SIAM J. MATH. DATA SCI.
Vol. 2, No. 4, pp. 944–966

Tensor Regression Using Low-Rank and Sparse Tucker Decompositions*

Talal Ahmed†, Haroon Raja‡, and Waheed U. Bajwa†

Abstract. This paper studies a tensor-structured linear regression model with a scalar response variable and tensor-structured predictors, such that the regression parameters form a tensor of order d (i.e., a d-fold multiway array) in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$. In particular, we focus on the task of estimating the regression tensor from m realizations of the response variable and the predictors where $m \ll n = \prod_i n_i$. Despite the seeming ill-posedness of this estimation problem, it can still be solved if the parameter tensor belongs to the space of sparse, low Tucker-rank tensors. Accordingly, the estimation procedure is posed as a nonconvex optimization program over the space of sparse, low Tucker-rank tensors, and a tensor variant of projected gradient descent is proposed to solve the resulting nonconvex problem. In addition, mathematical guarantees are provided that establish that the proposed method linearly converges to an appropriate solution under a certain set of conditions. Further, an upper bound on sample complexity of tensor parameter estimation for the model under consideration is characterized for the special case when the individual (scalar) predictors independently draw values from a sub-Gaussian distribution. The sample complexity bound is shown to have a polylogarithmic dependence on $\bar{n} = \max\{n_i : i \in \{1, 2, \ldots, d\}\}$; orderwise, it matches the bound one can obtain from a heuristic parameter counting argument. Finally, numerical experiments demonstrate the efficacy of the proposed tensor model and estimation method on a synthetic dataset and a collection of neuroimaging datasets pertaining to attention deficit hyperactivity disorder (ADHD). Specifically, the proposed method exhibits better sample complexities on both synthetic and real datasets, demonstrating the usefulness of the model and the method in settings where $n \gg m$.

Key words. linear regression, sample complexity, sparsity, tensor regression, Tucker decomposition

AMS subject classifications. 41A52, 41A63, 62F10, 62J05

DOI. 10.1137/19M1299335

1. Introduction.

Many modern data science problems involve learning a high-dimensional regression model, where the number of predictors is much larger than the number of samples. We focus on *tensor-structured* regression models, where the predictors appear naturally in the form of a tensor. Such regression models find applications within hyperspectral imaging [27, 6], climatology [24], neuroscience [1, 28], sentiment analysis [34], and computer vision [12]. In this work, we specifically consider a *linear* tensor-structured regression model with response variable $y \in \mathbb{R}$, tensor (multiway array) of predictors $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, tensor of regression parameters $\mathbf{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, and noise $\eta \in \mathbb{R}$ such that $y = \langle \mathbf{X}, \mathbf{B} \rangle + \eta$, where $d \in \mathbb{Z}^+$,

*Received by the editors November 13, 2019; accepted for publication (in revised form) July 14, 2020; published electronically October 13, 2020.

https://doi.org/10.1137/19M1299335

Funding: This work was supported in part by the National Science Foundation (NSF) under awards CCF-1453073 and CCF-1910110, and by the Army Research Office (ARO) under award W911NF-17-1-0546.

†Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08854 USA (talal.ahmed@rutgers.edu, waheed.bajwa@rutgers.edu).

‡Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (hraja@umich.edu).

and $\langle \cdot, \cdot \rangle$ denotes the canonical inner product. Among the various applications of this model, a major one appears in neuroimaging data analysis, where the voxels (predictors) in a brain image naturally appear in the form of a tensor, and the associated disease outcome (response) appears as a scalar variable [29, 28, 20, 38].

Mathematically, let us define $\{\mathbf{X}_i\}_{i=1}^m$, $\{y_i\}_{i=1}^m$, and $\{\eta_i\}_{i=1}^m$ to be the realizations of $\mathbf{X}$, y, and η, respectively, where m refers to the number of observations/measurements such that $m \lll n := \prod_i n_i$. Then, the realizations of the linear regression model can be expressed as

$$y_i = \langle \mathbf{X}_i, \mathbf{B} \rangle + \eta_i, \quad i \in \{1, 2, \ldots, m\}. \tag{1.1}$$

In this paper, given $\{\mathbf{X}_i\}_{i=1}^m$ and $\{y_i\}_{i=1}^m$, we focus on the task of learning the regression model in (1.1), which is equivalent to estimating $\mathbf{B}$. Since we are considering the high-dimensional setting of $m \lll n$ in this work, the learning task is ill-posed without the imposition of additional constraints on the parameter tensor $\mathbf{B}$. We now discuss how this challenge has been addressed in prior work.

1.1. Relationship to prior work. One simple approach to estimating $\mathbf{B}$ is to vectorize the regression tensor $\mathbf{B}$ and the realizations $\{\mathbf{X}_i\}_{i=1}^m$ of the predictor tensor such that the model in (1.1) can be equivalently expressed as $y_i = \langle \text{vec}(\mathbf{X}_i), \text{vec}(\mathbf{B}) \rangle + \eta_i$, where $\text{vec}(\cdot)$ denotes the vectorization procedure. Since this reduces the original model to a vector-valued regression model, any of the traditional sparsity promoting techniques in the literature, such as forward selection/matching pursuit [23], least absolute shrinkage and selection operator (LASSO) [43], elastic net [50], adaptive LASSO [49], and Dantzig selector [8], can be employed for estimating $\mathbf{b} := \text{vec}(\mathbf{B}) \in \mathbb{R}^n$. However, a potential drawback of the vectorization operation is that the spatial correlation structure in tensor data might be lost, and a natural question is if we can explicitly exploit this structure for learning $\mathbf{B}$.

Among the various notions of tensor decompositions that capture spatial relationships among entries of a tensor, a popular decomposition is the Tucker decomposition [25, 39]. Specifically, the concept of low Tucker rank, which is the notion of rank associated with Tucker decomposition, has been successfully imposed on the regression tensor $\mathbf{B}$ for sample-efficient learning of tensor-structured regression models [13, 44, 36]. Some early convex approaches for estimating $\mathbf{B}$ in this regard were based on minimization of the sum of nuclear norms of matricizations of tensor $\mathbf{B}$ in each mode [30, 13, 44, 33]. To understand the sample complexity of such learning methods, consider the special case where the d-tuple $(r, r, \ldots, r)$ is the Tucker rank of $\mathbf{B}$ and the entries in $\mathbf{X}_i$ independently draw values from a Gaussian distribution for $i \in \{1, 2, \ldots, m\}$. Under this special case, it was shown that convex approaches based on the sum of nuclear norm minimizations require $\Omega(r\bar{n}^{(d-1)})$ samples for estimating $\mathbf{B}$ [33], where $\bar{n} := \max\{n_i : i \in \{1, 2, \ldots, d\}\}$. Since the number of degrees of freedom in $\mathbf{B}$ is on the order of $r^d + \bar{n}rd$ in this case, such a sample complexity bound is clearly suboptimal. Thus, more recently, focus has shifted to solving *nonconvex* formulations of the learning problem for various tensor-valued regression models, in the hope of achieving better sample complexity [47, 11, 36]. In one such recent work that studies the imposition of low Tucker rank on $\mathbf{B}$ [36], it was shown that $\mathbf{B}$ can be learned using $\mathcal{O}\big((r^d + \bar{n}rd) \log d\big)$ observations, which is order optimal up to a logarithmic factor.

Although the imposition of low Tucker rank on $\mathbf{B}$ allows for efficient learning, the sample

complexity requirement of $\mathcal{O}\big((r^d+\bar{n}rd)\log d\big)$ poses a linear dependence on $\bar{n}$, where this linear dependence can easily become prohibitive in many application domains. For example, consider a case from neuroimaging data analysis, where a typical MRI image has size $256 \times 256 \times 256$ with $r = 3$ and $d = 3$ [48]. Clearly, $\bar{n} \gg r$ and $\bar{n} \gg d$ in this case, and the question arises of whether we can tighten the aforementioned sample complexity bound. This goal cannot be achieved with the imposition of low Tucker rank alone on $\mathbf{B}$, since the degrees of freedom in $\mathbf{B}$, in this case, scale linearly with $\bar{n}$. Another challenge with the imposition of low Tucker rank on $\mathbf{B}$ is that the resulting regression model does not encompass the typical situation where the response depends on only a few of the (scalar) predictors in the model (i.e., the sparsity assumption). In this work, we address both of these challenges simultaneously by studying the imposition of multiple structures on $\mathbf{B}$, as explained next.

1.2. Our contributions. We study the regression model in (1.1) under the assumption that (i) the regression tensor $\mathbf{B}$ has *low* Tucker rank (to be made precise later), and (ii) the factor matrices corresponding to the Tucker decomposition of $\mathbf{B}$ are sparse. This simultaneous imposition of structure on $\mathbf{B}$ allows us to address both of the aforementioned challenges. First, the imposed sparse, low-rank structure massively reduces the number of degrees of freedom in $\mathbf{B}$, which helps get rid of the linear dependence of sample complexity on $\bar{n}$. Second, the imposition of sparsity on the factor matrices induces sparsity in the regression tensor $\mathbf{B}$, which reflects the a priori belief that the response variable typically does not depend on all the (scalar) predictors, and facilitates model interpretability. Note that this simultaneous tensor structure is reminiscent of the notion of sparse principal component analysis (PCA) from the matrix decomposition literature [51].

From a computational perspective, we formulate the problem of learning the sparse, low Tucker-rank $\mathbf{B}$ as a nonconvex problem, and we propose a projected gradient descent-based method to solve it. Furthermore, in our theoretical analysis, we show that the proposed computational procedure—under a certain restricted isometry assumption on realizations of the predictor tensor—converges linearly to an approximately correct solution. In contrast, prior works that study recovery of simultaneously structured $\mathbf{B}$ either (i) formulate a convex problem for learning the parameter tensor [35], or (ii) impose a sparse, low canonical polyadic (CP)-rank structure on $\mathbf{B}$ [16, 17], where [16] imposes a certain cubic structure on realizations of the predictor tensor and [17] lacks sample complexity guarantees.

We also evaluate the introduced restricted isometry condition for the case of independently and identically distributed (i.i.d.) sub-Gaussian (tensor-structured) predictors, and in the process, we characterize the sample complexity of parameter estimation for the case of sparse, low Tucker-rank regression tensor. We show that our sample complexity bound has only a polylogarithmic dependence on $\bar{n} := \max\{n_i : i \in \{1, 2, \dots, d\}\}$. On the other hand, in similar prior works, the sample complexity requirement has been shown to be either linear or superlinear in $\bar{n}$ [44, 33, 36]. We also employ synthetic data experiments to demonstrate the efficacy of the proposed computational procedure. Finally, we conduct real-data experiments on a collection of fMRI images pertaining to attention deficit hyperactivity disorder (ADHD) [32], and we show that the imposition of multiple structures on $\mathbf{B}$ allows for efficient neuroimaging analysis in the low sample size regime.

1.3. Notation. Bold uppercase letters ($\mathbf{Z}$), italic uppercase letters (Z), bold lowercase letters ($\mathbf{z}$), italic lowercase letters (z), and underlined italic letters ($\underline{z}$) are used to denote tensors, matrices, vectors, scalars, and tuples, respectively. For any tuple $\underline{z}$ and scalar α, we use $\alpha\underline{z}$ to denote the tuple obtained by multiplying each entry of $\underline{z}$ by α. For any scalar $q \in \mathbb{Z}_+$, we use $[[q]]$ as shorthand for $\{1, 2, \ldots, q\}$. Given any vector $\mathbf{u} \in \mathbb{R}^n$, $\|\mathbf{u}\|_0$ and $\|\mathbf{u}\|_2$ denote the ℓ_0 and ℓ_2 norms of vector $\mathbf{u}$, respectively. Given two vectors $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^n$ of the same dimension, $\mathbf{u} \circ \mathbf{v}$ denotes their outer product. Given any matrix U, the ith column is denoted by $U(:, i)$, the spectral norm is denoted by $\|U\|_2$, and $\max_i \|U(:, i)\|_2$ is denoted by $\|U\|_{1,2}$. Given any two matrices U_1 and U_2, $U_1 \otimes U_2$ denotes the Kronecker product. Given any tensor $\mathbf{Z}$, its $(i_1, i_2, \ldots, i_d)$th entry is given by $\mathbf{Z}(i_1, i_2, \ldots, i_d)$, the Frobenius norm $\|\mathbf{Z}\|_F$ is given by $\sqrt{\sum_{i_1,i_2,\ldots,i_d} \mathbf{Z}(i_1, i_2, \ldots, i_d)^2}$, the ℓ_1 norm $\|\mathbf{Z}\|_1$ is given by $\sum_{i_1,i_2,\ldots,i_d} |\mathbf{Z}(i_1, i_2, \ldots, i_d)|$, and the mode-$i$ matricization $\mathbf{Z}_{(i)}$ is the matrix obtained from column-arrangement of the mode-i fibers of $\mathbf{Z}$. The conjugate transpose of a linear map $\mathcal{X} : \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \to \mathbb{R}^m$ is denoted by $\mathcal{X}^* : \mathbb{R}^m \to \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$. Following the tensor notation in [25], for matrices $\widetilde{U}_i \in \mathbb{R}^{n_i \times r_i}$, $i \in [[d]]$, and tensor $\mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$, we define $\mathbf{S} \times_1 \widetilde{U}_1 \times_2 \widetilde{U}_2 \cdots \times_d \widetilde{U}_d$ as $\sum_{i_1,i_2,\ldots,i_d} \mathbf{S}(i_1, i_2, \ldots, i_d)\, \widetilde{U}_1(:, i_1) \circ \widetilde{U}_2(:, i_2) \circ \cdots \circ \widetilde{U}_d(:, i_d)$. Finally, $\mathbb{I}_q$ refers to an identity matrix of size q, where $q \in \mathbb{Z}_+$.

1.4. Organization. The rest of this paper is organized as follows. In section 2, we describe the regression model, which includes a formal definition of sparse, low Tucker-rank tensors, and then we present a nonconvex formulation of the problem for estimating the regression tensor. In section 3, we propose a method for solving the posed nonconvex problem, and in section 4, we provide mathematical guarantees for the proposed method, based on a certain restricted isometry property of the predictor tensors. In section 5, we evaluate the posed property for sub-Gaussian predictors and provide sample complexity bounds. In section 6, we report results of extensive numerical experiments on both synthetic and real data, while concluding remarks are presented in section 7. Finally, for the sake of space, in the supplementary material we present some of the proofs not central to understanding the implications of this work.

2. Problem formulation.

For ease of notation, let us define $\mathcal{W} := \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, $\mathcal{Y} := \mathbb{R}^m$, and let us denote the collection of tensors $\{\mathbf{X}_i\}_{i=1}^m$ in (1.1) by a linear map/measurement operator $\mathcal{X} : \mathcal{W} \to \mathcal{Y}$ such that (1.1) can be equivalently expressed as

$$\mathbf{y} = \mathcal{X}(\mathbf{B}) + \boldsymbol{\eta}, \tag{2.1}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_m]$, and $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_m]$. In this work, we impose the fact that the parameter tensor $\mathbf{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is structured in the sense that it is $\underline{r}$-rank and $\underline{s}$-sparse simultaneously. We formally define the notion of an $\underline{r}$-rank and $\underline{s}$-sparse tensor as follows.

Definition 2.1 ($\underline{r}$-rank and $\underline{s}$-sparse tensor). *Given a rank tuple $\underline{r} := (r_1, r_2, \ldots, r_d)$ and a sparsity tuple $\underline{s} := (s_1, s_2, \ldots, s_d)$, a tensor $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is said to be both $\underline{r}$-rank and $\underline{s}$-sparse if $\mathbf{Z}$ can be expressed as*

$$\mathbf{Z} = \mathbf{S} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d, \tag{2.2}$$

where $\mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$ and $U_i \in \mathbb{R}^{n_i \times r_i}$, with $\|U_i(:, j)\|_0 \le s_i$ $\forall i \in [[d]]$, $j \in [[r_i]]$. Notice that, trivially, $r_i \le n_i$ and $s_i \le n_i$.

Recall from [25] that (2.2) is expressing $\mathbf{Z}$ in terms of a Tucker decomposition, in which $\mathbf{S}$ is termed the core tensor and the U_i's are referred to as factor matrices, with additional sparsity constraints on the factor matrices. It can also be seen from (2.2) that for the special case when $s_i = n_i$, the mode-i matricization of $\mathbf{Z}$ has rank r_i: $\text{rank}(\mathbf{Z}_{(i)}) = r_i$; i.e., the $\underline{r}$-rank of $\mathbf{Z}$ is simply the Tucker rank of $\mathbf{Z}$. Further, note that we are defining sparsity of $\mathbf{Z}$ in terms of sparsity of the columns of the factor matrices $\{U_i(:, j)\}$, $i \in [[d]]$, $j \in r_i$, that are generating the tensor. This notion of sparsity is different from the conventional notion of sparsity, where it is defined as the number of nonzero entries for the data structure under consideration, i.e., tensor $\mathbf{Z}$ in this case. In contrast, the notion of sparsity in Definition 2.1 not only induces sparsity on $\mathbf{Z}$ but also dramatically reduces the number of free parameters in $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ from $n := \prod_i n_i$ to the order of $\prod_i r_i + \sum_i r_i s_i \log n_i$, which can be significantly smaller than n for $r_i \ll n_i$ and $s_i \ll n_i$. (Note that the $\log n_i$ factor arises from the need to encode the locations of the s_i nonzero entries in a given column of U_i.) This reduction in degrees of freedom allows us to learn the tensor regression model in (2.1) with lower sample complexity, as we show later.

Since we are requiring the unknown tensor $\mathbf{B}$ to be $\underline{r}$-rank and $\underline{s}$-sparse in our regression model (2.1), we formally define a set of such tensors as follows:

$$\begin{aligned} \mathcal{C} =& \{\mathbf{S} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_d U_d : \mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}, \text{ and} \\ & U_i \in \mathbb{R}^{n_i \times r_i}, \|U_i(:, j)\|_0 \le s_i, i \in [[d]], j \in [[r_i]]\}. \end{aligned} \tag{2.3}$$

Using the definition of constraint set $\mathcal{C}$, and given a known linear map $\mathcal{X}$, we can pose the following constrained optimization problem for recovery of $\mathbf{B}$ from noisy observations $\mathbf{y}$:

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{Z} \in \mathcal{C}} \frac{1}{2} \|\mathbf{y} - \mathcal{X}(\mathbf{Z})\|_2^2. \tag{2.4}$$

We can see that the optimization problem posed in (2.4) is nonconvex because of nonconvexity of the constraint set $\mathcal{C}$. In contrast, most of the prior works in tensor parameter estimation focus on solving convex relaxations of the tensor recovery problem for various notions of low-dimensional tensor structures [30, 13, 44, 33], and hence benefit from the rich literature on theory and algorithms for convex optimization. But the issue with convex relaxation-based solutions is that *convex relaxations can be suboptimal in terms of number of measurements required to solve the problem* [33]. While posing and solving the tensor recovery problem in a nonconvex form tends to circumvent this issue, it brings about difficulties in terms of theoretically characterizing the behavior of the associated recovery algorithm. In the next section, we present our proposed method for solving (2.4), while theoretical characterization of the proposed approach follows in sections 4 and 5.

3. Estimation of $\underline{r}$-rank and $\underline{s}$-sparse regression tensors.

In this section, we present a method for estimation of the structured parameter tensor $\mathbf{B}$ in the regression model (2.1), given the linear map $\mathcal{X}$, response vector $\mathbf{y}$, and the assumption that $\mathbf{B}$ is $\underline{r}$-rank and $\underline{s}$-sparse. Our method is inspired by the various projected gradient descent-based methods in the literature, where such methods have been employed for recovery of sparse vectors [7], low-rank matrices [22], and, more recently, low-rank tensors [36, 47]. The method, termed *tensor projected gradient descent* (TPGD), is summarized in Algorithm 3.1. The TPGD method

Algorithm 3.1. Tensor Projected Gradient Descent (TPGD).

1: **Input:** Linear map $\mathcal{X}$, response vector $\mathbf{y}$, step size μ, sparsity tuple $\underline{s}$, rank tuple $\underline{r}$
2: **Initialize:** Tensor $\mathbf{B}^0$ and $k \leftarrow 0$
3: **while** Stopping criterion **do**
4: $\quad \widetilde{\mathbf{B}}^k \leftarrow \mathbf{B}^k - \mu \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y})$
5: $\quad \mathbf{B}^{k+1} \leftarrow \mathcal{H}(\widetilde{\mathbf{B}}^k)$
6: $\quad k \leftarrow k + 1$
7: **end while**
8: **return** Tensor $\mathbf{B}^* = \mathbf{B}^k$

consists of two steps. First, we perform gradient descent iteration over the objective function in (2.4) (step 4, Algorithm 3.1), and then we project the iterate onto set $\mathcal{C}$, which is the set of $\underline{r}$-rank and $\underline{s}$-sparse tensors (step 5, Algorithm 3.1). The projection operator, $\mathcal{H} : \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, in step 5 of Algorithm 3.1 is defined as

$$\mathcal{H}(\widetilde{\mathbf{B}}) := \arg\min_{\widehat{\mathbf{B}} \in \mathcal{C}} \|\widetilde{\mathbf{B}} - \widehat{\mathbf{B}}\|_F^2. \tag{3.1}$$

In general, computation of the best low-rank approximation of a given tensor is considered to be an NP-hard problem [19, 36]. Despite that, several algorithms have been proposed in the literature for computing low-rank tensor approximations corresponding to various notions of tensor decompositions [25, 2, 14, 40]. Although these approximation algorithms do not come with mathematical guarantees regarding the accuracy of tensor approximation, they have been employed successfully in practice for tensor approximation within various methods for estimating tensor-structured parameters in regression models [47, 36]. The mathematical guarantees for these parameter estimation methods *assume* the goodness of the tensor approximation step, since the corresponding approximation algorithms are not guaranteed to compute the best approximation.

In a similar vein, in the mathematical guarantees for Algorithm 3.1 (section 4), we *assume* that the best low-rank and sparse approximation (projection step in step 5, Algorithm 3.1) can be exactly computed. However, in our numerical simulations (section 6), we employ Algorithm 3.2 for computation of the projection step, where Algorithm 3.2 is essentially the sparse higher-order singular value decomposition (SVD) method [2]. Moreover, within Algorithm 3.2, we employ [18] for computation of the factor matrices $\{\bar{U}_j\}_{j=1}^d$ (step 3, Algorithm 3.2). Later, in section 6, our numerical simulations show that Algorithm 3.2 can indeed be effectively employed with Algorithm 3.1 to efficiently learn the regression model in (2.1) under certain conditions despite the lack of mathematical guarantees for Algorithm 3.2.

4. Convergence analysis of tensor projected gradient descent.

In this section we provide theoretical guarantees for TPGD (Algorithm 3.1), which, as explained earlier, is a projected gradient method for solving (2.4). Variants of the projected gradient method have been analyzed for recovery of sparse vectors [7], low-rank matrices [22], and low-rank tensors [47, 36, 11] under the assumption that the linear map/measurement operator satisfies some variant of the restricted isometry property (RIP) [9]. Since different tensor decompositions induce different notions of tensor rank [36, 16], and different regression models lead to different

Algorithm 3.2. Sparse Higher-Order SVD.

Input: Tensor $\widetilde{\mathbf{B}}$, sparsity tuple $\underline{s}$, rank tuple $\underline{r}$
for $j = 1, \dots, d$ **do**
  $\bar{U}_j \leftarrow$ First r_j, s_j-sparse principal components of $\widetilde{\mathbf{B}}_{(j)}$
end for
$\bar{\mathbf{S}} \leftarrow \widetilde{\mathbf{B}} \times_1 \bar{U}_1 \times_2 \bar{U}_2 \times_3 \cdots \times_d \bar{U}_d$
return Tensor $\bar{\mathbf{B}} = \bar{\mathbf{S}} \times_1 \bar{U}_1 \times_2 \bar{U}_2 \times_3 \cdots \times_d \bar{U}_d$

measurement operators [36, 47], various notions of RIP have also been posed for various tensor decompositions and regression models. Before we present the notion of RIP assumed on the linear map in this work, let us define a set of $\underline{r}$-rank and $\underline{s}$-sparse tensors, with additional constraints on (i) the ℓ_1 norm of the associated core tensor, and (ii) the ℓ_2 norm of columns of the associated factor matrices:

$$\begin{aligned}\mathcal{G}_{\underline{r},\underline{s},\tau} =& \{\mathbf{S} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_d U_d : \mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}, \|\mathbf{S}\|_1 \le \tau, \text{ and} \\ & U_i \in \mathbb{R}^{n_i \times r_i}, \|U_i(:,j)\|_0 \le s_i, \|U_i(:,j)\|_2 \le 1, i \in [[d]], j \in [[r_i]]\}. \end{aligned} \tag{4.1}$$

From the ℓ_1 norm constraint on $\mathbf{S}$ and ℓ_2 norm constraint on $U_i(:,j)$, where $i \in [[d]]$ and $j \in [[r_i]]$, it follows that $\|Z\|_F \le \tau$ for any $Z \in \mathcal{G}_{\underline{r},\underline{s},\tau}$. These norm constraints in (4.1) allow us to bound the covering number of the set $\mathcal{G}_{\underline{r},\underline{s},\tau}$, which enables us to obtain a sample complexity bound for tensor recovery, as follows in the next section. Specifically, in order to derive a bound on the covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$ in the next section, our mathematical analysis requires bounds on $\|\mathbf{S}\|_1$, $\|Z\|_F$, and $\|U_i(:,j)\|_2$ for $i \in [[d]]$, $j \in [[r_i]]$. Since the ℓ_1 norm constraint on $\mathbf{S}$ and the ℓ_2 norm constraint on $U_i(:,j)$ result in $\|Z\|_F \le \tau$, the constraints in (4.1) suffice to evaluate a bound on the covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$.

For the recovery of $\underline{r}$-rank and $\underline{s}$-sparse tensors considered in this work, we consider the following notion of RIP on the linear map $\mathcal{X}$.

Definition 4.1 ($(\underline{r}, \underline{s}, \tau, \delta_{\underline{r},\underline{s},\tau})$-restricted isometry property). *The restricted isometry constant $\delta_{\underline{r},\underline{s},\tau} \in (0,1)$ of a linear map $\mathcal{X} : \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \to \mathbb{R}^m$ acting on tensors of order d is the smallest quantity such that*

$$(1 - \delta_{\underline{r},\underline{s},\tau})\|\mathbf{Z}\|_F^2 \le \|\mathcal{X}(\mathbf{Z})\|_2^2 \le (1 + \delta_{\underline{r},\underline{s},\tau})\|\mathbf{Z}\|_F^2 \tag{4.2}$$

for all tensors $\mathbf{Z} \in \mathcal{G}_{\underline{r},\underline{s},\tau}$.

In the following we provide our first main theoretical result that characterizes the convergence behavior of TPGD under the assumption of an exact projection step (step 5, Algorithm 3.1).

Theorem 4.2 (convergence of TPGD). *Let $\mathbf{y} = \mathcal{X}(\mathbf{B}) + \boldsymbol{\eta}$, and let $\mathbf{B}^0 \in \mathcal{G}_{\underline{r},\underline{s},\tau}$ be the tensor initialization in Algorithm 3.1. For some fixed $\gamma \in (0,1)$, suppose $\mathcal{X} : \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \to \mathbb{R}^m$ satisfies RIP in Definition 4.1 with $\delta_{2\underline{r},\underline{s},2\tau} < \frac{\gamma}{4+\gamma}$. Then, fixing the step size $\mu = \frac{1}{1+\delta_{2\underline{r},\underline{s},2\tau}}$ and defining $b := \frac{1+3\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}$, the estimation error in TPGD algorithm's (Algorithm 3.1) iterate,*

$\mathbf{B}^k$, *after* k *iterations is given by*

$$\|\mathbf{B}^k - \mathbf{B}\|_F^2 \leq \frac{2\gamma^k}{1-\delta_{2\underline{r},\underline{s},2\tau}} \left\|\mathbf{y} - \mathcal{X}(\mathbf{B}^0)\right\|_2^2 + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big(1 + \frac{b}{1-\gamma}\Big).$$

4.1. Discussion of Theorem 4.2. Let $c_0 := \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\underline{r},\underline{s},2\tau}}(1 + \frac{b}{1-\gamma})$. Next, define the closed ball $\mathcal{B}(c_0, \mathbf{B})$ with center at $\mathbf{B}$ and radius c_0 as the set of all the tensors $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ such that $\|\mathbf{Z} - \mathbf{B}\|_F^2 \leq c_0$. Theorem 4.2 shows that starting from an initial estimate $\mathbf{B}^0$, the solution of TPGD converges linearly to the set $\mathcal{B}(c_0, \mathbf{B})$ at the rate of γ^k. Additionally, Theorem 4.2 also characterizes the impact of noise power $\|\boldsymbol{\eta}\|_2^2$ and RIP constant $\delta_{2\underline{r},\underline{s},2\tau}$ on the convergence behavior of the TPGD algorithm. First, the radius of ball $\mathcal{B}(c_0, \mathbf{B})$ scales linearly with the noise power $\|\boldsymbol{\eta}\|_2^2$. Thus, the more noise power, the less accurate the solution of TPGD and vice versa. Second, Theorem 4.2 shows that the smaller the RIP constant $\delta_{2\underline{r},\underline{s},2\tau}$, the smaller the radius of ball $\mathcal{B}(c_0, \mathbf{B})$. Thus, the larger the value of $\delta_{2\underline{r},\underline{s},2\tau}$, the less accurate may the solution of TPGD be and vice versa. We conclude by noting that although the mathematical guarantees in Theorem 4.2 depend on the $(\underline{r}, \underline{s}, \tau, \delta_{\underline{r},\underline{s},\tau})$-RIP property in Definition 4.1, we evaluate this property for a known family of linear maps in the next section.

4.2. Remarks on the proof of Theorem 4.2. A key step in proving Theorem 4.2 is to show that any linear combination of two $\underline{r}$-rank and $\underline{s}$-sparse tensors has rank at most $2\underline{r}$ and sparsity $\underline{s}$. We formally describe this in terms of a lemma that appears in the analysis of any step that involves a linear combination of $\underline{r}$-rank and $\underline{s}$-sparse tensors.

Lemma 4.3. *Let* $\mathbf{Z}_a \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ *and* $\mathbf{Z}_b \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ *be members of the set* $\mathcal{G}_{\underline{r},\underline{s},\tau}$, *where* $\underline{r} := (r_1, r_2, \ldots, r_d)$, $\underline{s} := (s_1, s_2, \ldots, s_d)$, *and* $\tau \in \mathbb{R}^+$. *Define* $\mathbf{Z}_c = \gamma_a \mathbf{Z}_a + \gamma_b \mathbf{Z}_c$, *where* $\gamma_a, \gamma_b \in \mathbb{R}$. *Then,* $\mathbf{Z}_c$ *is a member of the set* $\mathcal{G}_{2\underline{r},\underline{s},\kappa}$, *where* $\kappa = (|\gamma_a| + |\gamma_b|)\tau$.

The proof of this lemma is provided in Appendix A, and the proof of Theorem 4.2 follows in Appendix B.

5. Evaluating the restricted isometry property for sub-Gaussian linear maps. In the previous section, we provided theoretical guarantees for recovery of the parameter tensor $\mathbf{B}$ using the TPGD method, based on the assumption of the RIP (Definition 4.1). In this section, we provide examples of linear maps that satisfy this property. Specifically, in (2.1) we consider linear maps $\mathcal{X}$ that denote the collection of tensors in (1.1), $\{\mathbf{X}_i\}_{i=1}^m$, such that the entries of each $\mathbf{X}_i$ are independently drawn from zero-mean, unit-variance sub-Gaussian distributions. We call such linear maps sub-Gaussian linear maps. Before we evaluate the condition in Definition 4.1 for these maps, let us recall the definition of a sub-Gaussian random variable.

Definition 5.1. *A zero-mean random variable* $\mathcal{Z}$ *is said to follow a sub-Gaussian distribution* $subG(\alpha)$ *if there exists a sub-Gaussian parameter* $\alpha > 0$ *such that* $\mathbb{E}[\exp(\lambda \mathcal{Z})] \leq \exp(\frac{\alpha^2 \lambda^2}{2})$ *for all* $\lambda \in \mathbb{R}$.

In words, a $subG(\alpha)$ random variable is one whose moment generating function is dominated by that of a Gaussian random variable. Some common examples of sub-Gaussian random variables include

- $\mathcal{Z} \sim \mathcal{N}(0, \alpha^2) \;\Rightarrow\; \mathcal{Z} \sim subG(\alpha)$,
- $\mathcal{Z} \sim \text{unif}(-\alpha, \alpha) \;\Rightarrow\; \mathcal{Z} \sim subG(\alpha)$,

- $|\mathcal{Z}| \leq \alpha, \mathbb{E}[\mathcal{Z}] = 0 \;\Rightarrow\; \mathcal{Z} \sim subG(\alpha)$,
- $\mathcal{Z} \sim \begin{cases} \alpha, & \text{with prob. } \frac{1}{2}, \\ -\alpha, & \text{with prob. } \frac{1}{2}, \end{cases} \;\Rightarrow\; \mathcal{Z} \sim subG(\alpha)$.

We now evaluate the RIP (Definition 4.1) for sub-Gaussian linear maps. An outline of the proof of the following result is provided in section 5.2, while its detailed proof is presented in the supplementary material.

Theorem 5.2. *Let the entries of $\{\mathbf{X}_i\}_{i=1}^m$ be independently drawn from zero-mean, $\frac{1}{m}$-variance subG(α) distributions. Define $\bar{n} := \max\{n_i : i \in [[d]]\}$. Then, for any $\delta, \varepsilon \in (0,1)$, the linear map $\mathcal{X}$ satisfies $\delta_{\underline{r},\underline{s},\tau} \leq \delta$ with probability at least $1-\varepsilon$ as long as*

$$m \geq \delta^{-2} \max\left\{ K_1 \tau^2 \left(\prod_{i=1}^{d} r_i + \sum_{i=1}^{d} s_i r_i \right) \Big(\log(3\bar{n}d) \Big)^2,\; K_2 \log(\varepsilon^{-1}) \right\},$$

where the constants K_1, $K_2 > 0$ depend on τ and α.

5.1. Discussion. We compare the result in Theorem 5.2 with sample complexity bounds from the literature for estimation of the parameter tensor $\mathbf{B}$ in (2.1). Theoretically, we can pose the estimation problem as (i) low Tucker-rank recovery problem [36], or (ii) sparse recovery problem [37]. Thus, in this section, we first compare the sample complexity bound in Theorem 5.2 with complexity bounds from the low-rank recovery and sparse recovery literature. For ease of comparison, define $\bar{r} := \max\{r_1, r_2, \ldots, r_d\}$ and $\bar{s} := \max\{s_1, s_2, \ldots, s_d\}$. With these definitions, the sample requirement in Theorem 5.2 can be written as $\mathcal{O}\big((\bar{r}^d + \bar{s}\,\bar{r}\,d)(\log(3\,\bar{n}\,d))^2\big)$. We now compare this result with complexity bounds from prior works.

5.1.1. Low Tucker-rank recovery. Among the many works that study the problem of estimating $\mathbf{B}$ under the imposition of low Tucker rank on $\mathbf{B}$ [13, 44, 33, 36], the most tight sample complexity bound has been shown to be $\mathcal{O}\big((\bar{r}^d + \bar{n}\,\bar{r}\,d)\log(d)\big)$ [36]. If we apply this complexity bound for estimating the parameter tensor $\mathbf{B}$ in (2.1), the sample complexity requirement scales linearly with $\bar{n}$. In contrast, since we consider sparsity on columns of the factor matrices within Tucker decomposition of $\mathbf{B}$, our sample complexity bound has a linear dependence on $\bar{s}$ and only a polylogarithmic dependence on $\bar{n}$, where $\bar{s} \ll \bar{n}$.

5.1.2. Sparse recovery. The regression model in (2.1) or, equivalently, the model in (1.1), can be vectorized such that the model can be expressed as $y_i = \langle \mathrm{vec}(\mathbf{X}_i), \mathrm{vec}(\mathbf{B})\rangle + \eta_i$, $i \in [[m]]$, and the problem of recovering $\mathbf{B}$ can be posed as a sparse recovery problem. It has been shown that if the entries of $\mathrm{vec}(\mathbf{X}_i)$, $i \in [[m]]$, draw values from a Gaussian distribution, $\mathrm{vec}(\mathbf{B})$ can be recovered using $\mathcal{O}(k \log(\bar{n}^d/k))$ samples [3], where k is the number of nonzero entries in $\mathrm{vec}(\mathbf{B})$. The number of nonzero entries in $\mathrm{vec}(\mathbf{B})$ are upper bounded by $(\bar{s}\,\bar{r})^d$, which leads to the worst-case sample complexity requirement of $\mathcal{O}(d\,(\bar{s}\,\bar{r})^d \log(\bar{n}/\bar{s}\,\bar{r}))$. Thus, the sparse signal recovery literature poses a worst-case sample complexity requirement that has linear dependence on $d\,(\bar{s}\,\bar{r})^d$. In contrast, since we consider the multidimensional structure within $\mathbf{B}$, our sample complexity requirement has only linear dependence on $\bar{r}^d + \bar{s}\,\bar{r}\,d$.

Finally, note that the number of free parameters in the parameter tensor $\mathbf{B}$ is on the order of $\prod_i r_i + \sum_i r_i s_i \log n_i$, where the $\log n_i$ factor encodes for the s_i nonzero entries in each of

the r_i columns of the ith factor matrix of the tensor $\mathbf{B}$. More compactly, this number of free parameters can be expressed as $\bar{r}^d + \bar{s}\,\bar{r}\,d\log\bar{n}$. Thus, the posed sample complexity requirement of $\mathcal{O}\big((\bar{r}^d + \bar{s}\,\bar{r}\,d)\big(\log(3\,\bar{n}\,d)\big)^2\big)$ in Theorem 5.2 is order-optimal up to a polylogarithmic factor.

5.2. Outline of the proof. The general idea of the proof of Theorem 5.2 is similar to that of [22, Theorem 4.2], [10, Theorem 2.3], [26, Theorem 4.1], and [36, Theorem 2], where the main analytic challenge is to analyze the complexity of the set that is hypothesized to contain the regression parameters. In this work, the challenge translates into characterizing the complexity of the set $\mathcal{G}_{\underline{r},\underline{s},\tau}$, for which we employ the notion of ϵ-nets and covering numbers, defined as follows.

Definition 5.3 (ϵ-nets and covering numbers). *Let $(\mathbb{V}, h)$ be a metric space, and let $\mathbb{T} \subset \mathbb{V}$. The set $X \subset \mathbb{T}$ is called an ϵ-net of $\mathbb{T}$ with respect to the metric h if for any $T_i \in \mathbb{T}$, $\exists X_i \in X$ such that $h(X_i, T_i) \leq \epsilon$. The minimum cardinality of an ϵ-net of $\mathbb{T}$ (with respect to the metric h) is called the covering number of $\mathbb{T}$ with respect to the metric h and is denoted by $\Psi(\mathbb{T}, h, \epsilon)$ in this paper.*

Next, we provide an outline to the proof of Theorem 5.2. In the first step, we provide an upper bound on the covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$ with respect to the Frobenius norm, which forms our main contribution. In the second step, we employ a deviation bound from prior works [36, 26] to complete the proof of this theorem. A formal proof of Theorem 5.2 is given in section SM3 of the supplementary material.

5.2.1. Bound on covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$. The following lemma provides a bound on the covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$ with respect to the Frobenius norm.

Lemma 5.4. *For tuples $\underline{r} := (r_1, r_2, \ldots, r_d)$, $\underline{s} := (s_1, s_2, \ldots, s_d)$ and for any $\tau > 0$, the covering number of*

$$\begin{aligned}\mathcal{G}_{\underline{r},\underline{s},\tau} = \{&\mathbf{S} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_d U_d : \mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}, \|\mathbf{S}\|_1 \leq \tau, \text{ and} \\ &U_i \in \mathbb{R}^{n_i \times r_i}, \|U(:,j)\|_2 \leq 1, \|U_i(:,j)\|_0 \leq s_i, i \in [[d]], j \in [[r_i]]\}\end{aligned}$$

with respect to the metric $h_{\mathcal{G}}$ satisfies

$$\Psi(\mathcal{G}_{\underline{r},\underline{s},\tau}, h_{\mathcal{G}}, \epsilon) \leq \Big(\frac{3\tau(d+1)}{\epsilon}\Big)^{\prod_{i=1}^{d} r_i} \Big(\frac{3\bar{n}\tau(d+1)}{\epsilon}\Big)^{\sum_{i=1}^{d} s_i r_i}, \quad \epsilon \in (0,1),$$

where $\bar{n} := \max\{n_i : i \in [[m]]\}$ and $h_{\mathcal{G}}(\mathbf{G}^{(1)}, \mathbf{G}^{(2)}) = \|\mathbf{G}^{(1)} - \mathbf{G}^{(2)}\|_F$ for any $\mathbf{G}^{(1)}, \mathbf{G}^{(2)} \in \mathcal{G}_{\underline{r},\underline{s},\tau}$.

Let us provide an outline to the proof of Lemma 5.4, while a formal proof is provided in section SM1 of the supplementary material. Define the Cartesian product of metric spaces $(\mathcal{D}_{\mathbf{S}}, h_{\mathbf{S}})$, $(\mathcal{D}_{U_1}, h_{U_1})$, $(\mathcal{D}_{U_2}, h_{U_2})$, $\ldots$, $(\mathcal{D}_{U_d}, h_{U_d})$, that is,

$$\mathcal{D}_P := \mathcal{D}_{\mathbf{S}} \times \mathcal{D}_{U_1} \times \mathcal{D}_{U_2} \times \cdots \times \mathcal{D}_{U_d}, \tag{5.1}$$

where $\mathcal{D}_{\mathbf{S}} := \{\mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d} : \|\mathbf{S}\|_1 \leq \tau\}$, $h_{\mathbf{S}}(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}) := \frac{1}{\tau}\|\mathbf{S}^{(1)} - \mathbf{S}^{(2)}\|_1$ for any $\mathbf{S}^{(1)}, \mathbf{S}^{(2)} \in \mathcal{D}_{\mathbf{S}}$, $\mathcal{D}_{U_i} := \{U \in \mathbb{R}^{n_i \times r_i} : \|U(:,j)\|_2 \leq 1, \|U(:,j)\|_0 \leq s_i, j \in [[r_i]]\}$, and

$h_{U_i}(U_i^{(1)}, U_i^{(2)}) = \|U_i^{(1)} - U_i^{(2)}\|_{1,2}$ for any $U_i^{(1)}, U_i^{(2)} \in \mathcal{D}_{U_i} \forall i \in [[d]]$. First, we need to compute an upper bound on the covering number of $\mathcal{D}_P$ with respect to the metric h_P defined as

$$h_P(P^{(1)}, P^{(2)}) = \max\left\{ \max_{i\in[[d]]} \{h_{U_i}(U_i^{(1)}, U_i^{(2)})\}, h_{\mathbf{S}}(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}) \right\}, \tag{5.2}$$

where $P^{(1)}, P^{(2)} \in \mathcal{D}_P$, $\mathbf{S}^{(1)}, \mathbf{S}^{(2)} \in \mathcal{D}_{\mathbf{S}}$, and $U_i^{(1)}, U_i^{(2)} \in \mathcal{D}_{U_i}$, $i \in [[d]]$. Specifically, using Lemma E.2, a bound on $\Psi(\mathcal{D}_P, h_P, \epsilon)$ can be obtained as

$$\Psi(\mathcal{D}_P, h_P, \epsilon) \leq \Psi(\mathcal{D}_{\mathbf{S}}, h_{\mathbf{S}}, \epsilon) \prod_{i=1}^{d} \Psi(\mathcal{D}_{U_i}, h_{U_i}, \epsilon). \tag{5.3}$$

Thus, to compute an upper bound on $\Psi(\mathcal{D}_P, h_P, \epsilon)$, we need upper bounds on $\Psi(\mathcal{D}_{\mathbf{S}}, h_{\mathbf{S}}, \epsilon)$ and $\Psi(\mathcal{D}_{U_i}, h_{U_i}, \epsilon)$, respectively. To obtain a bound on $\Psi(\mathcal{D}_{\mathbf{S}}, h_{\mathbf{S}}, \epsilon)$, we employ the following lemma, which is proved in Appendix C.

Lemma 5.5. *Define* $\mathcal{D}_{\mathbf{S}} := \{\mathbf{S} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d} : \|\mathbf{S}\|_1 \leq \tau\}$ *with distance measure* $\|\,.\,\|_1$. *Then the covering number of* $\mathcal{D}_{\mathbf{S}}$ *(with respect to the norm* $\|\,.\,\|_1$*) satisfies the bound*

$$\Psi(\mathcal{D}_{\mathbf{S}}, \|\,.\,\|_1, \epsilon) \leq \left(\frac{3\,\tau}{\epsilon}\right)^{\prod_{i=1}^{d} r_i}, \epsilon \in (0, 1).$$

Similarly, to obtain a bound on $\Psi(\mathcal{D}_{U_i}, h_{U_i}, \epsilon)$ for any $i \in [[d]]$, we employ the following lemma, which is proved in Appendix D.

Lemma 5.6. *Define* $\mathcal{D}_U := \{U \in \mathbb{R}^{n\times r} : \|U(:,j)\|_2 \leq 1, \|U(:,j)\|_0 \leq s \ \forall j \in [[r]]\}$ *with distance measure* h_U, *where* $h_U(U^{(1)}, U^{(2)}) = \|U^{(1)} - U^{(2)}\|_{1,2}$ *for any* $U^{(1)}, U^{(2)} \in \mathcal{D}_U$. *Then the covering number of* $\mathcal{D}_U$ *with respect to the metric* h_U *satisfies the bound*

$$\Psi(\mathcal{D}_U, h_U, \epsilon) \leq \left(\frac{3n}{\epsilon}\right)^{sr}, \quad \epsilon \in (0, 1).$$

Therefore, the bound in (5.3) is evaluated using Lemmas 5.5 and 5.6.

Given a bound on $\Psi(\mathcal{D}_P, h_P, \epsilon)$ from (5.3), we are ready to derive a bound on the covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$ with respect to the metric $h_{\mathcal{G}}$. To this end, define a mapping Φ such that

$$\Phi(\mathbf{S}, U_1, U_2, \ldots, U_d) = \mathbf{S} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_d U_d,$$

where $(\mathbf{S}, U_1, U_2, \ldots, U_d) \in \mathcal{D}_P$. Note from this definition that $\Phi : \mathcal{D}_P \to \mathcal{G}_{\underline{r},\underline{s},\tau}$. We now employ the following lemma, which is formally proved in section SM2 of the supplementary material, to establish that this mapping Φ is Lipschitz with a Lipschitz constant of $\tau\,(d+1)$.

Lemma 5.7. *Consider* $(\mathcal{D}_P, h_P)$ *to be the Cartesian product of metric spaces* $(\mathcal{D}_{\mathbf{S}}, h_{\mathbf{S}})$, $(\mathcal{D}_{U_1}, h_{U_1})$, $(\mathcal{D}_{U_2}, h_{U_2})$, $\ldots$, $(\mathcal{D}_{U_d}, h_{U_d})$, *as defined in* (5.1) *and* (5.2). *Define mapping* Φ *such that*

$$\Phi(\mathbf{S}, U_1, U_2, \ldots, U_d) = \mathbf{S} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_d U_d,$$

where $(\mathbf{S}, U_1, U_2, \ldots, U_d) \in \mathcal{D}_P$. *Further, define metric space* $(\mathcal{G}_{\underline{r},\underline{s},\tau}, h_{\mathcal{G}})$, *where* $\mathcal{G}_{\underline{r},\underline{s},\tau}$ *is defined as in* (4.1), *and* $h_{\mathcal{G}}(\mathbf{G}^{(1)}, \mathbf{G}^{(2)}) = \|\mathbf{G}^{(1)} - \mathbf{G}^{(2)}\|_F$ *for any* $\mathbf{G}^{(1)}, \mathbf{G}^{(2)} \in \mathcal{G}_{\underline{r},\underline{s},\tau}$. *Then, given* $P^{(1)}, P^{(2)} \in \mathcal{D}_P$, *we have*

$$h_{\mathcal{G}}(\Phi(P^{(1)}), \Phi(P^{(2)})) \leq \tau(d+1) h_P(P^{(1)}, P^{(2)}). \tag{5.4}$$

Finally, the application of Lemma 5.7 with (5.3) and Lemma E.3, where (5.3) follows from Lemma E.2, establishes the statement of Lemma 5.4.

5.2.2. Deviation bound. Since $\delta_{\underline{r},\underline{s},\tau} = \sup_{\mathbf{Z} \in \mathcal{G}_{\underline{r},\underline{s},\tau}} \left| \|\mathcal{X}(\mathbf{Z})\|_2^2 - \mathbb{E}\big[\|\mathcal{X}(\mathbf{Z})\|_2^2\big] \right|$, we derive a probabilistic bound on the right-hand side of this equality to evaluate the condition in (4.1). To this end, we use techniques similar to those in [36, 26]. Specifically, define $\boldsymbol{\xi}$ to be a random vector in $\mathbb{R}^{n_1 n_2 \cdots n_d m}$ with independent entries from zero-mean, unit-variance, *subG*(B) random variables. Further, let $\mathbf{Z} \in \mathcal{G}_{\underline{r},\underline{s},\tau}$, and define $V_{\mathbf{Z}}$ to be a matrix in $\mathbb{R}^{m \times n_1 n_2 \cdots n_d m}$ such that

$$V_{\mathbf{Z}} = \frac{1}{\sqrt{m}} \; \mathbb{I}_m \otimes \mathbf{z}^\top,$$

where $\mathbf{z} \in \mathbb{R}^{n_1 n_2 \cdots n_d \times 1}$ is the vectorized version of $\mathbf{Z}$. Then, we have the equivalence relationship $\mathcal{X}(\mathbf{Z}) = V_{\mathbf{Z}} \boldsymbol{\xi}$. For ease of notation, let us further define a set $\mathcal{M} := \{V_{\mathbf{Z}} : \mathbf{Z} \in \mathcal{G}_{\underline{r},\underline{s},\tau}\}$. With this additional notation, we have $\delta_{\underline{r},\underline{s},\tau} = \sup_{M \in \mathcal{M}} \left| \|M\boldsymbol{\xi}\|_2^2 - \mathbb{E}\big[\|M\boldsymbol{\xi}\|_2^2\big] \right|$, and we apply the following theorem to obtain a deviation bound on the right-hand side of this equality.

Theorem 5.8 ([26, 36]). *Let* $\mathcal{M}_0$ *be a set of matrices, and let* $\boldsymbol{\xi}_{\mathbf{0}}$ *be a random vector with independent entries from zero-mean, unit-variance, subG*(α_0) *random variables. For the set* $\mathcal{M}_0$, *define*

$$d_F(\mathcal{M}_0) := \sup_{M \in \mathcal{M}_0} \|M\|_F, \;\; d_{2\to 2}(\mathcal{M}_0) := \sup_{M \in \mathcal{M}_0} \|M\|_2,$$
$$\text{and } d_4(\mathcal{M}_0) := \sup_{M \in \mathcal{M}_0} \|M\|_{S_4} = \sup_{M \in \mathcal{M}_0} \Big(tr\big[(M^\top M)^2\big]\Big)^{\frac{1}{4}}.$$

Furthermore, let $\gamma_2(\mathcal{M}_0, \|\,.\,\|_2)$ *be the Talagrand's* γ_2*-functional* [42]. *Finally, set*

$$E_0 = \gamma_2(\mathcal{M}_0, \|\,.\,\|_2)(\gamma_2(\mathcal{M}_0, \|\,.\,\|_2) + d_F(\mathcal{M}_0)) + d_F(\mathcal{M}_0) d_{2\to 2}(\mathcal{M}_0),$$
$$V_0 = d_4^2(\mathcal{M}_0), \text{ and } U_0 = d_{2\to 2}^2(\mathcal{M}_0).$$

Then, for $t > 0$,

$$\mathbb{P}\bigg(\sup_{M \in \mathcal{M}_0} \Big| \|M\boldsymbol{\xi}\|_2^2 - \mathbb{E}\big[\|M\boldsymbol{\xi}\|_2^2\big] \Big| \geq c_3 E_0 + t \bigg) \leq 2 \exp\bigg(- c_4 \min\Big\{ \frac{t^2}{V_0^2}, \frac{t}{U_0} \Big\} \bigg),$$

where the positive constants c_3, c_4 *depend on* α_0.

For the application of Theorem 5.8, we need to evaluate bounds on the metrics $d_F(\mathcal{M})$, $d_{2\to 2}(\mathcal{M})$, $d_4(\mathcal{M})$, and $\gamma_2(\mathcal{M}, \|\,.\,\|_2)$. However, the main analytical challenge in this application is evaluation of a bound on the Talagrand's γ_2-functional $\gamma_2(\mathcal{M}, \|\,.\,\|_2)$, which encompasses a geometric characterization of the metric space $(\mathcal{M}, \|\,.\,\|_2)$. We obtain a bound on the Talagrand's γ_2-functional using the following inequality [42, 36]:

$$\gamma_2(\mathcal{M}, \|\,.\,\|_2) \leq C \int_0^{d_{2\to 2}(\mathcal{M})} \sqrt{\log \Psi(\mathcal{M}, \|\,.\,\|_2, \epsilon)}\, d\epsilon, \tag{5.5}$$

where $C > 0$, and $\Psi(\mathcal{M}, \|\,.\,\|_2, u)$ denotes the covering number of the metric space $(\mathcal{M}, \|\,.\,\|_2)$ with respect to the metric $\|\,.\,\|_2$. Thus, we employ the bound on covering number of $\mathcal{G}_{\underline{r},\underline{s},\tau}$ from Lemma 5.4 to evaluate (5.5), which enables us to obtain a bound on $\sup_{M\in\mathcal{M}} \Big| \|M\boldsymbol{\xi}\|_2^2 - \mathbb{E}\big[\|M\boldsymbol{\xi}\|_2^2\big] \Big|$ using Theorem 5.8. A formal proof of Theorem 5.2 is presented in section SM3 of the supplementary material.

6. Numerical experiments.

In this section, we perform experiments on synthetic and real-world data to analyze the performance of the proposed TPGD method (Algorithm 3.1), which, as explained before, is a tensor variant of the projected gradient descent (PGD) method. We compare TPGD with learning methods based on (i) vectorization of the parameter tensor, (ii) imposition of low Tucker-rank, and (iii) imposition of low CP-rank [25] on the parameter tensor $\mathbf{B}$. To analyze linear vectorization-based methods, we employ LASSO [43] and linear support vector machine regression (SVR) [21]. To analyze imposition of low Tucker-rank and low CP-rank, we employ Tucker-rank and CP-rank variants of the TPGD method, respectively. Specifically, in the first variant, projection is performed on a set of low Tucker-rank tensors [36], and we call this method PGD-Tucker. In the second variant, projection is performed on a set of low CP-rank tensors [48], and we call this method PGD-CP. Thus, we draw comparisons of TPGD (Algorithm 3.1) with LASSO, SVR, PGD-Tucker, and PGD-CP.

Some relevant implementation details for these learning methods are as follows. For computation of the projection step $\mathcal{H}$ in Algorithm 3.1, we employ Algorithm 3.2, within which we employ the inverse power method from [18] for computation of step 3. For computation of the projection steps in the Tucker-rank (PGD-Tucker) and the CP-rank (PGD-CP) based methods, we employ the tensor toolboxes in [45] and [4], respectively. Finally, we employ the MATLAB built-in `fitrlinear` function [31] for implementing LASSO and SVR methods.

6.1. Synthetic experiments.

For synthetic-data experiments, we generate the $\underline{r}$-rank and $\underline{s}$-sparse tensor $\mathbf{B} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ in (2.1) as follows. We set $d = 3$, $n_1 = 50$, $n_2 = 50$, and $n_3 = 30$, and in (2.2), we set $s_1 = 6$, $s_2 = 6$, and $s_3 = 4$, with $r_1 = 3$, $r_2 = 3$, and $r_3 = 3$. For each $j \in [[d]]$, we generate the column vector $U_j(:, i)$, for each $i \in [[r]]$, such that $\|U_j(:, i)\|_0 \leq s_j$. The locations of the s_j nonzero entries in $U_j(:, i)$ are chosen uniformly at random from $[[n_j]]$. Setting $a = 0.5$, we sample the nonzero entries in $U_j(:, i)$ from $(-1)^u(a + |z|)$, where u is drawn from a Bernoulli distribution with parameter 0.5, and z is drawn from a standard Gaussian distribution, i.e., Gaussian$(0, 1)$. Finally, to generate the parameter tensor $\mathbf{B}$, the entries of the core tensor $\mathbf{S}$ are sampled from a uniform distribution with parameters 0 and 1, and the tensor $\mathbf{B}$ is generated as in (2.2). To generate the response vector $\mathbf{y}$, the tensors

$\{\mathbf{X}_i\}_{i=1}^m$ are generated such that their entries are i.i.d. Gaussian$(0, 1/m)$, the noise vector $\boldsymbol{\eta}$ is sampled from Gaussian$(0, \sigma_z^2 I)$, and then the response vector $\mathbf{y}$ is generated as in (2.1).

The aforementioned experiment is performed for various values of m, repeating each experiment for increasing value of σ_z to analyze the impact of increasing noise power. For each value of m and σ_z, (i) the parameter tensor $\mathbf{B}$, the linear map $\mathcal{X}$, and the response vector $\mathbf{y}$ are generated as explained above; and (ii) a parameter estimate $\mathbf{B}^*$ is computed using each of the learning methods. The algorithmic parameters for each of the learning methods are set using separate validation experiments. The performance of each learning method is characterized using the normalized estimation error, which is defined as $\frac{\|\mathbf{B}-\mathbf{B}^*\|_F}{\|\mathbf{B}\|_F}$. For each value of m and σ_z, this experimental procedure is repeated 50 times, and the median estimation error is reported in Figure 1(a)–1(c), along with the 25th and 75th percentiles of estimation error. Further, in order to compare the "failure" rates of different methods, which correspond to relatively large estimation errors, we plot the histograms of estimation errors for $\sigma_z = 0.1$ and $m = 1100$ in Figure 1(d) for the three methods TPGD, PGD-Tucker, and PGD-CP. It can be seen from these histograms that the failure rates of both TPGD and PGD-Tucker are quite small. Finally, in order to characterize the empirical distributions of the estimation errors over all experiments, we also report violin plots of the estimation errors for the three methods in Figure 1(e) for values of m around the phase transition region for the TPGD algorithm. Note that LASSO and SVR perform considerably worse than the other learning methods; thus, they are not included in Figure 1 for clarity of the plots.

We gain two interesting insights from Figure 1. First, the plots show that the projection step $\mathcal{H}$ in Algorithm 3.1 (TPGD method) can be computed accurately enough by Algorithm 3.2, enabling the TPGD method to achieve better sample complexity compared with the other learning methods by exploiting the low-rank and sparse structure in the parameter tensor. In other words, despite the lack of theoretical guarantees for Algorithm 3.2, it can be employed in practice to compute the projection step (3.1) in step 5 of Algorithm 3.1. Second, comparing Figures 1(a), 1(b), and 1(c), we see that as the noise power decreases, the accuracy of the solution of TPGD increases. This is also reflected in the statement of Theorem 4.2: the lower the noise power, the more accurate the solution of TPGD, and vice versa.

Finally, we numerically investigate the computational complexity of our specific implementations of TPGD, PGD-Tucker, and PGD-CP for various values of n_1, n_2, and n_3. Toward this end, we repeat the aforementioned experiment for $m = 500$, $\sigma_z = 0.1$, and $n_1 = n_2 = n_3 =: n$, we fix the maximum number of iterations to 100 and vary n to have values of (i) $n = 10$, (ii) $n = 20$, and (iii) $n = 40$. For each of these values of n, we report the per-iteration computational time of each method in Table 1. It can be seen from this table that the mean computational time is comparable for the three tensor-based methods.

6.2. Neuroimaging data analysis. We also analyze the performance of TPGD for predicting attention deficit hyperactivity disorder (ADHD) diagnosis, using a preprocessed repository of ADHD-200 fMRI images [32] from the Donders Institute (NeuroImage), the Kennedy Krieger Institute (KKI), and the NYU Child Study Center (NYU). Specifically, we use preprocessed brain maps of fractional amplitude of low-frequency fluctuations (fALFF) [52] that were obtained using the Athena pipeline [5]. Note that fALFF is defined as the ratio of power within the low-frequency range (0.01–0.1 Hz) to that of the entire frequency range, and as

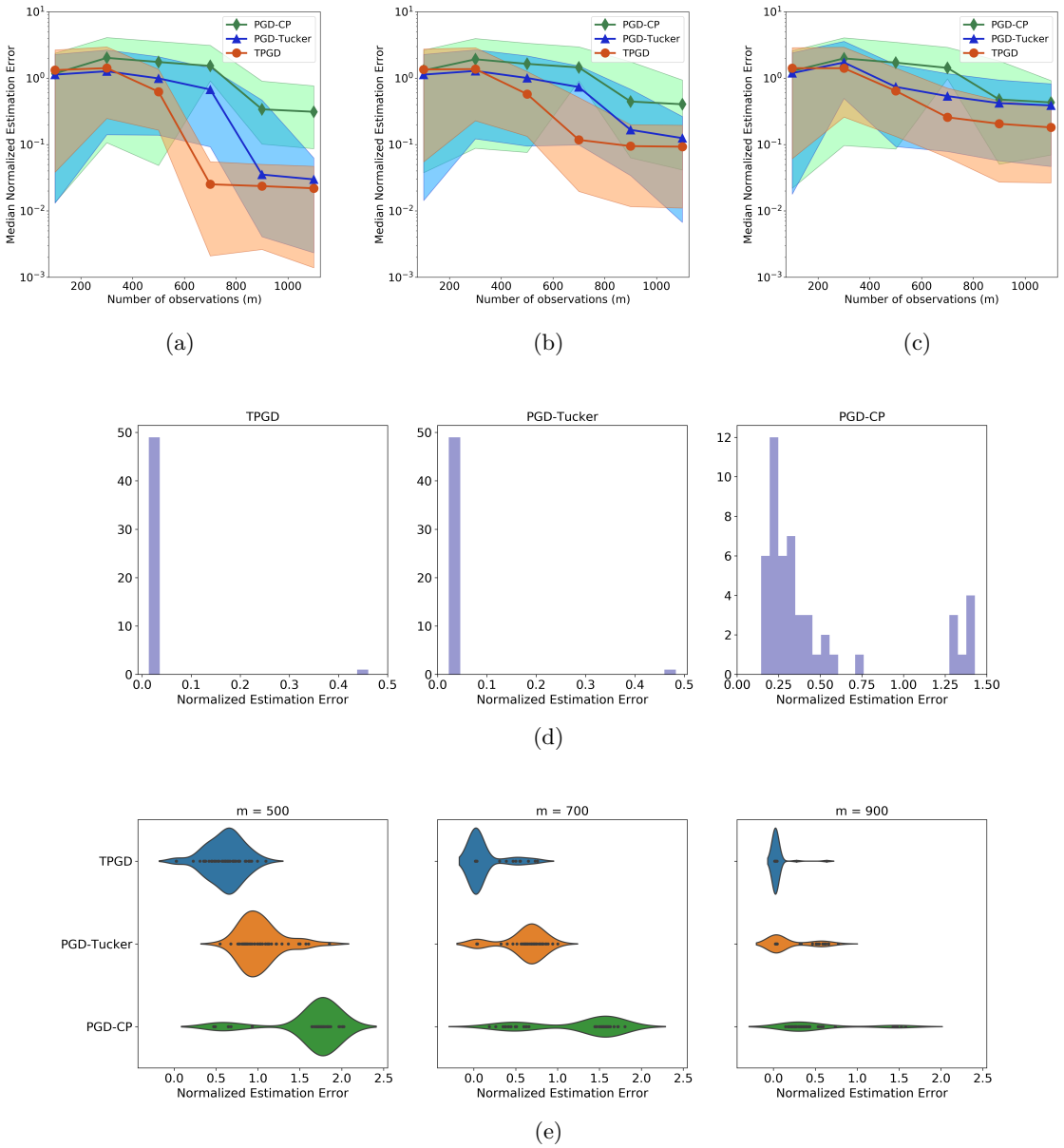

Figure 1. *Comparison of TPGD to PGD-Tucker and PGD-CP over synthetic data for* (a) $\sigma_z = 0.1$, (b) $\sigma_z = 0.4$, *and* (c) $\sigma_z = 0.7$. *For each value of* m, *the markers in* (a)–(c) *correspond to the median estimation error over* 50 *experiments, whereas the shaded region for each marker pertains to the* 25*th and* 75*th percentiles of the estimation error. Note that we report median error since an occasional failure in recovering the parameter tensor may lead to a spike in mean error. In order to analyze the failure rates,* (d) *shows histograms of estimation errors for* $\sigma_z = 0.1$ *and* $m = 1100$ *for the three methods. In addition,* (e) *shows empirical distributions of the estimation errors for* $\sigma_z = 0.1$ *in terms of violin plots, corresponding to the values of* m *around the phase transition region of TPGD.*

Table 1

Per-iteration computational time (in seconds) of TPGD, PGD-Tucker, and PGD-CP implementations for (i) $n = 10$, (ii) $n = 20$, *and* (iii) $n = 40$, *reported as an average over the* 50 *experiments. The variance is also reported in parentheses in each cell of the table.*

	$n = 10$	$n = 20$	$n = 40$
TPGD	0.0096 (1e-6)	0.031 (5e-7)	0.32 (8e-5)
PGD-Tucker	0.0021 (1e-7)	0.021 (5e-7)	0.24 (4e-5)
PGD-CP	0.047 (3e-4)	0.060 (2e-4)	0.27 (8e-5)

such it characterizes the intensity of spontaneous brain activity. Altered levels of fALFF have been reported in a sample of children with ADHD relative to controls [46], so fALFF brain maps form a useful feature space for predicting ADHD diagnosis.

The train data consists of fALFF brain maps for individuals pertaining to NeuroImage, KKI, and NYU. For each of these imaging sites, each individual's fALFF map forms a third-order tensor $\mathbf{X}_i \in \mathbb{R}^{49\times 58\times 47}$, and the ADHD diagnosis y_i (1 = ADHD, 0 = normal control) forms the response, where $i \in [[m]]$, and m is the number of train samples. In our experiments, we have $m = 39$ for NeuroImage (ADHD = 17, control = 22), $m = 78$ for KKI (ADHD = 20, control = 58), and $m = 188$ for NYU (ADHD = 97, control = 91), and we learn a regression model for each site independently. Given fALFF maps $\{\mathbf{X}_i\}_{i=1}^m$ and responses $\{y_i\}_{i=1}^m$ for each site, the task of learning the regression model in (1.1) is equivalent to learning the parameter tensor $\mathbf{B}$. We estimate the unknown parameter tensor using TPGD, PGD-Tucker, PGD-CP, LASSO, and SVR.

To analyze the performance of these learning methods, we employ separately provided test datasets for NeuroImage, KKI, and NYU pertaining to fALFF maps of 25, 11, and 41 test subjects, respectively. To analyze the performance for each method, we use the estimate of $\mathbf{B}$ to compute the responses for the test subjects using (1.1). If the computed response is more than 0.5 for a test subject, the subject is labeled with ADHD and vice versa. To evaluate the predictive power of each method using test data, we use the notion of (i) specificity, which is the ratio of subjects not diagnosed with ADHD that are correctly labeled as normal controls, and (ii) sensitivity, which is the ratio of subjects diagnosed with ADHD that are correctly labeled with ADHD. The explained experimental procedure is repeated 50 times for each method and imaging site, and the median results on test data are reported in Table 2, along with the harmonic mean of reported specificity and sensitivity. The TPGD method tends to perform well in the low sample size regime, given that it provides the highest harmonic mean on test data for the NeuroImage and the KKI sites, respectively. Moreover, we observe that vectorization-based methods LASSO and SVR perform poorly on the KKI test dataset, which entails a challenging prediction task because of the high class imbalance in the KKI train dataset. For the NYU imaging site, the PGD-Tucker method tends to work best; however, the performance of the PGD-CP and TPGD methods is not much worse. The slightly worse performance of TPGD compared to PGD-Tucker is attributable to differences in implementations of the projection steps for each method.

7. Conclusion.

In this work, we studied a tensor-structured linear regression model, with simultaneous imposition of a sparse and low Tucker-rank structure on the parameter tensor. We formulated the parameter estimation problem as a nonconvex program, and then we

Table 2

Comparison of TPGD to PGD-Tucker, PGD-CP, LASSO, and SVR for predicting diagnosis of test subjects corresponding to (a) *Donders Institute (NeuroImage),* (b) *Kennedy Krieger Institute (KKI), and* (c) *New York University Child Study Center (NYU), respectively.*

(a) The Donders Institute (NeuroImage)

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Specificity	0.68	0.57	0.57	1	0.89
Sensitivity	0.73	0.45	0.64	0.18	0.36
Harmonic mean	**0.70**	0.50	0.60	0.31	0.51

(b) Kennedy Krieger Institute (KKI)

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Specificity	0.63	0.50	0.50	1	1
Sensitivity	0.67	0.33	0.33	0	0
Harmonic mean	**0.65**	0.40	0.40	0	0

(c) New York University Child Study Center (NYU)

	TPGD	PGD-Tucker	PGD-CP	LASSO	SVR
Specificity	0.58	0.67	0.67	0.42	0.17
Sensitivity	0.52	0.52	0.48	0.55	0.59
Harmonic mean	0.55	**0.59**	0.56	0.48	0.26

proposed a projected gradient descent-based method to solve it. In our analysis, we provided mathematical guarantees for the proposed method based on the restricted isometry property. Furthermore, we evaluated the property for the case of sub-Gaussian predictors, characterizing the sample complexity of parameter estimation in the process. Finally, in our experiments with real-world data, we demonstrated that the simultaneously structured tensor regression model is not restrictive, and it can be effectively employed for neuroimaging data analysis.

Appendix A. Proof of Lemma 4.3.

Since $\mathbf{Z}_a \in \mathcal{G}_{\underline{r},\underline{s},\tau}$, it can be expressed as

$$\mathbf{Z}_a = \mathbf{S}_a \times_1 U_{a,1} \times_2 U_{a,2} \times \cdots \times_d U_{a,d},$$

where $\mathbf{S}_a \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$ such that $\|\mathbf{S}_a\|_1 \leq \tau$, and $U_{a,i} \in \mathbb{R}^{n_i \times r_i}$, with $\|U_{a,i}(:,j)\|_0 \leq s_i$, $\forall i \in [[d]]$, $j \in [[r_i]]$. Similarly, since $\mathbf{Z}_b \in \mathcal{G}_{\underline{r},\underline{s},\tau}$, it can be expressed as

$$\mathbf{Z}_b = \mathbf{S}_b \times_1 U_{b,1} \times_2 U_{b,2} \times \cdots \times_d U_{b,d},$$

where $\mathbf{S}_b \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$ such that $\|\mathbf{S}_b\|_1 \leq \tau$, and $U_{b,i} \in \mathbb{R}^{n_i \times r_i}$, with $\|U_{b,i}(:,j)\|_0 \leq s_i$, $\forall i \in [[d]]$, $j \in [[r_i]]$. Let $\mathbf{Z}_c = \gamma_a \mathbf{Z}_a + \gamma_b \mathbf{Z}_c$, where $\gamma_a \in \mathbb{R}$, $\gamma_b \in \mathbb{R}$, so that $\mathbf{Z}_c$ is some linear combination of $\mathbf{Z}_a$ and $\mathbf{Z}_b$. Define the Cartesian product $\mathcal{D}_P := [[r_1]] \times [[r_2]] \times \cdots \times [[r_d]]$. Using the definition of $\mathcal{D}_P$, define $\mathbf{S}_c \in \mathbb{R}^{2r_1 \times 2r_2 \times \cdots \times 2r_d}$, where

$$\mathbf{S}_c(i_1, i_2, \ldots, i_d) = \begin{cases} \gamma_a \, \mathbf{S}_a(i_1, i_2, \ldots, i_d) & : (i_1, i_2, \ldots, i_d) \in \mathcal{D}_P, \\ \gamma_b \, \mathbf{S}_b(i_1, i_2, \ldots, i_d) & : (i_1 - r_1, i_2 - r_2, \ldots, i_d - r_d) \in \mathcal{D}_P, \\ 0 & : \text{otherwise} \end{cases}$$

for $(i_1, i_2, \ldots, i_d) \in [[2r_1]] \times [[2r_2]] \times \cdots \times [[2r_d]]$. Note that $\|\mathbf{S}_c\|_1 = \|\gamma_a \mathbf{S}_a\|_1 + \|\gamma_b \mathbf{S}_b\|_1 \leq (|\gamma_a| + |\gamma_b|)\tau$. Furthermore, for $i \in [[d]]$, define $U_{c,i} \in \mathbb{R}^{n_i \times 2r_i}$ such that $U_{c,i} := [U_{a,i} \;\; U_{b,i}]$.

Finally, with these definitions, $\mathbf{Z}_c$ can be expressed as

$$\mathbf{Z}_c = \mathbf{S}_c \times_1 U_{c,1} \times_2 U_{c,2} \cdots \times_d U_{c,d},$$

where $\mathbf{S}_c \in \mathbb{R}^{2r_1 \times 2r_2 \times \cdots \times 2r_d}$ such that $\|\mathbf{S}_c\|_1 \leq (|\gamma_a| + |\gamma_b|)\tau$, and $U_{c,i} \in \mathbb{R}^{n_i \times 2r_i}$ such that $\|U_{c,i}(:, j)\|_0 \leq s_i$, for all $i \in [[d]]$, $j \in [[2r_i]]$. Therefore, $\mathbf{Z}_c$ is a member of the set $\mathcal{G}_{2\underline{r},\underline{s},\kappa}$, where $\kappa = (|\gamma_a| + |\gamma_b|)\tau$.

Appendix B. Proof of Theorem 4.2. Let $\mathcal{L}(\mathbf{Z}) := \|\mathbf{y} - \mathcal{X}(\mathbf{Z})\|_2^2$ be the loss function for any $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$. Then, we have

$$\begin{aligned}
\mathcal{L}(\mathbf{B}^{k+1}) - \mathcal{L}(\mathbf{B}^k) &= \|\mathbf{y} - \mathcal{X}(\mathbf{B}^{k+1})\|_2^2 - \|\mathbf{y} - \mathcal{X}(\mathbf{B}^k)\|_2^2 \\
&= \|\mathcal{X}(\mathbf{B}^{k+1})\|_2^2 - \|\mathcal{X}(\mathbf{B}^k)\|_2^2 - 2\langle \mathbf{y}, \mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\rangle \\
&= \|\mathcal{X}(\mathbf{B}^{k+1})\|_2^2 + \|\mathcal{X}(\mathbf{B}^k)\|_2^2 - 2\|\mathcal{X}(\mathbf{B}^k)\|_2^2 - 2\langle \mathbf{y}, \mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\rangle \\
&= \|\mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\|_2^2 + 2\langle \mathcal{X}(\mathbf{B}^k), \mathcal{X}(\mathbf{B}^{k+1})\rangle - 2\langle \mathcal{X}(\mathbf{B}^k), \mathcal{X}(\mathbf{B}^k)\rangle \\
&\quad - 2\langle \mathbf{y}, \mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\rangle \\
&= \|\mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\|_2^2 + 2\langle \mathcal{X}(\mathbf{B}^k) - \mathbf{y}, \mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\rangle \\
&= \|\mathcal{X}(\mathbf{B}^{k+1} - \mathbf{B}^k)\|_2^2 + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y}), \mathbf{B}^{k+1} - \mathbf{B}^k\rangle \\
&\leq (1 + \delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{B}^{k+1} - \mathbf{B}^k\|_F^2 + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y}), \mathbf{B}^{k+1} - \mathbf{B}^k\rangle, \qquad \text{(B.1)}
\end{aligned}$$

where the last inequality follows from application of Definition 4.1 with Lemma 4.3.

For any $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, define

$$\begin{aligned}
g(\mathbf{Z}) &:= (1 + \delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{Z} - \mathbf{B}^k\|_F^2 + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y}), \mathbf{Z} - \mathbf{B}^k\rangle \\
&\overset{\text{(a)}}{=} (1 + \delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{Z} - \widetilde{\mathbf{B}}^k + \mu\mathcal{X}^*(\mathbf{y} - \mathcal{X}(\mathbf{B}^k))\|_F^2 \\
&\quad + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y}), \mathbf{Z} - \widetilde{\mathbf{B}}^k + \mu\mathcal{X}^*(\mathbf{y} - \mathcal{X}(\mathbf{B}^k))\rangle \\
&\overset{\text{(b)}}{=} (1 + \delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{Z} - \widetilde{\mathbf{B}}^k\|_F^2 - \frac{1}{1 + \delta_{2\underline{r},\underline{s},2\tau}}\|\mathcal{X}^*(\mathbf{y} - \mathcal{X}(\mathbf{B}^k))\|_F^2, \qquad \text{(B.2)}
\end{aligned}$$

where (a) follows by substituting $\mathbf{B}^k = \widetilde{\mathbf{B}}^k + \mu\mathcal{X}^*(\mathcal{X}(\mathbf{B}^k) - \mathbf{y})$, and (b) follows by substituting $\mu = \frac{1}{1+\delta_{2\underline{r},\underline{s},2\tau}}$. Then, since $\|\mathbf{B}^{k+1} - \widetilde{\mathbf{B}}^k\|_F \leq \|\mathbf{B} - \widetilde{\mathbf{B}}^k\|_F$, which follows from $\mathbf{B}^{k+1} = \mathcal{H}(\widetilde{\mathbf{B}}^k)$,

we have $g(\mathbf{B}^{k+1}) \leq g(\mathbf{B})$. Using $g(\mathbf{B}^{k+1}) \leq g(\mathbf{B})$ with (B.1), we obtain

$$\begin{aligned}
\mathcal{L}(\mathbf{B}^{k+1}) - \mathcal{L}(\mathbf{B}^k) &\leq (1+\delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{B}-\mathbf{B}^k\|_F^2 + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k)-\mathbf{y}), \mathbf{B}-\mathbf{B}^k\rangle \\
&= 2\delta_{2\underline{r},\underline{s},2\tau}\|\mathbf{B}-\mathbf{B}^k\|_F^2 + (1-\delta_{2\underline{r},\underline{s},2\tau})\|\mathbf{B}-\mathbf{B}^k\|_F^2 \\
&\quad + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k)-\mathbf{y}), \mathbf{B}-\mathbf{B}^k\rangle \\
&\leq 2\delta_{2\underline{r},\underline{s},2\tau}\|\mathbf{B}-\mathbf{B}^k\|_F^2 + \|\mathcal{X}(\mathbf{B}-\mathbf{B}^k)\|_2^2 \\
&\quad + 2\langle \mathcal{X}^*(\mathcal{X}(\mathbf{B}^k)-\mathbf{y}), \mathbf{B}-\mathbf{B}^k\rangle \\
&= 2\delta_{2\underline{r},\underline{s},2\tau}\|\mathbf{B}-\mathbf{B}^k\|_F^2 + \|\mathcal{X}(\mathbf{B}-\mathbf{B}^k)\|_2^2 \\
&\quad + 2\langle \mathcal{X}(\mathbf{B}^k), \mathcal{X}(\mathbf{B}-\mathbf{B}^k)\rangle - 2\langle \mathbf{y}, \mathcal{X}(\mathbf{B}-\mathbf{B}^k)\rangle \\
&= 2\delta_{2\underline{r},\underline{s},2\tau}\|\mathbf{B}-\mathbf{B}^k\|_F^2 + \|\mathcal{X}(\mathbf{B})\|_2^2 - \|\mathcal{X}(\mathbf{B}^k)\|_2^2 - 2\langle \mathbf{y}, \mathcal{X}(\mathbf{B}-\mathbf{B}^k)\rangle \\
&= 2\delta_{2\underline{r},\underline{s},2\tau}\|\mathbf{B}-\mathbf{B}^k\|_F^2 + \|\mathbf{y}-\mathcal{X}(\mathbf{B})\|_2^2 - \|\mathbf{y}-\mathcal{X}(\mathbf{B}^k)\|_2^2 \\
&\leq \frac{2\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}\|\mathcal{X}(\mathbf{B}-\mathbf{B}^k)\|_2^2 + \mathcal{L}(\mathbf{B}) - \mathcal{L}(\mathbf{B}^k), \qquad \text{(B.3)}
\end{aligned}$$

where the last two inequalities follow from application of Definition 4.1 with Lemma 4.3. Thus, we have

$$\mathcal{L}(\mathbf{B}^{k+1}) \leq \frac{2\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}\|\mathcal{X}(\mathbf{B}-\mathbf{B}^k)\|_2^2 + \mathcal{L}(\mathbf{B}). \qquad \text{(B.4)}$$

Using $\mathcal{X}(\mathbf{B}) = \mathbf{y} - \boldsymbol{\eta}$, we have

$$\|\mathcal{X}(\mathbf{B}-\mathbf{B}^k)\|_2^2 = \|\mathbf{y}-\mathcal{X}(\mathbf{B}^k)-\boldsymbol{\eta}\|_2^2 \leq 2(\|\mathbf{y}-\mathcal{X}(\mathbf{B}^k)\|_2^2 + \|\boldsymbol{\eta}\|_2^2) = 2\big(\mathcal{L}(\mathbf{B}^k) + \|\boldsymbol{\eta}\|_2^2\big), \qquad \text{(B.5)}$$

where the inequality follows since $(u+v)^2 \leq 2(u^2+v^2)$ for all $u, v \in \mathbb{R}$. Using (B.4) with (B.5) and observing $\mathcal{L}(\mathbf{B}) = \|\boldsymbol{\eta}\|_2^2$, we obtain

$$\begin{aligned}
\mathcal{L}(\mathbf{B}^{k+1}) &\leq \frac{4\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big(\mathcal{L}(\mathbf{B}^k) + \|\boldsymbol{\eta}\|_2^2\Big) + \|\boldsymbol{\eta}\|_2^2 \\
&= \frac{4\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}\mathcal{L}(\mathbf{B}^k) + \Big(1 + \frac{4\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big)\|\boldsymbol{\eta}\|_2^2. \qquad \text{(B.6)}
\end{aligned}$$

Using $\delta_{2\underline{r},\underline{s},2\tau} < \frac{\gamma}{4+\gamma}$, $\gamma < 1$, and $b = \frac{1+3\delta_{2\underline{r},\underline{s},2\tau}}{1-\delta_{2\underline{r},\underline{s},2\tau}}$ yields

$$\mathcal{L}(\mathbf{B}^{k+1}) \leq \gamma\mathcal{L}(\mathbf{B}^k) + b\|\boldsymbol{\eta}\|_2^2. \qquad \text{(B.7)}$$

Iterative application of this inequality leads to

$$\mathcal{L}(\mathbf{B}^k) \leq \gamma^k\mathcal{L}(\mathbf{B}^0) + \frac{b}{1-\gamma}\|\boldsymbol{\eta}\|_2^2 \qquad \text{(B.8)}$$

for all $k \geq 1$.

Next, using Definition 4.1 with Lemma 4.3, we obtain

$$\|\mathbf{B}^k - \mathbf{B}\|_F^2 \leq \frac{1}{1-\delta_{2\underline{r},\underline{s},2\tau}}\|\mathcal{X}(\mathbf{B}^k-\mathbf{B})\|_2^2 \leq \frac{2}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big(\mathcal{L}(\mathbf{B}^k) + \|\boldsymbol{\eta}\|_2^2\Big), \qquad \text{(B.9)}$$

where the last inequality follows from (B.5). Finally, using (B.8), we get

$$\begin{aligned}\|\mathbf{B}^k - \mathbf{B}\|_F^2 &\le \frac{2}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big(\gamma^k \mathcal{L}(\mathbf{B}^0) + \frac{b}{1-\gamma}\|\boldsymbol{\eta}\|_2^2 + \|\boldsymbol{\eta}\|_2^2\Big) \\ &= \frac{2\gamma^k}{1-\delta_{2\underline{r},\underline{s},2\tau}}\mathcal{L}(\mathbf{B}^0) + \frac{2\|\boldsymbol{\eta}\|_2^2}{1-\delta_{2\underline{r},\underline{s},2\tau}}\Big(1+\frac{b}{1-\gamma}\Big). \end{aligned} \tag{B.10}$$

Appendix C. Proof of Lemma 5.5. Define $\mathcal{D}_{\widehat{\mathbf{S}}} := \{\frac{\mathbf{S}}{\tau} : \mathbf{S} \in \mathcal{D}_{\mathbf{S}}\}$, so that $\|\widehat{\mathbf{S}}\|_1 \le 1$ for all $\widehat{\mathbf{S}} \in \mathcal{D}_{\widehat{\mathbf{S}}}$. By application of Lemma E.1, we have

$$\Psi(\mathcal{D}_{\widehat{\mathbf{S}}}, \|\,.\,\|_1, \epsilon) \le \Big(\frac{3}{\epsilon}\Big)^{\prod_{i=1}^d r_i}$$

for $\epsilon \in (0,1)$. The bound on $\Psi(\mathcal{D}_{\mathbf{S}}, \|\,.\,\|_1, \epsilon)$ follows from a volume comparison argument between $\mathcal{D}_{\mathbf{S}}$ and $\mathcal{D}_{\widehat{\mathbf{S}}}$.

Appendix D. Proof of Lemma 5.6. The set $\mathcal{D}_U$ can be expressed as the Cartesian product of the sets $\mathcal{D}_U^{(j)} := \{x \in \mathbb{R}^n : \|x\|_0 \le s, \|x\|_2 \le 1\}$, $j \in [[r]]$. For any $j \in [[r]]$, since there are $\binom{n}{s}$ ways to choose the support of an s-sparse vector, we have

$$\Psi(\mathcal{D}_U^{(j)}, \|\,.\,\|_2, \epsilon) \le \binom{n}{s}\Big(\frac{3}{\epsilon}\Big)^s, \tag{D.1}$$

with the application of Lemma E.1. Then, the covering number of $\mathcal{D}_U$ with respect to the metric h_U, for any $\epsilon \in (0,1)$, satisfies the bound

$$\begin{aligned}\Psi(\mathcal{D}_U, h_U, \epsilon) &\overset{(a)}{\le} \prod_{j=1}^r \Psi(\mathcal{D}_U^{(j)}, \|\,.\,\|_2, \epsilon) \overset{(b)}{\le} \left[\binom{n}{s}\Big(\frac{3}{\epsilon}\Big)^s\right]^r \le \frac{n^{sr}}{(s!)^r}\Big(\frac{3}{\epsilon}\Big)^{sr} \\ &= \Big(\frac{3n}{(s!)^{\frac{1}{s}}\epsilon}\Big)^{sr} \le \Big(\frac{3n}{\epsilon}\Big)^{sr},\end{aligned}$$

where (a) and (b) follow from Lemma E.2 and (D.1), respectively.

Appendix E. Auxiliary lemmas.

Lemma E.1 ([10]). *For any fixed notion of norm $\|\,.\,\|$, define a unit-norm ball $\mathcal{B}_1 := \{x \in \mathbb{R}^n : \|x\| \le 1\}$ with distance measure $\|\,.\,\|$. Then the covering number of $\mathcal{B}_1$ (with respect to the norm $\|\,.\,\|$) satisfies the bound*

$$\Psi(\mathcal{B}_1, \|\,.\,\|, \epsilon) \le \Big(\frac{3}{\epsilon}\Big)^n, \quad \epsilon \in (0,1).$$

Lemma E.2 ([15]). *Define metric spaces $(\mathcal{D}_1, h_1)$, $(\mathcal{D}_2, h_2), \ldots, (\mathcal{D}_p, h_p)$. Further, define the Cartesian product $\mathcal{D}_0 := \mathcal{D}_1 \times_1 \mathcal{D}_2 \times_2 \cdots \times_p \mathcal{D}_p$ with respect to the norm $h_0(D_0^1, D_0^2) = \max_{j\in[[p]]}\{h_j(D_j^1, D_j^2)\}$, where $D_0^1, D_0^2 \in \mathcal{D}_0$ such that $D_0^1 = D_1^1 \times_1 D_2^1 \times_2 \cdots \times_p D_p^1$, $D_0^2 = D_1^2 \times_1 D_2^2 \times_2 \cdots \times_p D_p^2$, and $D_j^1, D_j^2 \in \mathcal{D}_j$ for any $j \in [[p]]$. Then, $\Psi(\mathcal{D}_0, h_0, \epsilon)$ satisfies the bound $\Psi(\mathcal{D}_0, h_0, \epsilon) \le \prod_{j=1}^d \Psi(\mathcal{D}_j, h_j, \epsilon)$.*

Lemma E.3 ([41]). *Define sets $\mathcal{D}_1$ and $\mathcal{D}_2$ with distance measures h_1 and h_2, respectively. Further, define map $\Phi : \mathcal{K} \to \mathcal{D}_2$ such that $\mathcal{K} \subset \mathcal{D}_1$. Then, for some $L > 0$, if Φ satisfies*

$$h_2(\Phi(K_1), \Phi(K_2)) \leq L\, h_1(K_1, K_2) \quad \text{for } K_1, K_2 \in \mathcal{K},$$

i.e., Φ is a Lipschitz map with constant L, then for any $\epsilon > 0$, we have

$$\Psi(\Phi(\mathcal{K}), h_2, L\,\epsilon) \leq \Psi(\mathcal{K}, h_1, \epsilon).$$

REFERENCES

[1] E. Acar, C. Aykut-Bingol, H. Bingol, R. Bro, and B. Yener, *Multiway analysis of epilepsy tensors*, Bioinformatics, 23 (2007), pp. i10–i18.

[2] G. Allen, *Sparse higher-order principal components analysis*, in Proc. 15th International Conference on Artificial Intelligence and Statistics (AISTATS), La Palma, Canary Islands, JMLR 22, 2012, pp. 27–36.

[3] K. D. Ba, P. Indyk, E. Price, and D. P. Woodruff, *Lower bounds for sparse recovery*, in Proc. 21st Annu. ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2010, pp. 1190–1197, https://doi.org/10.1137/1.9781611973075.95.

[4] B. W. Bader, T. G. Kolda et al., *MATLAB Tensor Toolbox,* version 3.0-dev., available online at https://www.tensortoolbox.org, 2017.

[5] P. Bellec, C. Chu, F. Chouinard-Decorte, Y. Benhajali, D. S. Margulies, and R. C. Craddock, *The Neuro Bureau ADHD-200 Preprocessed repository*, Neuroimage, 144 (2017), pp. 275–286.

[6] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, *Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches*, IEEE J. Select. Topics Appl. Earth Observ. Remote Sensing, 5 (2012), pp. 354–379.

[7] T. Blumensath and M. E. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 265–274.

[8] E. Candès and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n*, Ann. Stat., 35 (2007), pp. 2313–2351.

[9] E. J. Candès, *The restricted isometry property and its implications for compressed sensing*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 589–592.

[10] E. J. Candès and Y. Plan, *Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements*, IEEE Trans. Inform. Theory, 57 (2011), pp. 2342–2359.

[11] H. Chen, G. Raskutti, and M. Yuan, *Non-convex projected gradient descent for generalized low-rank tensor regression*, J. Mach. Learn. Res., 20 (2019), pp. 172–208.

[12] Y. Fu, G. Guo, and T. S. Huang, *Age synthesis and estimation via faces: A survey*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 1955–1976.

[13] S. Gandy, B. Recht, and I. Yamada, *Tensor completion and low-n-rank tensor recovery via convex optimization*, Inverse Problems, 27 (2011), 025010.

[14] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78.

[15] R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert, *Sample complexity of dictionary learning and other matrix factorizations*, IEEE Trans. Inform. Theory, 61 (2015), pp. 3469–3486.

[16] B. Hao, A. Zhang, and G. Cheng, *Sparse and Low-Rank Tensor Estimation via Cubic Sketchings*, preprint, https://arxiv.org/abs/1801.09326, 2018.

[17] L. He, K. Chen, W. Xu, J. Zhou, and F. Wang, *Boosted sparse and low-rank tensor regression*, in Proc. Advances in Neural Information Processing Systems, Curran Associates, 2018, pp. 1009–1018.

[18] M. Hein and T. Bühler, *An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA*, in Proc. Advances in Neural Information Processing Systems, Curran Associates, 2010, pp. 847–855.

[19] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM (JACM), 60 (2013), 45.
[20] C. Hinrichs, V. Singh, L. Mukherjee, G. Xu, M. K. Chung, and S. C. Johnson, *Spatially augmented LPboosting for AD classification with evaluations on the ADNI dataset*, Neuroimage, 48 (2009), pp. 138–149.
[21] C.-H. Ho and C.-J. Lin, *Large-scale linear support vector regression*, J. Mach. Learn. Res., 13 (2012), pp. 3323–3348.
[22] P. Jain, R. Meka, and I. S. Dhillon, *Guaranteed rank minimization via singular value projection*, in Proc. Advances in Neural Information Processing Systems, Curran Associates, 2010, pp. 937–945.
[23] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, Springer Texts Statist. 103, Springer, 2013.
[24] I. T. Jolliffe and D. B. Stephenson, *Forecast Verification: A Practitioner's Guide in Atmospheric Science*, John Wiley & Sons, 2012.
[25] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X.
[26] F. Krahmer, S. Mendelson, and H. Rauhut, *Suprema of chaos processes and the restricted isometry property*, Comm. Pure Appl. Math., 67 (2014), pp. 1877–1904.
[27] D. Landgrebe, *Hyperspectral image data analysis*, IEEE Signal Process. Mag., 19 (2002), pp. 17–28.
[28] N. Lazar, *The Statistical Analysis of Functional MRI Data*, Springer Science+Business Media, 2008.
[29] M. A. Lindquist, *The statistical analysis of fMRI data*, Statist. Sci., 23 (2008), pp. 439–464.
[30] J. Liu, P. Musialski, P. Wonka, and J. Ye, *Tensor completion for estimating missing values in visual data*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2012), pp. 208–220.
[31] MATLAB, *Version* 9.4.0 *(R*2018*a)*, The MathWorks Inc., 2018.
[32] M. P. Milham et al., *The ADHD-*200 *Consortium: A model to advance the translational potential of neuroimaging in clinical neuroscience*, Frontiers Syst. Neurosci., 6 (2012), 62.
[33] C. Mu, B. Huang, J. Wright, and D. Goldfarb, *Square deal: Lower bounds and improved relaxations for tensor recovery*, in Proc. 31st International Conference on Machine Learning (ICML), Beijing, China, JMLR 32, 2014, pp. 73–81.
[34] B. Pang and L. Lee, *Opinion mining and sentiment analysis*, Found. Trends Inform. Retrieval, 2 (2008), pp. 1–135.
[35] G. Raskutti, M. Yuan, and H. Chen, *Convex regularization for high-dimensional multi-response tensor regression*, Ann. Statist., 47 (2019), pp. 1554–1584.
[36] H. Rauhut, R. Schneider, and Ž. Stojanac, *Low rank tensor recovery via iterative hard thresholding*, Linear Algebra Appl., 523 (2017), pp. 220–262.
[37] I. Rish and G. Grabarnik, *Sparse Modeling: Theory, Algorithms, and Applications*, CRC Press, 2014.
[38] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon, *Sparse logistic regression for whole-brain classification of fMRI data*, Neuroimage, 51 (2010), pp. 752–764.
[39] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, *Tensor decomposition for signal processing and machine learning*, IEEE Trans. Signal Process., 65 (2017), pp. 3551–3582.
[40] W. W. Sun, J. Lu, H. Liu, and G. Cheng, *Provable sparse tensor decomposition*, J. Roy. Statist. Soc. Ser. B Statist. Methodol., 79 (2017), pp. 899–916.
[41] S. Szarek, *Metric entropy of homogeneous spaces*, in Quantum Probability, Banach Center Publ. 43, Polish Acad. Sci. Inst. Math., 1998, pp. 395–410.
[42] M. Talagrand, *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*, Ergeb. Math. Grenzgeb. (3) 60, Springer Science+Business Media, 2014.
[43] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. B, 58 (1996), pp. 267–288.
[44] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima, *Statistical performance of convex tensor decomposition*, in Proc. Advances in Neural Information Processing Systems, Curran Associates, 2011, pp. 972–980.
[45] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab* 3.0, available online at http://www.tensorlab.net, 2016.

[46] H. Yang, Q.-Z. Wu, L.-T. Guo, Q.-Q. Li, X.-Y. Long, X.-Q. Huang, R. C. Chan, and Q.-Y. Gong, *Abnormal spontaneous brain activity in medication-naive ADHD children: A resting state fMRI study*, Neurosci. Lett., 502 (2011), pp. 89–93.
[47] R. Yu and Y. Liu, *Learning from multiway data: Simple and efficient tensor regression*, in Proc. 33rd International Conference on Machine Learning (ICML), New York, NY, JMLR 48, 2016, pp. 373–381.
[48] H. Zhou, L. Li, and H. Zhu, *Tensor regression with applications in neuroimaging data analysis*, J. Amer. Statist. Assoc., 108 (2013), pp. 540–552.
[49] H. Zou, *The adaptive lasso and its oracle properties*, J. Amer. Statist. Assoc., 101 (2006), pp. 1418–1429.
[50] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, J. Roy. Statist. Soc., 67 (2005), pp. 301–320.
[51] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, J. Comput. Graphical Statist., 15 (2006), pp. 265–286.
[52] Q.-H. Zou, C.-Z. Zhu, Y. Yang, X.-N. Zuo, X.-Y. Long, Q.-J. Cao, Y.-F. Wang, and Y.-F. Zang, *An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: Fractional ALFF*, J. Neurosci. Methods, 172 (2008), pp. 137–141.