$ denotes statistical independence, but can also be replaced with $\perp$ that denotes uncorrelatedness. To facilitate understanding of this assumption, we consider the special case where $m = 1$ and the predictor $\boldsymbol{X}$ reduces to a $p_1 \times 1$ vector. Furthermore, let $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p_1 \times u_1}$ denote a basis matrix of $\mathcal{S}_1$, where u_1 is the dimension of $\mathcal{S}_1$ and $u_1 \leq p_1$. Let $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p_1 \times (p_1 - u_1)}$ denote a basis of the complement space of $\mathcal{S}_1$. Then assumption (6) states that the linear combinations $\boldsymbol{\Gamma}_0^{\mathsf{T}}\boldsymbol{X}$ are irrelevant to the regression $\boldsymbol{Y}|\boldsymbol{X}$, as $\boldsymbol{\Gamma}_0^{\mathsf{T}}\boldsymbol{X}$ are conditionally independent of $\boldsymbol{Y}$ given $\boldsymbol{\Gamma}_1^{\mathsf{T}}\boldsymbol{X}$, and do not affect the rest of predictors $\boldsymbol{\Gamma}_1^{\mathsf{T}}\boldsymbol{X}$. This leads to the notion of *sufficient dimension reduction* (Cook 1998; Li and Wang 2007).

It is also important to note that, although assumption (6) looks unfamiliar, it is closely related to the *sparsity principle* that is well known and commonly adopted in the context of *variable selection*, which assumes a subset of predictors are irrelevant to the regression. The two assumptions share exactly the same spirit that only part of information is deemed useful for regression and the rest irrelevant. They are different in that, whereas the usual sparsity principle focuses on individual variables, (6) permits *linear combinations* of the predictors to be irrelevant. As such we term this assumption the *generalized sparsity principle*, and we will show later that PLS essentially adopts this principle for dimension reduction and improved estimation of the regression coefficient.

Introducing (6) to model (3), we get the following parameterization.

Proposition 3.2. Under the tensor linear model (3), the assumption (6) implies the parameterization,

$$\begin{aligned}\boldsymbol{B} &= [\![\boldsymbol{\Theta}; \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_m, \boldsymbol{I}_r]\!] \quad \text{for some } \boldsymbol{\Theta} \in \mathbb{R}^{u_1 \times \cdots \times u_m}, \\ \boldsymbol{\Sigma}_k &= \boldsymbol{\Gamma}_k\boldsymbol{\Omega}_k\boldsymbol{\Gamma}_k^{\mathsf{T}} + \boldsymbol{\Gamma}_{0k}\boldsymbol{\Omega}_{0k}\boldsymbol{\Gamma}_{0k}^{\mathsf{T}}, \quad k = 1, \ldots, m,\end{aligned} \tag{7}$$

where $\boldsymbol{\Gamma}_k \in \mathbb{R}^{p_k \times u_k}$ denotes a basis of $\mathcal{S}_k$ such that $\boldsymbol{P}_k = \boldsymbol{\Gamma}_k\boldsymbol{\Gamma}_k^{\mathsf{T}}$, $\boldsymbol{\Gamma}_{0k} \in \mathbb{R}^{p_k \times (p_k - u_k)}$ denotes a basis of the complement space of $\mathcal{S}_k$ such that $\boldsymbol{Q}_k = \boldsymbol{\Gamma}_{0k}\boldsymbol{\Gamma}_{0k}^{\mathsf{T}}$, and $\boldsymbol{\Omega}_k \in \mathbb{R}^{u_k \times u_k}$, $\boldsymbol{\Omega}_{0k} \in \mathbb{R}^{(p_k - u_k) \times (p_k - u_k)}$ denote two symmetric positive definite matrices, $k = 1, \ldots, m$.

To address the issue of existence and uniqueness of $\mathcal{S}_k$ defined in (6), we adopt the definitions of *reducing subspace* and *envelope* of Cook, Li, and Chiaromonte (2010). The basic idea is to seek the intersection of all subspaces $\mathcal{S}_k$ that satisfy (6). In the light of Proposition 3.2, we arrive at the following two definitions, which are analogous to the definitions of response tensor envelope (Li and Zhang 2016).

Definition 3.1. The predictor envelope for $\boldsymbol{B}_{(k)}$ in model (3), denoted by $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$, is the intersection of all reducing subspaces $\mathcal{R}$ of $\boldsymbol{\Sigma}_k$ that contain $\text{span}(\boldsymbol{B}_{(k)})$.

Definition 3.2. The tensor predictor envelope for $\boldsymbol{B}$ in model (3), denoted by $\mathcal{T}_{\boldsymbol{\Sigma}_{\boldsymbol{X}}}(\boldsymbol{B})$, is defined as the intersection of all reducing subspaces $\mathcal{E}$ of $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$ that contain $\text{span}(\boldsymbol{B}_{(m+1)})$ and can be written as $\mathcal{E} = \mathcal{E}_m \otimes \cdots \otimes \mathcal{E}_1$ for $\mathcal{E}_k \subseteq \mathbb{R}^{p_k}$, $k = 1, \ldots, m$.

Definition 3.1 defines an individual envelope that concerns $\boldsymbol{\Sigma}_k$ and $\boldsymbol{B}_{(k)}$, and it echoes the usual envelope definition (Cook, Li, and Chiaromonte 2010). Definition 3.2 is our new concept of tensor predictor envelope that concerns a structured covariance $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_m \otimes \cdots \otimes \boldsymbol{\Sigma}_1$ and $\boldsymbol{B}$. The next proposition summarizes the relation between the tensor predictor envelope $\mathcal{T}_{\boldsymbol{\Sigma}_{\boldsymbol{X}}}(\boldsymbol{B})$ and the individual envelopes $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$.

Proposition 3.3. $\mathcal{T}_{\boldsymbol{\Sigma}_X}(\boldsymbol{B}) = \mathcal{E}_{\boldsymbol{\Sigma}_m}(\boldsymbol{B}_{(m)}) \otimes \cdots \otimes \mathcal{E}_{\boldsymbol{\Sigma}_1}(\boldsymbol{B}_{(1)})$.

Conceptually, the tensor predictor envelope $\mathcal{T}_{\boldsymbol{\Sigma}_X}(\boldsymbol{B})$ captures all the relevant predictor information, and regression with respect to $\mathcal{T}_{\boldsymbol{\Sigma}_X}(\boldsymbol{B})$ would lose no information.

3.3 Population Interpretation of Tensor PLS

Our key finding is that, tensor PLS actually seeks an estimation of the tensor predictor envelope $\mathcal{T}_{\boldsymbol{\Sigma}_X}(\boldsymbol{B})$, which in turn leads to effective dimension reduction and improved estimation. In other words, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ in (5) resulting from Algorithm 4 is an estimator of the coefficient tensor $\boldsymbol{B}$ in the tensor linear model (3) under the generalized sparsity principle (6). For this reason, we refer $\widehat{\boldsymbol{B}}_{\text{PLS}}$ as the *tensor envelope PLS estimator*.

To see that, we note, central to tensor PLS is identification of the factor matrices $\{\boldsymbol{W}_k\}_{k=1}^m$, because the reduced tensor $\boldsymbol{T}$ is constructed from $\{\boldsymbol{W}_k\}_{k=1}^m$ as $\boldsymbol{T} = [\![\boldsymbol{X}; \boldsymbol{W}_1^\mathsf{T}, \ldots, \boldsymbol{W}_m^\mathsf{T}]\!]$. As such, the interpretation of tensor PLS hinges on $\boldsymbol{W}_k$ and the subspace spanned by $\boldsymbol{W}_k$. Recall in Steps 2 and 3 of Algorithm 4, the column $\boldsymbol{w}_{ks}$, $s = 1, \ldots, d_k$, of $\boldsymbol{W}_k$ is the maximizer of a function of a deflated cross covariance matrix. The next lemma gives an explicit population characterization of those maximizers.

Lemma 3.3. The population solution of $\boldsymbol{w}_{sk}$, $k = 1, \ldots, m$, $s = 1, \ldots, d_k$, in Algorithm 4 is the dominant eigenvector of $\boldsymbol{Q}_{(s-1)k}\tilde{\boldsymbol{C}}_k\boldsymbol{Q}_{(s-1)k}$, where $\boldsymbol{Q}_{(s-1)k}$ is as defined in Step 3 of Algorithm 4 for $s > 1$, and $\boldsymbol{Q}_{0k} \equiv \boldsymbol{I}_{p_k}$ for $s = 0$, and $\tilde{\boldsymbol{C}}_k$ is defined as

$$\tilde{\boldsymbol{C}}_k = \boldsymbol{C}_{(k)}\left(\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{-1} \otimes \boldsymbol{\Sigma}_m^{-1} \otimes \cdots \otimes \boldsymbol{\Sigma}_{k+1}^{-1} \otimes \boldsymbol{\Sigma}_{k-1}^{-1} \otimes \cdots \otimes \boldsymbol{\Sigma}_1^{-1}\right)\boldsymbol{C}_{(k)}^\mathsf{T},$$

where $\boldsymbol{C}_{(k)}$ is the mode-k matricization of the cross covariance tensor $\boldsymbol{C} = \text{cov}(\boldsymbol{X}, \boldsymbol{Y})$.

Next, we define $\mathcal{W}_k^{(s)} \subseteq \mathbb{R}^{p_k}$ as the subspace spanned by the first s steps or directions, $(\boldsymbol{w}_{1k}, \ldots, \boldsymbol{w}_{sk}) \in \mathbb{R}^{p_k \times s}$, in the process of obtaining $\boldsymbol{W}_k$, $k = 1, \ldots, m$. We let $\mathcal{W}_k^{(0)} = 0$, and consider full number of steps from $s = 1$ to $s = p_k$. The next theorem formally summarizes the connection between the tensor PLS latent subspaces and the tensor envelope concept $\mathcal{T}_{\boldsymbol{\Sigma}_X}(\boldsymbol{B})$.

Theorem 3.1. For $k = 1, \ldots, m$, the subspaces $\mathcal{W}_k^{(s)}$, $s = 0, \ldots, p_k$, are nested such that,

$$\mathcal{W}_k^{(0)} \subset \cdots \subset \mathcal{W}_k^{(u_k)} = \mathcal{E}_{\boldsymbol{\Sigma}_k}\left(\boldsymbol{B}_{(k)}\right) = \mathcal{W}_k^{(u_k+1)} = \cdots = \mathcal{W}_k^{(p_k)},$$

where $u_k \leq p_k$ is the dimension of the envelope $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$.

This theorem clearly establishes a population interpretation of the tensor PLS algorithm; that is, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ estimates $\boldsymbol{B}$ in the tensor model (3) under the sparsity principle (6). The factor matrices $\boldsymbol{W}_k$ are indeed the basis matrices of the envelopes $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$, $k = 1, \ldots, m$. In addition, this theorem establishes a connection between tensor PLS and the notion of sufficient dimension reduction (Cook 1998; Li and Wang 2007). That is, for the kth facet $\boldsymbol{B}_{(k)}$ of $\boldsymbol{B}$, when tensor PLS reaches u_k latent components, where $u_k = \dim\{\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})\}$, then it fully captures all the regression information.

3.4 Asymptotics

Now with the established population interpretation of the tensor envelope PLS estimator $\widehat{\boldsymbol{B}}_{\text{PLS}}$, we next investigate its asymptotic properties under model (3). We assume the sample estimators $\widehat{\boldsymbol{\Sigma}}_k$ such as (1) and (2) are $\sqrt{n}$-consistent. This holds under very minor conditions that $(\boldsymbol{Y}_i, \boldsymbol{X}_i)$, $i = 1, \ldots, n$, are iid with finite fourth moments. Then, the next theorem establishes the $\sqrt{n}$-consistency of $\widehat{\boldsymbol{B}}_{\text{PLS}}$.

Theorem 3.2. If d_k is chosen as $u_k = \dim\{\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})\}$, $k = 1, \ldots, m$, then $\widehat{\boldsymbol{W}}_k$ is $\sqrt{n}$-consistent for $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$ in the sense that the projection onto $\text{span}(\widehat{\boldsymbol{W}}_k)$ is $\sqrt{n}$-consistent to the projection onto $\mathcal{E}_{\boldsymbol{\Sigma}_k}(\boldsymbol{B}_{(k)})$. Moreover, $\widehat{\boldsymbol{B}}_{\text{PLS}}$ is a $\sqrt{n}$-consistent estimator for $\boldsymbol{B}$ in model (3) for any d_k such that $d_k \geq u_k$.

As in any dimension reduction solution, selecting the PLS dimension is a crucial problem. This theorem says, as long as the selected dimension d_k exceeds the true dimension u_k, the resulting tensor envelope PLS estimator is consistent. On the other hand, a large value of d_k induces more unknown parameters and thus potentially larger variation. Its selection reflects the usual bias-variance tradeoff. In this article, we follow the classical PLS literature to select the PLS dimension based on the mean-squared error of prediction of the outcome $\boldsymbol{Y}$ via cross-validation. Likelihood based test is a viable alternative (Schott 2013); however, it requires development of a completely new tensor PLS algorithm based on a likelihood function, and we leave it as potential future research.

4. Comparison

In this section, we first compare our tensor envelope PLS solution with Cook, Helland, and Su (2013), mainly from the aspect of computational feasibility. We then analytically compare with Zhou, Li, and Zhu (2013). We will also compare the two solutions numerically in Section 5.

4.1 Vector Envelope PLS

Cook, Li, and Chiaromonte (2010) first proposed the concept of envelope, and Cook, Helland, and Su (2013) connected it with PLS. Our solution differs from Cook, Helland, and Su (2013) in that, we tackle a tensor predictor, whereas Cook, Helland, and Su (2013) deals with a vector predictor. Directly vectorizing a tensor predictor then applying Cook, Helland, and Su (2013) suffers several critical issues. First, the computation becomes almost infeasible. For instance, for our ADHD example with a $30 \times 36 \times 30$ predictor, the dimension of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is $32{,}400 \times 32{,}400$. This would require a software such as Matlab to allocate 7.8 GB memory for covariance computation alone, which essentially renders standard computing software inapplicable. By contrast, our solution enables a separable computation along each mode of the tensor, which is only 30 or 36 in the ADHD example, and thus is computationally feasible and fast. Second, directly applying Cook, Helland, and Su (2013) would require the sample size $n > \prod_{k=1}^m p_k$, whereas our method only requires $n \prod_{j \neq k} p_j > p_k$. Clearly, our approach is better suited to neuroimaging applications, where the sample size is usually limited. Finally, vectorization would

destroy all inherent structural information in tensor, while our tensor based solution respects and preserves the tensor structure.

4.2 Reduced Rank Tensor Regression

Next, we compare our method with that of Zhou, Li, and Zhu (2013). We have chosen Zhou, Li, and Zhu (2013), because both solutions share a similar model, both associate all image voxels jointly with clinical variables, and both employ substantial dimension reduction for effective parameter estimation. The comparison begins with an examination of the models imposed by the two methods. Specifically, Zhou, Li, and Zhu (2013) proposed a class of generalized linear models (GLM) with a scalar response and a tensor predictor. Under the identity link, their model becomes

$$Y = \langle \boldsymbol{B}, \boldsymbol{X} \rangle + \varepsilon, \tag{8}$$

which is the same as the tensor envelope PLS model (3). Both models associate all image voxels in $\boldsymbol{X}$ with the outcome Y. However, Zhou, Li, and Zhu (2013) treated $\boldsymbol{X}$ as fixed, whereas tensor envelope PLS treats $\boldsymbol{X}$ as random and further imposes the separable Kronecker covariance structure for $\boldsymbol{\Sigma}_{\boldsymbol{X}}$. In addition, Zhou, Li, and Zhu (2013) worked with GLM, while our work focuses on the normal linear model. On the other hand, Zhou, Li, and Zhu (2013) is not directly applicable to multivariate response variables, and additional extension is required. By contrast, the PLS model naturally incorporates multiple responses.

We next examine the dimension reduction strategies used in the two methods. The core strategy of Zhou, Li, and Zhu (2013) is to impose a *low rank structure* on the coefficient tensor $\boldsymbol{B}$. They employed a special case of the Tucker decomposition, named the *CP decomposition*, that requires the same number of columns, or rank, for all its factor matrices, plus a superdiagonal core tensor. This results in $\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_m^{(r)}$, where $\boldsymbol{\beta}_k^{(r)} \in \mathbb{R}^{p_k}$, $k = 1, \ldots, m$, $r = 1, \ldots, R$, are all column vectors, $\circ$ denotes the outer product, and R is the rank (Kolda and Bader 2009). We denote the resulting sample estimator as $\widehat{\boldsymbol{B}}_{\mathrm{CP}}$. By imposing such a low rank structure, the number of free parameters substantially reduces from the exponential order $\prod_k p_k$ to the linear order $R\sum_k p_k$. For a $30 \times 36 \times 30$ MRI image predictor, for instance, the number of parameters is reduced from 32,400 to 96 for a rank-1 CP model, and 288 for a rank-3 model. By contrast, the key strategy of our tensor envelope PLS method is to recognize the irrelevant predictor information, and to focus the estimation based on the relevant information only. It achieves this through exploring the predictor covariance structure $\boldsymbol{\Sigma}_{\boldsymbol{X}}$. As a result, the number of free parameters is substantially reduced as well. Substituting the envelope decomposition (7) into model (8), the regression mean function becomes

$$\mathrm{E}(Y \mid \boldsymbol{X}) = \boldsymbol{B}_{(m+1)}\mathrm{vec}(\boldsymbol{X}) = \boldsymbol{\Theta}_{(m+1)}\mathrm{vec}(\llbracket \boldsymbol{X}; \boldsymbol{\Gamma}_1^{\mathsf{T}}, \ldots, \boldsymbol{\Gamma}_m^{\mathsf{T}} \rrbracket).$$

Consequently, the regression of Y on $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$ can now focus on $\llbracket \boldsymbol{X}; \boldsymbol{\Gamma}_1^{\mathsf{T}}, \ldots, \boldsymbol{\Gamma}_m^{\mathsf{T}} \rrbracket \in \mathbb{R}^{u_1 \times \cdots \times u_m}$. The number of free parameters is reduced from $\prod_{k=1}^{m} p_k$ to $\prod_{k=1}^{m} u_k + \sum_{k=1}^{m} p_k u_k$. The second term in the latter accounts for estimation of the unknown basis matrices $\boldsymbol{\Gamma}_k$, or equivalently the PLS factor matrices $\boldsymbol{W}_k$, $k = 1, \ldots, m$. In practice, $\{u_k\}_{k=1}^{m}$ are often small, thus leading to substantial reduction. For the example of a $30 \times 36 \times 30$ MRI image predictor, the number of parameters is reduced from 32,400 to 97 if $u_1 = u_2 = u_3 = 1$, and to 315 if $u_1 = u_2 = u_3 = 3$.

In summary, both Zhou, Li, and Zhu (2013) and tensor envelope PLS reduce the vast number of unknown parameters to a manageable level through substantial dimension reduction. The difference is that the former achieves dimension reduction through the *reduced rank* approach, whereas the latter achieves it through the *generalized sparsity* principle and *tensor envelope* that eliminates the immaterial information in the tensor predictor and envelops the material information. Our numerical experiments in Section 5 have suggested that, Zhou, Li, and Zhu (2013) works best when the signal rank is small, the covariance structure $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is close to identity, and the sample size is relatively large. The tensor envelope PLS is more competitive when the covariance structure is complex, there is co-linearity present, and the sample size is moderate.

5. Simulations

In this section, we carry out simulations to investigate the empirical performance of our proposed tensor envelope PLS estimator, and compare with some alternative estimators. We consider both univariate and multivariate responses, 2-way and 3-way tensor predictors, and various scenarios that the PLS model assumptions hold or do not hold. The performance is evaluated on both estimation and prediction accuracy. For the 2-way predictor case, we also plot the estimated regression coefficient as a graphical evaluation.

5.1 Performance Under Different Scenarios: 2-Way Predictor

We first consider a model with a univariate response Y_i and a 2-way matrix predictor $\boldsymbol{X}_i \in \mathbb{R}^{64\times 64}$: $Y_i = \langle \boldsymbol{B}, \boldsymbol{X}_i \rangle + \varepsilon_i$. The coefficient matrix $\boldsymbol{B} \in \mathbb{R}^{64\times 64}$ takes value 0.1 or 1, where the ones form a particular shape, including square, cross, disk, triangle, and butterfly. See the first column of Figures 1 and 2 in Section 5.3. The numerical rank of $\boldsymbol{B}$ is 2 (square), 3 (cross), 9 (disk), 14 (triangle), and 30 (butterfly), respectively. Note that all elements in $\boldsymbol{B}$ are nonzero, which is challenging for a sparsity based estimation method. The error ε_i is generated from a standard normal distribution. The predictor $\boldsymbol{X}_i$ follows a matrix normal distribution with mean zero and covariance $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1$. We consider three different models for $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$, including a scenario that favors our tensor PLS method, a scenario that does *not* favor our method but the competing OLS and reduced rank solutions, and a scenario that violates key model assumptions and is challenging for all methods. The goal is to study and compare different estimation methods under both correct and incorrect model specification. We compare three estimators, the OLS estimator $\widehat{\boldsymbol{B}}_{\mathrm{OLS}}$ in (4) that respects the separable covariance structure (OLS), the tensor predictor regression estimator $\widehat{\boldsymbol{B}}_{\mathrm{CP}}$ of Zhou, Li, and Zhu (2013) that is built upon the low rank CP decomposition (CP), and the tensor envelope PLS estimator $\widehat{\boldsymbol{B}}_{\mathrm{PLS}}$ in (5). To obtain $\widehat{\boldsymbol{B}}_{\mathrm{CP}}$, we use a working rank $R = 3$ in all cases; see Zhou, Li, and Zhu (2013) for more discussion on the rank-3 approximation. To obtain $\widehat{\boldsymbol{B}}_{\mathrm{PLS}}$, we first use the true numerical

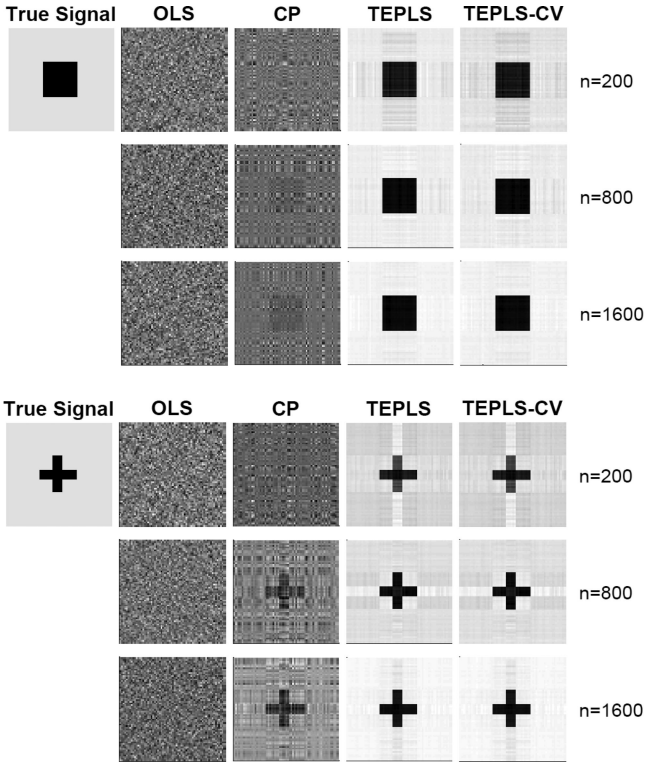

Figure 1. Consistency and comparison under relatively low rank images, where the true rank/envelope dimension u is 2 (square) and 3 (cross), respectively.

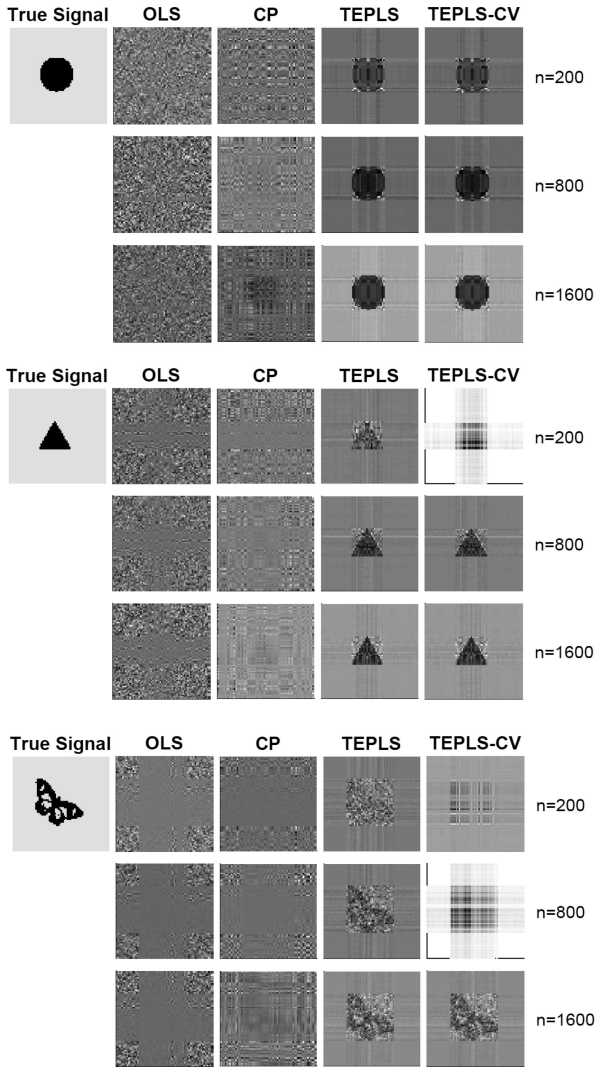

Figure 2. Consistency and comparison under relatively high rank images, where the true rank/envelope dimension u is 9 (disk), 14 (triangle), and 30 (butterfly), respectively.

rank (abbreviated as TEPLS), then use the estimated rank from cross-validation (TEPLS-CV).

Specifically, in Model-I, we construct $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ that comply with the envelope structure. Toward that end, we eigen-decompose $\boldsymbol{B} = \boldsymbol{G}_1\boldsymbol{D}\boldsymbol{G}_2^{\mathsf{T}}$, where $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathbb{R}^{64\times u}$. Then, we generate $\mathbf{\Gamma}_k = \boldsymbol{G}_k\boldsymbol{O}_k$, $k = 1, 2$, for random orthogonal matrices $\boldsymbol{O}_k \in \mathbb{R}^{u\times u}$. Here, $u = u_1 = u_2$ is the envelope dimension and is also equal to the numerical rank of the true signal $\boldsymbol{B}$. This way, it is guaranteed that $\text{span}(\boldsymbol{B}) \subseteq \text{span}(\mathbf{\Gamma}_1)$ and $\text{span}(\boldsymbol{B}^{\mathsf{T}}) \subseteq \text{span}(\mathbf{\Gamma}_2)$. We then orthogonalize $\mathbf{\Gamma}_k$ and obtained $\mathbf{\Gamma}_{0k}$. We set $\mathbf{\Sigma}_k = \mathbf{\Gamma}_k\mathbf{\Omega}_k\mathbf{\Gamma}_k^{\mathsf{T}} + \mathbf{\Gamma}_{0k}\mathbf{\Omega}_{0k}\mathbf{\Gamma}_{0k}^{\mathsf{T}}$, with $\mathbf{\Omega}_k = \boldsymbol{I}_{u_i}$ and $\mathbf{\Omega}_{0k} = 0.01\boldsymbol{I}_{p_i-u_i}$. The value 0.01 in $\mathbf{\Omega}_{0k}$ mimics the co-linearity that often appears among the predictors. In Model-II, we set $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \boldsymbol{I}_{64}$, such that it does *not* favor tensor envelope PLS but the alternative solutions. Under this setting, there is no correlation among the predictors, and there is no advantage in exploring the covariance structure. The only available reduction comes from the rank deficiency in $\boldsymbol{B}$. As such, this setting favors $\widehat{\boldsymbol{B}}_{\text{CP}}$, but not $\widehat{\boldsymbol{B}}_{\text{PLS}}$. Moreover, under this setting, $\boldsymbol{B} = \boldsymbol{C} = \text{cov}(\boldsymbol{X}, Y)$, as shown in Lemma 3.1, and as such $\widehat{\boldsymbol{B}}_{\text{OLS}}$ is expected to perform relatively well. In Model-III, we generate $\mathbf{\Sigma}_k$ as $\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}/\|\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}\|_F > 0$, where $\boldsymbol{A} \in \mathbb{R}^{64\times 64}$ is filled with random standard uniform numbers, $k = 1, 2$. This is a particularly challenging setting that favors neither of the three estimators.

Table 1 summarizes both the prediction and the estimation performance based on 100 data replications. The training sample size is set as $n = 200$. For prediction, we report the mean squared error $\sum_{i=1}^{n}(\widehat{Y}_i - Y_i)^2/n$ evaluated on an independently generated testing data of the same size. For estimation, we report the estimation error $\|\text{vec}(\boldsymbol{B} - \widehat{\boldsymbol{B}})\|_2$. It is seen from the table that our tensor envelope PLS clearly outperforms the two alternatives, by achieving a substantially smaller prediction and estimator error in all cases. The difference is more evident for those signals with a high rank, for instance, triangle or butterfly. It is also interesting to note that tensor envelope PLS performs well even when it is not favored by the true model (Model-II), or when its model assumptions are not satisfied (Model-III). Moreover, the PLS solutions under the true and estimated envelope dimension perform very similarly.

5.2 Performance Under Different Scenarios: 3-Way Predictor

We next consider a model with multivariate responses $\boldsymbol{Y}_i \in \mathbb{R}^5$ and a 3-way tensor predictor $\boldsymbol{X}_i \in \mathbb{R}^{20\times 20\times 20}$: $\boldsymbol{Y}_i = \boldsymbol{B}_{(m+1)}\text{vec}(\boldsymbol{X}_i) + \boldsymbol{\epsilon}_i$, $i = 1, \ldots, n$. The coefficient tensor $\boldsymbol{B} \in \mathbb{R}^{20\times 20\times 20\times 5}$ is of the structure, $\boldsymbol{B} = [\![\mathbf{\Theta}; \mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \mathbf{\Gamma}_3, \boldsymbol{I}_5]\!]$, where each $\mathbf{\Gamma}_k \in \mathbb{R}^{20\times u}$ is first filled with uniform random numbers between 0 and 1 then standardized to be semi-orthogonal, the elements of the core tensor $\mathbf{\Theta} \in \mathbb{R}^{u\times u\times u\times 5}$ are generated from a standard uniform distribution, and $u = 2$. The error

Table 1. Univariate response and 2-way predictor. Performance under various scenarios and comparison of estimators. OLS, CP, tensor envelope PLS with true and estimated envelope dimensions. Reported are the average and standard error (in parentheses) of the prediction mean squared error evaluated on an independent testing data, and the estimation error, all based on 100 data replications.

		Prediction			
Image	Model	OLS	CP	TEPLS	TEPLS-CV
Square	I	5184.1 (74.0)	3165.5 (331.7)	68.7 (1.4)	68.9 (1.4)
	II	182.8 (2.7)	297.4 (24.3)	8.6 (0.1)	8.6 (0.1)
	III	$>10^6$	$>10^4$	4.1 (0.3)	4.1 (0.3)
Cross	I	1956.7 (28.6)	1271.4 (191.9)	42.6 (0.5)	43.3 (0.6)
	II	111.3 (1.4)	180.3 (11.0)	5.3 (0.1)	5.3 (0.1)
	III	$>10^5$	$>10^4$	2.6 (0.1)	2.6 (0.1)
Disk	I	1032.6 (14.7)	1266.1 (106.7)	40.1 (1.4)	41.1 (1.4)
	II	165.9 (2.5)	254.7 (13.1)	7.7 (0.1)	7.8 (0.1)
	III	$>10^6$	$>10^4$	3.6 (0.2)	3.6 (0.2)
Triangle	I	412.4 (6.1)	574.2 (36.1)	16.8 (0.2)	17.3 (0.2)
	II	108.4 (1.8)	163.3 (11.5)	5.0 (0.1)	5.0 (0.1)
	III	$>10^5$	$>10^4$	2.6 (0.1)	2.6 (0.1)
Butterfly	I	341.6 (4.4)	536.1 (34.4)	15.3 (0.2)	15.7 (0.2)
	II	170.1 (3)	271.4 (16)	8.0 (0.1)	8.1 (0.1)
	III	$>10^6$	$>10^4$	3.6 (0.2)	3.6 (0.2)
		Estimation			
		OLS	CP	TEPLS	TEPLS-CV
Square	I	9821.8 (51.3)	5337.5 (281.9)	11.5 (0.1)	11.5 (0.1)
	II	107.4 (0.5)	131.1 (4.0)	22.0 (0.1)	22.0 (0.1)
	III	$>10^5$	$>10^4$	23.0 (0.1)	23.0 (0.1)
Cross	I	7306.9 (36.5)	3375.6 (192.2)	11.2 (0.1)	11.3 (0.1)
	II	83.9 (0.4)	103.2 (2.8)	16.5 (0.1)	16.5 (0.1)
	III	$>10^6$	$>10^5$	17.0 (0.1)	17.0 (0.1)
Disk	I	8316.7 (43.3)	3289.4 (115.6)	18.8 (0.1)	19.0 (0.1)
	II	102.0 (0.5)	123.9 (3.0)	20.8 (0.1)	20.8 (0.1)
	III	$>10^6$	$>10^5$	21.0 (0.1)	21.0 (0.1)
Triangle	I	5895.0 (33.1)	2075.7 (47.9)	15.0 (0.1)	15.2 (0.1)
	II	82.4 (0.5)	97.7 (3.0)	15.9 (0.1)	16.0 (0.1)
	III	$>10^5$	$>10^4$	16.0 (0.1)	16.0 (0.1)
Butterfly	I	5376.7 (28.2)	1741.2 (46.6)	20.7 (0.1)	20.9 (0.1)
	II	103.9 (0.6)	126.3 (3.2)	21.0 (0.1)	21.2 (0.1)
	III	$>10^5$	$>10^4$	21.0 (0.1)	21.0 (0.1)

$\boldsymbol{\varepsilon} \in \mathbb{R}^5$ follows a multivariate normal distribution with mean zero and covariance matrix $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} > 0$, where the elements of $\boldsymbol{A} \in \mathbb{R}^{5\times 5}$ are standard uniform. The covariance matrix of $\mathrm{vec}(\boldsymbol{X})$ is $\boldsymbol{\Sigma}_{\boldsymbol{X}} = \boldsymbol{\Sigma}_3 \otimes \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1$, and again we consider three model settings similar as those in Section 5.1 except we match the dimensions of those components accordingly. We compare the three estimators. Note that, tensor envelope PLS naturally works for a multivariate response vector, whereas for OLS and CP, we fit a model for one response variable at a time. The sample size for training and testing data is $n = 200$. Table 2 summarizes the prediction and estimation results based on 100 data replications. It is clearly seen again that the proposed tensor envelope PLS method is more competitive than the alternative solutions in terms of both prediction and estimation accuracy across all model scenarios. More simulations with a 3-way tensor predictor are reported in the supplement.

5.3 Consistency of Tensor Envelope PLS

We next investigate the consistency of our tensor envelope PLS numerically, and also compare to the two alternative solutions graphically. For that purpose, we reconsider the data generating Model-I in Section 5.1. Figure 1 presents a snapshot (a single replication) of the estimated signal by various methods for the signal shapes of a low rank structure, that is, square and cross (ranks 2 and 3). Figure 2 shows the signal shapes with a relatively high rank, that is, disk, triangle, and butterfly ((numerical rank 9, 14, and 30). Three rows of each signal correspond to the varying sample size $n = 200$, $n = 800$, and $n = 1600$, respectively. It is seen that, as the sample increases, all estimators improve, and to some extent, showing the consistency of all three methods. On the other hand, the CP regression seems to work best when the true signal rank is low, or the sample size is fairly large, whereas the proposed tensor envelope PLS works competitively across signals with different ranks, and under both moderate and large sample sizes.

Table 2. Multivariate responses and three-way predictor. Performance under various scenarios and comparison of estimators. OLS, CP, tensor envelope PLS with true and estimated envelope dimensions. Reported are the average and standard error (in parentheses) of the prediction mean squared error evaluated on an independent testing data, and the estimation error, all based on 100 data replications.

	Prediction			
Model	OLS	CP	TEPLS	TEPLS-CV
I	545.0 (7.1)	58.0 (0.7)	10.5 (0.1)	10.9 (0.1)
II	358.5 (4.3)	46.8 (0.6)	8.6 (0.1)	8.6 (0.1)
III	369.6 (4.0)	45.0 (0.5)	8.3 (0.1)	8.3 (0.1)
	Estimation			
Model	OLS	CP	TEPLS	TEPLS-CV
I	$>10^4$	$>10^3$	2.3 (0.1)	2.5 (0.1)
II	179.5 (0.8)	58.7 (0.4)	3.5 (0.1)	3.5 (0.1)
III	$>10^6$	$>10^5$	3.4 (0.1)	3.4 (0.1)

6. Neuroimaging Data Analysis

We analyzed two real neuroimaging datasets to illustrate the versatility of the new method. In both applications, the response is binary. Since the normality of the error term is not assumed or necessary for our proposed method, it is applicable to binary response. Specifically, we employed tensor envelope PLS to obtain the reduced dimensional latent variables, then applied linear discriminant analysis (LDA) on those variables. LDA has been a popular tool for discriminant analysis and has recently received revived attention (Mai and Zou 2013). In addition, PLS and LDA have been jointly applied in numerous genomics and chemometrics applications (Huang and Pan 2003; Barker and Rayens 2003; Boulesteix and Strimmer 2007). Our evaluation criterion is the misclassification error based on cross-validation or an independent testing data. We did not report the estimated tensor coefficient, because the main goals of PLS are dimension reduction and prediction. Additional Lasso-type regularization may be introduced into the current framework of tensor envelope PLS to identify disease relevant brain subregions, but it is beyond the scope of this article and will be our future work.

6.1 EEG Data

The first data contain electroencephalography (EEG) recordings for an alcoholism study. It includes 77 alcoholic individuals and 45 normal controls, and can be obtained from *https://archive.ics.uci.edu/ml/datasets/EEG+Database*. Each

Table 3. Misclassification error rate based on cross-validation for the EEG data.

	5-fold	10-fold	20-fold	Leave-one-out
Tensor envelope PLS	0.197	0.189	0.189	0.205
Regularized CP	0.279	0.271	0.295	0.230
Vectorized Lasso	0.262	0.246	0.246	0.238

individual performed 120 trials under three types of stimuli, and was measured with 64 electrodes placed on the scalp sampled at 256 Hz for one second. More information about data collection can be found in Zhang et al. (1995). We followed the analysis in Li, Kim, and Altman (2010) by focusing on the average of all trials under a single stimulus condition for each subject. The resulting predictor is a 64×256 matrix, and the response is a binary scalar indicating the alcoholic status.

We applied tensor envelope PLS to these data, obtained the envelope dimensions $u_1 = 2$ and $u_2 = 2$ by five-fold cross-validation. We then applied LDA on the reduced then vectorized $2 \times 2 = 4$ latent covariates. We report the misclassification error rate of leave-one-out, 5-fold, 10-fold, and 20-fold cross-validations in Table 3. As a comparison, we also included the two alternatives, OLS and CP. But given the fairly small-sample size of this dataset, we embedded both with Lasso regularization. That is, for OLS, we vectorized the matrix predictor then applied the usual Lasso; for CP, we used the Lasso regularized version of CP tensor regression of Zhou, Li, and Zhu (2013). From the table, we see that tensor envelope PLS achieved an improved classification accuracy compared to the two alternatives. In addition, Li, Kim, and Altman (2010) applied their proposed dimension folding then quadratic discriminant analysis to the same dataset, and reported a leave-one-out misclassification error rate 0.205, the same as our PLS result. However, their method required a preprocessing step to first reduce the 64×256 matrix to a smaller scale such as 15×15 or 9×9 before performing their dimension reduction. By contrast, our tensor envelope PLS can be directly applied to the 64×256 matrix.

6.2 ADHD Data

The second data contain magnetic resonance imaging (MRI) for a study of attention deficit hyperactivity disorder (ADHD). It was produced by the ADHD-200 Sample Initiative, then preprocessed by the Neuro Bureau and made available at *http://neurobureau.projects.nitrc.org/ADHD200/Data.html*. The data was pre-divided into a training set of 776 subjects, among whom 285 are combined ADHD subjects and 491 are normal controls, and a testing set of 169 subjects, among whom 76 ADHD subjects and 93 controls. T1-weighted images were acquired for each subject and were preprocessed by standard steps. We removed 47 subjects in the training data due to missing values or poor image quality, then downsized the MRI images from $256 \times 198 \times 256$ to $30 \times 36 \times 30$, which is to serve as our three-way tensor predictor. The age and gender of each subject were also included as two additional predictors. The response is the binary ADHD status indicator.

We applied tensor envelope PLS and obtained the envelope dimensions $(u_1, u_2, u_3) = (3, 3, 3)$ by five-fold cross-validation. This results in $3 \times 3 \times 3 + 2 = 29$ latent covariates, on which we applied LDA. We then evaluated our classifier on the independent testing data and obtained the classification error rate 34.91%. We also applied Lasso LDA to the vectorized predictors and the Lasso regularized CP regression with rank three, resulting an error rate of 39.05% and 42.60%, respectively. Again, we see that tensor envelope PLS achieves both substantial dimensional reduction and a competitive prediction accuracy.

7. Discussion

In this article, we have proposed a new PLS algorithm for regression with a tensor predictor. We have developed a population interpretation and established statistical properties of the PLS estimators. Numerical analyses have demonstrated the efficacy of the new method. We make a few remarks regarding the new method. First, having a population interpretation for the proposed PLS solution offers insights on why and when the method works, and under what situation the method is not suitable, and such insights are practically useful. It also leads to a better understanding of its connection with other dimension reduction methods and a quantification of the asymptotic behavior of the estimator. Second, as we have found out in the population model development, the PLS method uses, implicitly, a generalized sparsity principle. By exploring the covariance structure, it essentially removes the immaterial information and focuses the estimation on the material part. This is related to but also different from the sparsity principle commonly used in the variable selection literature. As shown in the simulations, the true signal does not have to be exactly sparse. Our method works reasonably well even when many elements of the coefficient signal are small. Third, the PLS method works best when the ranks or envelope dimensions $\{u_k\}_{k=1}^m$ are small, or in other words, when there is substantial immaterial information to reduce. In practice, the true signal is rarely of an exact low dimension, and our method essentially offers an approximation of the truth, which is governed by the usual bias-variance tradeoff. Fourth, similar to most existing PLS solutions, our PLS method demonstrates a competitive performance in terms of prediction of future response variables. Our asymptotic analysis also shows that it offers a consistent estimator of the true regression parameter of interest. However, our experience has suggested that, to obtain a good visual recovery of the coefficient, either a strong signal or a relatively large-sample size is required. For that reason, our performance evaluation in real-data analysis has primarily focused on prediction. Finally, our numerical experiments have shown the clear advantage of our proposal compared to some existing solutions, and thus it offers a useful addition to the currently growing literature on tensor regressions.

Our work also points to a number of potential extensions. Cook, Forzani, and Zhang (2015) recently proposed a hybrid estimator that combines the envelope based generalized sparsity principle with the reduced rank principle. This suggests a possible combination of our tensor envelope PLS with the reduced rank based tensor predictor regression of Zhou, Li, and Zhu (2013). Moreover, our current solution focuses envelope reduction on the predictor side only. One may consider simultaneous envelope reduction on both the response and predictor for additional efficiency gain, following a similar idea as Cook and Zhang (2015). Finally, selecting optimal number of PLS latent components is a crucial and challenging problem. We adopt the same selection strategy as in the usual PLS literature by cross-validation. However, asymptotic properties of such

selection remain untapped and warrant additional investigation. We plan to pursue these lines of work as our future research.

Supplementary Materials

Proofs and additional simulations: Technical proofs and additional simulations are provided in the online supplement to this article.
Computer code and data: The Matlab code and the real datasets can be downloaded from the first author's website.

Acknowledgments

The authors thank the Editor, the Associate Editor, and two referees for their constructive comments. Zhang's research was supported in part by NSF grants DMS-1613154 and CCF-1617691. Li's research was supported in part by NSF grants DMS-1310319 and DMS-1613137.

References

Ahn, M., Shen, H., Lin, W., and Zhu, H. (2015), "A Sparse Reduced Rank Framework for Group Analysis of Functional Neuroimaging Data," *Statistica Sinica*, 25, 295–312. [426]

Barker, M., and Rayens, W. (2003), "Partial Least Squares for Discrimination," *Journal of Chemometrics*, 17, 166–173. [434]

Boulesteix, A.-L., and Strimmer, K. (2007), "Partial Least Squares: A Versatile Tool for the Analysis of High-Dimensional Genomic Data," *Briefings in Bioinformatics*, 8, 32–44. [434]

Bro, R. (1996), "Multiway Calibration. Multilinear Pls," *Journal of Chemometrics*, 10, 47–61. [426,428]

Calhoun, V. D., Liu, J., and Adalı, T. (2009), "A Review of Group ICA for fMRI Data and ICA for Joint Inference of Imaging, Genetic, and ERP Data," *NeuroImage*, 45(1), S163–S172. [426]

Cook, R. D. (1998), *Regression Graphics,* Wiley Series in Probability and Statistics: Probability and Statistics. Ideas for Studying Regressions through Graphics, New York: Wiley. [426,430,431]

Cook, R. D., Forzani, L., and Zhang, X. (2015), "Envelopes and Reduced-Rank Regression," *Biometrika*, 102, 439–456. [435]

Cook, R. D., Helland, I. S., and Su, Z. (2013), "Envelopes and Partial Least Squares Regression," *Journal of the Royal Statistical Society*, Series B, 75, 851–877. [426,427,428,431]

Cook, R. D., Li, B., and Chiaromonte, F. (2010), "Envelope Models for Parsimonious and Efficient Multivariate Linear Regression," *Statistica Sinica*, 20, 927–960. [426,430,431]

Cook, R. D., and Zhang, X. (2015), "Simultaneous Envelopes for Multivariate Linear Regression," *Technometrics*, 57, 11–25. [435]

de Jong, S. (1993), "Simpls: An Alternative Approach to Partial Least 925 Squares Regression," *Chemometrics and Intelligent Laboratory Systems*, 18, 1188–1201. [427]

Ding, S., and Cook, R. D. (2015), "Tensor Sliced Inverse Regression," *Journal of Multivariate Analysis*, 133, 216–231. [427]

Eliseyev, A., and Aksenova, T. (2013), "Recursive *n*-way Partial Least Squares for Brain-Computer Interface," *PloS One*, 8, e69962. [426,428]

Fosdick, B. K., and Hoff, P. D. (2014), "Separable Factor Analysis with Applications to Mortality Data," *The Annals of Applied Statistics*, 8, 120–147. [429]

Frank, L. E., and Friedman, J. H. (1993), "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109–135. [426]

Goldsmith, J., Huang, L., and Crainiceanu, C. (2014), "Smooth Scalar-on-Image Regression via Spatial Bayesian Variable Selection," *Journal of Computational and Graphical Statistics*, 23, 46–64. [427]

Guo, R., Ahn, M., and Zhu, H. (2015), "Spatially Weighted Principal Component Analysis for Imaging Classification," *Journal of Computational and Graphical Statistics*, 24, 274–296. [426]

Helland, I. S. (1988), "On the Structure of Partial Least Squares Regression," *Communications in Statistics: Simulation and Computation*, 17, 581–607. [426,428]

——— (1992), "Maximum Likelihood Regression on Relevant Components," *Journal of the Royal Statistical Society*, Series B, 54, 637–647. [426]

——— (2001), "Some Theoretical Aspects of Partial Least Squares Regression," *Chemometrics and Intelligent Laboratory Systems*, 58, 97–107. [426]

Hoff, P. D. (2011), "Separable Covariance Arrays via the Tucker Product, with Applications to Multivariate Relational Data," *Bayesian Analytics*, 6, 179–196. [429]

Huang, X., and Pan, W. (2003), "Linear Regression and Two-Class Classification with Gene Expression Data," *Bioinformatics*, 19, 2072–2078. [434]

Kolda, T. G., and Bader, B. W. (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500. [427,432]

Li, B., Kim, M. K., and Altman, N. (2010), "On Dimension Folding of Matrix-Or array-Valued Statistical Objects," *The Annals of Statistics*, 38, 1094–1121. [435]

Li, B., and Wang, S. (2007), "On Directional Regression for Dimension Reduction," *Journal of the American Statistical Association*, 102, 997–1008. [426,430,431]

Li, L., Cook, R. D., and Tsai, C.-L. (2007), "Partial Inverse Regression," *Biometrika*, 94, 615–625. [426]

Li, L., and Zhang, X. (2016), "Parsimonious Tensor Response Regression," *Journal of the American Statistical Association*, accepted, DOI: 10.1080/01621459.2016.1193022. [429,430]

Mai, Q., and Zou, H. (2013), "A note on the Connection and Equivalence of Three Sparse Linear Discriminant Analysis Methods," *Technometrics*, 55, 243–246. [434]

Manceur, A. M., and Dutilleul, P. (2013), "Maximum Likelihood Estimation for the Tensor Normal Distribution: Algorithm, Minimum Sample Size, and Empirical Bias and Dispersion," *Journal of Computational and Applied Mathematics*, 239, 37–49. [429]

Martens, H., and Næs, T. (1989), *Multivariate Calibration,* Chichester: Wiley. [426]

Næs, T., and Helland, I. S. (1993), "Relevant Components in Regression," *Scandinavian Journal of Statistics*, 20, 239–250. [426]

Naik, P. A., and Tsai, C.-L. (2005), "Constrained Inverse Regression for Incorporating Prior Information," *Journal of the American Statistical Association*, 100, 204–211. [426]

Reiss, P., and Ogden, R. (2010), "Functional Generalized Linear Models with Images as Predictors," *Biometrics*, 66, 61–69. [427]

Schott, J. R. (2013), "On the Likelihood Ratio Test for Envelope Models in Multivariate Linear Regression," *Biometrika*, 100, 531–537. [431]

Sun, Q., Zhu, H., Liu, Y., and Ibrahim, J. G. (2015), "SPReM: Sparse Projection Regression Model for High-Dimensional Linear Regression," *Journal of the American Statistical Association*, 110, 289–302. [427]

Wang, X., Nan, B., Zhu, J., and Koeppe, R. (2014), "Regularized 3D Functional Regression for Brain Image Data via Haar Wavelets," *The Annals of Applied Statistics*, 8, 1045–1064. [427]

Wold, S. (1966), "Estimation of Principal Components and Related Models by Iterative Models Iterative Least Squares," in *Multivariate Analysis*, New York: Academic Press, pp. 391–420. [427]

Zhang, X., Begleiter, H., Porjesz, B., Wang, W., and Litke, A. (1995), "Event Related Potentials During Object Recognition Tasks," *Brain Research Bulletin*, 38, 531–538. [435]

Zhao, J., and Leng, C. (2014), "Structured Lasso for Regression with Matrix Covariates," *Statistica Sinica*, 24, 799–814. [427]

Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., Zhang, L., and Cichocki, A. (2013), "Higher Order Partial Least Squares (hopls): A Generalized Multilinear Regression Method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35, 1660–1673. [426,428]

Zhong, W., Xing, X., and Suslick, K. (2015), "Tensor Sufficient Dimension Reduction," *Wiley Interdisciplinary Reviews: Computational Statistics*, 7, 178–184. [427]

Zhou, H., and Li, L. (2014), "Regularized Matrix Regression," *Journal of the Royal Statistical Society*, Series B, 76, 463–483. [427]

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. [427,431,432,435]

Zhu, H., Fan, J., and Kong, L. (2014), "Spatially Varying Coefficient model for Neuroimaging Data with Jump Discontinuities," *Journal of the American Statistical Association*, 109, 1084–1098. [427]