Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Low-rank, Orthogonally Decomposable Tensor Regression With Application to Visual Stimulus Decoding of fMRI Data

J. C. Poythress, Jeongyoun Ahn & Cheolwoo Park

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2022, VOL. 31, NO. 1, 190–203
https://doi.org/10.1080/10618600.2021.1951741

Taylor & Francis
Taylor & Francis Group

Low-rank, Orthogonally Decomposable Tensor Regression With Application to Visual Stimulus Decoding of fMRI Data

J. C. Poythress[a], Jeongyoun Ahn[b], and Cheolwoo Park[c]

[a]Department of Mathematics and Statistics, University of New Hampshire, Durham, NC; [b]Department of Industrial and Systems Engineering, KAIST, Daejeon, South Korea; [c]Department of Mathematical Sciences, KAIST, Daejeon, South Korea

ABSTRACT
We consider the problem of fitting a generalized linear model with a three-dimensional image covariate, such as one obtained by functional magnetic resonance imaging (fMRI). A major challenge for fitting such a model is that the image is a multidimensional array, called a tensor, containing tens of thousands of elements, called voxels. Because there is a parameter associated with each voxel, fitting the model entails estimating tens of thousands of parameters with a typical sample size on the order of hundreds of subjects. We propose to reduce the dimensionality of the problem by imposing a low-rank assumption on the parameter tensor, which not only reduces the number of parameters to estimate, but also exploits the implicit spatial information in fMRI data. In addition, we assume the parameter tensor is orthogonally decomposable, enabling us to penalize the so-called tensor "singular values" to obtain a low-rank estimate. We develop algorithms based on projected gradient descent and the proximal gradient method to estimate the model. In simulation, the proposed methods yielded estimates with smaller squared errors than an existing method based on alternating minimization. For visual stimulus decoding of a real fMRI dataset, the proposed methods resulted in better classification accuracies than the existing method. Visualization of the estimates revealed compact clusters of voxels with relatively large values, potentially representing brain regions relevant to the task. Supplementary files for this article are available online.

ARTICLE HISTORY
Received May 2020
Revised June 2021

KEYWORDS
High-dimensional low sample size; Imaging data; Low-rank approximation; Nuclear norm; Tensor decomposition

1. Introduction

Modern technology allows researchers to collect data that can be represented as images. For example, images such as those obtained by structural magnetic resonance imaging (MRI), diffusion tensor imaging (DTI), electroencephalography (EEG), and functional MRI (fMRI) are frequently used by psychologists, neuroscientists, and others within health and medical disciplines to diagnose or study diseases. The need to analyze such imaging data poses new challenges for statisticians because many traditional statistical methods do not readily accommodate imaging data. Different approaches have been proposed depending on the type of question the researcher would like to answer. Our work considers research questions in which the image may be considered a covariate, and one would like to know how the image covariate relates to or affects a response variable, which may be discrete or continuous. Some examples include whether an image obtained by fMRI can be used to predict whether a person would be diagnosed with a particular psychiatric disorder (such as schizophrenia), or whether a structural MRI can be used to predict disease status (e.g., Alzheimer's). In either example, the response may be binary (1=disorder/disease, 0=healthy) or continuous (e.g., symptom scale scores). Determining whether an image can be used to make such a prediction and what elements of the image are important to the prediction can help researchers identify the neural circuitry involved in the disorder/disease and potentially contribute to a better understanding of its etiology.

From the statistical perspective, images can be treated as matrix-valued (two-dimensional) or array-valued (three-dimensional or higher) data. However, images are challenging to model using traditional methods because of the implicit spatial structure and large number of parameters to estimate. For example, if one wanted to model the relationship between disease status and an fMRI covariate using a generalized linear model (GLM), one would need to estimate $40 \times 48 \times 38 = 72{,}960$ parameters—the dimension of a typical image obtained by fMRI. One solution is to fit a regularized GLM with a sparsity-inducing penalty, which effectively reduces the number of parameters to estimate. However, the results from penalized regression are often unsatisfactory when the data are ultra-high dimensional. Moreover, many of the classical penalized regression techniques do not account for the spatial structure inherent in imaging data.

Several researchers have proposed tensor regression models to handle array-valued imaging data, where the term "tensor" refers to a higher-order array. Tensor regression models exploit the expected spatial dependency of the array-valued image by assuming the corresponding tensor of model parameters can be well-approximated by a low-rank structure. For example,

CONTACT Jeongyoun Ahn *jyahn@kaist.ac.kr* Department of Industrial and Systems Engineering, KAIST, Daejeon 34141, Korea.

Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JCGS*.

if there are features that distinguish images belonging to healthy subjects from images belonging to subjects diagnosed with a disorder, we might expect the features to be few and to occur in spatially compact locations. The presence of stronger signal in a few, spatially compact locations are indicative of low-rank structure in the overall signal. Note that although tensor regression models are useful for the type of spatial dependence we expect to see in neuroimaging data, they do not explicitly model the spatial structure, and so they may not be useful for more general forms of spatial dependency.

Tensor regression models make use of tensor decompositions. Tensor decompositions have a long history of applications in the fields of psychometrics, chemometrics, and signal processing (Kolda and Bader 2009). In those applications, the objective is often to approximate an observed tensor by one of lower rank. In contrast, for applications in statistics and machine learning, the tensor to approximate by one of lower rank is usually not observed. In a tensor regression model, the parameter tensor is assumed to admit a low-rank structure. That is, the problem is to simultaneously estimate the parameter tensor and approximate it by a tensor of lower rank. Thus, the problem of estimating and decomposing an unobserved quantity is of a fundamentally different nature than finding an approximation to an observed quantity.

Various types of tensor regression models have been proposed. We focus on the case of a scalar response and tensor covariate. Many of the models rely on the canonical polyadic decomposition (CPD) (Hitchcock 1927). The CPD decomposes a tensor as a sum of rank-1 tensors. For illustration, the CPD decomposes the D-dimensional tensor $\mathcal{B}$ as

$$\mathcal{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr},$$

where $\circ$ denotes the vector outer product. The minimum number of terms in the sum needed to reconstruct the tensor exactly is called the tensor rank. When the number of terms R is truncated so that R is less than the actual tensor rank, the CPD yields a rank-R approximation of the tensor. For convenience of notation, we may also collect the vectors comprising the outer product into matrices such that $B_1 = (\boldsymbol{\beta}_{11}, \ldots, \boldsymbol{\beta}_{1R}), \ldots, B_D = (\boldsymbol{\beta}_{D1}, \ldots, \boldsymbol{\beta}_{DR})$. The matrices $B_1, \ldots, B_D$ are referred to as "factor matrices." We may then write $\mathcal{B} = [[B_1, \ldots, B_D]]$.

For classification problems, Hung and Wang (2012) and Tan et al. (2013) studied logistic regression with a matrix and tensor covariate, respectively. Hung and Wang (2012) formulated the systematic part of the model as a bilinear form in the parameters, which is equivalent to a rank-1 CPD. Tan et al. (2013) considered the more general case of tensors of arbitrary dimension and a rank-R CPD. Hu et al. (2020) exploited the relationship between linear discriminant analysis and regression to classify 2D images, using a penalized nuclear norm approach to impose low-rank structure. Guo, Kotsia, and Patras (2012) studied linear regression for continuous responses and support vector machine for binary responses. In both cases, they used the CPD with an additional ridge penalty or group sparsity penalty. Guhaniyogi, Qamar, and Dunson (2017) constructed a Gaussian linear regression model under a Bayesian framework by specifying priors for the vectors comprising the rank-1 terms in the CPD. Low-rank techniques also serve as the basis for some methods designed for the case of matrix- or tensor-valued responses with vector-valued covariates (Sun and Li 2017; Kong et al. 2020; Hu et al. 2021).

The present work focuses on tensor regression in the context of GLMs. Zhou, Li, and Zhu (2013) assumed a rank-R CPD for the parameter tensor in a GLM. The systematic component of their model can be expressed as

$$g(\mu_i) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \mathcal{B}, \mathcal{X}_i \rangle \text{ such that}$$
$$\mathcal{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr}, \tag{1}$$

where $g(\mu_i)$ is the link function for the mean of the response, α is a constant parameter, $\boldsymbol{z}$ is a vector of regular covariates with corresponding parameter $\boldsymbol{\gamma}$, and $\mathcal{X}$ is a tensor covariate (e.g., an fMRI image) with corresponding parameter tensor $\mathcal{B}$. The tensor inner product $\langle \cdot, \cdot \rangle$ is equivalent to vectorizing the two tensors and taking the vector inner product. The model assumes that the true parameter tensor can be well-approximated by a rank-R CPD. The rank-R assumption reduces the number of parameters associated with the tensor covariate from $\prod_{d=1}^{D} p_d$ to $R \sum_{d=1}^{D} p_d$, where p_d denotes the size along one dimension. For a typical fMRI image, a rank-1 assumption would reduce the number of parameters from $40 \times 48 \times 38 = 72,960$ to $40 + 48 + 38 = 126$.

The model (1), which we refer to as the CPD model, requires a priori specification of the rank R, so the estimate obtained is a fixed-rank approximation of the parameter. In the case of two-dimensional images, the parameter is a matrix, so the fixed-rank approximation can be thought of as hard thresholding the matrix singular values. Zhou and Li (2014) proposed a convex relaxation of the matrix rank by penalizing the objective function with the ℓ_1 norm of the singular values, which they called spectral-regularized matrix regression. Their approach involves soft thresholding the singular values, which yields a low-rank estimate of the parameter matrix by shrinking some of its singular values to exactly zero. The objective function for their model can be written as

$$\min_{B} \; -\ell\ell(B) + P_\lambda\left(\sigma(B)\right), \tag{2}$$

where B is the (unrestricted, full rank) parameter matrix, $-\ell\ell(\cdot)$ is the negative log-likelihood of a GLM with link function $g(\mu_i) = \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle B, X_i \rangle$, $\sigma(\cdot)$ extracts the singular values of a matrix, and $P_\lambda(\cdot)$ is a sparsity-inducing penalty function. For example, if $P_\lambda(\cdot)$ is the LASSO penalty (Tibshirani 1996), then $P_\lambda(\sigma(B))$ is the nuclear norm of B, $\|B\|_* = \sum_{r=1}^{\min(p_1,p_2)} \sigma_r$, multiplied by the tuning parameter λ. One of the main advantages of model (2) over model (1) is convexity. The estimates for both models must be obtained numerically, and the non-convexity of Equation (1) makes it challenging to obtain an estimate near the global minimizer of the objective function.

The major limitation of model (2) is that it can only be fitted for two-dimensional image covariates. In Section 2, we propose two versions of a low-rank, orthogonally decomposable tensor regression model that can be applied to three-dimensional (or higher) images. Both models use a notion of tensor "singular values." One version of the model assumes a fixed tensor rank, which serves as a necessary intermediate step toward developing a second version of the model that relaxes the fixed-rank assumption and instead penalizes the tensor singular values. The penalized version of the model is similar in principle to (2). However, a key difference is that matrices always admit an orthogonal decomposition (by the singular value decomposition, SVD), whereas higher-order tensors are not guaranteed to be orthogonally decomposable ("odeco" for short). We illustrate some properties of the proposed model and compare to Zhou et al.'s CPD model in a simulation experiment in Section 3. We apply the proposed methods to intra-subject visual stimulus decoding of a real fMRI dataset in Section 4. We provide some concluding remarks in Section 5.

2. Proposed Methodology

2.1. Background: An Orthogonal Tensor Decomposition

Both versions of the proposed orthogonally decomposable tensor regression model make use of a tensor decomposition studied by Chen and Saad (2009) called a low rank, orthogonal approximation to tensors (LROAT). The LROAT decomposition solves

$$\begin{aligned} &\min_{\sigma_1,\ldots,\sigma_R,B_1,\ldots,B_D} \left\| \mathcal{B} - \sum_{r=1}^{R} \sigma_r \boldsymbol{\beta}_{1r} \circ \cdots \circ \boldsymbol{\beta}_{Dr} \right\|_F \quad \text{s.t.} \\ &B_1^T B_1 = \cdots = B_D^T B_D = I_R \qquad (3) \\ &\text{and } \sigma_r \geq 0,\ r = 1, \ldots, R, \end{aligned}$$

where $\| \cdot \|_F$ denotes the Frobenius norm (defined in an analogous way for tensors as it is for matrices). Chen and Saad (2009) proposed a block coordinate descent algorithm to solve Equation (3); see their article for the details.

Some important properties of the LROAT decomposition include:

1. The factor matrices $B_1, \ldots, B_D$ are orthogonal.
2. The tensor rank is bounded by the smallest size among the modes:
   $R = \text{rank}(\mathcal{B}) \leq \min(p_1, p_2, \ldots, p_D)$.
3. The "singular values" are nonnegative and ordered: $0 \leq \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_R$.
4. The tensor rank equals the number of nonzero singular values:
   $R = \#\{\sigma_r : \sigma_r > 0,\ r = 1, \ldots, \min(p_1, p_2, \ldots, p_D)\}$.
5. The decomposition is unique up to signs, even when some singular values are repeated.

Many of the properties listed above are shared with the matrix SVD. The key difference between LROAT for higher-order tensors and the matrix SVD is that not all tensors are odeco. From property 1, a tensor with $\text{rank}(\mathcal{B}) > \min(p_1, p_2, \ldots, p_D)$ does not admit an orthogonal decomposition. Moreover, even if $\text{rank}(\mathcal{B}) \leq \min(p_1, p_2, \ldots, p_D)$, there is no guarantee that $\mathcal{B}$ is odeco. Thus, in real data applications, the LROAT decomposition is nearly always an approximation of the original tensor (hence "approximation" is part of the acronym).

2.2. Fixed, Low-rank Orthogonally Decomposable Tensor Regression

To obtain the fixed-rank version of our low-rank, orthogonally decomposable tensor regression model, which we refer to as the LODTR model, we incorporate the LROAT decomposition as an additional assumption in a GLM. The systematic component of the model can be specified as

$$\begin{aligned} g(\mu_i) &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \langle \mathcal{B},\ \mathcal{X}_i \rangle \ \text{ s.t. rank}(\mathcal{B}) = R \text{ and odeco} \\ &= \alpha + \boldsymbol{\gamma}^T \boldsymbol{z}_i + \left\langle \sum_{r=1}^{R} \sigma_r \boldsymbol{\beta}_{1r} \circ \boldsymbol{\beta}_{2r} \circ \cdots \circ \boldsymbol{\beta}_{Dr},\ \mathcal{X}_i \right\rangle \ \text{ s.t.} \\ B_1^T B_1 &= \cdots = B_D^T B_D = I_R. \qquad (4) \end{aligned}$$

The model (4) is similar to the model (1), except that the factor matrices are restricted to be orthogonal. In fact, when $D = 2$ (i.e., the tensors are matrices), the two models are essentially equivalent. However, the estimated factor matrices $\widehat{B_1}, \widehat{B_2}$ will not necessarily be identical because of the orthogonality constraint in Equation (4). The interpretation is that the parameter spaces of models (1) and (4) are the same (namely the set of all real-valued matrices $\mathcal{B}$), but model (1) allows any basis for the column and row spaces of $\mathcal{B}$, while model (4) requires an orthogonal basis. When $D \geq 3$, the models are not equivalent because the parameter space of model (1) is all rank-R tensors of size $p_1 \times \cdots \times p_D$, while the parameter space of model (4) is all rank-R odeco tensors of size $p_1 \times \cdots \times p_D$.

We propose to estimate model (4) with a projected gradient descent algorithm. For a general problem $\min f(x)$ s.t. $x \in \mathcal{S}$, where $\mathcal{S}$ denotes a set of constraints, the updates of the projected gradient descent algorithm take the form

$$x^{(k+1)} = \prod\nolimits_{\mathcal{S}} \left(x^k - \delta^k \nabla f(x^k) \right),$$

where δ^k is a step size and $\prod_{\mathcal{S}}$ denotes the projection operator onto the feasible set $\mathcal{S}$. The projection operator can be defined as $\prod_{\mathcal{S}}(x) = \arg\min_{z \in \mathcal{S}} \|x - z\|_F$. The projected gradient descent method can be interpreted as the usual gradient descent method with the additional modification that each update is required to lie in the feasible set. The projection step ensures that each update lies in the feasible set while minimizing the distance between the update and the update that would be obtained from an unconstrained version of the problem.

The constraint set in model (4) is the space of rank-R odeco tensors. To project onto that set, recall problem (3). Problem (3) is identical to the problem defined by projecting onto the space of rank-R odeco tensors. This suggests that the algorithm described by Chen and Saad (2009) may be used for the projection step in a projected gradient descent algorithm for fitting model (4). The gradient descent step may be based on a suitable loss function, such as the negative log-likelihood. The details of the algorithm for fitting model (4) are included in the supplementary material.

2.3. Penalized Orthogonally Decomposable Tensor Regression

To avoid a priori specification of the rank in the estimation of model (4), we penalize the objective function with a function of the tensor singular values, which we call penalized orthogonally decomposable tensor regression (PODTR). For ease of exposition, we refer to the penalized objective function as the PODTR "model." The PODTR model can be written as

$$\min_{\mathcal{B}} \; -\ell\ell(\mathcal{B}) + P_\lambda\left(\sigma(\mathcal{B})\right), \tag{5}$$

where $\mathcal{B}$ is the parameter tensor (assumed to be odeco), $-\ell\ell(\cdot)$ is the negative log-likelihood of a GLM, $\sigma(\cdot)$ extracts the singular values of an odeco tensor, and $P_\lambda(\cdot)$ is a sparsity-inducing penalty function. For this study, we only consider the LASSO penalty [i.e., $P_\lambda\left(\sigma(\mathcal{B})\right)$ is the ℓ_1 norm of the tensor singular values, multiplied by λ]. By Property 4 of the LROAT decomposition (Section 2.1), the LASSO penalty applied to the tensor singular values will yield a low-rank estimate of $\mathcal{B}$.

Note that for $D = 2$, the model (5) is identical to model (2) considered by Zhou and Li (2014). Thus, the solution simply involves shrinking the singular values of $\mathcal{B}$. However, for $D \geq 3$, the solution involves first projecting onto the set of odeco tensors, then shrinking the singular values. More precisely, the problem constrains the magnitudes of the singular values but places no restriction on the parameter space of matrices when $D = 2$, but constrains both the magnitudes of the singular values and the parameter space of tensors when $D \geq 3$. Unlike model (2), model (5) is non-convex for the tensor case as a consequence of the odeco constraint.

To estimate model (5), we adapt Zhou and Li's algorithm for solving Equation (2), which is based on the proximal gradient method. The standard proximal gradient method applied to Equation (2) splits the problem into two steps: one that minimizes the negative log-likelihood by gradient descent, and one that applies the proximal operator associated with $P_\lambda(\sigma(B))$. The proximal operator associated with the LASSO penalty is the soft thresholding operator, so solving Equation (2) involves shrinking the singular values of B to zero. Zhou and Li's algorithm modifies the standard proximal gradient method by incorporating an extrapolation step due to Nesterov (1983) that achieves an accelerated convergence rate.

We must further modify the proximal gradient method to apply it to (5). Namely, we must account for the fact that tensors of order $D \geq 3$ must be projected onto the set of odeco tensors before shrinking the singular values. We propose to include a step that projects onto the set of rank-$\min(p_1, \ldots, p_D)$ odeco tensors—the largest rank possible for the LROAT decomposition of a tensor of size $p_1 \times \cdots \times p_D$. When the actual tensor rank $R < \min(p_1, \ldots, p_D)$, this does not pose a problem because we will have $\sigma_{R+1} = \cdots = \sigma_{\min(p_1,\ldots,p_D)} = 0$. A brief sketch of the updates for the algorithm is

Gradient descent step: $\dot{\mathcal{B}}^{(k+1)} = \mathcal{B}^{(k)} - \delta^k \nabla f\left(\mathcal{B}^{(k)}\right)$

Projection step:

$$\ddot{\mathcal{B}}^{(k+1)} = \arg\min \left\| \dot{\mathcal{B}}^{(k+1)} - \textstyle\sum_{r=1}^{\min(p_1,\ldots,p_D)} \sigma_r \boldsymbol{\beta}_{1r} \circ \cdots \circ \boldsymbol{\beta}_{Dr} \right\|_F$$

s.t. $B_1^T B_1 = \cdots = B_D^T B_D = I_{\min(p_1,\ldots,p_D)}$

Soft thresholding step:

$$\mathcal{B}^{(k+1)} = \textstyle\sum_{r=1}^{\min(p_1,\ldots,p_D)} \left(\ddot{\sigma}_r - \delta^k \lambda\right)_+ \ddot{\boldsymbol{\beta}}_{1r} \circ \cdots \circ \ddot{\boldsymbol{\beta}}_{Dr},$$

where the function f denotes the negative log-likelihood and $(\cdot)_+ = \max(0, \cdot)$. The complete details of the proposed algorithm for fitting model (5) are included in the supplementary material.

We note that users may wish to choose the value of the tuning parameter λ in practice. As in other regularized regression problems, cross validation may be employed for that purpose, albeit at considerable computational cost. In fact, cross-validation could also be used to choose the rank for fixed-rank versions of the model, with the additional limitations that (i) it is not possible to obtain parameter estimates that appear intermediate in rank and 2) the largest rank model one may consider will be dictated by the sample size (i.e., R cannot be so large that $R\sum_{d=1}^{D} p_d \geq n$). However, we strongly encourage users not to pick a single value of the tuning parameter. The major strength of the proposed method, and tensor regression models in general, is the ability to visualize the estimated parameter tensor. The visualization is more effective when one fits the model over a range of values of the tuning parameter and compares how the estimates change as the amount of regularization varies. The merits of visualizing the estimates under a variety of levels of regularization can be seen in both the simulation experiment and real data application.

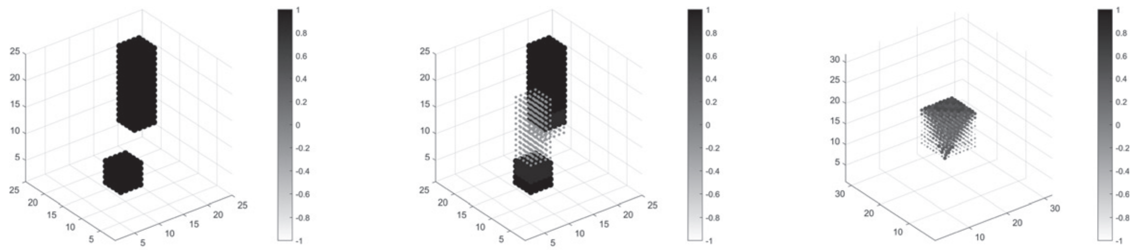
3. Simulation Experiment

We used simulated data to illustrate some of the properties of the LODTR and PODTR models compared to the CPD model. For $i = 1, \ldots, N$, we generated the response as $Y_i \sim \text{Normal}\left(\beta_0 + \langle \mathcal{X}_i, \mathcal{B}\rangle,\ 1\right)$ and the "image" covariate $(\mathcal{X}_i)_{jkl} \sim \text{Normal}(0,\ 1)$. We set $\beta_0 = 1$ in all of our experiments.

The simulation experiment (as well as the real data application in Section 4) were performed in MATLAB. The CPD model was fit with functions from the TensorReg Toolbox (Zhou 2017). MATLAB scripts implementing the proposed methods are included in the Supplementary Material. Some of the scripts rely on functions from the Tensor Toolbox (Bader et al. 2021).

We compare what happens when the true parameter tensor $\mathcal{B}$ is odeco vs. not. We constructed several different parameter tensors:

1. A $25 \times 25 \times 25$ rank-2 non-odeco tensor in which the signal region was comprised of two hyperrectangles (labelled $\mathcal{B}_1$ in Figure 1(i)). The values of the signal elements were set to 1 and all other elements were set to 0.
2. Odeco and non-odeco versions of a $32 \times 32 \times 32$ rank-4 tensor with two signal regions: one comprised of a cross and the other comprised of nested cubes (see the supplementary material for the visualization). For the non-odeco version, the values of the signal elements of the cross, inner cube and outer cube were set to 1, 1.5, and 0.75, respectively, with all other elements set to 0. Chen and Saad's algorithm was used to obtain the odeco version.
3. A $32 \times 32 \times 32$ rank-9 odeco tensor using the singular values and corresponding singular vectors from the SVD of a two-dimensional image of a triangle (labelled $\mathcal{B}_2$ in Figure 1(iii)). See the accompanying MATLAB code in the supplementary material for additional details about constructing the odeco tensor. An additional signal tensor was constructed by modifying the singular values so that they taper slowly to zero.



(i) True $\mathcal{B}_1$ (not odeco)

(ii) closest orthogonal approximation to $\mathcal{B}_1$

(iii) True $\mathcal{B}_2$ (odeco)

Figure 1. True parameter tensors for the simulation experiment.

4. A 32 × 32 × 32 rank-10 non-odeco tensor in which the signal region was constructed to look like a triangular pyramid (see the supplementary material for the visualization).

We simulated data from the models 100 times and calculated the squared error of the estimates, defined as $\|\mathcal{B} - \widehat{\mathcal{B}}\|_F^2$. The quartiles of the squared errors are shown in Table 1. The LODTR and PODTR estimates typically have smaller squared errors than the CPD estimates, even when the true parameter tensor is not odeco. Most of the poor performance of the CPD model can be attributed to its sensitivity to the initial value, as evidenced by the large discrepancies between Q1 and Q3. For the hyperrectangles parameter, the first quartile of the squared errors of the rank-2 CPD estimates was much lower than those for LODTR and PODTR, indicating that the relaxed assumptions of the CPD model can be advantageous when the initialization happens to be good (which seems to be infrequently). The CPD model also performs well when the sample size is large relative to the number of parameters to estimate (see the two scenarios with $N = 1000$ in Table 1), and can outperform both the LODTR and PODTR models when the sample size is large and the parameter tensor is non-odeco (see the $N = 1000$, rank-10 non-odeco triangle scenario in Table 1). The advantage of the PODTR model becomes apparent when the sample size is very limited and the singular values of the parameter tensor are not dominated by a few, relatively large values (see the $N = 350$, modified rank-9 odeco triangle scenario in Table 1). The better performance of PODTR when the sample size is limited is also consistent with the results from the real data application in the next section. In contrast, the better performance of LODTR for the majority of the simulation scenarios can be attributed to the fact that the LODTR models more closely resemble the models from which the data were actually generated.

Computational summaries for the methods (computation time, iterations until convergence, and the objective function evaluated at the solution) and a thorough discussion of computational aspects can be found in the supplementary material. In brief, the computational aspects of the algorithms used to fit the LODTR and CPD models tended to be similar. For the PODTR algorithm, the projection onto the set of rank-min$(p_1, \dots, p_D)$ odeco tensors significantly increases the number of iterations until convergence and computation time relative to the two other methods. However, after the estimate is obtained for the first value of λ in a sequence, the algorithm may be warm started for each subsequent value of λ. Warm starting reduces the computation time to comparable, and sometimes faster, times than those required by the CPD and LODTR algorithms.

If we visualize the estimates, then several additional properties of the LODTR and PODTR models vs. the CPD model become apparent. Figure 1 shows the true hyperrectangle and triangle parameters (labelled $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively), while Figures 2 and 3 show several instances of their corresponding estimates (see the supplementary material for additional estimates). For illustration, we also show the closest orthogonal approximation to the hyperrectangle parameter (Figure 1(ii)), which we obtained by applying Chen and Saad's algorithm. Notice that the closest orthogonal approximation captures the main features of the true signal, but there are some unavoidable artifacts caused by the orthogonality constraint. Comparing the true parameters to their estimates, we can observe that

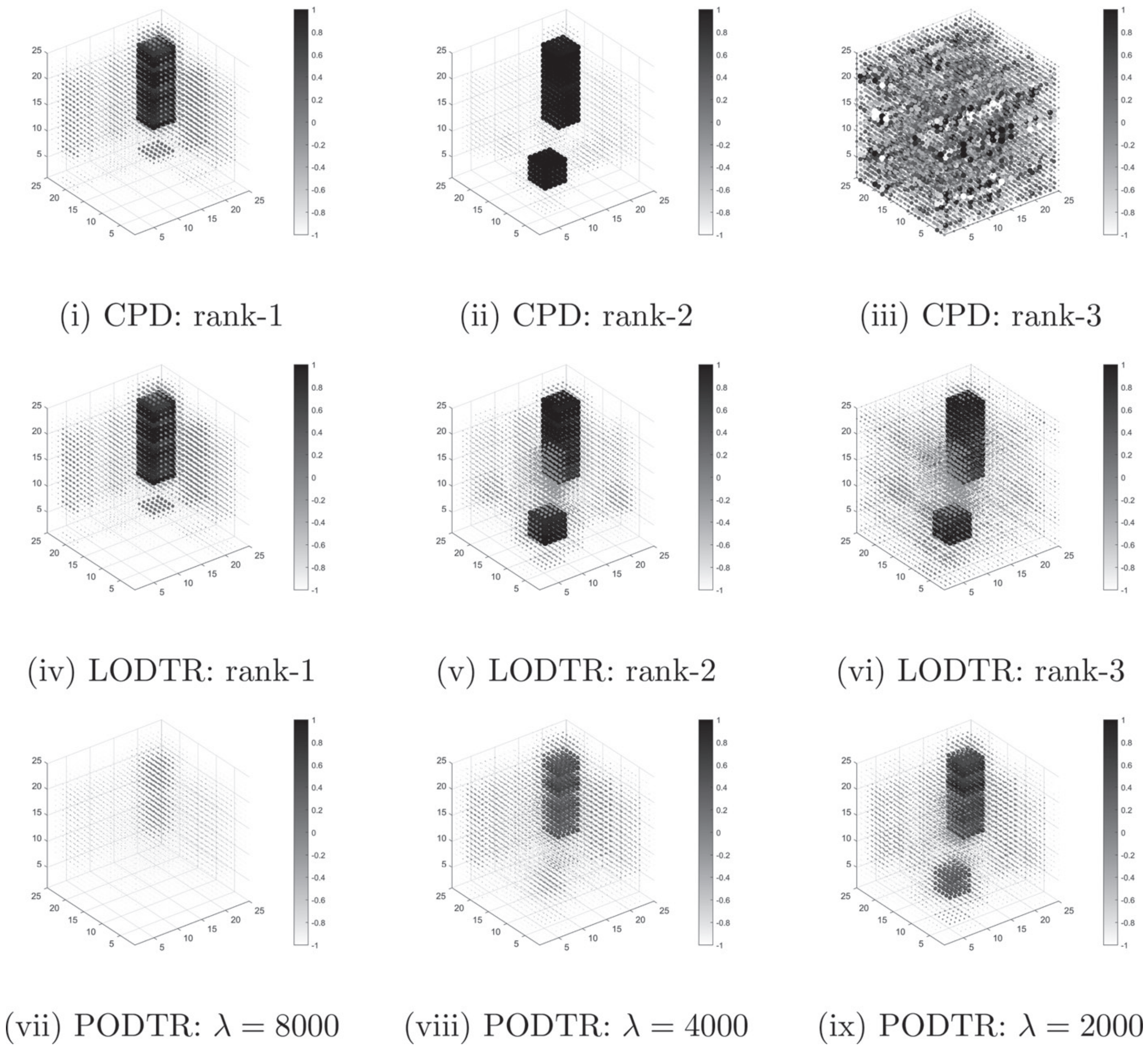
1. The CPD model performs well when the sample size is large relative to the number of parameters to estimate, but breaks down quickly as the number of parameters approaches the sample size (e.g., Figure 2(iii) and Figure 3(iii)).
2. When the true parameter tensor is not odeco, the LODTR and PODTR estimates improve as higher rank models are fitted, but reach a point beyond which no further improvement is possible (e.g., Figure 2(iv)–(vi)). However, the higher rank estimates closely resemble the closest orthogonal approximation of the true parameter tensor, so the estimates still manage to capture most of the main features.
3. When the true parameter tensor is odeco, the LODTR and PODTR models perform better the CPD model, especially for higher rank models (e.g., Figure 3(ii) vs. Figure 3(v)).
4. The PODTR model is able to produce estimates that appear intermediate in rank. For example, Figure 2(vii) is intermediate between the zero tensor and the rank-1 estimates from the other models, and Figure 2(viii) is intermediate between the rank-1 and rank-2 estimates from the other models. Such a property is a result of continuous shrinkage of the singular values as opposed to fitting discrete rank models.

4. Application to Intra-Subject Visual Stimulus Decoding

One application of the proposed methods is intra-subject visual stimulus decoding. One first obtains a set of fMRI images as a

Table 1. Summary of the squared errors of the estimated parameter tensors. Quartiles are calculated based on the squared errors from 100 simulations; the best value in each row is highlighted in bold. The sample size *N* for each simulation is given in the first column.

		CPD					LODTR					PODTR					
		Rank					Rank					Tuning Parameter λ					
		1	2	3	4	5	1	2	3	4	5	2500	2000	1500	1000	250	100
Hyperrectangles	Q1	152.9	**0.6**	404.5	1323.7	2480	152.3	18.1	30.7	45.3	56.6	123.4	93.6	67.2	54.1	45.1	46.1
(rank-2, non-odeco)	Q2	157.8	254	1344.4	2090.6	3017.9	155.9	**18.9**	33.1	48	60.3	129.8	98.2	73.4	59.4	50.7	52
$N = 400$	Q3	168	964.5	1660.4	2371.3	3329.4	159.6	**19.6**	35.4	52	68	139.2	107.1	80.3	64.4	55.1	58
Cross + nested cubes	Q1	76.6	131.9	525.1	859.5	1070.1	73.1	**31.5**	54.4	79.8	96.5	77.4	67.7	62.5	59.6	60.2	64.9
(rank-4, non-odeco)	Q2	277.2	419.9	639.5	929.7	1200.8	75.1	**32.7**	58.7	86.7	102	82	72	66.6	63.5	67.2	74.3
$N = 500$	Q3	291.4	447.9	691.5	1032.6	1364.4	77.2	**34.4**	62.9	92	111.4	86	77.3	71.4	69.2	76.8	98.8
Cross + nested cubes	Q1	59.4	96.5	496.7	818.9	1056.8	57.8	12.4	**3.5**	6.5	9.6	65	50.3	38.4	32.4	28.4	30.2
(rank-4, odeco)	Q2	260.7	392.3	598.4	889	1151.1	59.4	12.7	**3.8**	8.1	11.7	68.3	53.3	41.4	35.6	31.4	33
$N = 500$	Q3	277.6	419.8	648.8	955.7	1263.1	60.4	13.2	**4.1**	9.2	13.1	72.1	56.5	44.6	38.6	35.2	37.7
Triangle	Q1	11.2	20.1	38.3	229.4	290.8	11	**9.5**	17.4	24.2	29	31.5	25	19.2	15.1	14.6	15.5
(rank-9, odeco)	Q2	11.8	21.9	169.5	257.3	316.9	11.3	**10.9**	19.4	28	33.3	33	26.1	19.8	16.1	16	16.6
$N = 500$	Q3	77.8	122.5	182.5	275.6	352.7	**11.6**	18	27.1	34.3	39.5	35.2	27.9	21.2	17.1	16.8	17.6
Triangle	Q1	112.5	190.1	371	1719.9	2196.8	**110**	179	261	331.2	380.8	148.7	147.6	148.9	152.9	162.4	166
(rank-10, non-odeco)	Q2	117.5	204.7	1253.2	1872.4	2418.8	**112.3**	185.6	275.7	353.3	410.2	156.5	155.1	156.4	159.9	169.4	173.7
$N = 500$	Q3	583.8	879.6	1364.1	2039.1	2649.7	**115.4**	193	292.3	385.9	443.8	164.2	162.5	163.1	168	175.8	181
Triangle	Q1	9.8	6.3	7.9	10.5	15.3	9.8	6	4.5	**4.3**	5.6	15.7	13	10	7.3	6.3	7.4
(rank-9, odeco)	Q2	9.9	12.3	8.6	12.1	25.1	9.9	6.1	**4.6**	5	6.7	16.3	13.4	10.3	7.6	6.8	8.2
$N = 1000$	Q3	10.1	12.8	16.5	21.3	27.4	10	6.3	**4.7**	6.3	8.2	16.7	13.9	10.7	8	7.3	9.4
Triangle	Q1	98.6	58.7	**44.8**	58.2	101.5	98.6	123.7	154.2	189.3	228	115.2	117.1	120.1	126.2	143.1	151.8
(rank-10, non-odeco)	Q2	99.4	**60**	69.6	69.5	129.7	99.5	126.2	158.1	193.4	232.8	117.6	119.9	123.4	129.6	146.3	154.7
$N = 1000$	Q3	100.7	**61.7**	79.4	105.5	144.6	100.6	128	161.5	199.2	239.7	120	122.2	125.9	131.6	149.9	159
Triangle, modified	Q1	208.3	369.8	620.3	NA	NA	175.2	283.1	367.4	NA	NA	111.9	**111**	111.9	115.6	125.3	125.1
(rank-9, odeco)	Q2	219.7	394.7	666.6	NA	NA	201.4	307.6	395.3	NA	NA	**116.8**	117.2	119.4	122.7	129.3	128.2
$N = 350$	Q3	227.5	415.3	745.4	NA	NA	213.4	331.1	432.2	NA	NA	**118.8**	119.9	121.6	124.7	131.5	132.6



(i) CPD: rank-1

(ii) CPD: rank-2

(iii) CPD: rank-3

(iv) LODTR: rank-1

(v) LODTR: rank-2

(vi) LODTR: rank-3

(vii) PODTR: $\lambda = 8000$

(viii) PODTR: $\lambda = 4000$

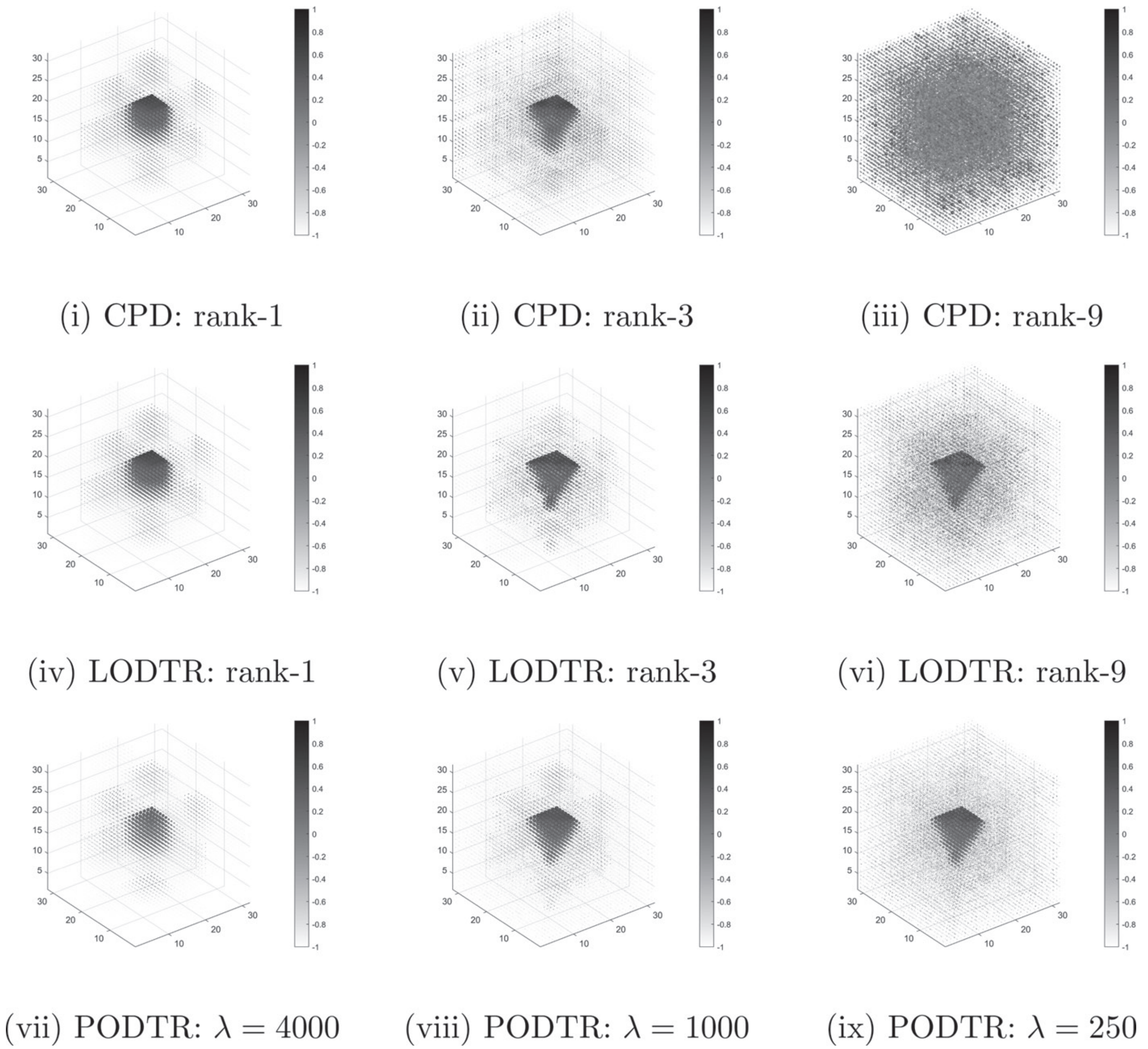
(ix) PODTR: $\lambda = 2000$

Figure 2. Estimates of the non-odeco tensor $\mathcal{B}_1$ under the CPD, LODTR, and PODTR models. Sample size: $N = 400$.

subject performs several visual stimulus tasks (typically, viewing different types or categories of images). Then, in the encoding step of visual stimulus decoding, one uses the fMRI images to train a classification model in which the fMRI image itself serves as the predictor and the type of image being viewed (i.e., the task) serves as the response. For the decoding step, one uses the model to predict the type of image being viewed for a separate test set of fMRI images. The goal is to identify the brain regions involved for one task versus another.

Haxby et al. (2001) showed that the visual object recognition (VOR) dataset can be used for visual stimulus decoding. The VOR data (Haxby et al. 2001; Hanson, Matsuka, and Haxby 2004; O'Toole et al. 2005) is freely available through the OpenNeuro project (openneuro.org) and contains fMRI data for six subjects as they viewed different types of images (viewing each type of image is considered a task). There were eight different types of images: bottles, scissors, shoes, chairs, houses, cats, faces, and scrambled images. Data were recorded for each subject for 12 runs (except Subject 5, who had data for 11 runs). For each run, the image types were shown sequentially in blocks of 24 s duration with 12 s of rest between blocks of different image type. Within a block, images of the same type were shown every 2 s (images of the same type includes different objects of the same type and different views of the same object).

The data for one run consists of two parts: (i) an events file giving the time of onset of an image and the type of image and (ii) a $40 \times 64 \times 64 \times 121$ fMRI tensor containing raw BOLD signal. The $40 \times 64 \times 64$ component of the fMRI tensor represents the spatial component with voxel dimensions 3.50 mm × 3.75 mm × 3.75 mm. The component of length 121 represents the time component with dimension 2.5 s. For each block of image type, we extracted the time slices corresponding to 6 s after the onset of the first image (to account for the delay in the hemodynamic response) up to the onset of the last image



(i) CPD: rank-1 (ii) CPD: rank-3 (iii) CPD: rank-9

(iv) LODTR: rank-1 (v) LODTR: rank-3 (vi) LODTR: rank-9

(vii) PODTR: $\lambda = 4000$ (viii) PODTR: $\lambda = 1000$ (ix) PODTR: $\lambda = 250$

Figure 3. Estimates of the odeco tensor $\mathcal{B}_2$ under the CPD, LODTR, and PODTR models. Sample size: $N = 1200$.

in the block. Because the resolution of the time component of the fMRI tensor does not match up exactly with the duration of a block, the number of time slices we obtained for a block varied slightly across image types, but was usually 6 or 7 time slices per block.

We applied the CPD, LODTR, and PODTR models to the VOR data for binary visual stimulus decoding. We used a logistic regression model with the indicator of image type as the response and the subset of the fMRI tensor corresponding to the visual cortex as the predictor. The visual cortex is located in the posterior part of the brain. By visualizing a whole-brain fMRI image, we determined that indices 1–40 along the first spatial dimension, indices 9–30 along the second spatial dimension, and indices 23–48 along the third spatial dimension should conservatively capture the visual cortex. The size of the resulting image was $40 \times 22 \times 26$. We trained the models on the data for 11 runs ($N^{\text{train}} \approx 145$), then obtained the classification accuracies based on the predictions for the remaining run ($N^{\text{test}} \approx 13$). We repeated the process for all 6 subjects and each of their 11 or 12 runs, for a total of 71 fitted models per method. For clarity of exposition, we refer to the results for run j as those for the models trained with the data from run j left out.

For our classification tasks, we focused on classifying scrambled images vs. each of the other image types. We compared scrambled images against each of the others because the signal should be strongest for those comparisons, and so they serve as good examples to illustrate the uses of the proposed models. Scrambled images should result in general activation of the visual cortex, whereas images of other types may result in stronger activation of specific subregions within the visual cortex; visualizing the parameter estimates from the trained models can help identify the subregions of the visual cortex that are associated with a particular task. For LODTR and CPD, we were restricted to fitting rank-1 models. The limited sample

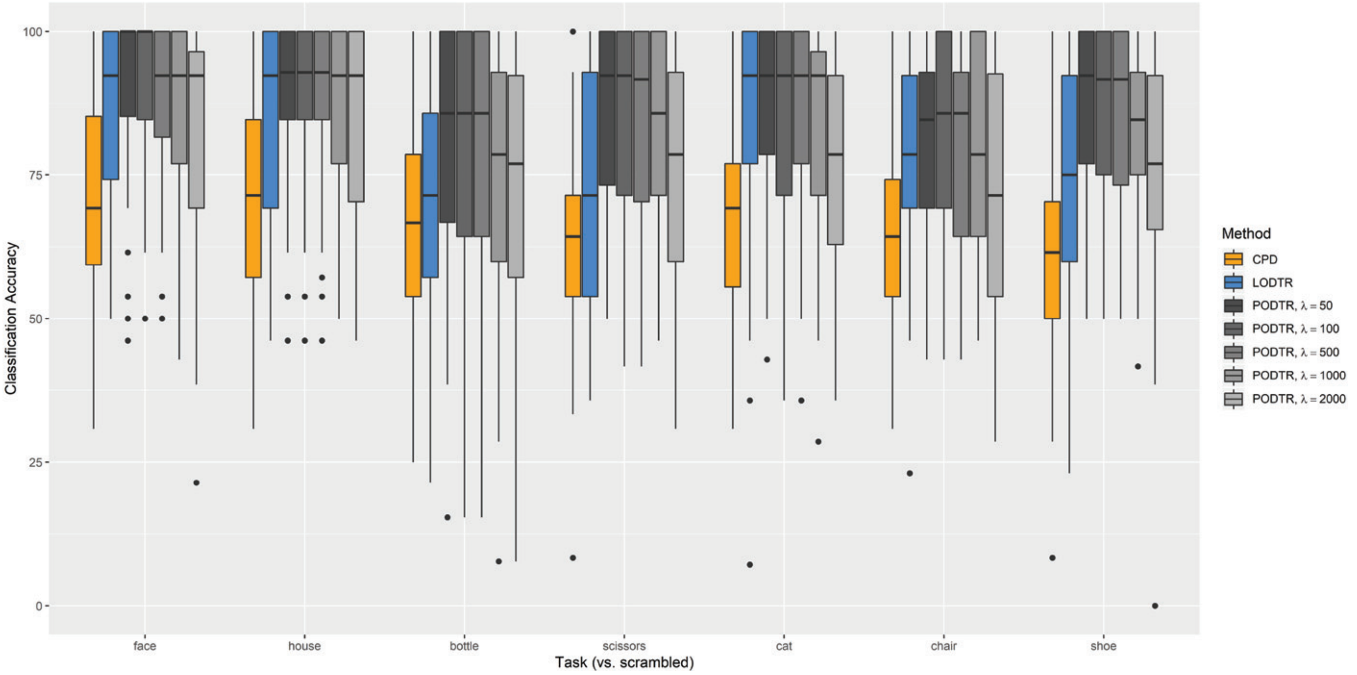


Figure 4. Classification accuracy (%) for the CPD, LODTR, and PODTR models applied to the VOR data. Boxplots are accuracies for 6 subjects × 11 or 12 runs per subject = 71 fitted models per method per task.

size relative to the dimension of the fMRI tensor precluded fitting higher rank models. For PODTR, we fitted the model for a sequence of values of the tuning parameter. For each type of image comparison, we reported the classification accuracy achieved for the test dataset. We classified an image as 1 if $\hat{p} > 0.5$ and 0 if $\hat{p} < 0.5$, where $\hat{p}$ was defined as

$$\hat{p} = \frac{1}{1 + \exp\left[-\hat{\beta}_0 - \langle \mathcal{X}_i, \hat{\mathcal{B}} \rangle\right]}.$$

The classification accuracies achieved by each method are shown in Figure 4, where the accuracies are aggregated across the six subjects. Average classification accuracies disaggregated by subject can be found in the supplementary material. Using the fMRI image as input, all three models were able to predict which type of visual stimulus was being viewed better than by random guessing. The LODTR and PODTR models performed better than the CPD model, sometimes substantially so. The PODTR model performed better than the LODTR model except for some instances in which a large amount of regularization was used. In general, the classification accuracy of the PODTR model tended to be better with less regularization, and its performance declined as more regularization was added.

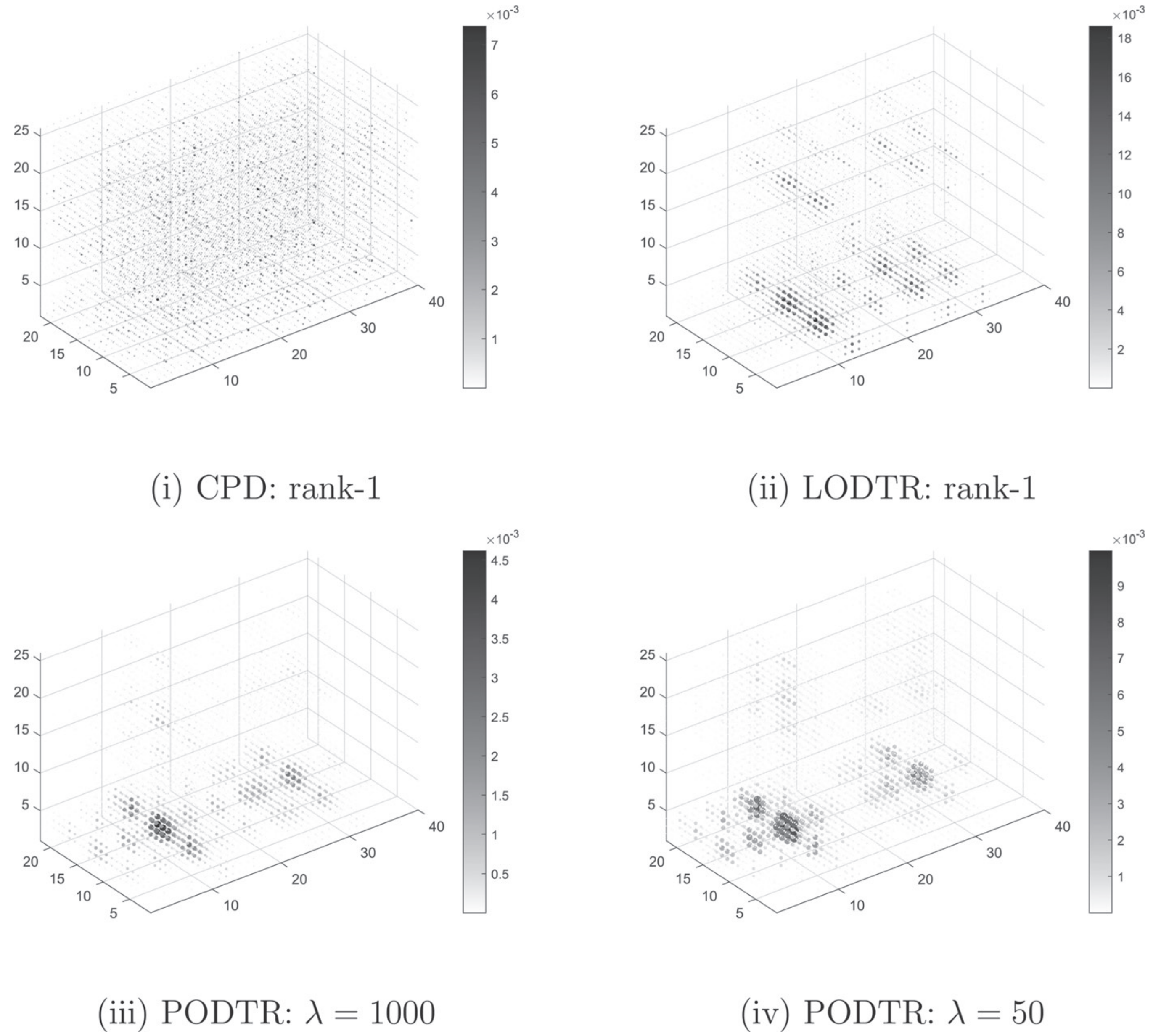
In addition to making accurate predictions, another goal of the proposed models is to understand what components of the image are most related to the response. For visual stimulus decoding, that can be interpreted as understanding which areas of the brain are responsible for discriminating between one type of viewed image and another, as measured by the BOLD signal. If we visualize the estimated parameter tensor, then we hope to find compact clusters of voxels with relatively large magnitude values, which we can interpret as regions of the brain that may be associated with performing one task versus another.

Figures 5–7 show the absolute values of the average estimate of the parameter tensor over Subject 1's 12 runs for selected tasks. Results for the other tasks can be found in the supplementary material. The PODTR model estimates are shown for two values of the tuning parameter, representing different extremes with respect to the amount of shrinkage ($\lambda = 1000$ results in large shrinkage and lower rank estimates, while $\lambda = 50$ results in less shrinkage and higher rank estimates). Note that the colorbars for the PODTR estimates are typically on a smaller scale than those for the rank-1 CPD and LODTR estimates, reflecting the powerful effect of shrinkage. The size of the dots in the PODTR estimates have been magnified to a comparable level as the CPD and LODTR estimates to enable easier identification of regions with (relatively) large values.

In all of the figures, regions of the estimated tensor with large magnitude values (relative to other regions within the same estimate) can be interpreted as possible signal regions—that is, regions that are strongly associated with the type of image being viewed. The CPD estimates (Figures 5(i)–7(i)) are noisy, with the largest values occurring in scattered voxels rather than in spatially compact locations. Such estimates are difficult to interpret because no clearly defined regions stand out.

The rank-1 LODTR estimates (Figures 5(ii)–7(ii)) are less noisy than the CPD estimates, especially for the scrambled vs. face (Figure 5(ii)) and scrambled vs. house (Figure 6(ii)). For those estimates, distinct, spatially compact regions of the estimate stand out, suggesting that the corresponding region of the brain may be relevant to the task. For scrambled vs. bottle (Figure 7(ii)), the voxels with relatively large magnitude values are few and they do not form compact clusters, making them difficult to interpret.

The PODTR estimates (Figures 5(iii)–7(iii) and 5(iv)–7(iv)) are easiest to interpret and become more interpretable with less shrinkage. The same regions that stood out in the rank-1 LODTR estimates for scrambled vs. face and scrambled vs. house stand out in the PODTR estimates as well. The most



(i) CPD: rank-1

(ii) LODTR: rank-1

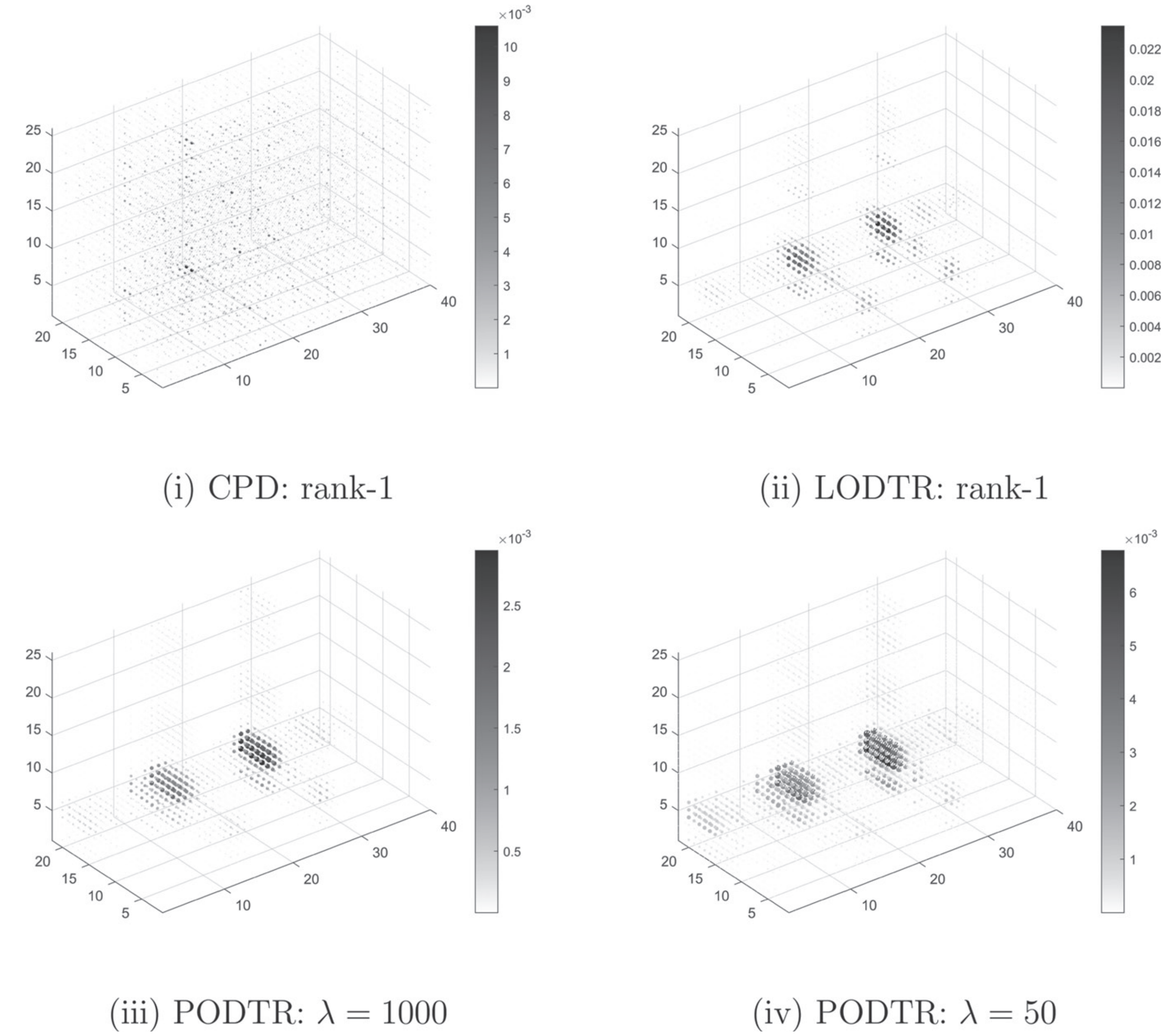
(iii) PODTR: $\lambda = 1000$

(iv) PODTR: $\lambda = 50$

Figure 5. Scrambled vs. Face: Average $\widehat{\mathcal{B}}$ for Subject 1's 12 runs.

improvement occurs for the scrambled vs. bottle task under less shrinkage. Although certain regions of the estimates stand out to some extent with greater shrinkage (Figure 7(iii)), they are much more well-defined with less shrinkage (Figure 7(iv)). From Table 4 in the supplementary materials, the scrambled vs. bottle estimates obtained with less shrinkage had better classification accuracies than those obtained with greater shrinkage. Thus, we can say that the estimates with large magnitude values located in compact clusters tended to yield better classification accuracies for predictions made on new data.

The clusters of voxels with relatively large values in Figures 5–7 represent regions within the visual cortex that the models identified as being associated with the tasks. A natural question is: Do the clusters of voxels have biological relevance? Namely, do they correspond to regions of the brain that are already known to be associated with the tasks? Figure 8 shows a 2D projection of the PODTR estimate with $\lambda = 50$ from Figures 5–7 overlain on one of Subject 1's anatomical slices. The blue clusters (scrambled vs. face estimate) appear to correspond to the fusiform face area, the magenta clusters (scrambled vs. house estimate) to the parahippocampal place area, and the green clusters (scrambled vs. bottle estimate) to the lateral occipital complex. The fusiform face area, parahippocampal place area, and lateral occipital complex have been shown to be associated with viewing images of faces, locations and things (Kanwisher et al. 2001), respectively, which is consistent with the tasks shown. We caution that such an assessment is tentative at best. Expert judgment would be necessary to confirm that the regions corresponding to the clusters of voxels are indeed those that have been identified by others. See the supplementary material for a precise description of how Figure 8 was created, as well as the corresponding figures for the CPD and LODTR estimates.



(i) CPD: rank-1

(ii) LODTR: rank-1

(iii) PODTR: $\lambda = 1000$

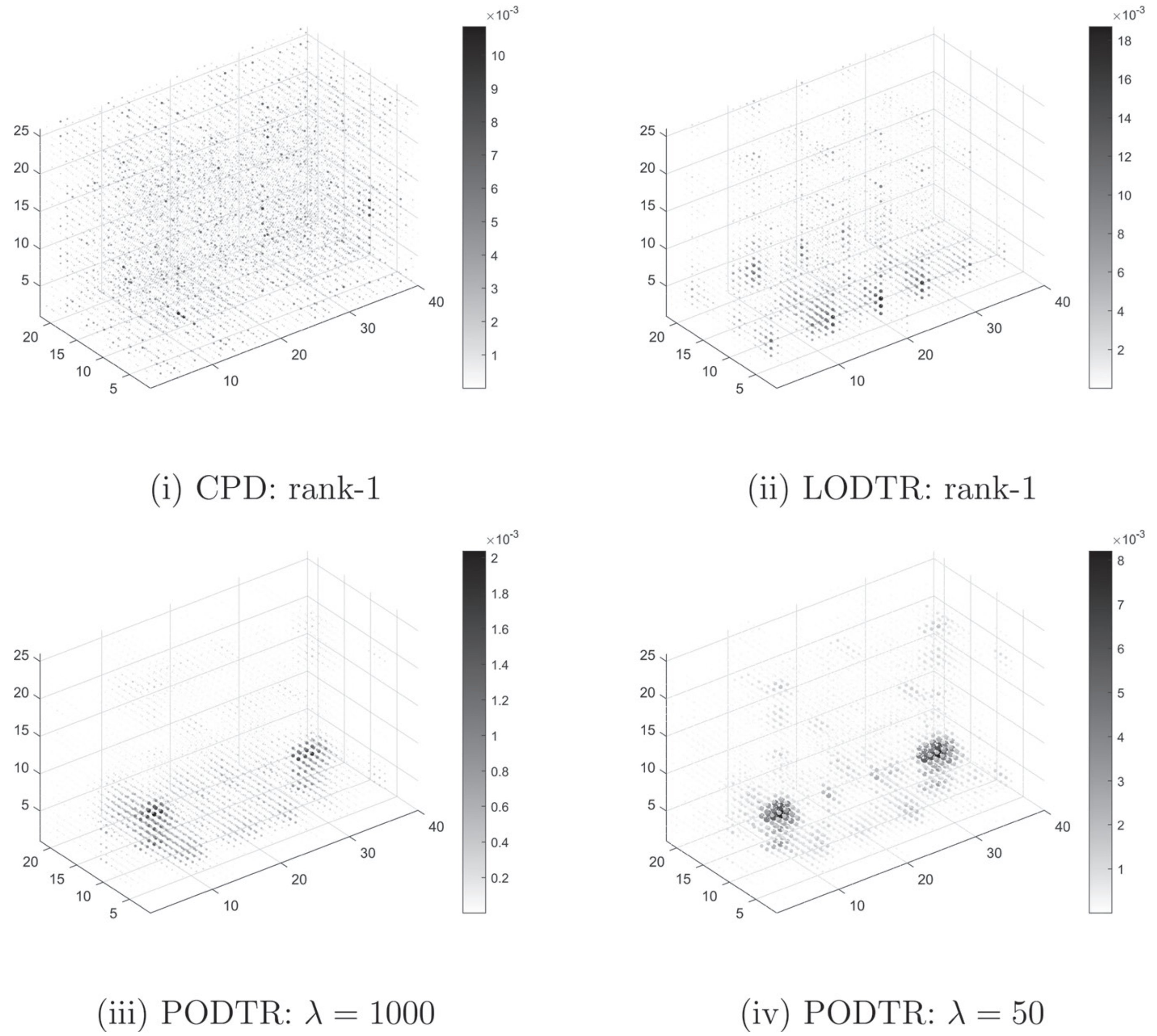
(iv) PODTR: $\lambda = 50$

Figure 6. Scrambled vs. House: Average $\widehat{\mathcal{B}}$ for Subject 1's 12 runs.

5. Conclusions and Future Directions

Through a simulation experiment and the analysis of real fMRI dataset, we demonstrated that the proposed low-rank, orthogonally decomposable tensor regression models have the potential to perform better than existing methods both with respect to predictive performance and interpretation. Because the LODTR model is a special case of the CPD model to which we compared (and the two are in fact equivalent in some instances), the superior performance of the LODTR model can be attributed to some extent to the estimation algorithm rather than the model itself. The projected gradient descent algorithm used to fit the LODTR model yielded parameter estimates with smaller squared errors in simulation and higher classification accuracies in our real data analysis. In both the simulation and real data analysis, the parameter estimates tended to be easier to interpret when visualized. In contrast, the alternating minimization algorithm used to fit the CPD model yielded parameter estimates that were easy to interpret only for lower rank models in the simulation; the algorithm completely failed to produce interpretable estimates in the real data analysis and higher rank models in the simulation. Both algorithms used the same initial value, tolerance criterion, and tolerance value, so the differences cannot be attributed to differences in the initialization or stopping rule. Furthermore, the LODTR and CPD models fitted for the real data analysis were all rank 1, and in that case the LODTR and CPD models are equivalent. One possible explanation for the poor performance of the alternating minimization algorithm is that alternating among parameters makes the algorithm more likely to find local optima, especially when the number of parameters is large relative to the sample size. The projected gradient descent algorithm updates all parameters at once, perhaps making it less likely to get stuck in a local optimum.

The results from the real data application were also consistent with some of the properties of the models demonstrated in the



(i) CPD: rank-1

(ii) LODTR: rank-1

(iii) PODTR: $\lambda = 1000$

(iv) PODTR: $\lambda = 50$

Figure 7. Scrambled vs. Bottle: Average $\widehat{\mathcal{B}}$ for Subject 1's 12 runs.

simulation experiment. In the simulation, the performance of the CPD model was observed to break down quickly as the number of parameters approached the sample size. In the real data application, fitting the logistic regression model entailed estimating 88 parameters with only 145 observations, so it is not surprising that the CPD model produced poor estimates. The simulation also revealed that the LODTR and PODTR models are not capable of estimating the true parameter tensor perfectly when it is not odeco. Real data do not follow a model, so there is no concept of a "true" parameter tensor, and even if there was, we wouldn't know what it would look like. Then, we do not know whether the clusters of voxels evident in Figures 5–7 are actually voxels associated with the tasks, or whether some of the clusters are potentially artifacts caused by the orthogonality constraints. Besides not knowing what a "good" estimate of the parameter should look like, we also cannot compare with the CPD estimates (which are unconstrained) because the CPD model performed so poorly. At the least, we can say that the estimates with the most clearly defined clusters also produced the best classification accuracies in Table 4, suggesting that the clusters of voxels are in fact associated with the tasks.

In future work we hope to explore the convergence properties of the algorithms we have proposed for fitting the LODTR and PODTR models. Empirically, the algorithms appear to converge based on the simulated and real examples we have tried. Moreover, the solutions are frequently better than the solutions from the alternating minimization algorithm of Zhou, Li, and Zhu (2013). Unfortunately, there are myriad challenges for proving global convergence for tensors of dimension ≥ 3. One challenge is that both of the algorithms we have proposed rely on Chen and Saad's algorithm for the LROAT decomposition. They were unable to prove global convergence of their algorithm. In addition, their algorithm is not guaranteed to converge to the global optimum of problem (3). Thus, the projection steps in

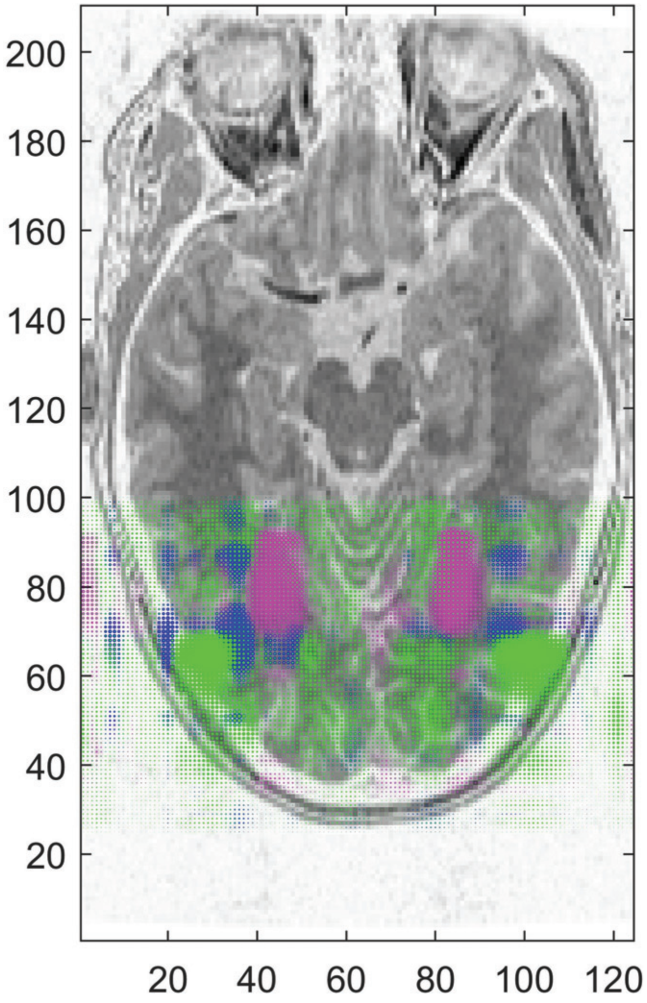


Figure 8. 2D projection of the PODTR estimate with $\lambda = 50$ from Figures 5–7 overlain on one of Subject 1's anatomical slices. Blue: scrambled vs. face estimate; Magenta: scrambled vs. house estimate; Green: scrambled vs. bottle estimate.

the proposed algorithms are not necessarily carried out exactly. That creates a difficulty for applying any known results about the convergence of projected gradient descent with non-convex constraints. Another challenge is in dealing with the non-convex constraints imposed by the low-rank and orthogonality assumptions. Some results regarding the convergence of projected gradient descent are known in the context of low-rank matrices (see Jain and Kar (2017)) and have been applied in the context of the Tucker decomposition for tensors (e.g., Chen, Raskutti, and Yuan (2019)). However, given that not all tensors admit an orthogonal decomposition, it is not clear how one might extend those results to the LROAT decomposition utilized in the proposed algorithms. Recent algorithmic developments for symmetric odeco tensors (Anandkumar et al. 2014; Wang and Song 2017), which have been applied in the context of Gaussian mixture models and topic models, may prove useful for developing algorithms with established convergence properties for decomposing general odeco tensors, which could then replace Chen and Saad's algorithm in the proposed methods. Finally, a major challenge in proving convergence of the algorithm for fitting for the PODTR model is that it is neither a projected gradient descent algorithm nor a proximal gradient algorithm in the strict sense, but rather a mixture of the two. Although superficially similar to Zhou and Li's algorithm for spectral-regularized matrix regression, the proposed algorithm involves an additional projection step onto the set of odeco tensors. For matrices, this is simply the SVD and is a change of basis rather than a projection. For tensors, it is a nonconvex projection for all but the rare cases in which the tensor already admits an orthogonal decomposition. Since the proposed algorithm adds a nontrivial modification to the standard proximal gradient method, it may not be possible to apply any known convergence results for the proximal gradient method.

Supplementary Material

Algorithms: A detailed description of the algorithms used to estimate the LODTR and PODTR models. (.pdf file)

Additional results: Complete set of figures and results associated with the simulation experiment and VOR data analysis. We also include some computational results related to the algorithms, such as the number of iterations until convergence and computation time. (.pdf file) *MATLAB code:* Collection of scripts and functions necessary to reproduce the simulation results and real data analysis. (.m files)

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and M. Telgarsky (2014), "Tensor Decompositions for Learning Latent Variable Models," *Journal of Machine Learning Research*, 15, 2773–2832. [202]

Bader, B. W., Kolda, T. G. (2021), "Tensor Toolbox for MATLAB, Version 3.2." Available at *www.tensortoolbox.org*. [193]

Chen, H., Raskutti, G., and Yuan, M. (2019), "Non-Convex Projected Gradient Descent for Generalized Low-Rank Tensor Regression," *Journal of Machine Learning Research*, 20, 1–37. [202]

Chen, J., and Saad, Y. (2009), "On the Tensor SVD and the Optimal Low Rank Orthogonal Approximation of Tensors," *SIAM Journal on Matrix Analysis and Applications*, 30, 1709–1734. [192]

Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2017), "Bayesian Tensor Regression," *Journal of Machine Learning Research*, 18, 1–31. [191]

Guo, W., Kotsia, I., and Patras, I. (2012), "Tensor Learning for Regression," *IEEE Transactions on Image Processing*, 21, 816–827. [191]

Hanson, S. J., T. Matsuka, and J. V. Haxby (2004), "Combinatorial Codes in Ventral Temporal Lobe for Object Recognition: Haxby (2001) Revisited: Is There a "face" Area?" *NeuroImage*, 23, 156 – 166. [196]

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001), "Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex," *Science*, 293, 2425–2430. [196]

Hitchcock, F. L. (1927), "The Expression of a Tensor or a Polyadic as a Sum of Products," *Journal of Mathematics and Physics*, 6, 164–189. [191]

Hu, W., Pan, T., Kong, D., and Shen, W. (2021), "Nonparametric Matrix Response Regression With Application to Brain Imaging Data Analysis," *Biometrics*. *https://doi.org/10.1111/biom.13362* [191]

Hu, W., Shen, W., Zhou, H., and Kong, D. (2020), "Matrix Linear Discriminant Analysis," *Technometrics*, 62, 196–205. [191]

Hung, H., and Wang, C.-C. (2012), "Matrix Variate Logistic Regression Model With Application to EEG Data," *Biostatistics*, 14, 189–202. [191]

Jain, P., and P. Kar (2017), "Non-Convex Optimization for Machine Learning," *Foundations and Trends in Machine Learning*, 10, 142–363. [202]

Kanwisher, N., Downing, P., Epstein, R., and Kourtzi, Z. (2001), "Functional Neuroimaging of Visual Recognition," in *Handbook of Functional Neuroimaging of Cognition* (1st ed., Chapter 5), eds. R. Cabeza and A. Kingstone. Cambridge, MA: Massachusetts Institute of Technology Press, pp. 109–152. [199]

Kolda, T. G., and B. W. Bader (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500. [191]

Kong, D., An, B., Zhang, J., and Zhu, H. (2020), "L2rm: Low-Rank Linear Regression Models for High-Dimensional Matrix Responses," *Journal of the American Statistical Association*, 115, 403–424. [191]

Nesterov, Y. (1983), "A Method of Solving a Convex Programming Problem With Convergence Rate O(1/k2)," *Soviet Mathematics Doklady 27*(2), 372–376. [193]

O'Toole, A. J., Jiang, F., Abdi, H., and Haxby, J. V. (2005), "Partially Distributed Representations of Objects and Faces in Ventral Temporal Cortex," *Journal of Cognitive Neuroscience*, 17, 580–590. [196]

Sun, W. W., and Li, L. (2017), "STORE: Sparse Tensor Response Regression and Neuroimaging Analysis," *Journal of Machine Learning Research*, 18, 4908–4944. [191]

Tan, X., Zhang, Y., Tang, S., Shao, J., Wu, F., and Zhuang, Y. (2013), "Logistic Tensor Regression for Classification," in *Intelligent Science and Intelligent Data Engineering*, J. Yang, F. Fang, and C. Sun (Eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 573–581. [191]

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [191]

Wang, M., and Song, Y. (2017), "Tensor Decompositions Via Two-Mode Higher-Order SVD (HOSVD)," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, eds. A. Singh and J. Zhu, *Proceedings of Machine Learning Research*, Fort Lauderdale, FL: PMLR, pp. 614–622. [202]

Zhou, H. (2017), "Matlab TensorReg Toolbox Version 1.0." Available at *https://hua-zhou.github.io/TensorReg/*. [193]

Zhou, H., and Li, L. (2014), "Regularized Matrix Regression," *Journal of the Royal Statistical Society*, Series B, 76, 463–483. [191,193]

Zhou, H., Li, L., and Zhu, H. (2013), "Tensor Regression With Applications in Neuroimaging Data Analysis," *Journal of the American Statistical Association*, 108, 540–552. [191,201]