Stat Biosci (2018) 10:520–545
https://doi.org/10.1007/s12561-018-9215-6

Tucker Tensor Regression and Neuroimaging Analysis

Xiaoshan Li[1] · **Da Xu**[3] · **Hua Zhou**[2] · **Lexin Li**[3]

Received: 11 October 2016 / Accepted: 28 February 2018 / Published online: 7 March 2018

Abstract Neuroimaging data often take the form of high-dimensional arrays, also known as tensors. Addressing scientific questions arising from such data demands new regression models that take multidimensional arrays as covariates. Simply turning an image array into a vector would both cause extremely high dimensionality and destroy the inherent spatial structure of the array. In a recent work, Zhou et al. (J Am Stat Assoc, 108(502):540–552, 2013) proposed a family of generalized linear tensor regression models based upon the CP (CANDECOMP/PARAFAC) decomposition of regression coefficient array. Low-rank approximation brings the ultrahigh dimensionality to a manageable level and leads to efficient estimation. In this article, we propose a tensor regression model based on the more flexible Tucker decomposition. Compared to the CP model, Tucker regression model allows different number of factors along each mode. Such flexibility leads to several advantages that are particularly suited to neuroimaging analysis, including further reduction of the number of free parameters, accommodation of images with skewed dimensions, explicit modeling of interactions, and a principled way of image downsizing. We also compare the Tucker model with CP numerically on both simulated data and real magnetic resonance imaging data, and demonstrate its effectiveness in finite sample performance.

Keywords CP decomposition · Magnetic resonance image · Multidimensional array · Tucker decomposition · Tensor regression

✉ Lexin Li
lexinli@berkeley.edu

1 Wells Fargo & Company, Charlotte 28202, USA

2 University of California, Los Angeles, USA

3 Division of Biostatistics, University of California, Berkeley 94720, USA

1 Introduction

Medical imaging routinely produces data in the form of multidimensional array, also known as *tensor*. Examples include electroencephalography (EEG, 2D matrix), anatomical magnetic resonance images (MRI, 3D array), and functional magnetic resonance images (fMRI, 4D array). It is of common scientific interest to identify associations between brain regions and clinical outcomes, to diagnose neurodegenerative disorders, and to predict onset of neuropsychiatric diseases. These problems can be collectively formulated as regression with the clinical outcome as the response, and the image or tensor as the predictor. However, the sheer size and complex structure of the image covariate pose unusual challenges. Most classical regression models take a vector as the predictor. Naively turning an image array into a vector is evidently unsatisfactory. For instance, a typical MRI image of size $128 \times 128 \times 128$ requires $128^3 = 2{,}097{,}152$ regression parameters, which severely compromises the computability and theoretical guarantee of the classical regression models. More seriously, vectorizing an array destroys the inherent spatial structure of the image array that usually possesses rich information.

One class of solutions build a regression model *one-voxel-at-a-time* [8], which totally ignores all spatial correlations among the voxels [14,23]. Another class of typical solutions in the literature first extracts a vector of features from images, either through subject knowledge such as predefined regions of interest, or data-driven approach such as principal component analysis (PCA, [3]). Then the extracted feature vector is fed into a classical regression model. Although such solutions are intuitive, there is no consensus on what choice best summarizes a brain image, whereas unsupervised dimension reduction like principal components could result in potential information loss. More recently, there have been a sequence of developments aiming at regression with a tensor predictor, sometimes also referred as *scalar-on-image regression* ([9,15,17,22,24], among others), which enjoy varying degrees of success and reflect the increasing importance of this family of problems.

In particular, Zhou et al. [25] proposed a class of generalized linear *tensor regression* models. For a response variable Y, a vector predictor $\boldsymbol{Z} \in \mathbb{R}^{p_0}$ and a D-dimensional tensor predictor $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$, the response is assumed to belong to an exponential family where the linear systematic part is of the form,

$$g(\mu) = \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{Z} + \langle \boldsymbol{B}, \boldsymbol{X} \rangle. \tag{1}$$

Here $g(\cdot)$ is a strictly increasing link function, $\mu = E(Y|\boldsymbol{X}, \boldsymbol{Z})$, $\boldsymbol{\gamma} \in \mathbb{R}^{p_0}$ is the regular regression coefficient vector, $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ is the coefficient array that captures the effects of tensor covariate $\boldsymbol{X}$, and the inner product between two arrays is defined as $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \mathrm{vec}\boldsymbol{B}, \mathrm{vec}\boldsymbol{X} \rangle = \sum_{i_1,\ldots,i_D} \beta_{i_1 \ldots i_D} x_{i_1 \ldots i_D}$. This model, if with no further simplification, is prohibitive given its gigantic dimensionality: $p_0 + \prod_{d=1}^{D} p_d$. Zhou et al. [25] introduced a low-rank structure on the coefficient $\boldsymbol{B}$, in that $\boldsymbol{B}$ is assumed to follow a rank-R CANDECOMP/PARAFAC (CP) decomposition [10],

$$\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)}, \tag{2}$$

where $\boldsymbol{\beta}_d^{(r)} \in \mathbb{R}^{p_d}$ are all column vectors, $d = 1, \ldots, D, r = 1, \ldots, R$, and $\circ$ denotes outer product such that $\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D$ of D vectors $\boldsymbol{b}_d \in \mathbb{R}^{p_d}$ forms the $p_1 \times \cdots \times p_D$ array with entries $(\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D)_{i_1 \cdots i_D} = \prod_{d=1}^{D} b_{di_d}$. For convenience, this CP decomposition is often represented by a shorthand $\boldsymbol{B} = [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$, where $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R)}] \in \mathbb{R}^{p_d \times R}$, $d = 1, \ldots, D$. Combining (2) with (1) reduces the ultrahigh dimensionality of the model to a more manageable scale of $p_0 + R \times \sum_{d=1}^{D} p_d$, which in turn results in efficient estimation and prediction. For instance, for regression with a 128-by-128-by-128 MRI image and 5 usual covariates, the dimensionality is reduced from the order of $2,097,157 = 5 + 128^3$ to $389 = 5 + 128 \times 3$ for a rank-1 model, and to $1157 = 5 + 3 \times 128 \times 3$ for a rank-3 model. Zhou et al. [25] showed that this low-rank tensor model could provide a sound recovery of even high-rank signals.

In the tensor literature, there has been an important development parallel to CP decomposition, which is termed Tucker decomposition or higher-order singular value decomposition (HOSVD, [10]). In this article, we propose a class of *Tucker tensor regression models*. To differentiate, we call the models of Zhou et al. [25] *CP tensor regression models*. Specifically, we continue to adopt the model (1), but assume that the coefficient array $\boldsymbol{B}$ follows a Tucker decomposition,

$$\boldsymbol{B} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_D=1}^{R_D} g_{r_1,\ldots,r_D} \boldsymbol{\beta}_1^{(r_1)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r_D)}, \tag{3}$$

where $\boldsymbol{\beta}_d^{(r_d)} \in \mathbb{R}^{p_d}$ are all column vectors, $d = 1, \ldots, D, r_d = 1, \ldots, R_d$, and $g_{r_1,\ldots,r_D}$ are constants. It is often abbreviated as $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$, where $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ is the D-dimensional *core tensor* with entries $(\boldsymbol{G})_{r_1 \ldots r_D} = g_{r_1,\ldots,r_D}$, and $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ are the factor matrices. $\boldsymbol{B}_d$'s are usually orthogonal and can be thought of as the *principal components* in each dimension, and thus the name, HOSVD. The number of parameters of a Tucker tensor model is in the order of $p_0 + \sum_{d=1}^{D} R_d \times p_d + \prod_{d=1}^{D} R_d$. Comparing the two decompositions (2) and (3), the key difference is that CP fixes the number of basis vectors R along each dimension of $\boldsymbol{B}$ so that all $\boldsymbol{B}_d$'s have the *same* number of columns (ranks). In contrast, Tucker allows the number R_d to differ along different dimensions and $\boldsymbol{B}_d$'s could have *different* ranks.

This difference between the two decompositions seems minor; however, in the context of tensor regression modeling and neuroimaging analysis, it has profound implications, which essentially motivates this article. On one hand, the Tucker tensor regression model shares the advantages of the CP model, in that it exploits the special structure of the tensor data, reduces the dimensionality to enable efficient model estimation, and provides a sound low-rank approximation to a potentially high-rank signal. On the other hand, Tucker tensor regression offers a much more *flexible* modeling framework than CP regression, by allowing a distinct order along each dimension. It includes CP model as a special case, where the core tensor $\boldsymbol{G}$ is super-diagonal. This flexibility leads to several improvements that are particularly useful for neuroimaging analysis. First, a Tucker model could be more parsimonious than a CP model thanks to the flexibility of different orders. For instance, suppose a 3D signal $\boldsymbol{B} \in \mathbb{R}^{16 \times 16 \times 16}$ admits a Tucker decomposition (3) with $R_1 = R_2 = 2$ and $R_3 = 5$. It can only be

recovered by a CP decomposition with $R = 5$, costing 230 parameters. In contrast, the Tucker model is more parsimonious with only 131 parameters. This reduction of free parameters is valuable for medical imaging studies, as the sample size is often limited. Second, the freedom in the choice of different orders is useful when the tensor data are skewed in dimensions, which is not uncommon in neuroimaging data. For instance, in EEG, the temporal dimension often far exceeds the spatial dimension. Third, even when all tensor modes have comparable sizes, the Tucker formulation explicitly models the interactions between the factor matrices $\boldsymbol{B}_d$'s, and as such allows a finer grid search within a larger model space, which in turn may explain more trait variance. Finally, as we show in Sect. 2.3, there exists a duality regarding the Tucker tensor model. Thanks to this duality, a Tucker tensor decomposition naturally lends itself to a principled way of imaging data downsizing, which again can be practically very useful.

For these reasons, we feel it important to develop a complete methodology of Tucker tensor regression and its associated theory. The resulting Tucker tensor model performs dimension reduction through low-rank tensor decomposition but in a supervised fashion, and thus reduces potential information loss in regression. It works for general array-valued image modalities and/or any combination of them, and for various types of responses, including continuous, binary, and count data. Besides, a highly scalable algorithm is developed for the associated maximum likelihood estimation, where scalability is crucial considering the massive size of imaging data. In addition, regularization is studied in conjunction with the proposed model, yielding a collection of regularized Tucker tensor models. Particularly, the one with a lasso penalty encourages sparsity of the core tensor, improves interpretability of the model, and is of vital scientific interest.

The rest of the article is organized as follows. Section 2 presents the Tucker tensor regression model, and Sect. 3 develops the maximum likelihood and regularized estimation. Section 4 derives the theoretical properties including consistency and asymptotic normality. Section 5 presents the numerical results, and Sect. 6 concludes with a discussion. All technical proofs are delegated to Appendix.

2 Model

2.1 Notation

We begin with a brief review of some key array notations and operations. An extensive reference can be found in the survey paper [10]. A *tensor* is a multidimensional array. *Fiber* of a tensor is defined by fixing every index but one, and is the higher-order analogue of matrix row and column. The *vec operator* stacks the entries of a D-dimensional tensor $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ into a column vector, such that an entry $b_{i_1 \ldots i_D}$ maps to the j-th entry of $\operatorname{vec} \boldsymbol{B}$ where $j = 1 + \sum_{d=1}^{D} (i_d - 1) \prod_{d'=1}^{d-1} p_{d'}$. The *mode-d matricization*, $\boldsymbol{B}_{(d)}$, maps a tensor $\boldsymbol{B}$ into a $p_d \times \prod_{d' \neq d} p_{d'}$ matrix, such that the $(i_1, \ldots, i_D)$ element of the array $\boldsymbol{B}$ maps to the (i_d, j) element of the matrix $\boldsymbol{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d} (i_{d'} - 1) \prod_{d'' < d', d'' \neq d} p_{d''}$. When $D = 1$, we observe that $\operatorname{vec} \boldsymbol{B}$ is the same as vectorizing the mode-1 matricization $\boldsymbol{B}_{(1)}$. The *mode-(d, d') matricization* $\boldsymbol{B}_{(dd')} \in \mathbb{R}^{p_d p_{d'} \times \prod_{d'' \neq d, d'} p_{d''}}$ is defined in a similar fashion. The *mode-d multiplication*

of the tensor $\boldsymbol{B}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{p_d \times q}$, denoted by $\boldsymbol{B} \times_d \boldsymbol{U} \in \mathbb{R}^{p_1 \times \cdots \times q \times \cdots \times p_D}$, is the multiplication of the mode-d fibers of $\boldsymbol{B}$ by $\boldsymbol{U}$. In other words, the mode-d matricization of $\boldsymbol{B} \times_d \boldsymbol{U}$ is $\boldsymbol{U}\boldsymbol{B}_{(d)}$. For a tensor $\boldsymbol{B}$ that admits Tucker decomposition (3), its mode-d matricization can be expresses as

$$\boldsymbol{B}_{(d)} = \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^{\mathsf{T}},$$

where $\otimes$ denotes the Kronecker product of matrices, and

$$\mathrm{vec}\boldsymbol{B} = \mathrm{vec}\boldsymbol{B}_{(1)} = \mathrm{vec}(\boldsymbol{B}_1 \boldsymbol{G}_{(1)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_2)^{\mathsf{T}}) = (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)\mathrm{vec}\boldsymbol{G}.$$

2.2 Tucker Regression Model

We elaborate on the Tucker tensor regression model introduced in Sect. 1. We assume that Y belongs to an exponential family with probability mass function or density [16],

$$p(y_i|\theta_i, \phi) = \exp\left\{ \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right\}$$

with the first two moments $E(Y_i) = \mu_i = b'(\theta_i)$ and $\mathrm{Var}(Y_i) = \sigma_i^2 = b''(\theta_i)a_i(\phi)$. θ and $\phi > 0$ are, respectively, the natural and dispersion parameters. We assume the systematic part of GLM is of the form,

$$g(\mu) = \eta = \boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{Z} + \left\langle \sum_{r_1=1}^{R_1} \cdots \sum_{r_D=1}^{R_D} g_{r_1,\ldots,r_D} \boldsymbol{\beta}_1^{(r_1)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r_D)}, \boldsymbol{X} \right\rangle. \tag{4}$$

That is, we impose a Tucker structure on the array coefficient $\boldsymbol{B}$. We make a few remarks. First, in this article, we consider the problem of estimating the core tensor $\boldsymbol{G}$ and the factor matrices $\boldsymbol{B}_d$ simultaneously given the response Y and covariates $\boldsymbol{X}$ and $\boldsymbol{Z}$. This is a *supervised* version of the classical unsupervised Tucker decomposition [20]. It can also be viewed as a supervised version of principal component analysis for higher-order multidimensional array. Unlike a two-stage solution that first performs principal component analysis and then fits a regression model, the basis (principal components) $\boldsymbol{B}_d$ in our models are estimated under the guidance (supervision) of the response variable. Second, the CP model of Zhou et al. [25] corresponds to a special case of the Tucker model (4) with $g_{r_1,\ldots,r_D} = 1_{\{r_1=\cdots=r_D\}}$ and $R_1 = \ldots = R_D = R$. In other words, the CP model is a special Tucker model with a super-diagonal core tensor $\boldsymbol{G}$. The CP model has a rank at most R, while the general Tucker model can have a rank as high as R^D.

2.3 Duality and Tensor Basis Pursuit

The next result concerns the inner product between a general tensor and a tensor that admits a Tucker decomposition.

Lemma 1 (Duality) *Suppose a tensor* $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ *admits Tucker decomposition* $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$. *Then, for any tensor* $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$, $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \boldsymbol{G}, \tilde{\boldsymbol{X}} \rangle$, *where* $\tilde{\boldsymbol{X}}$ *admits a Tucker decomposition* $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^{\mathrm{T}}, \ldots, \boldsymbol{B}_D^{\mathrm{T}}]\!]$.

This duality gives some important insights to the Tucker tensor regression model. First, if we consider $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ as fixed and known basis matrices, then Lemma 1 says fitting the Tucker tensor regression model (4) is equivalent to fitting a tensor regression model in $\boldsymbol{G}$ with the *transformed* data $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^{\mathrm{T}}, \ldots, \boldsymbol{B}_D^{\mathrm{T}}]\!] \in \mathbb{R}^{R_1 \times \cdots \times R_D}$. When $R_d \ll p_d$, the transformed data $\tilde{\boldsymbol{X}}$ effectively *downsize* the original data. We further illustrate this downsizing feature in the real data analysis in Sect. 5.4. Second, in applications where the numbers of basis vectors R_d are unknown, we can utilize possibly over-complete basis matrices $\boldsymbol{B}_d$ such that $R_d \geq p_d$, and then estimate $\boldsymbol{G}$ with sparsity regularization. This leads to a tensor version of the classical basis pursuit problem [4]. Take fMRI data as the examples. We can adopt the wavelet basis for the three image dimensions and the Fourier basis for the time dimension. Regularization on $\boldsymbol{G}$ can be achieved by either imposing a low rank decomposition (CP or Tucker) on $\boldsymbol{G}$ (hard thresholding) or penalized regression (soft thresholding). We investigate Tucker regression regularization in details in Sect. 3.3.

2.4 Model Size: Tucker Versus CP

In this section we study the size of the Tucker tensor model. Comparison with the size of the CP model helps gain a better understanding of both models. In addition, it provides a basis for data-adaptive selection of appropriate orders in a Tucker model.

First we quickly review the number of free parameters p_{C} for a CP model $\boldsymbol{B} = [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d]\!]$, with $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R}$. For $D = 2$, $p_{\mathrm{C}} = R(p_1 + p_2) - R^2$, and for $D > 2$, $p_{\mathrm{C}} = R(\sum_{d=1}^{D} p_d - D + 1)$. For $D = 2$, the term $-R^2$ adjusts for the nonsingular transformation indeterminacy for model identifiability; for $D > 2$, the term $R(-D + 1)$ adjusts for the scaling indeterminacy in the CP decomposition. See Zhou et al. [25] for more details. Following similar arguments, we obtain that the number of free parameters p_{T} in a Tucker model $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_d]\!]$, with $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_d}$ and $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$, is

$$p_{\mathrm{T}} = \sum_{d=1}^{D} p_d R_d + \prod_{d=1}^{D} R_d - \sum_{d=1}^{D} R_d^2,$$

for any D. Here the term $-\sum_{d=1}^{D} R_d^2$ adjusts for the nonsingular transformation indeterminacy in the Tucker decomposition. We summarize these results in Table 1.

Next we compare the two model sizes (degrees of freedom) under an additional assumption that $R_1 = \cdots = R_d = R$. Now the difference becomes

$$p_{\mathrm{T}} - p_{\mathrm{C}} = \begin{cases} 0 & \text{when } D = 2, \\ R(R-1)(R-2) & \text{when } D = 3, \\ R(R^3 - 4R + 3) & \text{when } D = 4, \\ R(R^{D-1} - DR + D - 1) & \text{when } D > 4. \end{cases}$$

Table 1 Number of free parameters in Tucker and CP models

	CP	Tucker
$D = 2$	$R(p_1 + p_2) - R^2$	$p_1 R_1 + p_2 R_2 + R_1 R_2 - R_1^2 - R_2^2$
$D > 2$	$R(\sum_d p_d - D + 1)$	$\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2$

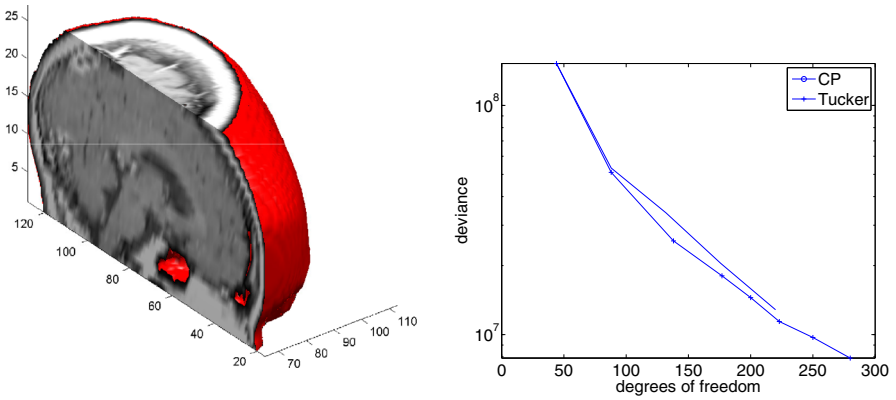

Fig. 1 Left: half of the true signal array $\boldsymbol{B}$. Right: deviances of CP regression estimates at $R = 1, \ldots, 5$, and Tucker regression estimates at orders $(R_1, R_2, R_3) = (1, 1, 1)$, $(2, 2, 2)$, $(3, 3, 3)$, $(4, 4, 3)$, $(4, 4, 4)$, $(5, 4, 4)$, $(5, 5, 4)$, and $(5, 5, 5)$. The sample size is $n = 1000$

When $D = 2$, the Tucker model is essentially the same as the CP model. When $D = 3$, Tucker has the same number of parameters as CP for $R = 1$ or $R = 2$, but costs $R(R-1)(R-2)$ more parameters for $R > 2$. When $D > 3$, Tucker and CP are the same for $R = 1$, but Tucker costs substantially more parameters than CP for $R > 2$. However, this comparison assumes $R_1 = \cdots = R_d = R$. In reality, Tucker could require *fewer* free parameters than CP, as shown in the illustrative example given in Sect. 1, since Tucker is more flexible and allows a different order R_d along each dimension.

Figure 1 shows an example with a $D = 3$-dimensional array covariate. Half of the true signal (brain activity map) $\boldsymbol{B}$ is displayed in the left panel, which is by no means a low-rank signal. Suppose 3D images $\boldsymbol{X}_i$ are taken on $n = 1000$ subjects. We simulate image traits $\boldsymbol{X}_i$ from an independent standard normal distribution and quantitative traits Y_i from an independent normal with the mean $\langle \boldsymbol{X}_i, \boldsymbol{B} \rangle$ and the unit variance. Given the limited sample size, the hope is to infer a reasonable low-rank approximation to the activity map from the 3D image covariate. The right panel displays the model deviance versus the degrees of freedom of a series of CP and Tucker model estimates. The CP model is estimated at ranks $R = 1, \ldots, 5$, respectively. The Tucker model is fitted at orders $(R_1, R_2, R_3) = (1, 1, 1)$, $(2, 2, 2)$, $(3, 3, 3)$, $(4, 4, 3)$, $(4, 4, 4)$, $(5, 4, 4)$, $(5, 5, 4)$, and $(5, 5, 5)$. We see from the plot that, under the same number of free parameters, the Tucker model could generally achieve a better model fit with a smaller deviance. Note that the deviance is in the log scale, so a small discrepancy between the two lines translates to a large value of difference in deviance.

Algorithm 1 Block relaxation algorithm for fitting the Tucker tensor regression.

Initialize: $\boldsymbol{\gamma}^{(0)} = \text{argmax}_{\boldsymbol{\gamma}}\, \ell(\boldsymbol{\gamma}, \mathbf{0}, \ldots, \mathbf{0})$, $\boldsymbol{B}_d^{(0)} \in \mathbb{R}^{p_d \times R_d}$ a random matrix for $d = 1, \ldots, D$, and $\boldsymbol{G}^{(0)} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ a random matrix.
repeat
  for $d = 1, \ldots, D$ **do**
    $\boldsymbol{B}_d^{(t+1)} = \text{argmax}_{\boldsymbol{B}_d}\, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_{d-1}^{(t+1)}, \boldsymbol{B}_d, \boldsymbol{B}_{d+1}^{(t)}, \ldots, \boldsymbol{B}_D^{(t)}, \boldsymbol{G}^{(t)})$
  end for
  $\boldsymbol{G}^{(t+1)} = \text{argmax}_{\boldsymbol{G}}\, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G})$
  $\boldsymbol{\gamma}^{(t+1)} = \text{argmax}_{\boldsymbol{\gamma}}\, \ell(\boldsymbol{\gamma}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G}^{(t+1)})$
until $\ell(\boldsymbol{\theta}^{(t+1)}) - \ell(\boldsymbol{\theta}^{(t)}) < \epsilon$

3 Estimation and Regularization

3.1 Maximum Likelihood Estimation

We pursue the maximum likelihood estimation (MLE) for the Tucker tensor regression model and develop a scalable estimation algorithm in this section. Finding the MLE is difficult due to nonconcavity. However, we make a key observation that, although the systematic part (4) is not linear in $\boldsymbol{G}$ and $\boldsymbol{B}_d$ *jointly*, it is linear in them *separately*. This naturally suggests a block ascent algorithm, which updates each factor matrix $\boldsymbol{B}_d$ and the core tensor $\boldsymbol{G}$ *alternately*.

The algorithm consists of two core steps. First, when updating $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ with the rest of $\boldsymbol{B}_{d'}$'s and $\boldsymbol{G}$ fixed , we rewrite the array inner product in (4) as

$$\begin{aligned}\langle \boldsymbol{B}, \boldsymbol{X} \rangle &= \langle \boldsymbol{B}_{(d)}, \boldsymbol{X}_{(d)} \rangle \\ &= \langle \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^\top, \boldsymbol{X}_{(d)} \rangle \\ &= \langle \boldsymbol{B}_d, \boldsymbol{X}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1) \boldsymbol{G}_{(d)}^\top \rangle.\end{aligned}$$

Then the problem turns into a GLM regression with $\boldsymbol{B}_d$ as the "parameter" and the term $\boldsymbol{X}_{(d)}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(d)}^\top$ as the "predictor". It is a low-dimensional GLM with only $p_d R_d$ parameters and thus is easy to solve. Second, when updating $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ with all $\boldsymbol{B}_d$'s fixed,

$$\begin{aligned}\langle \boldsymbol{B}, \boldsymbol{X} \rangle &= \langle \text{vec}\boldsymbol{B}, \text{vec}\boldsymbol{X} \rangle \\ &= \langle (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1) \text{vec}\boldsymbol{G}, \text{vec}\boldsymbol{X} \rangle \\ &= \langle \text{vec}\boldsymbol{G}, (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)^\top \text{vec}\boldsymbol{X} \rangle.\end{aligned}$$

This implies a GLM regression with $\text{vec}\boldsymbol{G}$ as the "parameter" and the term $(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)^\top \text{vec}\boldsymbol{X}$ as the "predictor". Again this is a low-dimensional regression problem with $\prod_d R_d$ parameters. For completeness, we summarize the above alternating estimation procedure in Algorithm 1. The orthogonality between the columns of the factor matrices $\boldsymbol{B}_d$ is not enforced as in unsupervised HOSVD, because our primary goal is approximating tensor signal instead of finding the principal components along each mode.

As the block ascent algorithm monotonically increases the objective value, the stopping criterion is well defined and the convergence properties of iterates follow from the standard theory for monotone algorithms [5,12]. The proof of the next result is given in Appendix.

Proposition 1 *Assume (i) ℓ is coercive, i.e., the set $\{\boldsymbol{\theta} : \ell(\boldsymbol{\theta}) \geq \ell(\boldsymbol{\theta}^{(0)})\}$ is compact, and bounded above, (ii) the stationary points (modulo nonsingular transformation indeterminacy) of ℓ are isolated, (iii) the algorithmic mapping is continuous, (iv) $\boldsymbol{\theta}$ is a fixed point of the algorithm if and only if it is a stationarity point of ℓ, and (v) $\ell(\boldsymbol{\theta}^{(t+1)}) \geq \ell(\boldsymbol{\theta}^{(t)})$ with equality if and only if $\theta^{(t)}$ is a fixed point of the algorithm. We have the following results.*

1. *(Global Convergence) The sequence $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\gamma}^{(t)}, \boldsymbol{G}^{(t)}, \boldsymbol{B}_1^{(t)}, \ldots, \boldsymbol{B}_D^{(t)})$ generated by Algorithm 1 converges to a stationary point of $\ell(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$.*
2. *(Local Convergence) Let $\boldsymbol{\theta}^{(\infty)} = (\boldsymbol{\gamma}^{(\infty)}, \boldsymbol{G}^{(\infty)}, \boldsymbol{B}_1^{(\infty)}, \ldots, \boldsymbol{B}_D^{(\infty)})$ be a strict local maximum of ℓ. The iterates generated by Algorithm 1 are locally attracted to $\boldsymbol{\theta}^{(\infty)}$ for $\boldsymbol{\theta}^{(0)}$ sufficiently close to $\boldsymbol{\theta}^{(\infty)}$.*

3.2 Tucker Order Selection

For our Tucker model, one needs to select the order R_d along each direction. One solution is to view this as a model selection problem. Accordingly, we suggest to use Bayesian information criterion (BIC), $-2 \log \ell + \log(n) p_e$. Here ℓ is the log-likelihood, and $p_e = p_{\mathrm{T}}$ is the effective number of parameters of the Tucker model as given in Table 1. We illustrate this BIC criterion in the simulation Sect. 5.1. Alternatively, one may employ the spectral regularization to adaptively select the order of the Tucker decomposition, following a similar idea as [24].

Practically, we have found the following rule useful when selecting the Tucker orders. Specifically, at each step of GLM model fit, we ensure that the ratio between the sample size n and the number of parameters under estimation in that step, $p_d \times R_d$, satisfies a heuristic rule of greater than two in normal models and greater than five in logistic models. Moreover, we also ensure the ratio between n and the number of parameters in the core tensor estimation $\prod_d R_d$ satisfies this rule. Our numerical experiments seem to suggest this is a useful practical guideline, especially when the data are noisy.

3.3 Regularized Estimation

Regularization plays a crucial role in neuroimaging analysis for several reasons. First, even after substantial dimension reduction by imposing a Tucker structure, the number of parameters p_{T} can still exceed the number of observations n. Second, even when $n > p_{\mathrm{T}}$, regularization could potentially be useful for stabilizing the estimates and improving the risk property. Finally, regularization is an effective way to incorporate prior scientific knowledge about brain structures. For instance, the smoothness property of the brain spatial structure may be incorporated via a fused type regularization.

In our context of Tucker regularized regression, there are multiple types of regularizations. In this section, we illustrate the regularization on the core tensor $\boldsymbol{G}$ only. It helps achieve sparsity in the number of outer products in Tucker decomposition (3) and shrinkage, and is useful for tensor basis pursuit as discussed in Sect. 2.3. One can also impose the sparsity regularization on both $\boldsymbol{G}$ and $\boldsymbol{B}_d$ simultaneously, which can help select brain subregions that are highly relevant to the disease outcome. Moreover, one may introduce the spectral regularization to select Tucker orders in a soft-thresholding way [24]. For the sake of space, we leave those alternative regularizations for future research.

Specifically, in this article, we propose to maximize the regularized log-likelihood

$$\ell(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) - \sum_{r_1,\ldots,r_D} P_\eta(|g_{r_1,\ldots,r_D}|, \lambda),$$

where $P_\eta(|x|, \lambda)$ is a scalar penalty function, λ is the penalty tuning parameter, and η is an index for the penalty family. Note that the penalty term above only involves elements of the core tensor, and thus regularization on $\boldsymbol{G}$ only. This formulation includes a large class of penalty functions, including the power family [7], where $P_\eta(|x|, \lambda) = \lambda|x|^\eta$, $\eta \in (0, 2]$, such as lasso [19] ($\eta = 1$) and ridge ($\eta = 2$); elastic net [26], where $P_\eta(|x|, \lambda) = \lambda[(\eta - 1)x^2/2 + (2 - \eta)|x|]$, $\eta \in [1, 2]$; SCAD [6], where $\partial/\partial|x| P_\eta(|x|, \lambda) = \lambda \left\{1_{\{|x|\le\lambda\}} + (\eta\lambda - |x|)_+/(\eta - 1)\lambda 1_{\{|x|>\lambda\}}\right\}$, $\eta > 2$; among many others.

Two aspects of the proposed regularized Tucker regression, parameter estimation and tuning, deserve some discussion. For regularized estimation, it incurs only slight changes in Algorithm 1, i.e., when updating $\boldsymbol{G}$, we fit a penalized GLM regression problem,

$$\boldsymbol{G}^{(t+1)} = \text{argmax}_{\boldsymbol{G}}\, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G}) - \sum_{r_1,\ldots,r_D} P_\eta(|g_{r_1,\ldots,r_D}|, \lambda),$$

for which many software packages exist. Other steps of Algorithm 1 remain unchanged. For the regularization to remain legitimate, we constrain the column norms of $\boldsymbol{B}_d$ to be one when updating factor matrices $\boldsymbol{B}_d$. Moreover one can employ general cross-validation or Bayesian information criterion to tune the penalty parameter λ.

4 Theory

We study the usual large n asymptotics of the proposed Tucker tensor regression. Although the usually limited sample size of neuroimaging studies makes the large n asymptotics seem irrelevant, we feel it is still useful for several reasons. First, when the sample size n is considerably larger than the effective number of parameters p_{T}, the asymptotic study tells us that the model is consistently estimating the best Tucker structure approximation to the full array model in the sense of Kullback–Liebler distance. Second, the explicit formula for the score and information are not only useful for asymptotic theory but also for computation, while the identifiability

issue has to be properly dealt with. Finally, the regular asymptotics can be of practical relevance, for instance, can be useful in a likelihood ratio type test in a replication study.

4.1 Score and Information

We derive the score and information for the tensor regression model, which are essential for statistical estimation and inference. For simplicity, we drop the classical covariate $\boldsymbol{Z}$ in this section, but all the results can be straightforwardly extended to include $\boldsymbol{Z}$. The following standard calculus notations are used. For a scalar function f, ∇f is the (column) gradient vector, $df = [\nabla f]^{\mathrm{T}}$ is the differential, and $d^2 f$ is the Hessian matrix. For a multivariate function $g : \mathbb{R}^p \mapsto \mathbb{R}^q$, $Dg \in \mathbb{R}^{p\times q}$ denotes the Jacobian matrix holding partial derivatives $\frac{\partial g_j}{\partial x_i}$. We start from the Jacobian and Hessian of the systematic part $\eta \equiv g(\mu)$ in (4).

Lemma 2 *1. The gradient* $\nabla\eta(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \in \mathbb{R}^{\prod_d R_d + \sum_{d=1}^D p_d R_d}$ *is*

$$\nabla\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) = [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ \boldsymbol{J}_2 \ \cdots \ \boldsymbol{J}_D]^{\mathrm{T}}(\mathrm{vec}\boldsymbol{X}),$$

where $\boldsymbol{J}_d \in \mathbb{R}^{\prod_{d=1}^D p_d \times p_d R_d}$ *is the Jacobian*

$$\boldsymbol{J}_d = D\boldsymbol{B}(\boldsymbol{B}_d) = \boldsymbol{\Pi}_d\{[(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(d)}^{\mathrm{T}}] \otimes \boldsymbol{I}_{p_d}\} \tag{5}$$

and $\boldsymbol{\Pi}_d$ *is the* $(\prod_{d=1}^D p_d)$*-by-*$(\prod_{d=1}^D p_d)$ *permutation matrix that reorders* $\mathrm{vec}\boldsymbol{B}_{(d)}$ *to obtain* $\mathrm{vec}\boldsymbol{B}$*, i.e.,* $\mathrm{vec}\boldsymbol{B} = \boldsymbol{\Pi}_d\,\mathrm{vec}\boldsymbol{B}_{(d)}$.

2. Let the Hessian $d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \in \mathbb{R}^{(\prod_d R_d + \sum_d p_d R_d)\times(\prod_d R_d + \sum_d p_d R_d)}$ *be partitioned into four blocks* $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{G}} \in \mathbb{R}^{\prod_d R_d \times \prod_d R_d}$, $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}} = \boldsymbol{H}_{\boldsymbol{B},\boldsymbol{G}}^{\mathrm{T}} \in \mathbb{R}^{\prod_d R_d \times \sum_d p_d R_d}$ *and* $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}} \in \mathbb{R}^{\sum_d p_d R_d \times \sum_d p_d R_d}$*. Then* $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{G}} = \boldsymbol{0}$*,* $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}}$ *has entries*

$$h_{(r_1,\ldots,r_D),(i_d,s_d)} = 1_{\{r_d = s_d\}} \sum_{j_d = i_d} x_{j_1,\ldots,j_D} \prod_{d' \neq d} \beta_{j_{d'}}^{(r_{d'})},$$

and $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}}$ *has entries*

$$h_{(i_d,r_d),(i_{d'},r_{d'})} = 1_{\{d \neq d'\}} \sum_{j_d = i_d, j_{d'} = i_{d'}} x_{j_1,\ldots,j_D} \sum_{s_d = r_d, s_{d'} = r_{d'}} g_{s_1,\ldots,s_D} \prod_{d'' \neq d,d'} \beta_{j_{d''}}^{(s_{d''})}.$$

Furthermore, $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}}$ *can be partitioned in* D^2 *sub-blocks as*

$$\begin{pmatrix} \boldsymbol{0} & * & * & * \\ \boldsymbol{H}_{21} & \boldsymbol{0} & * & * \\ \vdots & \vdots & \ddots & * \\ \boldsymbol{H}_{D1} & \boldsymbol{H}_{D2} & \cdots & \boldsymbol{0} \end{pmatrix}.$$

The elements of sub-block $\boldsymbol{H}_{dd'} \in \mathbb{R}^{p_d R_d \times p_{d'} R_{d'}}$ *can be retrieved from the matrix*

$$\boldsymbol{X}_{(dd')}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_{d'+1} \otimes \boldsymbol{B}_{d'-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(dd')}^{\mathsf{T}}.$$

$\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}}$ *can be partitioned into* D *sub-blocks as* $(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_D)$. *The sub-block* $\boldsymbol{H}_d \in \mathbb{R}^{\prod_d R_d \times p_d R_d}$ *has at most* $p_d \prod_d R_d$ *nonzero entries which can be retrieved from the matrix*

$$\boldsymbol{X}_{(d)}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1).$$

Let $\ell(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D | y, \boldsymbol{x}) = \ln p(y|\boldsymbol{x}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ be the log-density of GLM. Next result derives the score function, Hessian, and Fisher information of the Tucker tensor regression model.

Proposition 2 *Consider the tensor regression model defined by* (4) *and* (4).

1. *The score function (or score vector) is*

$$\nabla \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) = \frac{(y-\mu)\mu'(\eta)}{\sigma^2} \nabla \eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \tag{6}$$

   with $\nabla \eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ *given in Lemma 2.*
2. *The Hessian of the log-density* ℓ *is*

$$\begin{aligned} & H(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &= -\left[\frac{[\mu'(\eta)]^2}{\sigma^2} - \frac{(y-\mu)\theta''(\eta)}{\sigma^2}\right] \nabla \eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) d\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &\quad + \frac{(y-\mu)\theta'(\eta)}{\sigma^2} d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D), \end{aligned} \tag{7}$$

   with $d^2\eta$ *defined in Lemma 2.*
3. *The Fisher information matrix is*

$$\begin{aligned} & \boldsymbol{I}(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &= E[-H(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] \\ &= \mathrm{Var}[\nabla \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) d\ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] \\ &= \frac{[\mu'(\eta)]^2}{\sigma^2} [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} (\mathrm{vec}\boldsymbol{X})(\mathrm{vec}\boldsymbol{X})^{\mathsf{T}} [\boldsymbol{B}_D \\ &\quad \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]. \end{aligned} \tag{8}$$

Remark 1 For the canonical link, $\theta = \eta$, $\theta'(\eta) = 1$, $\theta''(\eta) = 0$, and the second term of Hessian vanishes. For the classical GLM with a linear systematic part ($D = 1$), $d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ is zero and thus the third term of Hessian vanishes. For the classical GLM ($D = 1$) with a canonical link, both second and third terms of the Hessian vanish and thus the Hessian is nonstochastic, coinciding with the information matrix.

4.2 Identifiability

The Tucker decomposition (3) is nonidentifiable due to the nonsingular transformation indeterminacy. That is

$$[\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!] = [\![\boldsymbol{G} \times_1 \boldsymbol{O}_1^{-1} \times \cdots \times_D \boldsymbol{O}_D^{-1}; \boldsymbol{B}_1\boldsymbol{O}_1, \ldots, \boldsymbol{B}_D\boldsymbol{O}_D]\!]$$

for any nonsingular matrices $\boldsymbol{O}_d \in \mathbb{R}^{R_d \times R_d}$. This implies that the number of free parameters for a Tucker model is $\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2$, with the last term adjusting for nonsingular indeterminacy. Therefore the Tucker model is identifiable only in terms of the equivalency classes.

For asymptotic consistency and normality, it is necessary to adopt a specific constrained parameterization. It is common to impose the orthonormality constraint on the factor matrices $\boldsymbol{B}_d^{\mathsf{T}}\boldsymbol{B}_d = \boldsymbol{I}_{R_d}, d = 1, \ldots, D$. However the resulting parameter space is a manifold and much harder to deal with. We adopt an alternative parameterization that fixes the entries of the first R_d rows of $\boldsymbol{B}_d$ to be ones

$$\boldsymbol{B} = \{[\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!] : \beta_{i_d}^{(r)} = 1, i_d = 1, \ldots, R_d, d = 1, \ldots, D\}.$$

The formulae for score, Hessian, and information in Proposition 2 require changes accordingly. The entries in the first R_d rows of $\boldsymbol{B}_d$ are fixed at ones and their corresponding entries, rows and columns in score, Hessian, and information need to be deleted. Choice of the restricted space $\boldsymbol{B}$ is obviously arbitrary, and excludes arrays with any entries in the first rows of $\boldsymbol{B}_d$ equal to zeros. However the set of such exceptional arrays has Lebesgue measure zero. In specific applications, subject knowledge may suggest alternative restrictions on the parameters.

Given a finite sample size, the conditions for global identifiability of the parameters are in general hard to obtain except in the linear case ($D = 1$). Local identifiability essentially requires linear independence between the "collapsed" vectors $[\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}}\text{vec}\boldsymbol{x}_i \in \mathbb{R}^{\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2}$.

Proposition 3 (Identifiability) *Given iid data points* $\{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$ *from the Tucker tensor regression model. Let* $\boldsymbol{B}_0 \in \boldsymbol{B}$ *be a parameter point and assume there exists an open neighborhood of* $\boldsymbol{B}_0$ *in which the information matrix has a constant rank. Then* $\boldsymbol{B}_0$ *is locally identifiable if and only if*

$$I(\boldsymbol{B}_0) = [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \left[\sum_{i=1}^{n} \frac{\mu'(\eta_i)^2}{\sigma_i^2} (\text{vec}\,\boldsymbol{x}_i)(\text{vec}\,\boldsymbol{x}_i)^{\mathsf{T}}\right] [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]$$

is nonsingular.

4.3 Asymptotics

The asymptotics for tensor regression follow from those for MLE or M-estimation [13,21].

Theorem 1 *Assume $\boldsymbol{B}_0 \in \boldsymbol{B}$ is (globally) identifiable up to permutation and the array covariates $\boldsymbol{X}_i$ are iid from a bounded underlying distribution.*

1. *(Consistency) The MLE is consistent, i.e., $\hat{\boldsymbol{B}}_n$ converges to $\boldsymbol{B}_0$ in probability, in following models. (1) Normal tensor regression with a compact parameter space $\boldsymbol{B}_0 \subset \boldsymbol{B}$. (2) Binary tensor regression. (3) Poisson tensor regression with a compact parameter space $\boldsymbol{B}_0 \subset \boldsymbol{B}$.*
2. *(Asymptotic Normality) For an interior point $\boldsymbol{B}_0 \in \boldsymbol{B}$ with nonsingular information matrix $\boldsymbol{I}(\boldsymbol{B}_0)$ (8) and $\hat{\boldsymbol{B}}_n$ is consistent, $\sqrt{n}(\mathrm{vec}\hat{\boldsymbol{B}}_n - \mathrm{vec}\boldsymbol{B}_0)$ converges in distribution to a normal with mean zero and covariance matrix $\boldsymbol{I}^{-1}(\boldsymbol{B}_0)$.*

In practice it is rare that the true regression coefficient $\boldsymbol{B}_{\text{true}} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ is exactly a low-rank tensor. However the MLE of the rank-R tensor model converges to the maximizer of function $M(\boldsymbol{B}) = \mathbb{P}_{\boldsymbol{B}_{\text{true}}} \ln p_{\boldsymbol{B}}$, or equivalently, $\mathbb{P}_{\boldsymbol{B}_{\text{true}}} \ln(p_{\boldsymbol{B}}/p_{\boldsymbol{B}_{\text{true}}})$. In other words, the MLE consistently estimates the best approximation (among models in $\boldsymbol{B}$) of $\boldsymbol{B}_{\text{true}}$ in the sense of Kullback–Leibler distance.

5 Numerical Study

We have carried out numerical experiments to study the finite sample performance of the Tucker regression. Our simulations focus on three aspects. First, we demonstrate the capacity of the Tucker regression in identifying various shapes of signals. Second, we study the consistency property of the method by gradually increasing the sample size. Third, we compare the performance of the Tucker regression with the CP regression [25] and some other alternative solutions. Finally, we analyze real MRI data to illustrate the Tucker downsizing and to further compare different methods.

5.1 Identification of Various Shapes of Signals

In our first example, we demonstrate that the proposed Tucker regression model, though with substantial reduction in dimension, can manage to identify a range of two-dimensional signal shapes with varying ranks. In Fig. 2, we list the 2D signals $\boldsymbol{B} \in \mathbb{R}^{64\times 64}$ in the first row, along with the estimates by Tucker tensor models in the second to fourth rows with orders (1, 1), (2, 2), and (3, 3), respectively. Note that, since the orders along both dimensions are made equal, the Tucker model is to perform essentially the same as a CP model in this example, and the results are presented here for completeness. Later examples examine differences of the two models. The regular covariate vector $\boldsymbol{Z} \in \mathbb{R}^5$ and image covariate $\boldsymbol{X} \in \mathbb{R}^{64\times 64}$ are randomly generated with all elements being independent standard normals. The response Y is generated from a normal model with mean $\mu = \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{Z} + \langle \boldsymbol{B}, \boldsymbol{X}\rangle$ and variance $\mathrm{var}(\mu)/10$. The coefficient $\boldsymbol{\gamma}$ has all elements equal to one, and $\boldsymbol{B}$ is binary, with the signal region equal

to one and the rest zero. Figure 2 shows that the Tucker model yields a sound recovery of the true signals, even for those of high rank or natural shape, e.g., "disk" and "butterfly."

5.2 Performance with Increasing Sample Size

The second example employs a model similar to Fig. 2, but with a three-dimensional image covariate. The dimension of $\boldsymbol{X}$ is set as $p_1 \times p_2 \times p_3$, with $p_1 = p_2 = p_3$ equal to 16 and 32, respectively. The signal array $\boldsymbol{B}$ is generated from a Tucker structure, with the elements of core tensor $\boldsymbol{G}$ and the factor matrices $\boldsymbol{B}$'s all coming from independent standard normals. The dimension of the core tensor $\boldsymbol{G}$ is set as $R_1 \times R_2 \times R_3$, with $R_1 = R_2 = R_3 = 2$, 5, and 8, respectively. We gradually increase the sample size, starting with an n that is in hundreds and no smaller than the degrees of freedom of the generating model. We aim to achieve two purposes with this example: first, we verify the consistency property of the proposed estimator, and second, we gain some practical knowledge about the estimation accuracy with different sample sizes. Figure 3 summarizes the results. It is clearly seen that the estimation improves with the increasing sample size. Meanwhile, we observe that, unless the core tensor dimension is small, one would require a relatively large sample size to achieve a good estimation accuracy. This is not surprising though, considering the number of parameters of the model and that regularization is not employed here. The proposed approach has been primarily designed for imaging studies with a reasonably large number of subjects. Recently, a number of large-scale brain imaging studies are emerging. For instance, the Attention Deficit Hyperactivity Disorder Sample Initiative [1] consists of over 900 participants from eight imaging centers. The Alzheimer's Disease Neuroimaging Initiative [2] accumulates over 3000 participants with MRI, fMRI, and genomics data. In addition, regularization discussed in Sect. 3.3 and the Tucker downsizing in Sect. 2.3 can both help improve estimation given a limited sample size.

5.3 Comparison of Different Methods

The third example compares the Tucker tensor model with the CP tensor model of Zhou et al. [25]. We consider two cases. First, we generate the normal response and the 3D tensor covariate $\boldsymbol{X}$ with all elements drawn from a standard normal distribution. The signal array $\boldsymbol{B}$ has dimensions p_1, p_2, p_3 and the d-ranks r_1, r_2, r_3, where the d-rank is defined as the column rank of the mode-d matricization $\boldsymbol{B}_{(d)}$ of $\boldsymbol{B}$. We set $p_1 = p_2 = p_3 = 16$ and 32, and $(r_1, r_2, r_3) = (5, 3, 3)$, $(8, 4, 4)$, and $(10, 5, 5)$, respectively. The sample size is 2000. We fit a Tucker model with $R_d = r_d$, and a CP model with $R = \max r_d$, $d = 1, 2, 3$. We report in Table 2 the degrees of freedom of the two models, as well as the root mean squared error (RMSE) out of 100 data replications. It is seen that the Tucker model requires a smaller number of free parameters, while it achieves a more accurate estimation compared to the CP model. Such advantages come from the flexibility of the Tucker decomposition that permits different orders R_d along directions.

Second, we employ the approach of Goldsmith et al. [9] to generate the 3D tensor covariate $\boldsymbol{X}$ based on the real data described in Sect. 5.4. The original image size

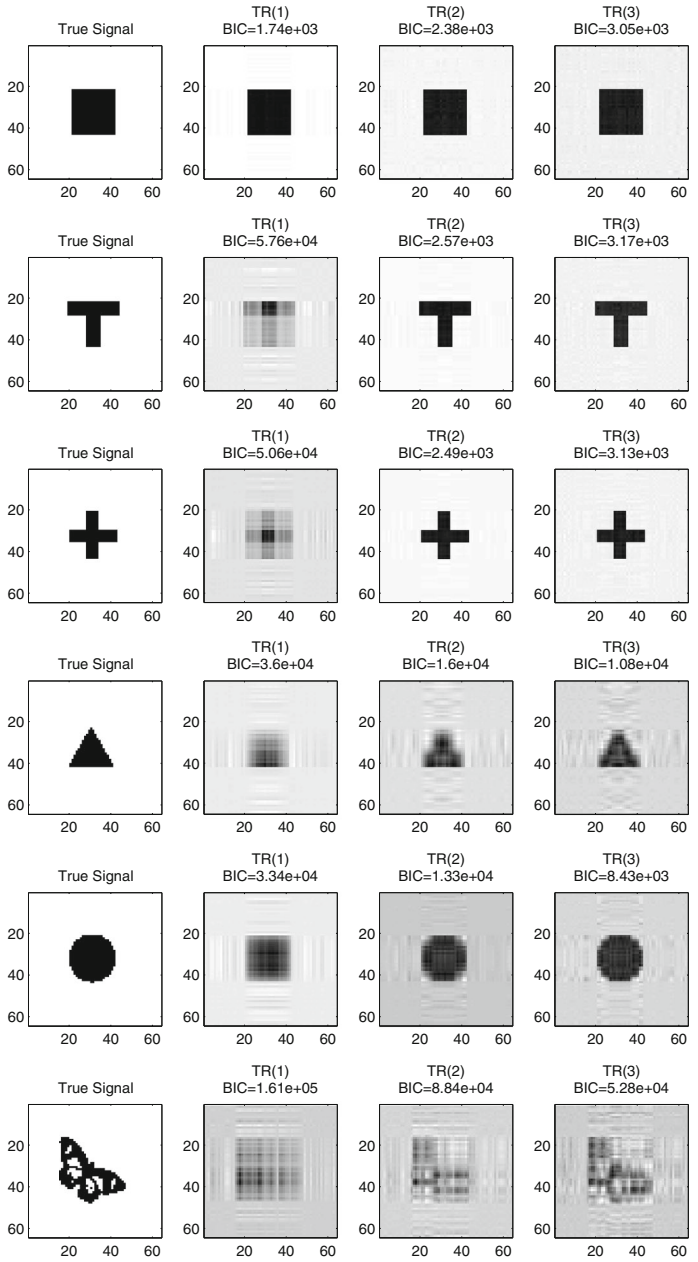

Fig. 2 True and recovered image signals by Tucker regression. The matrix variate has size 64 by 64 with entries generated as independent standard normals. The regression coefficient for each entry is either 0 (white) or 1 (black). The sample size is 1000. TR(r) means estimate from the Tucker regression with an r-by-r core tensor

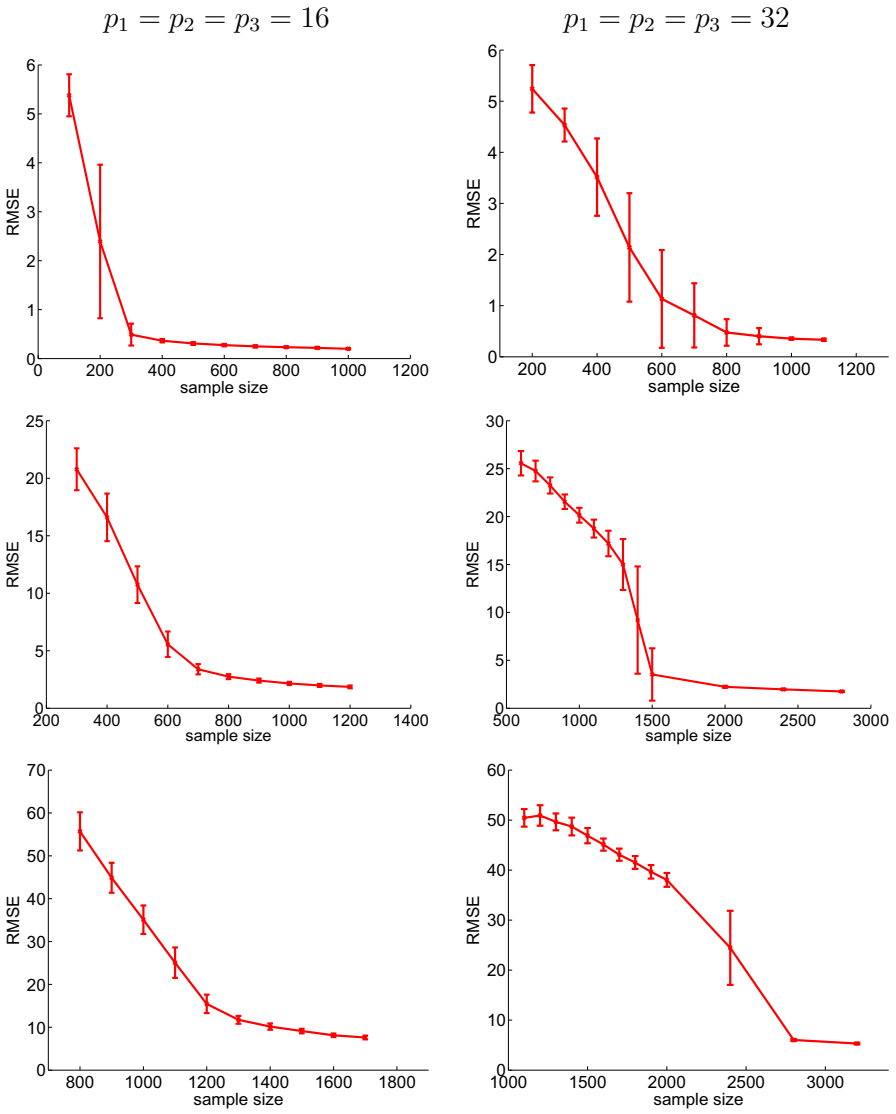

Fig. 3 Root mean squared error (RMSE) of the tensor parameter estimate versus the sample size. Reported are the average and standard deviation of RMSE based on 100 data replications. Top: $R_1 = R_2 = R_3 = 2$; middle: $R_1 = R_2 = R_3 = 5$; bottom: $R_1 = R_2 = R_3 = 8$

is $121 \times 145 \times 121$, and for computational simplicity, we downsize the image to $15 \times 18 \times 15$. We then extract the top 100 principal components $\{\boldsymbol{\phi}_j\}_{j=1}^{100}$ and the corresponding eigenvalues $\{\lambda_j\}_{j=1}^{100}$ of the the data matrix where the ith row is the vectorized image of the ith subject. The simulated image of the ith subject is obtained by first computing $\text{vec}(\boldsymbol{X}_i) = \sum_{j=1}^{100} c_{ij}\boldsymbol{\phi}_j$, then transforming to a $15 \times 18 \times 15$ array, where the loadings c_{ij} are generated from a normal distribution with mean 0

Table 2 Comparison of the Tucker and CP models based on the image tensor with normal elements

Dimension	Criterion	Model	(5, 3, 3)	(8, 4, 4)	(10, 5, 5)
$16 \times 16 \times 16$	Df	Tucker	178	288	420
		CP	230	368	460
	RMSE	Tucker	0.202 (0.013)	0.379 (0.017)	0.728 (0.030)
		CP	0.287 (0.033)	1.030 (0.081)	2.858 (0.133)
$32 \times 32 \times 32$	Df	Tucker	354	544	740
		CP	470	752	940
	RMSE	Tucker	0.288 (0.013)	0.570 (0.023)	1.234 (0.045)
		CP	0.392 (0.046)	1.927 (0.172)	16.24 (3.867)

Reported are the average and standard deviation (in the parenthesis) of the root mean squared error, all based on 100 data replications

Table 3 Comparison of different methods. Reported are the average and standard deviation (in the parenthesis) of the root mean squared error for both the estimated coefficient tensor $\boldsymbol{B}$ and the predicted response Y, all based on 100 data replications

	Tucker	CP	Bayes	PCA
Prediction of Y	0.149 (0.023)	0.308 (0.005)	2.456 (0.104)	0.855 (0.110)
Estimation of $\boldsymbol{B}$	16.78 (14.20)	16.41 (16.82)	18.39 (0.137)	N/A

and variance λ_j. The image array is also standardized by dividing all the entries by the maximum absolute value. The true signal array $\boldsymbol{B}$ is a 3D ball centered at the middle of the array with value one inside the ball and zero outside. The response is generated from a normal model with a unit variance. In addition to comparing the Tucker and CP models, we also compare with the Bayesian regression method of Goldsmith et al. [9] and the PCA method of Caffo et al. [3]. We set $(R_1, R_2, R_3) = (3, 3, 3)$ for the Tucker model, and $R = 3$ for CP, which yields comparable number of parameters. For the Bayesian method, we only tune the sigma beta parameter using cross-validation, and fix the rest of parameters following the rules in Goldsmith et al. [9]. Otherwise, the computation is prohibitive. For PCA, we keep the number of principal components retaining 95% of total variation. The sample size for training is 1000, and for testing is 500. We report in Table 3 the RMSE of both the estimated $\boldsymbol{B}$ and the predicted Y based on 100 data replications. We also plot in Fig. 4 the true (shown in red color) and estimated (green) $\boldsymbol{B}$ overlaid on a randomly chosen brain image. PCA is not reported for estimation of $\boldsymbol{B}$, since only a subset of principal components are retained. It is clearly seen in this example that the Tucker method outperforms both CP and the alternative PCA and Bayesian solutions.

5.4 Attention Deficit Hyperactivity Disorder Data Analysis and Running Time Analysis

Next we analyze the attention deficit hyperactivity disorder (ADHD) data from the ADHD-200 Sample Initiative [1] to illustrate our proposed method as well as Tucker

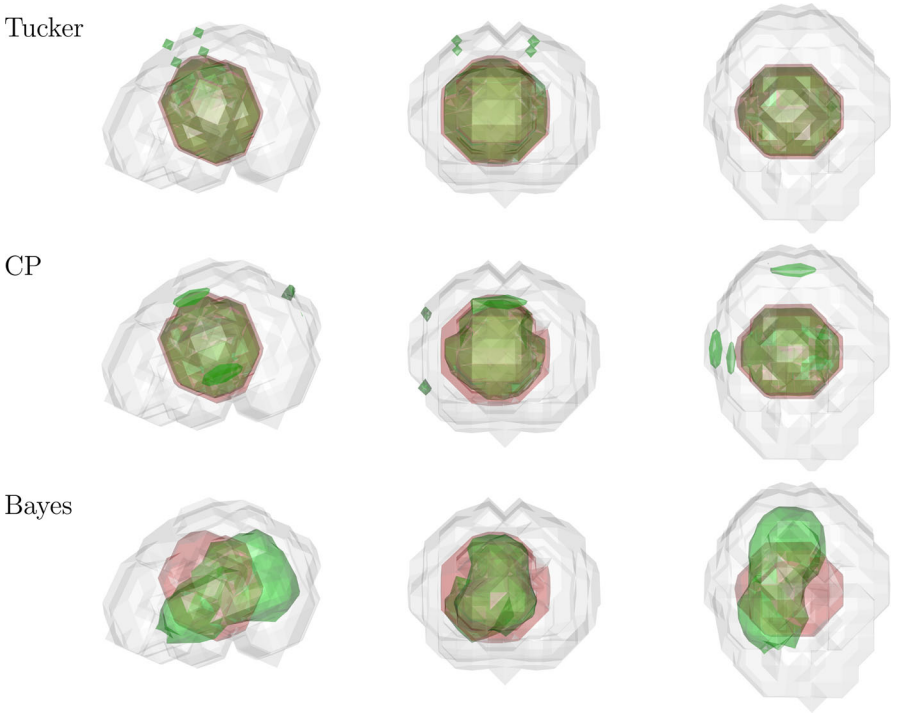

Fig. 4 True (in red color) and recovered (green) image signal overlaid on a randomly chosen brain image shown in three views. Under comparison are the Tucker, CP, and Bayesian method (Color figure online)

downsizing. ADHD is a common childhood disorder and can continue through adolescence and adulthood. Symptoms include difficulty in staying focused and paying attention, difficulty in controlling behavior, and over-activity. The data have been prepartitioned into a training data of 770 subjects and a testing data of 197 subjects. We remove those subjects with missing observations or poor image quality, resulting in 762 training subjects and 169 testing subjects. In the training set, there are 280 combined ADHD subjects, 482 normal controls, and the case–control ratio is about 3:5. In the testing set, there are 76 combined ADHD subjects, 93 normal controls, and the case–control ratio is about 4:5. T1-weighted images have been acquired for each subject and preprocessed by standard steps. The data are obtained from the Neuro Bureau (the Burner data, http://neurobureau.projects.nitrc.org/ADHD200/Data.html). In addition to the MRI image predictor, we also include the subjects' age and handiness as regular covariates. The response is the binary diagnosis status.

The original image size is $p_1 \times p_2 \times p_3 = 121 \times 145 \times 121$. We employ Tucker downsizing in Sect. 2.3. Specifically, we first choose a wavelet basis for $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times \tilde{p}_d}$, and then transform the image predictor from $\boldsymbol{X}$ to $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^\tau, \ldots, \boldsymbol{B}_D^\tau]\!]$. We prespecify the values of $\tilde{p}_d$'s that are about tenth of the original dimensions p_d, and equivalently, we fit a Tucker tensor regression with the image predictor dimension downsized to $\tilde{p}_1 \times \tilde{p}_2 \times \tilde{p}_3$. In our example, we have experimented with a set of

Table 4 ADHD testing data misclassification error. Methods under comparison are regularized Tucker, regularized CP, Tucker, CP, Bayesian model, and PCA

Basis	Dimension	Reg-Tucker	Reg-CP	Tucker	CP	Bayes	PCA
Haar	$12 \times 14 \times 12$	0.361	0.367	0.379	0.438	0.414	0.485
(D2)	$10 \times 12 \times 10$	0.343	0.390	0.379	0.408	0.420	0.485
Daubechies	$12 \times 14 \times 12$	0.337	0.385	0.385	0.414	0.402	0.391
(D4)	$10 \times 12 \times 10$	0.320	0.396	0.367	0.373	0.396	0.462

values of $\tilde{p}_d$'s, and the results are qualitatively similar. We report two sets, $\tilde{p}_1 = 12$, $\tilde{p}_2 = 14$, $\tilde{p}_3 = 12$, and $\tilde{p}_1 = 10$, $\tilde{p}_2 = 12$, $\tilde{p}_3 = 10$. We have also experimented with the Haar wavelet basis (Daubechies D2) and the Daubechies D4 wavelet basis, which again show similar qualitative patterns.

For $\tilde{p}_1 = 12$, $\tilde{p}_2 = 14$, $\tilde{p}_3 = 12$, we fit a Tucker tensor model with $R_1 = R_2 = R_3 = 3$, resulting in 114 free parameters, and fit a CP tensor model with $R = 4$, resulting in 144 free parameters. For $\tilde{p}_1 = 10$, $\tilde{p}_2 = 12$, $\tilde{p}_3 = 10$, we fit a Tucker tensor model with $R_1 = R_2 = 2$ and $R_3 = 3$, resulting in 71 free parameters, and fit a CP tensor model with $R = 4$, resulting in 120 free parameters. We have chosen those orders so that the numbers of free parameters of the Tucker and CP models are comparable. Besides, we follow the practical guideline as discussed in Sect. 3.2. We also fit the regularized version of the Tucker and CP model with the same orders. In addition, for comparison, we apply the Bayesian model of Goldsmith et al. [9] and the PCA method of Caffo et al. [3]. The parameters are tuned based on fivefold cross-validation.

We evaluate each method by comparing the misclassification error rate on the independent testing set. We also remark that, our method is designed not solely for prediction or classification purpose, but instead to obtain a parsimonious model characterizing the association between the image covariates and the clinical outcome. We focus on classification in this example, mainly because this dataset was originally released for a data competition where the classification accuracy is the main evaluation criterion. The results are shown in Table 4. We see from the table that Tucker outperforms CP and other alternative solutions. In addition, the regularized Tucker method performs the best in this example.

6 Discussion

We have proposed a tensor regression model based on the Tucker decomposition. The new model provides a flexible framework for regression with imaging covariates. We develop a fast estimation algorithm, a general regularization procedure, and the associated asymptotic properties. In addition, we provide a detailed comparison, both analytically and numerically, of the Tucker and CP tensor models.

We make some additional remarks regarding our proposed method. First, in a real imaging analysis, the signal hardly has an exact low rank. However, given the limited sample size, a low-rank estimate often provides a reasonable *approximation* to the true

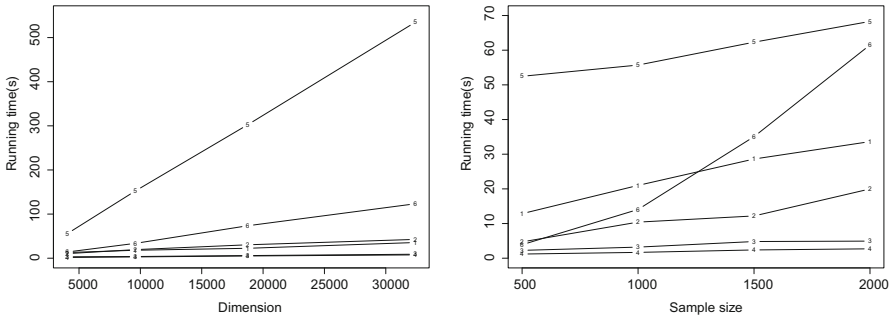

Fig. 5 The running time (in seconds) of different methods with varying dimensions and sample sizes. The left panel shows the varying dimensions, $p_1 \times p_2 \times p_3 = 15 \times 18 \times 15, 20 \times 24 \times 24, 25 \times 30 \times 25$, and $30 \times 36 \times 30$, respectively, when the sample size is fixed at $n = 1000$. The horizontal axis shows $p_1 \times p_2 \times p_3$. The right panel shows the varying sample size $n = 500, 1000, 1500, 2000$, and 2500, respectively, when the dimension is fixed at $p_1 \times p_2 \times p_3 = 15 \times 18 \times 15$. The lines, numbered from 1 to 6, indicate the running time of regularized Tucker, regularized CP, Tucker, CP, Bayesian model, and PCA, respectively

signal, as demonstrated by the simulation examples. Second, imaging downsizing is a tradeoff, in that it is to facilitate the computation, including the memory usage, and is to reduce the dimensionality of the estimation problem, but at the cost of sacrificing the image resolution. Third, even after substantial dimension reduction, the number of remaining parameters can still be large compared to the sample size. When the sample size is limited, the optimization algorithm is more likely to get trapped at a local rather than global minimum. Increasing the number of initializations is to facilitate the issue, but cannot avoid it completely. Finally, we report the computation time of the proposed Tucker model. Specifically, we employ the simulation example in Sect. 5.3, and investigate how the running time of different methods grows along with the image dimension and sample size. Figure 5 records the running time (in seconds) that are obtained on a standard laptop computer with a 2.2 GHz Intel Core i7. The left panel shows the results with the varying dimensions, $p_1 \times p_2 \times p_3 = 15 \times 18 \times 15, 20 \times 24 \times 24, 25 \times 30 \times 25$, and $30 \times 36 \times 30$, respectively, when the sample size is fixed at $n = 1000$. The right panel shows the varying sample size $n = 500, 1000, 1500, 2000$, and 2500, respectively, when the dimension is fixed at $p_1 \times p_2 \times p_3 = 15 \times 18 \times 15$. It is seen that the Tucker model and its regularized version are computationally more expensive than the counterpart of CP. We view this as a price that comes with the additional flexibility of the Tucker model. However, Tucker maintains a reasonable overall computation time, and is much faster than the alternative solutions such as the Bayesian model and PCA.

Appendix

Proof of Lemma 1

We rewrite the array inner product

$$\begin{aligned}\langle \boldsymbol{B}, \boldsymbol{X} \rangle &= \langle \boldsymbol{B}_{(d)}, \boldsymbol{X}_{(d)} \rangle = \langle \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T}, \boldsymbol{X}_{(d)} \rangle \\ &= \langle \boldsymbol{G}_{(d)}, \boldsymbol{B}_d^\mathsf{T} \boldsymbol{X}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1) \rangle \\ &= \langle \boldsymbol{G}_{(d)}, \tilde{\boldsymbol{X}}_{(d)} \rangle = \langle \boldsymbol{G}, \tilde{\boldsymbol{X}} \rangle,\end{aligned}$$

where the second and fourth equalities follow from (4) and the third follows from the invariance of trace function under cyclic permutation.

Proof of Proposition 1

It is easy to see that the block relaxation algorithm monotonically increases the objective values, i.e., $\ell(\boldsymbol{\theta}^{(t+1)}) \geq \ell(\boldsymbol{\theta}^{(t)})$ for all $t \geq 0$. Therefore its global convergence property follows from the standard theory for monotone algorithms [5,11,12]. Specifically global convergence is guaranteed under the following conditions: (i) ℓ is coercive, (ii) the stationary points of ℓ are isolated, (iii) the algorithmic mapping is continuous, (iv) $\boldsymbol{\theta}$ is a fixed point of the algorithm if and only if it is a stationarity point of ℓ, and (v) $\ell(\boldsymbol{\theta}^{(t+1)}) \geq \ell(\boldsymbol{\theta}^{(t)})$ with equality if and only if $\theta^{(t)}$ is a fixed point of the algorithm. Condition (i) is guaranteed by the compactness of the set $\{\boldsymbol{\theta} : \ell(\boldsymbol{\theta}) \geq \ell(\boldsymbol{\theta}^{(0)})$. Condition (ii) is assumed. Condition (iii) follows from the strict concavity assumption and implicit function theorem. By Fermat's principle, $\boldsymbol{\theta} = (\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ is a fixed point of the block relaxation algorithm if $D\ell(\boldsymbol{G}) = \boldsymbol{0}$ and $D\ell(\boldsymbol{B}_d) = \boldsymbol{0}$ for all d. Thus $\boldsymbol{\theta}$ is a fixed point if and only if it is a stationarity point of ℓ, i.e., condition (iv) is satisfied. Condition (v) follows from the monotonicity of the block relaxation algorithm. Local convergence follows from the classical Ostrowski theorem, which states that the algorithmic sequence $\boldsymbol{\theta}^{(t)}$ is local attracted to strictly local minimum $\boldsymbol{\theta}^{(\infty)}$ if the spectral radius of the differential of the algorithmic map $\rho[dM(\boldsymbol{\theta}^{(\infty)})]$ is strictly less than one. This follows from the strict concavity assumption of the block updates. See Zhou et al. [25] for more details.

Proof of Lemma 2

Assume $\boldsymbol{B}$ admits the Tucker decomposition (3). By (4),

$$\boldsymbol{B}_{(d)} = \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T}.$$

Using the well-known fact that $\mathrm{vec}(\boldsymbol{XYZ}) = (\boldsymbol{Z}^\mathsf{T} \otimes \boldsymbol{X})\mathrm{vec}(\boldsymbol{Y})$,

$$\mathrm{vec}\boldsymbol{B}_{(d)} = [(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(d)}^\mathsf{T} \otimes \boldsymbol{I}_{p_d}]\mathrm{vec}(\boldsymbol{B}_d).$$

Thus by the chain rule we have

$$\begin{aligned}\boldsymbol{J}_d &= D\boldsymbol{B}(\boldsymbol{B}_d) = D\boldsymbol{B}(\boldsymbol{B}_{(d)}) \cdot D\boldsymbol{B}_{(d)}(\boldsymbol{B}_d) = \boldsymbol{\Pi}_d \frac{\partial \mathrm{vec}\boldsymbol{B}_{(d)}}{\partial (\mathrm{vec}\boldsymbol{B}_d)^\mathsf{T}} \\ &= \boldsymbol{\Pi}_d \{[\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(d)}^\mathsf{T}] \otimes \boldsymbol{I}_{p_d}\}.\end{aligned}$$

Again by the chain rule, $D\eta(\boldsymbol{B}_d) = D\eta(\boldsymbol{B}) \cdot D\boldsymbol{B}(\boldsymbol{B}_d) = (\mathrm{vec}\boldsymbol{X})^\top \boldsymbol{J}_d$. For the derivative in $\boldsymbol{G}$, the duality Lemma 1 implies $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \boldsymbol{G}, \tilde{\boldsymbol{X}} \rangle$ for $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^\top, \ldots, \boldsymbol{B}_D^\top]\!]$. Then, by (4), we have

$$D\eta(\boldsymbol{G}) = (\mathrm{vec}\tilde{\boldsymbol{X}})^\top = (\mathrm{vec}\boldsymbol{X})^\top (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1).$$

Combining these results gives the gradient displayed in Lemma 2.

Next we consider the Hessian $d^2\eta$. Because $\boldsymbol{B}$ is linear in $\boldsymbol{G}$, the block $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{G}}$ vanishes. For the block $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}}$, the $(i_d, r_d, i_{d'}, r_{d'})$-entry is

$$\begin{aligned} h_{(i_d,r_d),(i_{d'},r_{d'})} &= \sum_{j_1,\ldots,j_D} x_{j_1,\ldots,j_D} \frac{\partial^2 b_{j_1,\ldots,j_D}}{\partial \beta_{i_d}^{(r)} \partial \beta_{i_{d'}}^{(r')}} \\ &= \sum_{j_1,\ldots,j_D} x_{j_1,\ldots,j_D} \sum_{s_1,\ldots,s_D} g_{s_1,\ldots,s_D} \frac{\partial^2 \beta_{j_1}^{(s_1)} \cdots \beta_{j_D}^{(s_D)}}{\partial \beta_{i_d}^{(r)} \partial \beta_{i_{d'}}^{(r')}}. \end{aligned}$$

The second derivative in the summand is nonzero only if $j_d = i_d$, $j_{d'} = i_{d'}$, $s_d = r_d$, $s_{d'} = r_{d'}$, and $d \neq d'$. Therefore

$$h_{(i_d,r_d),(i_{d'},r_{d'})} = 1_{\{d \neq d'\}} \sum_{j_d = i_d, j_{d'} = i_{d'}} x_{j_1,\ldots,j_D} \sum_{s_d = r_d, s_{d'} = r_{d'}} g_{s_1,\ldots,s_D} \prod_{d'' \neq d,d'} \beta_{j_{d''}}^{(s_{d''})}.$$

The first sum is over $\prod_{d'' \neq d,d'} p_{d''}$ terms and the second term is over $\prod_{d'' \neq d,d'} R_{d''}$ terms. A careful inspection reveals that the sub-block $\boldsymbol{H}_{dd'}$ shares the same entries as the matrix

$$\boldsymbol{X}_{(dd')} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_{d'+1} \otimes \boldsymbol{B}_{d'-1} \otimes \cdots \otimes \boldsymbol{B}_1) \boldsymbol{G}_{(dd')}^\top.$$

Finally, for the $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}}$ block, the $\{(r_1, \ldots, r_D), (i_d, r_d)\}$-entry is

$$\begin{aligned} h_{(r_1,\ldots,r_D),(i_d,s_d)} &= \sum_{j_1,\ldots,j_D} x_{j_1,\ldots,j_D} \frac{\partial^2 b_{j_1,\ldots,j_D}}{\partial g_{r_1,\ldots,r_D} \partial \beta_{i_d}^{(s_d)}} \\ &= \sum_{j_1,\ldots,j_D} x_{j_1,\ldots,j_D} \sum_{t_1,\ldots,t_D} \frac{\partial^2 g_{t_1,\ldots,t_D} \beta_{j_1}^{(t_1)} \cdots \beta_{j_D}^{(t_D)}}{\partial g_{r_1,\ldots,r_D} \partial \beta_{i_d}^{(s_d)}} \\ &= \sum_{j_1,\ldots,j_D} x_{j_1,\ldots,j_D} \frac{\partial \beta_{j_1}^{(r_1)} \cdots \beta_{j_D}^{(r_D)}}{\partial \beta_{i_d}^{(s_d)}} \\ &= 1_{\{r_d = s_d\}} \sum_{j_d = i_d} x_{j_1,\ldots,j_D} \prod_{d' \neq d} \beta_{j_{d'}}^{(r_{d'})}, \end{aligned}$$

where the sum is over $\prod_{d' \neq d} p_{d'}$ terms. The sub-block $\boldsymbol{H}_d \in \mathbb{R}^{\prod_d R_d \times p_d R_d}$ has at most $p_d \prod_d R_d$ nonzero entries. A close inspection suggests that the nonzero entries coincide with those in the matrix

$$\boldsymbol{X}_{(d)}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1).$$

Proof of Proposition 2

Since $\mu = b'(\theta)$, $d\mu/d\theta = b''(\theta) = \sigma^2/a(\phi)$ and

$$\begin{aligned}\nabla \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) &= \frac{y - b'(\theta)}{a(\phi)} \frac{d\theta}{d\mu} \frac{d\mu}{d\eta} \nabla \eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &= \frac{(y-\mu)\mu'(\eta)}{\sigma^2} [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} (\mathrm{vec} \boldsymbol{X})\end{aligned}$$

by Lemma 2. Further differentiating shows

$$\begin{aligned}&d^2 \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &= -\frac{1}{\sigma^2} \nabla \mu(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) d\mu(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) + \frac{y-\mu}{\sigma^2} d^2 \mu(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &= -\frac{[\mu'(\eta)]^2}{\sigma^2} ([\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \mathrm{vec} \boldsymbol{X})([\boldsymbol{B}_D \\ &\quad \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \mathrm{vec} \boldsymbol{X})^{\mathsf{T}} \\ &\quad + \frac{(y-\mu)\theta''(\eta)}{\sigma^2} ([\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \mathrm{vec} \boldsymbol{X})([\boldsymbol{B}_D \\ &\quad \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \mathrm{vec} \boldsymbol{X})^{\mathsf{T}} \\ &\quad + \frac{(y-\mu)\theta'(\eta)}{\sigma^2} d^2 \eta(\boldsymbol{B}).\end{aligned}$$

It is easy to see that $\mathbf{E}[\nabla \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] = \mathbf{0}$. Moreover, $\mathbf{E}[-d^2 \ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] = \boldsymbol{I}(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$. Then (8) follows.

Proof of Proposition 3

The proof follows from a classical result [18] that states that, if θ_0 be a regular point of the information matrix $I(\theta)$, then θ_0 is locally identifiable if and only if $I(\theta_0)$ is nonsingular. The regularity assumptions are satisfied by Tucker regression model: (1) the parameter space $\boldsymbol{B}$ is open, (2) the density $p(y, \boldsymbol{x}|\boldsymbol{B})$ is proper for all $\boldsymbol{B} \in \boldsymbol{B}$, (3) the support of the density $p(y, \boldsymbol{x}|\boldsymbol{B})$ is same for all $\boldsymbol{B} \in \boldsymbol{B}$, (4) the log-density $\ell(\boldsymbol{B}|y, \boldsymbol{x}) = \ln p(y, \boldsymbol{x}|\boldsymbol{B})$ is continuously differentiable, and (5) the information matrix

$$\begin{aligned}\boldsymbol{I}(\boldsymbol{B}) = [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^{\mathsf{T}} \left[\sum_{i=1}^{n} \frac{\mu'(\eta_i)^2}{\sigma_i^2} (\mathrm{vec}\, \boldsymbol{x}_i)(\mathrm{vec}\, \boldsymbol{x}_i)^{\mathsf{T}} \right] [\boldsymbol{B}_D \\ \otimes \cdots \otimes \boldsymbol{B}_1 \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]\end{aligned}$$

is continuous in $\boldsymbol{B}$ by Proposition 2. Therefore $\boldsymbol{B} \in \boldsymbol{B}$ is locally identifiable if and only if $\boldsymbol{I}(\boldsymbol{B})$ is nonsingular.

Proof of Theorem 1

The asymptotics for tensor regression follow from the standard theory of M-estimation. The key observation is that the nonlinear part of tensor model (4) is a degree-$(D+1)$ polynomial of parameters $\boldsymbol{G}$ and $\boldsymbol{B}_d$ and the collection of polynomials $\{\langle \boldsymbol{B}, \boldsymbol{X}\rangle, \boldsymbol{B} \in \boldsymbol{B}\}$ form a Vapnik–Červonenkis (VC) class. Then the classical uniform convergence theory applies [21]. The arguments in [25] extends the classical argument for GLM [21, Example 5.40] to the CP tensor regression model. The same proof also applies to the Tucker model with little changes and thus is omitted here. For the asymptotic normality, we need to establish that the log-likelihood function of Tucker regression model is quadratic mean differentiable (q.m.d.) [13]. By a well-known result [13, Theorem 12.2.2] or [21, Lemma 7.6], it suffices to verify that the density is continuously differentiable in parameter for μ-almost all x and that the Fisher information matrix exists and is continuous. The derivative of density is

$$\nabla p(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) = \nabla e^{\ell(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)} = p(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \nabla \ell(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D),$$

which is well defined and continuous by Proposition 2. The same proposition shows that the information matrix exists and is continuous. Therefore the Tucker regression model is q.m.d. and the asymptotic normality follows from the classical result for q.m.d. families [21, Theorem 5.39].

References

1. ADHD (2017) The ADHD-200 sample. http://fcon_1000.projects.nitrc.org/indi/adhd200/. Accessed Mar 2017
2. ADNI (2017) Alzheimer's disease neuroimaging initiative. http://adni.loni.ucla.edu. Accessed Mar 2017
3. Caffo B, Crainiceanu C, Verduzco G, Joel S, Mostofsky SH, Bassett S, Pekar J (2010) Two-stage decompositions for the analysis of functional connectivity for fMRI with application to Alzheimer's disease risk. Neuroimage 51(3):1140–1149
4. Chen SS, Donoho DL, Saunders MA (2001) Atomic decomposition by basis pursuit. SIAM Rev. 43(1):129–159
5. de Leeuw J (1994) Block-relaxation algorithms in statistics. In: Information systems and data analysis. Springer, Berlin, pp 308–325
6. Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96(456):1348–1360
7. Frank IE, Friedman JH (1993) A statistical view of some chemometrics regression tools. Technometrics 35(2):109–135
8. Friston K, Ashburner J, Kiebel S, Nichols T, Penny W (eds) (2007) Statistical parametric mapping: the analysis of functional brain images. Academic Press, London
9. Goldsmith J, Huang L, Crainiceanu C (2014) Smooth scalar-on-image regression via spatial bayesian variable selection. J Comput Graph Stat 23:46–64
10. Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500
11. Lange K (2004) Optimization. Springer texts in statistics. Springer, New York
12. Lange K (2010) Numerical analysis for statisticians. Statistics and computing, second edn. Springer, New York
13. Lehmann EL, Romano JP (2005) Testing statistical hypotheses. Springer texts in statistics, third edn. Springer, New York
14. Li Y, Zhu H, Shen D, Lin W, Gilmore JH, Ibrahim JG (2011) Multiscale adaptive regression models for neuroimaging data. J R Stat Soc 73:559–578

15. Li F, Zhang T, Wang Q, Gonzalez M, Maresh E, Coan J (2015) Spatial Bayesian variable selection and grouping in high-dimensional scalar-on-image regressions. Ann Appl Stat (in press)
16. McCullagh P, Nelder JA (1983) Generalized linear models. Monographs on statistics and applied probability. Chapman & Hall, London
17. Reiss P, Ogden R (2010) Functional generalized linear models with images as predictors. Biometrics 66:61–69
18. Rothenberg TJ (1971) Identification in parametric models. Econometrica 39(3):577–91
19. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc 58(1):267–288
20. Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31:279–311
21. van der Vaart AW (1998) Asymptotic statistics, volume 3 of Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge
22. Wang X, Nan B, Zhu J, Koeppe R (2014) Regularized 3D functional regression for brain image data via haar wavelets. Ann Appl Stat 8:1045–1064
23. Yue Y, Loh JM, Lindquist MA (2010) Adaptive spatial smoothing of fMRI images. Stat Interface 3:3–14
24. Zhou H, Li L (2014) Regularized matrix regression. J R Stat Soc 76:463–483
25. Zhou H, Li L, Zhu H (2013) Tensor regression with applications in neuroimaging data analysis. J Am Stat Assoc 108(502):540–552
26. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc 67(2):301–320