Statistics and Its Interface Volume 17 (2024) 249–274

Bayesian methods in tensor analysis

Yiyao Shi and Weining Shen[*]

Tensors, also known as multidimensional arrays, are useful data structures in machine learning and statistics. In recent years, Bayesian methods have emerged as a popular direction for analyzing tensor-valued data since they provide a convenient way to introduce sparsity into the model and conduct uncertainty quantification. In this article, we provide an overview of frequentist and Bayesian methods for solving tensor completion and regression problems, with a focus on Bayesian methods. We review common Bayesian tensor approaches including model formulation, prior assignment, posterior computation, and theoretical properties. We also discuss potential future directions in this field.

Keywords and phrases: Imaging analysis, Posterior inference, Recommender system, Tensor completion, Tensor decomposition, Tensor regression.

1. INTRODUCTION

Tensors, also known as multidimensional arrays, are higher dimensional analogues of two-dimensional matrices. Tensor data analysis has gained popularity in many scientific research and business applications, including medical imaging [8], recommender systems [81], relational learning [97], computer vision [86] and network analysis [55]. There is a vast literature on studying tensor-related problems such as tensor decomposition [49, 74, 93], tensor regression [28, 89], tensor completion [86], tensor clustering [8, 89], tensor reinforcement learning and deep learning [89]. Among them, tensor completion and tensor regression are two fundamental problems and we focus on their review in this article.

Tensor completion aims at imputing missing or unobserved entries in a partially observed tensor. Important applications of tensor completion include providing personalized services and recommendations in context-aware recommender systems (CARS) [81], restoring incomplete images collected from magnetic resonance imaging (MRI) and computerized tomography (CT) [23], and inpainting missing pixels in images and videos [59, 68]. In this review, we divide tensor completion methods into trace norm based methods and decomposition based methods, and introduce common approaches in each category.

Different from tensor completion, tensor regression investigates the association between tensor-valued objects and other variables. For example, medical imaging data such as brain MRI are naturally stored as a multi-dimensional array, and tensor regression methods are applied to analyze their relationship with clinical outcomes (e.g., diagnostic status, cognition and memory score) [54, 90]. Based on the role that the tensor-valued object plays in the regression model, tensor regression methods can be categorized into tensor predictor regression and tensor response regression.

Frequentist approaches have been successful in tensor analysis [102, 8]. In recent years, Bayesian approaches have also gained popularity as they provide a useful way to induce sparsity in tensor models and conduct uncertainty quantification for estimation and predictions. In this article, we will briefly discuss common frequentist approaches to solve tensor completion and regression problems and focus on Bayesian approaches. We also review two commonly used tensor decompositions, i.e., CANDECOMP/PARAFAC (CP) decomposition [45] and the Tucker decomposition [98], since they are the foundations for most Bayesian tensor models. For example, many Bayesian tensor completion approaches begin with certain decomposition structure on the tensor-valued data and then use Bayesian methods to infer the decomposition parameters and impute the missing entries. Based on the decomposition structures being utilized, we divide these methods into CP-based, Tucker-based, and nonparametric methods. For tensor regression methods, we classify the Bayesian tensor regression into Bayesian tensor predictor regression and Bayesian tensor response regression. For each category, we review the prior construction, model setup, posterior convergence property and sampling strategies.

The rest of this article is organized as follows. Section 2 provides a background introduction to tensor notations, operations and decompositions. Section 3 and 4 review common frequentist approaches for tensor completion and regression problems, respectively. Section 5 and 6 review Bayesian tensor completion and regression approaches, including the prior construction, posterior computing, and theoretical properties. Section 7 provides concluding remarks and discusses several future directions for Bayesian tensor analysis. Figure 1 shows an outline of our review.

2. BACKGROUND

In this section, we follow [49] and introduce notation, definitions, and operations related to tensors. We also discuss two popular tensor decomposition approaches and highlight some challenges in tensor analysis.

arXiv: 2302.05978

[*] Corresponding author.

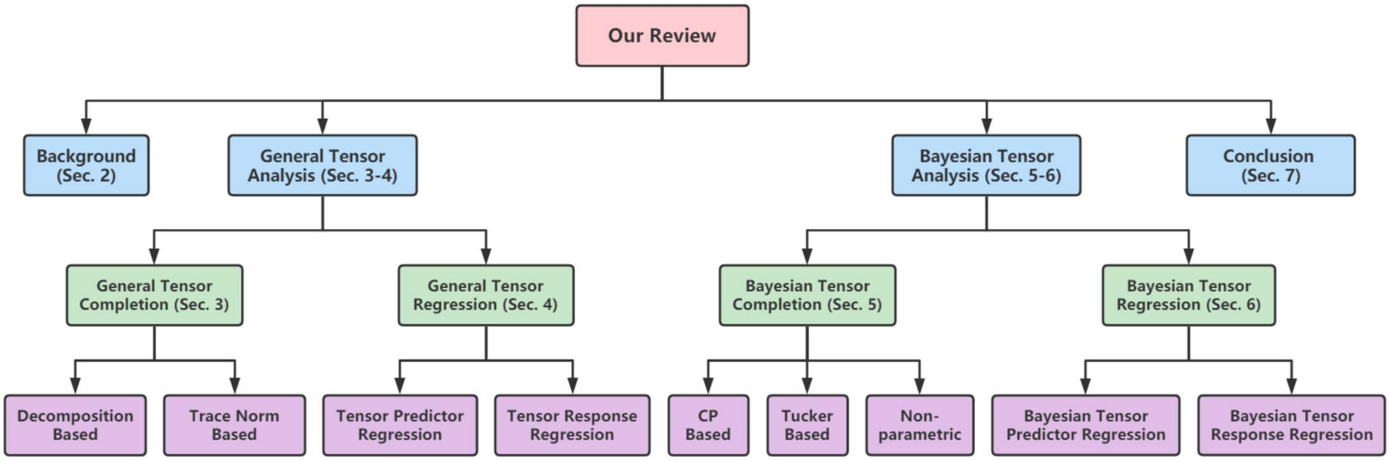

Our Review
Background (Sec. 2)
General Tensor Analysis (Sec. 3-4)
Bayesian Tensor Analysis (Sec. 5-6)
Conclusion (Sec. 7)
General Tensor Completion (Sec. 3)
General Tensor Regression (Sec. 4)
Bayesian Tensor Completion (Sec. 5)
Bayesian Tensor Regression (Sec. 6)
Decomposition Based
Trace Norm Based
Tensor Predictor Regression
Tensor Response Regression
CP Based
Tucker Based
Non-parametric
Bayesian Tensor Predictor Regression
Bayesian Tensor Response Regression

Figure 1. Outline of this survey.

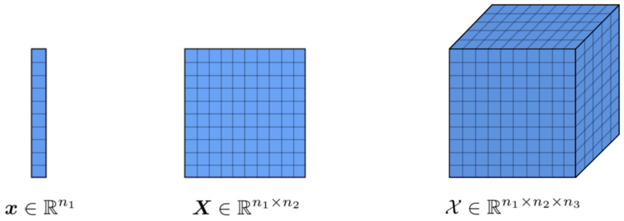

$\boldsymbol{x} \in \mathbb{R}^{n_1}$
$\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$
$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Figure 2. An example of first, second and third-order tensors.

2.1 Basics

Notation: A tensor is a multidimensional array. The dimension of a tensor is also known as *mode*, *way*, or *order*. A first-order tensor is a vector; a second-order tensor is a matrix; and tensors of order three and higher are referred to as higher-order tensors (see Figure 2). In this review, a tensor is denoted by Euler script letter $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$. Here d is the order of tensor $\mathcal{X}$, and n_k is the marginal dimension of the kth mode ($k = 1, 2, ..., d$). The $(i_1, i_2, ..., i_d)$th element of the tensor $\mathcal{X}$ is denoted by $x_{i_1 i_2 \ldots i_d}$ for $i_k = 1, 2, ..., n_k$ and $k = 1, 2, ..., d$. Subarrays of a tensor are formed through fixing a subset of indices in the tensor. A *fiber* is a vector defined by fixing all but one indices of a tensor, and a *slice* is a matrix created by fixing all the indices except for those of two specific orders in the tensor. For instance, a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ has column, row and tube fibers, which are respectively denoted by $\mathcal{X}_{:i_2 i_3}, \mathcal{X}_{i_1 : i_3}$, and $\mathcal{X}_{i_1 i_2 :}$ (see Figure 3(a)(b)(c)). A third-order tensor also has horizontal, lateral, and frontal slices, denoted by $\mathcal{X}_{i_1 ::}, \mathcal{X}_{: i_2 :}$ and $\mathcal{X}_{:: i_3}$, respectively (see Figure 3(d)(e)(f)).

Tensor operations: Here we introduce some tensor operations following [49]. The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$ is defined as the square root of the sum of the squares of all elements, i.e.,

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} x^2_{i_1 i_2 \ldots i_d}}. \tag{1}$$

For two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$, their *inner product* is the sum of products of their corresponding entries, i.e.,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_d=1}^{n_d} x_{i_1 i_2 \ldots i_d} y_{i_1 i_2 \ldots i_d}. \tag{2}$$

It immediately follows that $\langle \mathcal{X}, \mathcal{X} \rangle = \|\mathcal{X}\|^2$. The *tensor Hadamard product* of two tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ and $\mathcal{Y} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ is denoted by $\mathcal{X} *_H \mathcal{Y} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$; each entry of $\mathcal{X} *_H \mathcal{Y}$ is the product of the corresponding entries in tensors $\mathcal{X}$ and $\mathcal{Y}$:

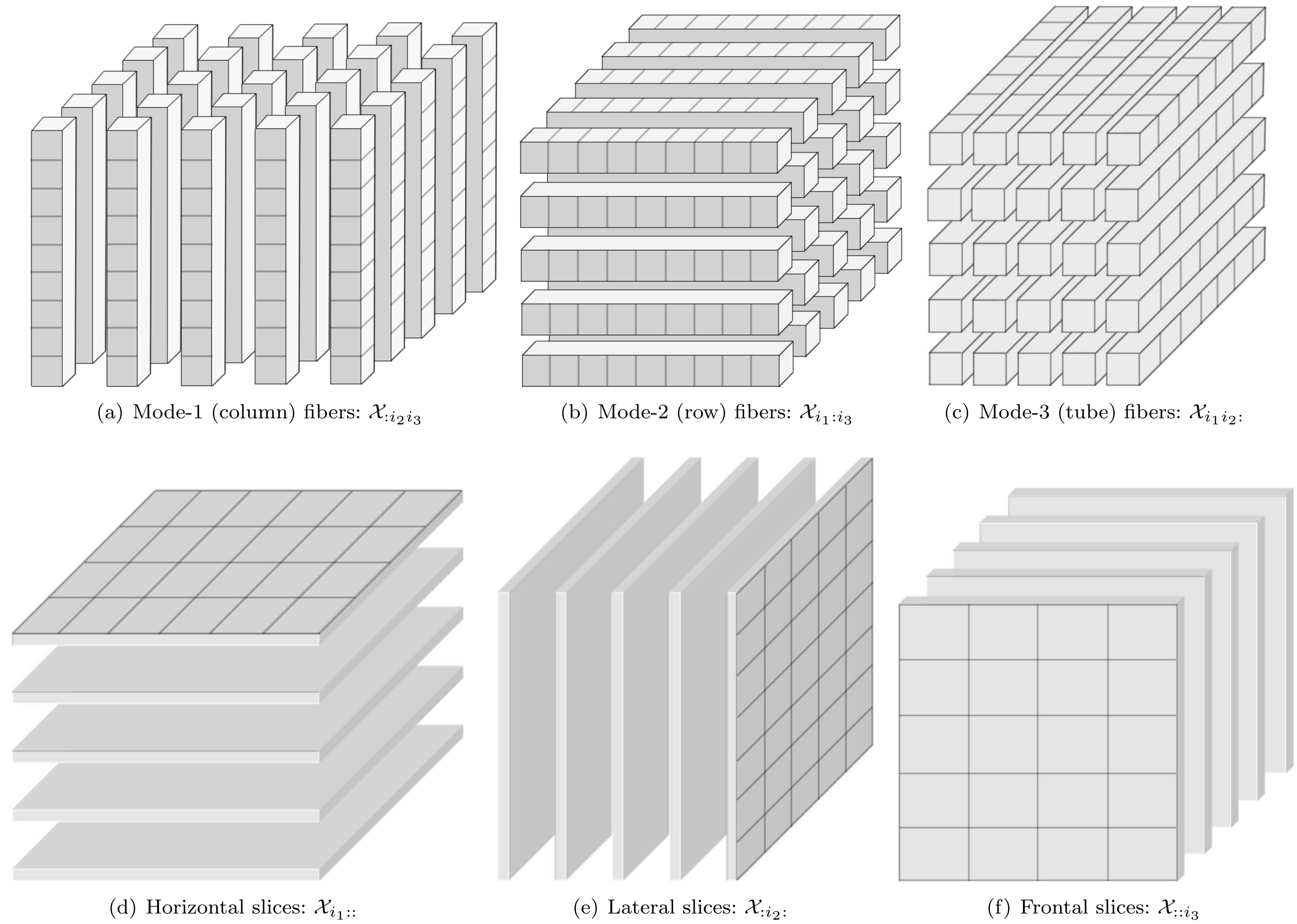
$$(\mathcal{X} *_H \mathcal{Y})_{i_1 \ldots i_d} = x_{i_1 \ldots i_d} \cdot y_{i_1 \ldots i_d}. \tag{3}$$

The *tensor contraction product*, also known as the *Einstein product*, of two tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d \times p_1 \times \ldots \times p_k}$ and $\mathcal{Y} \in \mathbb{R}^{p_1 \times \ldots \times p_k \times m_1 \times \ldots \times m_q}$ is denoted by $\mathcal{X} * \mathcal{Y} \in \mathbb{R}^{n_1 \times \ldots \times n_d \times m_1 \times \ldots \times m_q}$ and defined as

$$\begin{aligned}(\mathcal{X} * \mathcal{Y})&_{i_1, \ldots, i_d, j_1, \ldots, j_q} \\ &= \sum_{c_1=1}^{p_1} \cdots \sum_{c_k=1}^{p_k} x_{i_1, \ldots, i_d, c_1, \ldots, c_k} y_{c_1, \ldots, c_k, j_1, \ldots, j_q},\end{aligned} \tag{4}$$

where $i_g = 1, 2, ..., n_g$ for $g = 1, 2, ..., d$, and $j_s = 1, 2, ..., m_s$ for $s = 1, 2, ..., q$. Moreover, a dth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$ is *rank one* if it can be written as the outer product of d vectors, i.e,

$$\mathcal{X} = \boldsymbol{p}^1 \circ \boldsymbol{p}^2 \circ \cdots \circ \boldsymbol{p}^d,$$



(a) Mode-1 (column) fibers: $\mathcal{X}_{:i_2i_3}$

(b) Mode-2 (row) fibers: $\mathcal{X}_{i_1:i_3}$

(c) Mode-3 (tube) fibers: $\mathcal{X}_{i_1i_2:}$

(d) Horizontal slices: $\mathcal{X}_{i_1::}$

(e) Lateral slices: $\mathcal{X}_{:i_2:}$

(f) Frontal slices: $\mathcal{X}_{::i_3}$

Figure 3. Example of fibers and slices of third-order tensor. This figure is reproduced based on Figure 2.1 and 2.2 in [49].

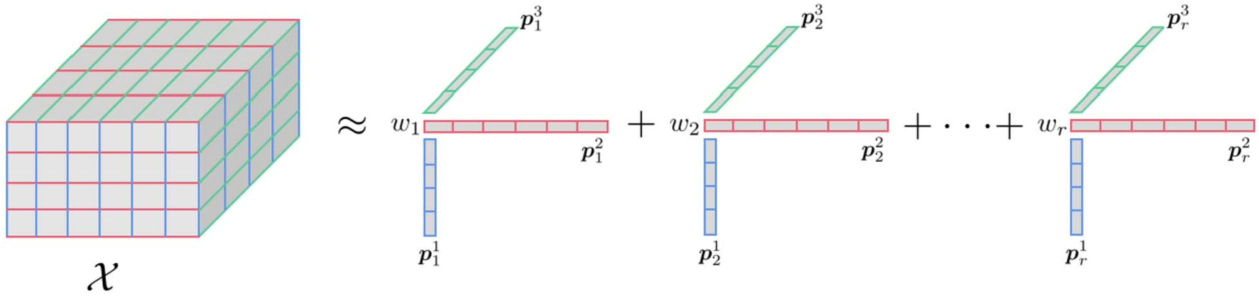


Figure 4. Rank-r CP decomposition for a third-order tensor: $\mathcal{X} \approx \sum_{j=1}^{r} w_j \boldsymbol{p}_j^1 \circ \boldsymbol{p}_j^2 \circ \boldsymbol{p}_j^3$.

where $\boldsymbol{p}^k = (p_1^k, p_2^k, ..., p_{n_k}^k) \in \mathbb{R}^{n_k}$ $(k = 1, 2, ..., d)$ is a vector, and the symbol "$\circ$" represents the vector outer product. It means that each element of the tensor $\mathcal{X}$ is the product of corresponding vector elements: $x_{i_1 i_2 ... i_d} = p_{i_1}^1 p_{i_2}^2 ... p_{i_d}^d$ for $i_k = 1, 2, ..., n_k$ and $k = 1, 2, ..., d$. A tensor $\mathcal{X}$ is *rank* r if r is the smallest number such that $\mathcal{X}$ is the sum of r outer products of vectors: $\mathcal{X} = \sum_{j=1}^{r} \boldsymbol{p}_j^1 \circ \boldsymbol{p}_j^2 \circ \cdots \circ \boldsymbol{p}_j^d$.

Tensor *matricization*, also known as tensor *unfolding* or *flattening*, is an operation that transforms a tensor into a matrix. Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_d}$, the kth-mode matricization arranges the mode-k fibers to be columns of the resulting matrix, which is denoted by $\boldsymbol{X}_{(k)}$ $(k = 1, 2, ..., d)$. The element $(i_1, i_2, ..., i_d)$ of tensor $\mathcal{X}$ corresponds to the entry (i_k, j) of $\boldsymbol{X}_{(k)}$, where $j = 1 + \sum_{t=1, t \neq k}^{d} (i_t - 1) J_t$ with $J_t = \prod_{m=1, m \neq k}^{t-1} n_m$. In addition, a tensor can be transformed into a vector through tensor *vectorization*. For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, the vectorization of $\mathcal{X}$ is denoted by $\mathrm{vec}(\mathcal{X}) \in \mathbb{R}^{\prod_{i=1}^{d} n_i}$. The element $(i_1, i_2, ..., i_d)$ of tensor $\mathcal{X}$ corresponds to the element $1 + \sum_{t=1}^{d}(i_t - 1)M_t$ of $\mathrm{vec}(\mathcal{X})$, where $M_t = \prod_{m=1}^{t-1} n_m$.

The *k-mode tensor matrix product* of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n_k}$ is denoted by $\mathcal{X} \times_k \boldsymbol{A}$, which is of size $n_1 \times \cdots \times n_{k-1} \times m \times n_{k+1} \times \cdots \times n_d$. Elementwise, we have $(\mathcal{X} \times_k \boldsymbol{A})_{i_1,...,i_{k-1},j,i_{k+1},...,i_d} = \sum_{i_k=1}^{n_k} \mathcal{X}_{i_1,...,i_d} \boldsymbol{A}_{j i_k}$. The *k-mode vector product* of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with a vector $\boldsymbol{a} \in \mathbb{R}^{n_k}$ is denoted by $\mathcal{X} \bar{\times}_k \boldsymbol{a}$, which is of size $n_1 \times \cdots \times n_{k-1} \times n_{k+1} \times \cdots \times n_d$. Elementwise, $(\mathcal{X} \bar{\times}_k \boldsymbol{a})_{i_1 \ldots i_{k-1} i_{k+1} \ldots i_d} = \sum_{i_k=1}^{n_k} x_{i_1 i_2 \ldots i_d} a_{i_k}$.

2.2 Tensor decompositions

Tensor decompositions refer to methods that express a tensor by a combination of simple arrays. Here we introduce two widely-used tensor decompositions and discuss their applications.

CP decomposition: The *CANDECOMP/PARAFAC decomposition* (*CP decomposition*) [45] factorizes a tensor into a sum of rank-1 tensors. For a dth-mode tensor $\mathcal{X}$, the rank-r CP decomposition is written as

$$\mathcal{X} \approx \sum_{j=1}^{r} w_j \boldsymbol{p}_j^1 \circ \boldsymbol{p}_j^2 \circ \cdots \circ \boldsymbol{p}_j^d, \tag{5}$$

where $w_j \in \mathbb{R}, \boldsymbol{p}_j^k \in \mathbb{S}^{n_k-1}, j = 1, ..., r, k = 1, 2, ..., d, \mathbb{S}^{n_k-1} = \{\boldsymbol{a} \in \mathbb{R}^{n_k} | \|\boldsymbol{a}\| = 1\}$, and $\circ$ is the outer product. See Figure 4 for a graphical illustration of CP decomposition. Sometimes the CP-decomposition is denoted by an abbreviation: $\mathcal{X} \approx [\![\boldsymbol{W}; \boldsymbol{P}^1, \boldsymbol{P}^2, ..., \boldsymbol{P}^d]\!]$, where $\boldsymbol{W} = \mathrm{diag}(w_1, ..., w_r) \in \mathbb{R}^{r \times r}$ is a diagonal matrix, and $\boldsymbol{P}^k = [\boldsymbol{p}_1^k, \boldsymbol{p}_2^k ..., \boldsymbol{p}_r^k] \in \mathbb{R}^{n_k \times r}$ are factor matrices. If tensor $\mathcal{X}$ admits a CP structure, then the number of free parameters changes from $\prod_{i=1}^{d} n_i$ to $r \times (\sum_{i=1}^{d} n_i - d + 1)$.

If Equation (5) attains equality, the decomposition is called an exact CP decomposition. Even for an exact CP decomposition, there is no straightforward algorithm to determine the rank r of a specific tensor, and in fact the problem is NP-hard [32]. In practice, most procedures numerically infer the rank by fitting CP models with different ranks and choosing the one with the best numerical performance.

Tucker decomposition: The *Tucker decomposition* factorizes a tensor into a core tensor multiplied by a matrix along each mode. Given a dth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$, the Tucker decomposition is defined as

$$\begin{aligned} \mathcal{X} &\approx \mathcal{C} \times_1 \boldsymbol{Q}^1 \times_2 \boldsymbol{Q}^2 \times_3 \cdots \times_d \boldsymbol{Q}^d \\ &= \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \cdots \sum_{j_d=1}^{m_d} c_{j_1 j_2 \ldots j_d} \boldsymbol{q}_{j_1}^1 \circ \boldsymbol{q}_{j_2}^2 \circ \cdots \circ \boldsymbol{q}_{j_d}^d, \end{aligned} \tag{6}$$

where $\mathcal{C} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$ is the core tensor, $\boldsymbol{Q}^k \in \mathbb{R}^{n_k \times m_k} (k = 1, 2, ..., d)$ are factor matrices, $c_{j_1 j_2 \ldots j_d} \in \mathbb{R}, \boldsymbol{q}_{j_k}^k \in \mathbb{S}^{n_k-1} (j_k = 1, 2, ..., m_k, k = 1, 2, ..., d)$. See Figure 5 for a graphical illustration of Tucker decomposition. The Tucker decomposition can be denoted as $\mathcal{X} \approx [\![\mathcal{C}; \boldsymbol{Q}^1, \boldsymbol{Q}^2, ..., \boldsymbol{Q}^d]\!]$. If $\mathcal{X}$ admits a Tucker structure, the number of free parameters in $\mathcal{X}$ changes from $\prod_{i=1}^{d} n_i$ to $\sum_{i=1}^{d}(n_i - 1) \times m_i + \prod_{i=1}^{d} m_i$.

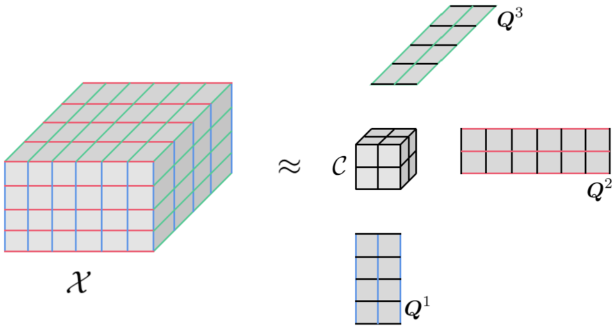


Figure 5. Tucker decomposition of the third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where $\mathcal{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ is the core tensor, and $\boldsymbol{Q}^k \in \mathbb{R}^{n_k \times m_k} (k = 1, 2, 3)$ are factor matrices.

The *k-rank* of $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, denoted by $\mathrm{rank}_k(\mathcal{X})$, is defined as the column rank of kth-mode matricization matrix $\boldsymbol{X}_{(k)}$. Let $R_k = \mathrm{rank}_k(\mathcal{X})$, then $\mathcal{X}$ is a rank-$(R_1, R_2, ..., R_d)$ tensor. Trivially, $R_k \leq n_k$ for $k = 1, 2, ..., d$. When the equality in Equation (6) is attained, the decomposition is called an exact Tucker decomposition. For a given tensor $\mathcal{X}$, there always exists an exact Tucker decomposition with core tensor $\mathcal{C} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$ where m_k is the true k-rank for $k = 1, 2, ..., d$. Nevertheless, for one or more k, if $m_k < R_k$, then the Tucker decomposition is not necessarily exact; and if $m_k > R_k$, the model will contain redundant parameters. Therefore, we usually want to identify the true tensor rank, i.e., $m_k = R_k$. While this job is easy for noiseless complete tensors, for tensors obtained in real-world applications, which are usually noisy or partially observed, the rank still needs to be determined by certain searching procedures.

2.3 Challenges in tensor analysis

In tensor analysis, the ultrahigh dimensionality of the tensor-valued coefficients and tensor data creates challenges such as heavy computational burden and vulnerability to model overfitting. Conventional approaches usually transform the tensors into vectors or matrices and utilize dimension reduction and low-dimensional techniques. However, these methods are usually incapable of accounting for the dependence structure in tensor entries. In the past decades, an increasing number of studies have imposed decomposition structures on the tensor-valued coefficients or data; thus naturally reducing the number of free parameters, and avoiding the issues brought by high dimensionality.

In this paper, we focus on tensor regression and tensor completion problems, where various decomposition structures including CP and Tucker have been widely used. Specifically, a large proportion of tensor completion methods are realized through inferring the decomposition structure based on the partially observed tensor, and then impute the missing values through the inferred decomposition structure. Also, tensor regression problems usually include tensor-valued coefficients, and decomposition structures are imposed on the coefficient tensor to achieve parsimony in parameters. In both situations, the decomposition is not performed on a completely observed tensor, thus the rank of the decomposition cannot be directly inferred from the data. Most optimization-based approaches determine the rank by various selection criteria, which may suffer from low stability issues. Bayesian approaches perform automatic rank inference through the introduction of sparsity-inducing priors. However, efficient posterior computing and study of theoretical properties of the posterior distributions are largely needed.

Low rankness and sparsity are commonly used assumptions in the literature to help reduce the number of free parameters. For non-Bayesian methods, oftentimes the task is formulated into an optimization problem, and the assumptions are enforced by sparsity-inducing penalty functions. In comparison, the Bayesian methods perform decompositions in the probabilistic setting, and enforce sparsity assumptions through sparsity priors. We will discuss more details about these approaches and how they resolve challenges in the following sections.

3. TENSOR COMPLETION

Tensor completion methods aim at imputing missing or unobserved entries from a partially observed tensor. It is a fundamental problem in tensor research and has wide applications in numerous domains. For instance, tensor completion techniques are extensively utilized in context-aware recommender systems (CARS) to provide personalized services and recommendations [43, 7, 92]. In ordinary recommender systems, the user-item interaction data are collected and formulated into a sparse interaction matrix, and the goal is to complete the matrix and thus recommend individualized items to the users. In CARS, the user-item interaction is collected with their contextual information (e.g., time and network), and the data are formulated as a high-order tensor where the modes respectively represent users, items, and contexts [2]. Therefore, the matrix completion problem in ordinary recommender systems is transformed into a tensor completion problem in CARS, and the purpose is to make personalized recommendations to users based on the collected user-item interaction and contextual information.

Apart from CARS, tensor completion is also applied in other research domains including healthcare, computer vision and chemometrics [86]. For example, medical images collected from MRI and CT play important roles in the clinical diagnosis process. Due to the high acquisition speed, oftentimes these high-order images are incomplete, thus necessitating the application of tensor completion algorithms [23, 5]. In the field of computer vision, color videos can be represented by a fourth-order tensor (length×width×channel×frame) by stacking the frames in time order (see Figure 6). Tensor completion can be adopted to impute the missing pixels and restore the lossy videos [59, 68]. As another example, chemometrics is a discipline that employs mathematical, statistical and other methods to improve chemical analysis. Tensor completion methods have been successfully applied on various benchmark chemometric datasets including semi-realistic amino acid fluorescence datasets [12] and flow injection datasets [69].

Tensor completion can be viewed as a generalization of matrix completion. Since the matrix completion problems

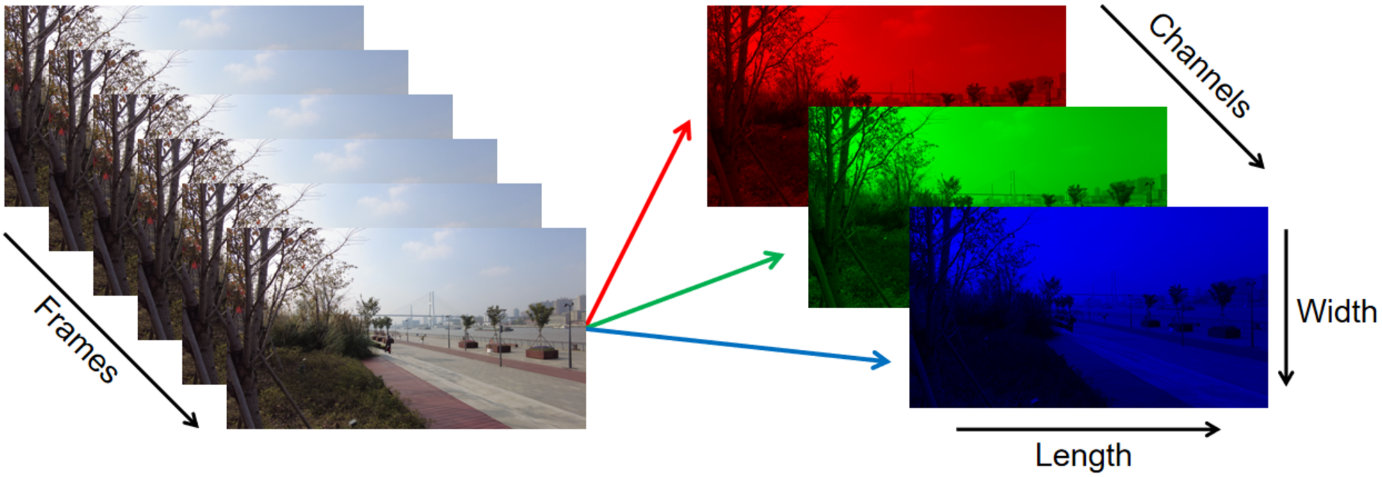


Figure 6. An illustration of color videos. Each frame of the video is formulated as a third-order tensor, where the modes are length, width and channels (RGB channels in this case). The frames are then stacked into a fourth-order tensor according to time order.

have been well-studied in the past few decades, a natural way to conduct tensor completion is to unfold or slice the tensor into a matrix (or matrices) and apply matrix completion methods to the transformed matrix (or matrices). Nevertheless, the performance and efficiency of such approaches are largely reduced by the loss of structural information during the matricization process and excessive computational cost due to the high dimensionality of the original tensor.

Under such circumstances, various methods that specifically focus on high-order tensor completion have been developed. Among these techniques, a classical group of approaches perform tensor completion through tensor decomposition. Generally speaking, these methods impose a decomposition structure on a tensor, and estimate the decomposition parameters based on the observed entries of the tensor. After that, the estimated decomposition structure is utilized to infer the missing entries of the tensor. Trace-norm based methods are another popular class of tensor completion methods. These methods first formulate tensor completion as a rank minimization problem, and then employ the tensor trace norm to further transform the task into a convex optimization problem. Finally, various optimization techniques are applied to solve the problem and thus complete the tensor. In this section we provide a brief review of decomposition based and trace norm based tensor completion methods. More details on these two methods and other variants of tensor completion approaches can be found in Song et al. [86].

3.1 Decomposition based methods

CP decomposition (5) and Tucker decomposition (6) are two of the most commonly used decomposition-based methods for tensor completion. In [95], the authors propose to perform CP decomposition on partially observed tensors by iteratively imputing the missing values and estimating the latent vectors in the CP structure. Specifically, in iteration s ($s \geq 1$), the partially observed tensor $\mathcal{X}$ is completed by:

$$\tilde{\mathcal{X}}^{(s)} = \mathcal{X} *_H \mathcal{M} + \mathcal{Y}^{(s)} *_H (\mathbf{1} - \mathcal{M}),$$

where $*_H$ is the tensor Hadamard product defined in (3), $\tilde{\mathcal{X}}^{(s)}, \mathcal{X}, \mathcal{Y}^{(s)}, \mathcal{M} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ are tensors of same size, $\tilde{\mathcal{X}}^{(s)}$ is the completed tensor, $\mathcal{Y}^{(s)}$ is the interim low-rank approximation based on CP decomposition, and $\mathcal{M}$ is the observation index tensor defined as

$$\mathcal{M}_{i_1 \dots i_d} = \begin{cases} 1 & \text{if } \mathcal{X}_{i_1 \dots i_d} \text{ is observed,} \\ 0 & \text{if } \mathcal{X}_{i_1 \dots i_d} \text{ is unobserved.} \end{cases}$$

After the tensor is completed, the decomposition parameters are estimated by alternating least square optimization (ALS). The loop of tensor completion and parameter estimation is repeated until convergence.

Similar approaches were adopted by Kiers et al. [46] and Kroonenberg [51] to impute missing entries. These methods are referred to as EM-like methods, because they can be viewed as a special expectation maximization (EM) method when the residuals independently follow a Gaussian distribution. While the EM-like methods are usually easy to implement, they may not perform well (e.g., slow convergence and converging to a local maximum) when there is a high proportion of missing values.

Also based on the CP decomposition, Bro et al. [13] propose another type of tensor completion method called the Missing-Skipping (MS) method. It conducts the CP decomposition based only on the observed entries in the tensor, and is typically more robust than the EM-like approaches when applied to tensors with a high proportion of missingness. In general, the MS methods seek to optimize the following objective function

$$L = \sum_{(i_1, i_2, \dots, i_d) \in \Omega} \mathcal{D}(\mathcal{X}_{i_1, \dots i_d}, \mathcal{Y}_{i_1, \dots, i_d}), \tag{7}$$

where $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots n_d}$ is the observed tensor, $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the estimated tensor with a CP structure, Ω is a set containing indices of all observed entries in tensor $\mathcal{X}$, and $\mathcal{D}$ is an error measure.

Under the optimization framework (7), Tomasi and Bro [95] define the error measure $\mathcal{D}$ to be the squared difference between the observed and estimated entry $\mathcal{D}(\mathcal{X}_{i_1, \dots i_d}, \mathcal{Y}_{i_1, \dots, i_d}) = (\mathcal{X}_{i_1, \dots i_d} - \mathcal{Y}_{i_1, \dots, i_d})^2$, and employ a modified Gauss-Newton iterative algorithm (i.e., Levenberg-Marquardt method) [53, 66] to solve the optimization problem. Acar et al. [1] utilize a weighted error and minimize the objective function based on the first-order gradient, which is shown to be more scalable to larger problem sizes than the second-order optimization method in [95]. Moreover, the optimization problem can be analyzed in a Bayesian setting by treating the error measure $\mathcal{D}$ to be the negative log-likelihood function. We will discuss more details about these probabilistic methods in Section 5.

Tucker decomposition is another widely utilized tool to conduct tensor completion. While the CP-based completion approaches enjoy nice properties including uniqueness (with the exception of elementary indeterminacies of scaling and permutation) and nice interpretability of latent vectors, methods that employ Tucker structure are able to accommodate more complex interaction among latent vectors and are more effective than CP-based methods. Therefore, in some real-world applications where the completion accuracy is prioritized over the uniqueness and latent vector interpretation, Tucker-based approaches are potentially more suitable than the CP-based methods.

Similar to CP-based methods, EM-like approaches and MS approaches are still two conventional ways for Tucker-based tensor completion algorithms. Walczak and Massart [100] and Andersson and Bro [3] discuss the idea of utilizing EM-like Tucker decomposition to solve tensor completion in their earlier works. This method is further combined with higher-order orthogonal iteration to impute missing data [25]. As an example of MS Tucker decomposition, Karatzoglou et al. [43] employ a stochastic gradient descent algorithm to optimize the loss function based only on the observed entries. There are also researches that develop MS-based methods under a Bayesian framework. See Section 5 for more details.

In recent years, several studies utilize hierarchical tensor (HT) representations to provide a generalization of classical Tucker models. Most of the HT representation based methods are implemented using projected gradient methods. For instance, Rauhut et al. [79, 80] employ a Riemannian gradient iteration method to establish an iterative hard thresholding algorithm in their model. The Riemannian optimization is utilized to construct the manifold for low-rank tensors in [17, 44, 50].

3.2 Trace norm based methods

In [59] and a subsequent paper [60], the authors generalize matrix completion to study tensors and solve the tensor completion problem by considering the following optimization:

$$\begin{aligned} &\min_{\mathcal{Y}} : \|\mathcal{Y}\|_*, \\ &\text{s.t.} : \mathcal{Y}_\Omega = \mathcal{X}_\Omega, \end{aligned} \tag{8}$$

where $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ is the observed tensor, $\mathcal{Y} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ is the estimated tensor, Ω is the set containing indices of all observed entries in tensor $\mathcal{X}$, and $\| \cdot \|_*$ is the tensor trace norm. The tensor trace norm is a relaxation of the tensor n-rank ($\text{rank}_n(\mathcal{X})$, see section 2.2), and is defined as a convex combination of the trace norms of all unfolding matrices. When the noises are included, the optimization problem is now described by

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \|\mathcal{Y}\|_* := \sum_{k=1}^{d} \alpha_k \|\boldsymbol{Y}_{(k)}\|_* \\ \text{subject to} \quad & \mathcal{Y}_\Omega = \mathcal{X}_\Omega + \mathcal{E}_\Omega \end{aligned} \tag{9}$$

where the α_k's are non-negative weights satisfying $\sum_{k=1}^d \alpha_k = 1$, and $\mathcal{E}_\Omega$ is the error. The optimization problem (9) is called a sum of nuclear norm (SNN) model. Note that we do not impose any data generation assumptions in (9). If the noise $\mathcal{E}_\Omega$ is assumed to be Gaussian, then by considering maximizing the likelihood function under the constraint, the SNN model becomes

$$\min_{\mathcal{Y}} \frac{\lambda}{2} \|\mathcal{P}_\Omega(\mathcal{Y} - \mathcal{X})\|^2 + \sum_{k=1}^{d} \alpha_k \|\boldsymbol{Y}_{(k)}\|_*, \tag{10}$$

where $\lambda > 0$ is a tuning parameter, $\mathcal{P}_\Omega(\cdot)$ denotes all the entries in the observed index set Ω, $\| \cdot \|$ is the tensor norm defined in (1), and $\| \cdot \|_*$ is the matrix trace norm [86]. This optimization problem can be solved by block coordinate descent algorithms [59] and splitting methods (e.g., Alternating Direction Method of Multipliers, ADMM) [23, 96, 85].

Using a similar model as (8), Mu et al. [68] propose to apply the trace norm on a balanced unfolding matrix instead of utilizing the summation of trace norms in (9). In the literature, it is also common to consider alternative norms such as the incoherent trace norm [107] and tensor nuclear norm [47, 109]. There are other studies that impose trace norms on the factorized matrices rather than unfolding matrices [61, 106, 65]; these approaches can be viewed as a combination of decomposition based and trace norm based completion methods.

4. TENSOR REGRESSION

In this section, we review tensor regression methods, where the primary goal is to analyze the association between tensor-valued objects and other variables. Based on the role that the tensor plays in the regression, the problem can be further categorized into tensor predictor regression (with tensor-valued predictors and a univariate or multivariate response variable) and tensor response regression (with tensor-valued response and predictors that can be a vector, a tensor or even multiple tensors).

4.1 Tensor predictor regression

Many tensor predictor regression methods are motivated by the need to analyze anatomical magnetic resonance imaging (MRI) data [31, 120]. Usually stored in the form of 3D images (see Figure 7 for an example), MRI presents the shape, volume, intensity, or developmental changes in brain tissues and blood brain barrier. These characteristics are closely related to the clinical outcomes including diagnostic status, and cognition and memory score. It is hence natural to formulate a tensor predictor regression to model the changes of these scalar or vector-valued clinical outcomes with respect to the tensor-valued MRI images.

In medical imaging analysis, conventional approaches are generally based on vectorized data, either by summarizing the image data through a small number of preidentified regions of interest (ROIs), or by transforming the entire image into a long vector. The former is highly dependent on the prior domain knowledge and does not fully utilize the information in the raw image, and the latter suffers from the high-dimensionality of voxels in the 3D image and abandons important spatial information during the vectorization process. In order to circumvent these limitations, a class of regression methods have been developed to preserve the tensor structure. Specifically, given a univariate response Y (e.g. memory test score, disease status) and a tensor-valued predictor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ (e.g. 3D image), Guo et al. [31] propose a linear regression model

$$Y = \langle \mathcal{W}, \mathcal{X} \rangle + b, \tag{11}$$

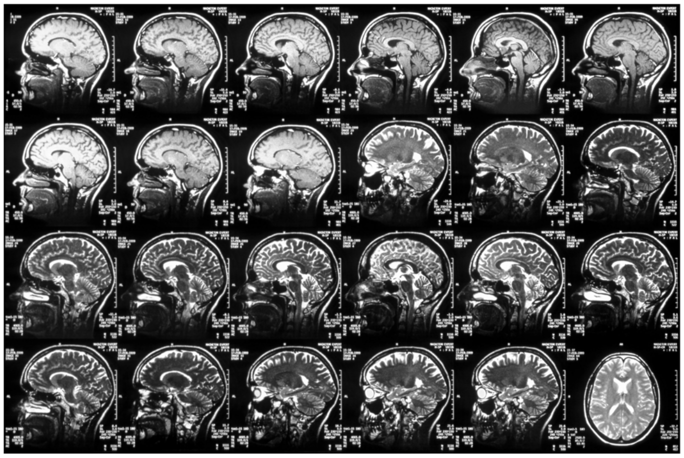


Figure 7. An example of 3D magnetic resonance imaging (MRI). The image is adapted with permissions from Science Photo Library. url: https://www.sciencephoto.com/media/306963/view.

where $\langle \cdot, \cdot \rangle$ is the tensor inner product defined in (2), $\mathcal{W}$ is the coefficient tensor, and b is the error. While model (11) is a direct extension of a classical linear regression model, the extension can result in the explosion of the number of unknown parameters. Specifically, the coefficient tensor $\mathcal{W}$ includes $\prod_{i=1}^{d} n_i$ free parameters, which far exceeds the typical sample size. To address this issue, Guo et al. [31] impose a rank-r CP structure (5) on $\mathcal{W}$, which reduces the number of parameters in $\mathcal{W}$ to $r \sum_{i=1}^{d} n_i$.

Li et al. [56] extend model (11) to the multivariate response $\boldsymbol{Y} = (Y_1, Y_2, ..., Y_q)^\top$ case, where each marginal response Y_k $(1 \leq k \leq q)$ is assumed to be the summation of $\langle \mathcal{X}, \mathcal{B}_k \rangle$ and an error term, where $\mathcal{X}$ is the predictor tensor, and $\mathcal{B}_k \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ is the coefficient tensor. Under the assumption that the coefficients share common features, the coefficient tensors are further formulated into a stack $\mathcal{B} = [\mathcal{B}_1, ..., \mathcal{B}_q] \in \mathbb{R}^{n_1 \times \ldots \times n_d \times q}$, on which a CP structure is imposed for parameter number reduction.

Additionally, Zhou et al. [120] integrate model (11) with the generalized linear regression framework, and incorporate the association between response and other adjusting covariates into the model. Consider a scalar response Y, a tensor-valued predictor $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ and vectorized covariates $\boldsymbol{z} \in \mathbb{R}^{n_0}$ (e.g., demographic features), the generalized linear model is given by

$$g\{\mathbb{E}(Y)\} = b + \boldsymbol{\gamma}^\top \boldsymbol{z} + \langle \mathcal{W}, \mathcal{X} \rangle, \tag{12}$$

where $\boldsymbol{\gamma}$ is the vector coefficient for $\boldsymbol{z}$, $g(\cdot)$ is a link function, and $\mathcal{W}$ is the coefficient tensor where a CP structure is assumed. In model (12), Li et al. [57] impose a Tucker decomposition on $\mathcal{W}$, and demonstrate that the Tucker structure allows for more flexibility.

In order to accommodate longitudinal correlation of the data in imaging analysis, Zhang et al. [110] extend model (12) in the generalized estimating equation setting and establish asymptotic properties of the method. Hao et al. [34] show that the linearity assumption in (11) may be violated in some applications, and propose a nonparametric extension of (11) that accommodates nonlinear interactions between the response and tensor predictor. Zhang et al. [111] use importance sketching to reduce the high computational cost associated with the low-rank factorization in tensor predictor regression, and establish the optimality of their method in terms of reducing mean squared error under the Tucker structure assumption and randomized Gaussian design. Beyond the regression framework, Wimalawarne et al. [102] propose a binary classification method by considering a logistic loss function and various tensor norms for regularization.

4.2 Tensor response regression

While the main focus of tensor predictor regression is analyzing the effects of tensors on the response variables, researchers are also interested in studying how tensor-valued outcomes change with respect to covariates. For example, an important question in MRI studies is to compare the scans of brains between subjects with neurological disorders (e.g., attention deficit disorder) and normal controls, after adjusting for other covariates such as age and sex [56]. This problem can be formulated as a tensor response regression problem where the MRI data, usually taking the form of a three-dimensional image, is the tensor-valued response, and other variables are predictors. Apart from medical imaging analysis, tensor response regression is also useful in the advertisement industry. For example, the click-through rate (CTR) of digital advertisements is often considered to be a significant indicator of the effectiveness of an advertisement campaign. Thus an important business question is to understand how CTR is affected by different features. Since the CTR data can be formulated as a high-dimensional tensor (see Figure 8), we can develop a regression model to address this problem, where the click-through rate on target audience is the tensor-valued response, and the features of advertisements are predictors of interest.

Given a dth-order tensor response $\mathcal{Y} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ and a vector predictor $\boldsymbol{x} \in \mathbb{R}^q$, Rabusseau and Kadri [75] and Sun and Li [90] propose a linear regression model

$$\mathcal{Y} = \mathcal{B} \bar{\times}_{d+1} \boldsymbol{x} + \mathcal{E}, \tag{13}$$

where $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d \times q}$ is an $(d+1)$th-order tensor coefficient, $\mathcal{E}$ is an error tensor independent of $\boldsymbol{x}$, and $\bar{\times}_{d+1}$ is the $(d+1)$-mode vector product. Without loss of generality, the intercept is set to be zero to simplify the presentation.

Both studies [75, 90] propose to estimate the coefficients $\mathcal{B}$ by solving an optimization problem, which consists of a squared tensor norm of the difference between observed and

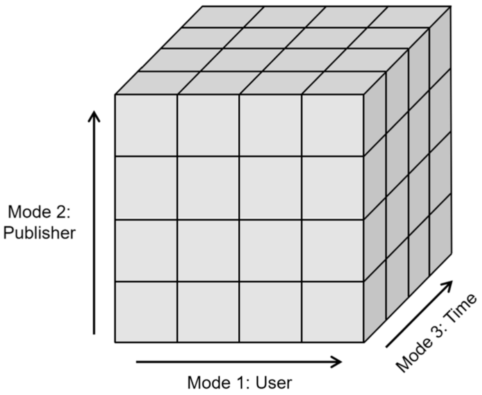


Figure 8. An illustration of click through rate data, which is formulated as a three-mode tensor, where each voxel represents the click-through rate of user i reacting to advertisements from publisher j at time k.

estimated response $\|\mathcal{Y}-\mathcal{B}\bar{\times}_{d+1}\boldsymbol{x}\|^2$ and a sparsity structure. In Rabusseau and Kadri [75], the sparsity is achieved by a L_2-penalty on parameters. In Sun and Li [90], the sparsity structure is realized through a hard-thresholding constraint on the coefficients. For both studies, decomposition structures are imposed on the tensor coefficient $\mathcal{B}$ to facilitate parsimonious estimation of high-dimensional parameters.

Lock [63] further extends (13) to a tensor-on-tensor regression model, allowing a predictor of arbitrary order. Given N independent samples, the responses can be stacked into a tensor $\mathcal{Y} \in \mathbb{R}^{N\times m_1\times m_2\times\ldots\times m_q}$, and the predictors are denoted by $\mathcal{X} \in \mathbb{R}^{N\times n_1\times n_2\times\ldots\times n_d}$. Lock [63] proposes the following model:

$$\mathcal{Y} = \mathcal{X} * \mathcal{B} + \mathcal{E}, \tag{14}$$

where $*$ is the tensor contraction product defined in (4), $\mathcal{B} \in \mathbb{R}^{n_1\times\ldots\times n_d\times m_1\times\ldots\times m_q}$ is the coefficient tensor and $\mathcal{E}$ denotes the error. A CP structure is imposed on $\mathcal{B}$ to achieve parsimony in parameters. The estimation of $\mathcal{B}$ is also transformed into an optimization problem, and a L_2-penalty is included in the loss function to prevent over-fitting. Under a similar modeling framework, Gahrooei et al. [22] develop a multiple tensor-on-tensor regression model, where the predictors are a set of tensors with various orders and sizes.

Based on (14), Li and Zhang [54] propose a tensor response regression that utilizes the envelope method to remove redundant information from the response. Raskutti et al. [78] analyze the tensor regression problem with convex and weakly decomposable regularizers. In their regression model, both the predictors and the responses can be tensors, and the low-rankness assumption is realized by a nuclear norm penalty. Zhou et al. [121] focus on tensor regression where the response is a partially observed dynamic tensor, and impose low-rankness, sparsity and temporal smoothness constraints in the optimization. Chen et al. [15] extend model (14) to the generalized tensor regression setting and utilize a projected gradient descent algorithm to solve the non-convex optimization.

5. BAYESIAN METHODS IN TENSOR COMPLETION

In Section 3.1, we mention that the tensor completion tasks can be realized by performing decomposition on partially observed tensors and using the inferred decomposition structure to impute the missing data (e.g., the Missing-Skipping methods). Bayesian tensor decomposition methods can be naturally applied to study partially observed tensors. Generally, a large proportion of Bayesian decomposition methods are based on CP (5) or Tucker decomposition (6). A class of nonparametric methods have also been proposed to model complex non-linear interactions among latent factors. Recently, more decomposition structures are analyzed under the Bayesian framework (e.g., tensor ring decomposition [64], tensor train decomposition [39] and neural decomposition [36]). A summary of the methods discussed in this section is given in Table 1.

5.1 Bayesian CP-based decomposition

Under the Bayesian framework, Xiong et al. [103] utilize a CP decomposition based method to model time-evolving relational data in recommender systems. In their study, the observed data are formed into a three-dimensional tensor $\mathcal{R} \in \mathbb{R}^{N\times M\times K}$, where each entry $\mathcal{R}_{ij}^k$ denotes user i's rate on item j given time k. A CP structure (5) is then imposed on $\mathcal{R}$:

$$\mathcal{R} \approx \sum_{d=1}^{D} \boldsymbol{U}_{d:} \circ \boldsymbol{V}_{d:} \circ \boldsymbol{T}_{d:} = [\![\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{T}]\!], \tag{15}$$

where $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{T}$ are latent factors corresponding to user, item, and time, respectively; and $\boldsymbol{U}_{d:}, \boldsymbol{V}_{d:}, \boldsymbol{T}_{d:}$ represent the dth-row of $\boldsymbol{U}, \boldsymbol{V}$ and $\boldsymbol{T}$. Xiong et al. [103] assume a Gaussian distribution for the continuous entries R_{ij}^k conditional on $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{T}$ as follows,

$$R_{ij}^k | \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{T} \sim \mathcal{N}(\langle \boldsymbol{U}_{:i}, \boldsymbol{V}_{:j}, \boldsymbol{T}_{:k}\rangle, \alpha^{-1}), \tag{16}$$

where α is the precision, and $\langle \boldsymbol{U}_{:i}, \boldsymbol{V}_{:j}, \boldsymbol{T}_{:k}\rangle$ is the inner product of three D-dimensional vectors defined as

$$\langle \boldsymbol{U}_{:i}, \boldsymbol{V}_{:j}, \boldsymbol{T}_{:k}\rangle = \sum_{d=1}^{D} U_{di} V_{dj} T_{dk}.$$

A complete Bayesian setting requires full specification of the parameter priors. In the study, multivariate Gaussian priors are put on the latent vectors corresponding to users and items

$$\boldsymbol{U}_i \sim \mathcal{N}(\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U^{-1}), \quad i = 1, 2, ..., N, \tag{17}$$

Table 1. Summary of Bayesian tensor decomposition methods.

Name	Decomposition Structure	Rank Specification	Posterior Inference	Data Type
BPTF [103]	CP Decomposition	Pre-specify	Gibbs	Continuous
PLTF [84]		Pre-specify	Gibbs	Binary
BGCP [14]		Pre-specify	Gibbs	Continuous
PTF [82]		Pre-specify	VB	Count
NeuralCP [62]		Pre-specify	AEVB	Continuous
MGP-CP [76]		Automatically inferred	Gibbs	Continuous/Binary
PGCP [77]		Automatically inferred	Gibbs/EM	Binary/Count
BNBCP [41]		Automatically inferred	Gibbs/VB	Count
ZTP-CP [40]		Automatically inferred	Gibbs	Binary
FBCP [112]		Automatically inferred	VB	Continuous
BRTF [115]		Automatically inferred	VB	Continuous
POST [18]		Pre-specify	SVB	Continuous/Binary
BRST [108]		Automatically inferred	SVB	Continuous
SBDT [20]		Pre-specify	ADF&EP	Continuous/Binary
pTucker [16]	Tucker Decomposition	Pre-specify	MAP/EM	Continuous
Hayashi et al. [35]		Pre-specify	EM	All
BPTD [83]		Pre-specify	Gibbs	Count
BTD [113]		Automatically inferred	VB	Continuous
BASS-Tucker [19]		Pre-specify	ADF&EP	Continuous
InfTucker [104]	Nonparametric	Pre-Specify	VEM	Binary/Continuous
Zhe et al. [117]			VEM	
DinTucker [118]			VEM	
Zhe et al. [119]			VI	
SNBTD [73]			ADF&EP	
POND [94]			VB	
Zhe and Du [116]			VEM	
Wang et al. [101]			VI	
BCTT [21]			EP	
TR-VBI [64]	Tensor Ring	Automatically inferred	VB	Continuous
KFT [39]	Tensor Train	N/A	VI	Continuous
He et al. [36]	Neural	N/A	AEVB	All

ADF: Assume-density filtering [11]. AEVB: Auto-Encoding Variational Bayes [48]. EM: Expectation maximization. EP: Expectation propagation [67]. Gibbs: Markov chain Monte Carlo (MCMC) with Gibbs sampling. MAP: Maximum a posteriori. SVB: Steaming variational Bayes. VB: Variational Bayes. VEM: Variational expectation maximization. VI: Variational Inference. N/A: Not applicable. Neural: Neural tensor decomposition.

$$\boldsymbol{V}_j \sim \mathcal{N}(\boldsymbol{\mu}_V, \boldsymbol{\Lambda}_V^{-1}), \quad j = 1, 2, ..., M, \tag{18}$$

and each time feature vector is assumed to depend only on its immediate predecessor due to temporal smoothness:

$$\boldsymbol{T}_k \sim \mathcal{N}(\boldsymbol{T}_{k-1}, \boldsymbol{\Lambda}_T^{-1}), \quad k = 1, 2, ..., K, \tag{19}$$

$$\boldsymbol{T}_0 \sim \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Lambda}_T^{-1}). \tag{20}$$

Moreover, Xiong et al. [103] consider a hierarchical Bayesian structure where the hyper-parameters $\boldsymbol{\Theta}_U \equiv \{\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U\}, \boldsymbol{\Theta}_V \equiv \{\boldsymbol{\mu}_V, \boldsymbol{\Lambda}_V\}$, and $\boldsymbol{\Theta}_T \equiv \{\boldsymbol{\mu}_T, \boldsymbol{\Lambda}_T\}$ are viewed as random variables, and their prior distributions (i.e., hyper-priors), denoted by $p(\cdot)$, are

$$\begin{aligned} &p(\boldsymbol{\Theta}_U) = \\ &\quad p(\boldsymbol{\mu}_U|\boldsymbol{\Lambda}_U)p(\boldsymbol{\Lambda}_U) = \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0\boldsymbol{\Lambda}_U)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_U|\boldsymbol{W}_0, \nu_0), \\ &p(\boldsymbol{\Theta}_V) = \\ &\quad p(\boldsymbol{\mu}_V|\boldsymbol{\Lambda}_V)p(\boldsymbol{\Lambda}_V) = \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0\boldsymbol{\Lambda}_V)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_V|\boldsymbol{W}_0, \nu_0), \\ &p(\boldsymbol{\Theta}_T) = \\ &\quad p(\boldsymbol{\mu}_T|\boldsymbol{\Lambda}_T)p(\boldsymbol{\Lambda}_T) = \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0\boldsymbol{\Lambda}_T)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_T|\boldsymbol{W}_0, \nu_0). \end{aligned} \tag{21}$$

Here $\mathcal{W}(\boldsymbol{\Lambda}|\boldsymbol{W}_0, \nu_0)$ is the Wishart distribution of a $D \times D$ random matrix $\boldsymbol{\Lambda}$ with ν_0 degrees of freedom and a $D \times D$ scale matrix $\boldsymbol{W}_0$:

$$\mathcal{W}(\boldsymbol{\Lambda}|\boldsymbol{W}_0, \nu_0) \propto |\boldsymbol{\Lambda}|^{(\nu_0 - D - 1)/2} \exp\left(-\frac{\mathrm{Tr}(\boldsymbol{W}_0^{-1}\boldsymbol{\Lambda})}{2}\right).$$

Also, a Wishart prior is put on the precision α

$$p(\alpha) = \mathcal{W}(\alpha|\tilde{W}_0, \tilde{\nu}_0). \tag{22}$$

The priors in (21) and (22) are conjugate priors for the Gaussian parameters to help simplify the posterior computation. The parameters $\boldsymbol{\mu}_0, \beta_0, \boldsymbol{W}_0, \nu_0, \tilde{W}_0$ and $\tilde{\nu}_0$ can be chosen by prior knowledge or tuned by model training.

The Bayesian model in (16)–(21) is called a Bayesian Probabilistic Tensor Factorization (BPTF). The posterior distribution of the BPTF model is obtained by Markov Chain Monte Carlo (MCMC) with Gibbs sampling [24]. While Xiong et al. [103] use the BPTF model to perform tensor decomposition on continuous rating data in recommender systems, similar priors have been adapted in other applications and data types. For example, Chen et al. [14] formulate the spatio-temporal traffic data as a third-order tensor (road segment×day×time of day), where a CP structure is assumed and a Gaussian-Wishart prior is put on the latent factors for conjugacy. A similar model has been used to study multi-relational network [84], where the interaction data form a partially symmetric third-order tensor and the tensor entries are binary indicators of whether a certain type of relationship exists. Correspondingly, a sigmoid function is employed in (16) to map the outer product of latent factors onto the range $[0, 1]$.

In addition, Schein et al. [82] develop a Poisson tensor factorization (PTF) method to deal with dyadic interaction data in social networks. Specifically, the interaction data are formulated as a fourth-order tensor $\mathcal{X}$, where $\mathcal{X}_{ijat}$ denotes the number of interactions within a discrete time interval t involving a particular sender i, receiver j, and action-type a. A Poisson distribution is employed to connect the CP structure to the count-valued data:

$$\mathcal{X}_{ijat} \sim \text{Poisson}(\sum_{k=1}^{K} \theta_{ik}^{s} \theta_{jk}^{r} \psi_{ak} \delta_{tk}). \tag{23}$$

Here $\theta_{ik}^{s}, \theta_{jk}^{r}, \psi_{ak}$ and δ_{tk} represent the latent factors corresponding to the sender, receiver, action-type and time interval, respectively. Gamma priors are then assigned to the latent factors,

$$\begin{aligned} \theta_{ik}^{s} &\sim \text{Gamma}(a, b), \\ \theta_{jk}^{r} &\sim \text{Gamma}(a, b), \\ \psi_{ak} &\sim \text{Gamma}(c, d), \\ \delta_{tk} &\sim \text{Gamma}(e, f). \end{aligned} \tag{24}$$

Schein et al. [82] then represent the Poisson likelihood (23) as a sum of K independent Poisson random variables, and derive a Variational Bayesian (VB) algorithm to make inference on the posterior distribution.

All the aforementioned methods assume that the interactions among the latent factors are multi-linear, which may not necessarily hold in practice. To address this issue, Liu et al. [62] consider a neural CP decomposition that exploits both neural networks and probabilistic methods to capture potential nonlinear interactions among the tensor entries. Given a tensor $\mathcal{X}$ and the latent matrices in its CP structure $\boldsymbol{U}^1, ..., \boldsymbol{U}^D$, the distribution of $\mathcal{X}$ conditional on $\boldsymbol{U}^1, ..., \boldsymbol{U}^D$ is given by

$$p(\mathcal{X}|\{\boldsymbol{U}^d\}_{d=1}^{D}) = \prod_{i_1,...,i_D} \mathcal{N}(x_{i_1...i_D}|\mu(\boldsymbol{u}_{i_1...i_D}), \sigma^2(\boldsymbol{u}_{i_1...i_D})),$$

where $\boldsymbol{u}_{i_1...i_D} = (U^1_{i_1:}, ..., U^D_{i_D:}) \in \mathbb{R}^{DR}$ is a long vector generated by concatenating the elements in the i_dth row of the factor matrix U^d. In order to accommodate nonlinear interactions between latent factors, μ and σ^2 are defined as functions of $\boldsymbol{u}_{i_1...i_D}$ ($\mu = \mu(\boldsymbol{u}_{i_1...i_D}), \sigma^2 = \sigma^2(\boldsymbol{u}_{i_1...i_D})$). In particular, the two functions $\mu(\cdot)$ and $\sigma^2(\cdot)$ are modeled by two neural networks with the same input $\boldsymbol{u}_{i_1...i_D}$,

$$\begin{aligned} \mu &= \boldsymbol{w}_{\mu}^{\top} \boldsymbol{h}(\boldsymbol{u}_{i_1...i_D}) + b_{\mu}, \\ \log \sigma^2 &= \boldsymbol{w}_{\sigma}^{\top} \boldsymbol{h}(\boldsymbol{u}_{i_1...i_D}) + b_{\sigma}, \end{aligned}$$

where $\boldsymbol{h}(\boldsymbol{u}_{i_1...i_D})$ is a nonlinear hidden layer shared by these two neural networks, and is defined as a *tanh* activation function in [62]:

$$\boldsymbol{h}(\boldsymbol{u}_{i_1...i_D}) = tanh(\boldsymbol{W}^{\top} \boldsymbol{u}_{i_1...i_D} + \boldsymbol{b}).$$

As discussed in Section 2.2, determining the rank of CP can be challenging in practice. Even for a noise-free tensor, its rank specification is an NP-hard problem [32]. In order to determine the CP rank, a common practice is to fit models with different ranks and choose the best rank based on certain criteria. Nevertheless, this approach may suffer from a low stability issue and a high computational cost. An alternative approach is to use sparsity-inducing priors. For example, in [76] and a subsequent work [77], the authors propose a Bayesian low-rank CP decomposition method, which utilizes the multiplicative gamma process (MGP) prior [6] to automatically infer the rank. Specifically, given a CP structure

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \cdot \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(K)},$$

the following priors are put on the vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_R)$:

$$\lambda_r \sim \mathcal{N}(0, \tau_r^{-1}), \quad 1 \le r \le R \tag{25}$$

$$\tau_r = \prod_{l=1}^{r} \delta_l, \quad \delta_l \sim \text{Gamma}(a_c, 1), \quad a_c > 1. \tag{26}$$

In MGP prior, as r increases, the precision τ_r takes large values hence shrinks λ_r towards zero. Small λ_r values indicate that the term $\lambda_r \cdot \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(K)}$ does not have a significant impact on the CP structure, hence could be removed from the model. Two generalizations of MGP prior are further developed, including truncation based variant MGP-CP^t and the adaptive variant MGP-CP^a, to automatically infer the rank R [76, 77].

Hu et al. [41] develop a Bayesian non-negative tensor factorization that deals with count-valued data and automatically infers the rank of CP decomposition. In their work, the Poisson distribution is utilized to establish a connection between the CP structure and the count-valued data. Given a tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_K}$ and its entries $\boldsymbol{i} = \{i_1, ..., i_K\}$, we have

$$\mathcal{Y}_{\boldsymbol{i}} \sim \text{Poisson}\left(\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}\right).$$

The non-negativity constraints on the factor matrices $\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}$ ($\boldsymbol{U}^{(k)} = [\boldsymbol{u}_1^{(k)}, ..., \boldsymbol{u}_R^{(k)}], k = 1, 2, ..., K$) are naturally satisfied by imposing Dirichlet priors on the factors $\boldsymbol{u}_r^{(k)} = [u_{1r}^{(k)}, ..., u_{i_k r}^{(k)}]^\top$:

$$\boldsymbol{u}_r^{(k)} \sim \text{Dir}(a^{(k)}, ..., a^{(k)}),$$

and a gamma-beta hierarchical prior is put on λ_r to promote the automatic rank specification:

$$\lambda_r \sim \text{Gamma}(g_r, \frac{p_r}{1 - p_r}), \tag{27}$$

$$p_r \sim \text{Beta}(c\epsilon, c(1 - \epsilon)) \quad \text{for some } c > 0. \tag{28}$$

Similar to the MGP prior in (25) and (26), the gamma-beta hierarchical prior in (27) and (28) also shrinks λ_r to zero as r increases, and is thus able to select the CP rank. This model is also extended to binary data by adding an additional layer $b_{\boldsymbol{i}} = \mathbf{1}(y_{\boldsymbol{i}} \geq 1)$, which takes a count-valued entry $y_{\boldsymbol{i}}$ in $\mathcal{Y}$ and thresholds this latent count at one to generate binary-valued entries $b_{\boldsymbol{i}}$ [40].

Instead of imposing sparsity priors on the core elements of CP structure, Zhao et al. [112] place a hierarchical prior over the latent factors. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ have a CP structure

$$\mathcal{X} = [\![\boldsymbol{A}^{(1)}, ..., \boldsymbol{A}^{(N)}]\!],$$

where $\boldsymbol{A}^{(n)} = [\boldsymbol{a}_1^{(n)}, ..., \boldsymbol{a}_{I_n}^{(n)}]$ ($n = 1, 2, ..., N$) are latent factors. Let $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_R]$ and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$. The prior distribution of $\boldsymbol{A}^{(n)}$ is

$$p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda}) = \prod_{i_n=1}^{I_n} \mathcal{N}(\boldsymbol{a}_{i_n}^{(n)}|\boldsymbol{0}, \boldsymbol{\Lambda}^{-1}), \quad n = 1, 2, \ldots, N.$$

A hyperprior is further defined over $\boldsymbol{\lambda}$, which is factorized over the latent dimensions

$$p(\boldsymbol{\lambda}) = \prod_{r=1}^{R} \text{Gamma}(\lambda_r | c_0^r, d_0^r).$$

Here R is a pre-specified maximum possible rank. The latent vectors (the rth row of all latent matrices) will shrink to a zero vector as λ_r^{-1}'s approach to zero. This model can also accommodate various types of outliers and non-Gaussian noise through the introduction of a sparsity structure, and the tradeoff between the low-rankness approximation and the sparse representation can be learned automatically by maximizing the model evidence [115].

In real-world applications including recommender systems, image/video data analysis and internet networks, the data are sometimes produced continuously (i.e., streaming data). Therefore it is of interest to generalize the tensor decomposition models to analyze such data in a real time manner, where the model parameters can be updated efficiently upon receiving new data without retrieving previous entries. To this end, a class of streaming tensor decomposition methods have been developed, and some are analyzed under the Bayesian CP framework [108, 18, 20]. In general, these algorithms start with a prior distribution of unknown parameters and then infer a posterior that best approximates the joint distribution of these parameters upon the arrival of new streaming data. The estimated posterior is then used as the prior for the next update. These methods are implemented either by streaming variational Bayes (SVB) [108, 18], or assume-density filtering (ADF) and expectation-propagation (EP) [20].

5.2 Tucker-based Bayesian decomposition methods

Compared to the CP decomposition, the Tucker structure (6) can model more complex interactions between latent factors. One of the early works that employs a probabilistic Tucker structure is proposed by Chu and Ghahramani [16], where a probabilistic framework called pTucker is developed to perform a decomposition on partially observed tensors. Given a continuous third-order tensor $\mathcal{Y} \in \mathbb{R}^{n \times m \times d}$, a Gaussian distribution is assigned to each entry of the tensor $\mathcal{Y}$,

$$\mathcal{Y}_{ijr} | \mathcal{T} \sim \mathcal{N}(\mathcal{F}_{ijr}, \sigma^2).$$

Here $\mathcal{F}$ has a Tucker structure with a core tensor $\mathcal{T}$

$$\mathcal{F}_{ijr} = \text{vec}(\mathcal{T})^\top (\boldsymbol{v}_r \otimes \boldsymbol{z}_j \otimes \boldsymbol{x}_i),$$

where $\otimes$ is the Kronecker product, and $\boldsymbol{v}_r, \boldsymbol{z}_j$ and $\boldsymbol{x}_i$ are latent vectors. Next, independent standard normal distributions are specified over the entries in $\mathcal{T}$ as priors:

$$\mathcal{T}_{kls} \sim \mathcal{N}(0, 1), \quad \forall k, l, s.$$

By integrating out the core tensor $\mathcal{T}$ from the joint distribution $\prod_{i,j,r} p(\mathcal{Y}_{ijr}|\mathcal{T}) \prod_{k,l,s} p(\mathcal{T}_{kls})$, the observational array still follows a Gaussian distribution:

$$\text{vec}(\mathcal{Y}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{U}\boldsymbol{U}^\top + \sigma^2 \boldsymbol{I}),$$

where vec($\mathcal{Y}$) is the vectorized tensor, σ^2 is the noise level, and $\boldsymbol{U} = \boldsymbol{V} \otimes \boldsymbol{Z} \otimes \boldsymbol{X}$, where $\boldsymbol{V}, \boldsymbol{Z}$ and $\boldsymbol{X}$ are latent matrices. To complete the Bayesian framework, standard normal distributions are further used as priors for latent components $\boldsymbol{X}, \boldsymbol{Z}$ and $\boldsymbol{V}$. Finally, the latent factors are estimated by maximum a posteriori (MAP) method with gradient descent.

While the MAP method provides an efficient alternative to perform point estimation for latent factors, it also has significant disadvantages including vulnerability to overfitting and incapability of quantifying parameter uncertainties. To this end, various approaches seek to provide a fully Bayesian treatment through inferring the posterior distribution of parameters. For instance, Hayashi et al. [35] utilize the expectation maximization (EM) method that combines the Laplace approximation and the Gaussian process to perform posterior inference on latent factors. They use the exponential family distributions to connect the Tucker structure with the observed tensor, thus developing a decomposition method that is compatible with various data types. In addition, Schein et al. [83] propose a Bayesian Poisson Tucker decomposition (BPTD) that uses MCMC with Gibbs sampling for posterior inference. That method mainly focus on modeling count-valued tensors by putting Poisson priors on the Tucker structure entries and Gamma priors on the latent factors. More recently, Fang et al. [19] develop a Bayesian streaming sparse Tucker decomposition (BASS-Tucker) method to deal with streaming data. BASS-Tucker assigns a spike-and-slab prior over entries of core tensor and employs an extended assumed density filtering (ADF) framework for posterior inference.

Similar to CP-based methods, an important task for Tucker decomposition based methods is to choose an appropriate tensor rank. Unfortunately, this problem is challenging especially when dealing with partially observed data corrupted with noise. Zhao et al. [113] employ hierarchical sparsity-inducing priors to perform automatic rank determination in their Bayesian tensor decomposition (BTD) model. Specifically, the observed tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is assumed to follow a Gaussian distribution with the mean following a Tucker structure:

$$\mathrm{vec}(\mathcal{Y})|\{\boldsymbol{U}^{(n)}\}, \mathcal{G}, \tau \sim \mathcal{N}((\bigotimes_n \boldsymbol{U}^{(n)})\mathrm{vec}(\mathcal{G}), \tau^{-1}\boldsymbol{I}),$$

where $\{\boldsymbol{U}^{(n)}\}$ are latent matrices, $\mathcal{G}$ is the core tensor, and τ is the precision. To allow a fully Bayesian treatment, hierarchical priors are placed over all model parameters. First, a noninformative Gamma prior is assigned to the precision parameter τ

$$\tau \sim \mathrm{Gamma}(a_0^\tau, b_0^\tau).$$

Next, a group sparsity prior is employed over the factor matrices, i.e., each $\boldsymbol{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, ..., \boldsymbol{u}_{I_n}^{(n)}]^\top$ ($\boldsymbol{u}_{i_n}^{(n)}$ are latent vectors) is governed by hyper-parameters $\boldsymbol{\lambda}^{(n)} = (\lambda_1^{(n)}, ..., \lambda_{R_n}^{(n)})$, where $\lambda_{r_n}^{(n)}$ controls the precision related to group r_n (i.e., r_nth column of $\boldsymbol{U}^{(n)}$). Let $\boldsymbol{\Lambda}^{(n)}$ =diag($\boldsymbol{\lambda}^{(n)}$), then the group sparsity prior is given by

$$\boldsymbol{u}_{i_n}^{(n)}|\boldsymbol{\lambda}^{(n)} \sim \mathcal{N}(\mathbf{0}, {\boldsymbol{\Lambda}^{(n)}}^{-1}), \quad \forall n, \forall i_n.$$

The sparsity assumption is also imposed on the core tensor $\mathcal{G}$. Considering the connection between latent factors and the corresponding entries of the core tensor, the precision parameter for $\mathcal{G}_{r_1,\ldots,r_N}$ can be viewed as the product of precisions over $\{\boldsymbol{u}_{\cdot r_n}^{(n)}\}_{n=1}^N$, which is represented by

$$\mathcal{G}_{r_1\ldots r_N}|\{\boldsymbol{\lambda}^{(n)}\}, \beta \sim \mathcal{N}(0, (\beta \prod_n \lambda_{r_n}^{(n)})^{-1}),$$

or equivalently,

$$\mathrm{vec}(\mathcal{G})|\{\boldsymbol{\lambda}^{(n)}\}, \beta \sim \mathcal{N}(\mathbf{0}, (\beta \bigotimes_n \boldsymbol{\Lambda}^{(n)})^{-1}),$$

where β is a scaling parameter on which a Gamma prior is placed

$$\beta \sim \mathrm{Gamma}(a_0^\beta, b_0^\beta).$$

The hyperprior for $\boldsymbol{\lambda}^{(n)}$ plays a key role for different sparsity-inducing priors. Two options (student-t and Laplace) are commonly used to achieve group sparsity:

$$\text{Student-}t: \lambda_{r_n}^{(n)} \sim \mathrm{Gamma}(a_0^\lambda, b_0^\lambda), \quad \forall n, \forall r_n;$$

$$\text{Laplace}: \lambda_{r_n}^{(n)} \sim \mathrm{IG}(1, \frac{\gamma}{2}), \quad \forall n, \forall r_n,$$
$$\gamma \sim \mathrm{Gamma}(a_0^\gamma, b_0^\gamma).$$

5.3 Nonparametric Bayesian decomposition methods

In addition to the aforementioned linear models, a class of nonparametric Bayesian approaches have been developed to capture the potential nonlinear relationship between tensor entries. One of the pioneering works is InfTucker proposed by Xu et al. [104]. Generally, InfTucker maps the latent factors onto an infinite feature space and then performs Tucker decomposition with the core tensor of an infinite size. Let $\mathcal{M} \in \mathbb{R}^{m_1 \times \ldots \times m_K}$ be a tensor following a Tucker structure with a core tensor $\mathcal{W}$ and latent factors $\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}$. One can assign an element-wise standard Gaussian prior over the core tensor $\mathcal{W}$ ($\mathrm{vec}(\mathcal{W}) \sim \mathcal{N}(\mathrm{vec}(\mathcal{W}); \mathbf{0}, \boldsymbol{I})$) and marginalize out $\mathcal{W}$. The marginal distribution of tensor $\mathcal{M}$ is then given by

$$p(\mathcal{M}|\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}) = \mathcal{N}(\mathrm{vec}(\mathcal{M}); \mathbf{0}, \boldsymbol{\Sigma}^{(1)} \otimes ... \otimes \boldsymbol{\Sigma}^{(K)})), \tag{29}$$

where $\boldsymbol{\Sigma}^{(K)} = \boldsymbol{U}^{(K)}\boldsymbol{U}^{(K)^\top}$. Since the goal is to capture the nonlinear relationships, each row $\boldsymbol{u}_t^k$ of the latent factors $\boldsymbol{U}^{(k)}$ is replaced by a nonlinear map $\phi(\boldsymbol{u}_t^k)$. Then a nonlinear covariance matrix $\boldsymbol{\Sigma}^{(k)} = k(\boldsymbol{U}^{(k)}, \boldsymbol{U}^{(k)})$ can be obtained, where $k(\cdot,\cdot)$ is a nonlinear covariance kernel function. In InfTucker [104], $k(\cdot,\cdot)$ is chosen as the radial basis function kernel. After feature mapping, the core tensor $\mathcal{W}$ has the size of the mapped feature vector $\boldsymbol{u}_t^k$ on mode k, which can be potentially infinity. Because the covariance of vec$(\mathcal{M})$ is a function of the latent factors $\mathcal{U} = \{\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}\}$, equation (29) actually defines a Gaussian process (GP) on tensor entries, where the input is based on the corresponding latent factors $\mathcal{U}$. To encourage sparse estimation, element-wise Laplace priors are assigned on $\mathcal{U}$:

$$\boldsymbol{u}_i^{(k)} \sim \mathcal{L}(\lambda) \propto \exp(-\lambda \|\boldsymbol{u}_i^{(k)}\|_1). \tag{30}$$

Finally, the observed tensor $\mathcal{Y}$ is sampled from a noisy model $p(\mathcal{Y}|\mathcal{M})$, of which the form depends on the data type of $\mathcal{Y}$. The joint distribution is then given by

$$p(\mathcal{Y}, \mathcal{M}, \mathcal{U}) = p(\mathcal{U})p(\mathcal{M}|\mathcal{U})p(\mathcal{Y}|\mathcal{M}),$$

where $p(\mathcal{U})$ is given by (30), and $p(\mathcal{M}|\mathcal{U})$ is given by (29) with $\boldsymbol{\Sigma}^{(k)} = k(\boldsymbol{U}^{(k)}, \boldsymbol{U}^{(k)})$.

Under a similar modeling framework, Zhe et al. [117] make two modifications to InfTucker. One is to assign a Dirichlet process mixture (DPM) prior [4] over the latent factors that allows a random number of latent clusters. The other is to utilize a local GP assumption instead of a global GP when generating the observed array given the latent factors, which enables fast computation over subarrays. Specifically, the local GP-based construction is realized by first breaking the whole array $\mathcal{Y}$ into smaller subarrays $\{\mathcal{Y}_1, .., \mathcal{Y}_N\}$. Then for each subarray $\mathcal{Y}_n$, a latent real-valued subarray $\mathcal{M}_n$ is generated by a local GP based on the corresponding subset of latent factors $\mathcal{U}_n = \{\boldsymbol{U}_n^{(1)}, ..., \boldsymbol{U}_n^{(K)}\}$, and the noisy observation $\mathcal{Y}_n$ is sampled according to $\mathcal{M}_n$,

$$\begin{aligned} p(\mathcal{Y}_n, \mathcal{M}_n|\mathcal{U}) &= p(\mathcal{M}_n|\mathcal{U}_n)p(\mathcal{Y}_n|\mathcal{M}_n) \\ &= \mathcal{N}(\text{vec}(\mathcal{M}_n); \boldsymbol{0}, \boldsymbol{\Sigma}_n^{(1)} \otimes ... \otimes \boldsymbol{\Sigma}_n^{(K)})p(\mathcal{Y}_n|\mathcal{M}_n), \end{aligned}$$

where $\boldsymbol{\Sigma}_n^{(k)} = k(\boldsymbol{U}_n^{(k)}, \boldsymbol{U}_n^{(k)})$ is the kth mode covariance matrix over the sub-factors $\mathcal{U}_n$.

Likewise, DinTucker [118] consider a local GP assumption and sample each of the subarrays $\{\mathcal{Y}_1, ..., \mathcal{Y}_n\}$ from a GP based on the latent factors $\tilde{\mathcal{U}}_n = \{\tilde{\boldsymbol{U}}_n^{(1)}, ..., \tilde{\boldsymbol{U}}_n^{(K)}\}$. Different from Zhe et al. [117], in DinTucker these latent factors are then tied to a set of common latent factors $\mathcal{U} = \{\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}\}$ via a prior distribution

$$p(\tilde{\mathcal{U}}_n|\mathcal{U}) = \prod_{k=1}^{K} \mathcal{N}(\text{vec}(\tilde{\boldsymbol{U}}_n^{(k)})|\text{vec}(\boldsymbol{U}^{(k)}), \lambda \boldsymbol{I}),$$

where λ is the variance parameter that controls the similarity between $\mathcal{U}$ and $\tilde{\mathcal{U}}_n$. Furthermore, DinTucker divides each subarray $\mathcal{Y}_n$ into T_n smaller subarrays $\mathcal{Y}_n = \{\mathcal{Y}_{n1}, ..., \mathcal{Y}_{nT_n}\}$ that share the same latent factors $\{\tilde{\mathcal{U}}_n\}$, and their joint probability is given by

$$\begin{aligned} p(\mathcal{U}, \{\tilde{\mathcal{U}}_n, \mathcal{M}_n, & \mathcal{Y}_n\}_{n=1}^N) \\ &= \prod_{n=1}^{N} p(\tilde{\mathcal{U}}_n|\mathcal{U}) \prod_{t=1}^{T_n} p(\mathcal{M}_{nt}|\tilde{\mathcal{U}}_n)p(\mathcal{Y}_{nt}|\mathcal{M}_{nt}), \end{aligned}$$

where $\mathcal{M}_{nt}$ is a latent subarray, and $\mathcal{M}_n = \{\mathcal{M}_{nt}\}_{t=1}^{T_n}$. The local terms require less memory and have a faster processing time than the global term. More importantly, the additive nature of these local terms in the log domain enables distributed inference, which is then realized through the MapReduce system.

While Zhe et al. [117] and DinTucker [118] improve the scalability of their GP-based approaches through modeling the subtensors, their methods can still run into challenges when the sparsity level is very high in observed tensors. To address this issue, a class of methods that do not rely on the Kronecker-product structure in the variance (29) are proposed based on the idea of selecting an arbitrary subset of tensor entries for training. Assume that the decomposition is performed on a sparsely observed tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times ... \times d_K}$. For each tensor entry $\boldsymbol{i} = (i_1, ..., i_K)$, Zhe et al. [119] first construct an input $\boldsymbol{x_i}$ by concatenating the corresponding latent factors from all the modes: $\boldsymbol{x_i} = [\boldsymbol{u}_{i_1}^{(1)}, ..., \boldsymbol{u}_{i_K}^{(K)}]$, where $\boldsymbol{u}_{i_k}^{(k)}$ is the i_kth row in the latent factor matrix $\boldsymbol{U}^{(k)}$ for mode k. Then each $\boldsymbol{x_i}$ is transformed to a scalar $m_{\boldsymbol{i}}$ through an underlying function $f : \mathbb{R}^{\sum_{j=1}^K d_j} \to \mathbb{R}$ such that $m_{\boldsymbol{i}} = f(\boldsymbol{x_i}) = f([\boldsymbol{u}_{i_1}^{(1)}, ..., \boldsymbol{u}_{i_K}^{(K)}])$. After that, a GP prior is assigned over f to learn the unknown function: for any set of tensor entries $S = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_N\}$, the function values $\boldsymbol{f}_S = \{f(\boldsymbol{x}_{\boldsymbol{i}_1}), ..., f(\boldsymbol{x}_{\boldsymbol{i}_N})\}$ are distributed according to a multivariate Gaussian distribution with mean $\boldsymbol{0}$ and the covariance determined by $\boldsymbol{X}_S = \{\boldsymbol{x}_{\boldsymbol{i}_1}, ..., \boldsymbol{x}_{\boldsymbol{i}_N}\}$:

$$p(\boldsymbol{f}_S|\mathcal{U}) = \mathcal{N}(\boldsymbol{f}_S|\boldsymbol{0}, k(\boldsymbol{X}_S, \boldsymbol{X}_S)), \tag{31}$$

where $\mathcal{U}$ is the latent factor, and $k(\cdot,\cdot)$ is a nonlinear covariance kernel. Note that this method is equivalent to InfTucker [104] if all entries are selected and a Kronecker-product structure is applied in the full covariance. A standard normal prior is assigned over the latent factors, and the observed entries $\boldsymbol{y} = [y_{\boldsymbol{i}_1}, ..., y_{\boldsymbol{i}_N}]$ are sampled from a model $p(\boldsymbol{y}|\boldsymbol{m})$, where $p(\cdot)$ is selected based on the data type.

Following the sparse GP framework (31), Pan et al. [73] propose the Streaming Nonlinear Bayesian Tensor Decomposition (SNBTD) that performs fast posterior updates upon receiving new tensor entries. Their model is augmented with feature weights to incorporate a linear structure, and the assumed-density-filtering (ADF) framework is extended to perform reliable streaming inference. Also based on (31), Tillinghast et al. [94] utilize convolutional neural networks to construct a deep kernel $k(\cdot,\cdot)$ for GP modeling, which is more powerful in estimating arbitrarily complicated relationships in data compared to the methods based on shallow kernel functions (e.g., RBF kernel).

In some applications, the tensor data are observed with additional temporal information. Various approaches have been proposed to preserve the accurate timestamps and take full advantage of the temporal information. Among these methods, Zhe and Du [116] and Wang et al. [101] perform decomposition based on event-tensors to capture complete temporal information, and Fang et al. [21] model the core tensor as a time-varying function, where GP prior is placed to estimate different types of temporal dynamics.

6. BAYESIAN METHODS IN TENSOR REGRESSION

Similar to the frequentist tensor regression methods discussed in Section 4, Bayesian tensor regression methods can be categorized into Bayesian tensor predictor regression and Bayesian tensor response regression. We discuss these two classes of methods in Section 6.1 and 6.2, and their theoretical properties in Section 6.3. We also review posterior computing in Section 6.4. A summary of the methods discussed in this section is given in Table 2.

6.1 Bayesian tensor predictor regression

In recent years, Bayesian tensor predictor regression models have gained an increasing attention. For example, Suzuki [91] develop a Bayesian framework based on the basic tensor linear regression model

$$Y_i = \langle \mathcal{W}, \mathcal{X}_i \rangle + \epsilon_i, \tag{32}$$

where $Y_i \in \mathbb{R}$ is a univariate response, $\mathcal{X}_i \in \mathbb{R}^{M_1 \times \cdots \times M_K}$ is a tensor-valued predictor, $\mathcal{W} \in \mathbb{R}^{M_1 \times \cdots \times M_K}$ is the coefficient tensor, and $\langle \cdot, \cdot \rangle$ is the tensor inner product (2). The error terms ϵ_i's are assumed i.i.d. following a normal distribution $\mathcal{N}(0, \sigma^2)$. To achieve parsimony in free parameters, a rank-r CP structure (5) is imposed on the coefficient tensor $\mathcal{W}$:

$$\mathcal{W} = [\![\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)}]\!],$$

where $\boldsymbol{U}^{(k)} \in \mathbb{R}^{r \times M_K}$ $(k = 1, 2, ..., K)$ are latent factors. To complete model specification, a Gaussian prior is placed on the latent matrices:

$$\pi(\boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(K)} | r) \propto \exp\Big\{ - \frac{r}{2\sigma_p^2} \sum_{k=1}^{K} \mathrm{Tr}[\boldsymbol{U}^{(k)\top} \boldsymbol{U}^{(k)}] \Big\},$$

and an independent prior is used for the rank r:

$$\pi(r) = \frac{1}{N_\xi} \xi^{r(M_1 + \cdots + M_K)},$$

where $0 < \xi < 1$ is a positive real number, and N_ξ is the normalizing constant.

In order to adjust for other covariates in the model and accommodate various data types of the response variable, Guhaniyogi et al. [29] propose a Bayesian method based on the generalized tensor predictor regression model (12). Given a scalar response y, vectorized predictors $\boldsymbol{z} \in \mathbb{R}^p$ and a tensor predictor $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$, the regression model is given by

$$y \sim f(\alpha + \boldsymbol{z}^\top \boldsymbol{\gamma} + \langle \mathcal{X}, \mathcal{B} \rangle, \sigma), \tag{33}$$

where $f(\mu, \sigma)$ is a family of distributions with location μ and scale σ, $\boldsymbol{\gamma} \in \mathbb{R}^p$ are coefficients for predictors $\boldsymbol{z}$, $\mathcal{B} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$ is the coefficient tensor, and $\langle \cdot, \cdot \rangle$ is the tensor inner product (2). A CP structure is imposed on the tensor

Table 2. Summary of Bayesian tensor regression methods.

Name	Predictor Type	Response Type	Tensor Structure	Algorithm
Suzuki [91]	Tensor	Scalar	CP	Gibbs
BTR [29]	Tensor+Vector	Scalar	CP	Gibbs
Zhao et al. [114]	Tensor	Scalar	Nonparametric	MAP
OLGP [38]	Tensor	Scalar	Nonparametric	OLGP
AMNR [42]	Tensor	Scalar	Nonparametric	MC
Yang and Dunson [105]	Vector (Categorical)	Scalar (Categorical)	Tucker	Gibbs
CATCH [72]	Tensor+Vector	Scalar (Categorical)	Tucker	MLE
BTRR [30]	Vector	Tensor	CP	Gibbs
Spencer et al. [87, 88]	Vector	Tensor	CP	Gibbs
SGTM [26]	Vector	Symmetric Tensor	CP	Gibbs
BSTN [52]	Vector	Tensor	Other	Gibbs
SGPRN [58]	Matrix	Tensor	Nonparametric	VI
MLTR [37]	Tensor	Tensor	Tucker	Gibbs
ART [10]	Tensor	Tensor	CP	Gibbs

Gibbs: MCMC with Gibbs sampling. MAP: Maximum a posteriori. MC: Monte Carlo Method. MLE: Maximum likelihood estimator. OLGP: Online local Gaussian process [71, 99]. VI: Variational Inference.

coefficient $\mathcal{B}$:

$$\mathcal{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)}.$$

Under the Bayesian framework, Guhaniyogi et al. [29] propose a multiway Dirichlet generalized double Pareto (M-DGDP) prior over the latent factors $\boldsymbol{\beta}_j^{(r)}$. This prior promotes the joint shrinkage on the global and local component parameters, as well as accommodates dimension reduction by favoring low-rank decompositions. Specifically, the M-DGDP prior first assigns a multivariate Gaussian prior on $\boldsymbol{\beta}_j^{(r)}$:

$$\text{(34)} \qquad \boldsymbol{\beta}_j^{(r)} \sim \mathcal{N}(\mathbf{0}, (\phi_r \tau)\boldsymbol{W}_{jr}), \; j = 1, \ldots, D.$$

The shrinkage across components is induced in an exchangeable way, with a global scale parameter $\tau \sim \text{Gamma}(a_\tau, b_\tau)$ adjusted in each component by ϕ_r for $r = 1, 2, ..., R$, where $\Phi = (\phi_1, ..., \phi_R) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_R)$ encourages shrinkage towards lower ranks in the CP structure. In addition, $\boldsymbol{W}_{jr} = \text{diag}(w_{jr,1}, \cdots, w_{jr,p_j})$, $j = 1, 2, ..., D$ and $r = 1, 2, ..., R$, are scale parameters for each component, where a hierarchical prior is used,

$$\text{(35)} \qquad w_{jr,k} \sim \text{Exp}(\lambda_{jr}^2/2), \quad \lambda_{jr} \sim \text{Gamma}(a_\lambda, b_\lambda).$$

In the M-DGDP prior, flexibility in estimating $\mathcal{B}_r = \{\boldsymbol{\beta}_j^{(r)}; 1 \le j \le D\}$ is achieved by modeling individual-level heterogeneity via element-specific scaling parameters $w_{jr,k}$'s. The common rate parameter λ_{jr} shares information between individual elements, hence leads to shrinkage at the local scale.

Besides linear models, a class of Gaussian process (GP) based nonparametric approaches have been proposed to model nonlinear relationships in the tensor-valued predictors. Given a dataset of N paired observations $\mathcal{D} = \{(\mathcal{X}_n, y_n) | n = 1, 2, ..., N\}$, Zhao et al. [114] aggregate all N tensor inputs $\mathcal{X}_n$ $(n = 1, 2, ..., N)$ into a design tensor $\mathcal{X} \in \mathbb{R}^{N \times I_1 \times \cdots \times I_M}$, and collect the responses in the vector form $\boldsymbol{y} = [y_1, ..., y_N]^\top$. The distribution of the response vector can be factored over the observations as

$$\text{(36)} \qquad \boldsymbol{y} \sim \prod_{n=1}^{N} \mathcal{N}(y_n | f(\mathcal{X}_n), \sigma^2).$$

Here $f(\cdot)$ is a latent function on which a GP prior is placed

$$\text{(37)} \qquad f(\mathcal{X}) \sim \text{GP}(m(\mathcal{X}), k(\mathcal{X}, \mathcal{X}') | \boldsymbol{\theta}),$$

where $k(\mathcal{X}, \mathcal{X}')$ is the covariance function (kernel), $\boldsymbol{\theta}$ is the associated hyperparameter vector, and $m(\mathcal{X})$ is the mean function which is set to be zero in [114]. The authors further propose to use the following product kernel in (37):

$$\text{(38)} \qquad k(\mathcal{X}, \mathcal{X}') = \alpha^2 \prod_{d=1}^{D} \exp\Big(\frac{D(p(\boldsymbol{x}|\Omega_d^{\mathcal{X}}) \parallel q(\boldsymbol{x}'|\Omega_d^{\mathcal{X}'}))}{-2\beta_d^2}\Big),$$

where α is a magnitude hyperparameter, β_d denotes the d-mode length-scale hyper-parameter, and D is the symmetric Kullback-Leibler (KL) divergence defined as

$$D(P||Q) = \text{KL}(P||Q) + \text{KL}(Q||P).$$

The distributions p and q in the symmetric KL divergence are characterized by the hyper-parameters Ω_d, which can be estimated from the d-mode unfolding matrix $\boldsymbol{X}_d$ of tensor $\mathcal{X}$ by treating each $\boldsymbol{X}_d$ as a generative model with I_d variables and $I_1 \times \cdots \times I_{d-1} \times I_{d+1} \times \cdots \times I_D$ observations. Given the prior construction, the hyperparameters $\boldsymbol{\theta} = \{\alpha, \beta_d | d = 1, 2, ..., D\}$ and σ are then estimated by maximum a posteriori (MAP). While the computational complexity of GP-based methods is usually excessive, Hou et al. [38] take advantage of the online local Gaussian Process (OLGP) and present a computationally-efficient approach for the nonparametric model in (36)-(38).

To further mitigate the burden of high-dimensionality, Imaizumi and Hayashi [42] propose an additive-multiplicative nonparametric regression (AMNR) method that concurrently decomposes the functional space and the input space. This method is referred to as a doubly decomposing nonparametric tensor regression method.

Denote a Sobolev space by $\mathcal{W}^\beta(\mathcal{X})$, which is a space of β-times differentiable functions with the support $\mathcal{X}$. Let $\mathcal{X} = \bigotimes_k \boldsymbol{x}_k := \boldsymbol{x}_1 \otimes \cdots \otimes \boldsymbol{x}_K$ be a rank-one tensor denoted by the outer product of vectors $\boldsymbol{x}_k \in \mathcal{X}^{(k)}$ ($\otimes$ is the outer product). Let $f \in \mathcal{W}^\beta(\bigotimes_k \mathcal{X}^{(k)})$ be a function on a rank-one tensor. For any f we can construct $\tilde{f}(\boldsymbol{x}_1, ..., \boldsymbol{x}_K) \in \mathcal{W}^\beta(\mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(k)})$ such that $\tilde{f}(\boldsymbol{x}_1, ..., \boldsymbol{x}_K) = f(\mathcal{X})$ using function decomposition as $\tilde{f} = f \circ h$ with $h : (\boldsymbol{x}_1, ..., \boldsymbol{x}_K) \to \bigotimes_k \boldsymbol{x}_k$. Then f can be decomposed into a set of local functions $\{f_m^k \in \mathcal{W}^\beta(\mathcal{X}^{(k)})\}_m$ following [33]:

$$\text{(39)} \qquad f(\mathcal{X}) = \tilde{f}(\boldsymbol{x}_1, ..., \boldsymbol{x}_K) = \sum_{m=1}^{M} \prod_{k=1}^{K} f_m^{(k)}(\boldsymbol{x}_k),$$

where M represents the complexity of f (i.e., the "rank" of the model).

Based on (39), for a rank-R tensor $\mathcal{X}$, Imaizumi and Hayashi [42] define the AMNR function as:

$$\text{(40)} \qquad f^{AMNR}(\mathcal{X}) := \sum_{m=1}^{M} \sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} f_m^{(k)}(\boldsymbol{x}_r^{(k)}),$$

which is obtained by first writing a rank-R tensor as the sum of R rank-one tensors, and then decomposing the function into a set of local functions for each rank-one tensor. Under the Bayesian framework, a GP prior is assigned to the local functions $f_m^{(k)}$, and the Gaussian distribution (36) is utilized to associate the scalar response Y_i with the function $f^{AMNR}(\mathcal{X}_i)$.

While the previous studies mainly deal with regression problems with continuous response variables, the probabilistic methods can also apply to categorical-response regression problems with tensor-valued predictors, i.e., the tensor classification problems. For example, Pan et al. [72] propose a covariate-adjusted tensor classification model (CATCH), which jointly models the relationship among the covariates, tensor predictors, and categorical responses. Given a categorical response $Y \in \{1, 2, ..., K\}$, a vector of covariates $\boldsymbol{U} \in \mathbb{R}^q$, and tensor-variate predictors $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$, the CATCH model is proposed as

$$\boldsymbol{U}|(Y = k) \sim \mathcal{N}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}) \tag{41}$$

$$\mathcal{X}|(\boldsymbol{U} = \boldsymbol{u}, Y = k) \sim \mathrm{TN}(\boldsymbol{\mu}_k + \boldsymbol{\alpha}\bar{\times}_{(M+1)}\boldsymbol{u}; \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_M), \tag{42}$$

where $\boldsymbol{\Phi}_k \in \mathbb{R}^q, \boldsymbol{\Psi} \in \mathbb{R}^{q \times q}$ is positive definite, $\boldsymbol{\alpha} \in \mathbb{R}^{p_1 \times \ldots \times p_M \times q}, \boldsymbol{\mu}_k \in \mathbb{R}^{p_1 \times \ldots \times p_M}$, and $\boldsymbol{\Sigma}_m \in \mathbb{R}^{p_m \times p_m}$ is positive definite for $m = 1, ..., M$. Here $\mathrm{TN}(\cdot)$ is the tensor normal distribution, and $\bar{\times}_{(M+1)}$ is the $(M+1)$-mode tensor vector product.

In equation (41), it is assumed that $\{Y, \boldsymbol{U}\}$ follow a classical LDA model, where $\boldsymbol{\Phi}_k$ is the mean of $\boldsymbol{U}$ within class k and $\boldsymbol{\Psi}$ is the common within class covariance of $\boldsymbol{U}$. Similarly, in equation (42) a common within class covariance structure of $\mathcal{X}$ is assumed (denoted by $\boldsymbol{\Sigma}_m, m = 1, 2, ..., M$), which does not depend on Y after adjusting for the covariates $\boldsymbol{U}$. The tensor coefficient $\boldsymbol{\alpha}$ characterizes the linear dependence of tensor predictor $\mathcal{X}$ on the covariates $\boldsymbol{U}$, and $\boldsymbol{\mu}_k$ is the covariate-adjusted within-class mean of $\mathcal{X}$ in class k.

While the goal is to predict Y given $\{\boldsymbol{U}, \mathcal{X}\}$, based on the Bayes' rule the optimal classifier under the CATCH model is derived by maximizing the posterior probability

$$\begin{aligned} \hat{Y} &= \arg \max_{k=1,2,...,K} P(Y = k|\mathcal{X} = \boldsymbol{x}, \boldsymbol{U} = \boldsymbol{u}) \\ &= \arg \max_{k=1,2,...,K} \pi_k f_k(\boldsymbol{x}, \boldsymbol{u}), \end{aligned} \tag{43}$$

where $\pi_k = P(Y = k)$ and $f_k(\boldsymbol{x}, \boldsymbol{u})$ is the joint density function of $\mathcal{X}$ and $\boldsymbol{U}$ conditional on $Y = k$. Combining (41) and (42), equation (43) is transformed into

$$\hat{Y} = \arg \max_{k=1,2,...,K} \{a_k + \boldsymbol{\gamma}_k^\top \boldsymbol{U} + \langle \mathcal{B}_k, \mathcal{X} - \boldsymbol{\alpha}\bar{\times}_{(M+1)}\boldsymbol{U} \rangle\},$$

where $\boldsymbol{\gamma}_k = \boldsymbol{\Psi}^{-1}(\boldsymbol{\Phi}_k - \boldsymbol{\Phi}_1), \mathcal{B}_k = [\![\boldsymbol{\mu}_k - \boldsymbol{\mu}_1; \boldsymbol{\Sigma}_1^{-1}, ..., \boldsymbol{\Sigma}_M^{-1}]\!]$ following a Tucker structure with the core tensor $\boldsymbol{\mu}_k - \boldsymbol{\mu}_1$ and latent matrices $\boldsymbol{\Sigma}_1^{-1}, ..., \boldsymbol{\Sigma}_M^{-1}$, and $a_k = \log(\pi_k/\pi_1) - \frac{1}{2}\boldsymbol{\gamma}_k^\top(\boldsymbol{\Phi}_k + \boldsymbol{\Phi}_1) - \langle \mathcal{B}_k, \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_1)\rangle$ is a scalar that does not depend on $\mathcal{X}$ or $\boldsymbol{U}$.

Given i.i.d. samples $\{Y^i, \boldsymbol{U}^i, \mathcal{X}^i\}_{i=1}^n$, the parameters $\{\pi_k, \boldsymbol{\Phi}_k, \boldsymbol{\gamma}_k, \boldsymbol{\mu}_k, \mathcal{B}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_m\}_{m=1}^M$ can be estimated to build an accurate classifier based on the data. Regularization is used when estimating $\mathcal{B}_k$ in order to facilitate sparsity.

Though not modeling tensor predictors, Yang and Dunson [105] employ tensor methods to deal with classification problems with categorical predictors. Specifically, [105] develop a framework for nonparametric Bayesian classification through performing decomposition on the tensor constructed from the conditional probability

$$P(Y = y|X_1 = x_1, ..., X_p = x_p),$$

with a categorical response $Y \in \{1, 2, ..., d_0\}$ and a vector of p categorical predictors $\boldsymbol{X} = (X_1, X_2, ..., X_p)^\top$. The conditional probability can be structured as a $d_0 \times d_1 \times \cdots \times d_p$-dimensional tensor, where d_j $(j = 1, 2, ..., p)$ denotes the number of levels of the jth categorical predictor X_j. This tensor is called a conditional probability tensor, and the set of all conditional probability tensors is denoted by $\mathcal{P}_{d_1,...,d_p}(d_0)$. Therefore, $\mathcal{P} \in \mathcal{P}_{d_1,...,d_p}(d_0)$ implies

$$\begin{aligned} &\mathcal{P}_{y,x_1,...,x_p} \geq 0 \quad \text{for every } y, x_1, ..., x_p; \\ &\sum_{y=1}^{d_0} \mathcal{P}_{y,x_1,...,x_p} = 1 \quad \text{for every } x_1, ..., x_p. \end{aligned}$$

Since all the conditional probabilities are entries in the conditional probability tensor, the classification problem is converted into a tensor decomposition problem. Additionally, Yang and Dunson [105] prove that every conditional probability tensor $\mathcal{P} \in \mathcal{P}_{d_1,...,d_p}(d_0)$ can be expressed by a Tucker structure

$$\begin{aligned} \mathcal{P}_{y,x_1,...,x_p} &= P(y|x_1, ..., x_p) \\ &= \sum_{h_1=1}^{k_1} \cdots \sum_{h_p=1}^{k_p} \lambda_{h_1 h_2 \ldots h_p}(y) \prod_{j=1}^{p} \pi_{h_j}^{(j)}(x_j), \end{aligned}$$

with all positive parameters satisfying

$$\begin{aligned} &\sum_{c=1}^{d_0} \lambda_{h_1 h_2 \ldots h_p}(c) = 1, \quad \text{for every } h_1, h_2, ..., h_p, \\ &\sum_{h=1}^{k_j} \pi_h^{(j)}(x_j) = 1, \quad \text{for every pair of } j, x_j. \end{aligned}$$

The inference of the Tucker coefficients is carried out under the Bayesian framework. Specifically, independent Dirichlet priors are assigned to the parameters $\boldsymbol{\Lambda} = \{\lambda_{h_1,...,h_p}(c), c = 1, 2, ..., d_0\}$ and $\boldsymbol{\pi} = \{\pi_{h_j}^{(j)}(x_j), h_j = 1, 2, ..., k_j\}$ $(x_j = 1, 2, ..., d_j, h_j = 1, 2, ..., k_j, j = 1, 2, ..., p)$:

$$\begin{aligned} &\left\{\lambda_{h_1,...,h_p}(1), ..., \lambda_{h_1,...,h_p}(d_0)\right\} \sim \mathrm{Dirichlet}(\frac{1}{d_0}, ..., \frac{1}{d_0}), \\ &\left\{\pi_1^{(j)}(x_j), ..., \pi_{k_j}^{(j)}(x_j)\right\} \sim \mathrm{Dirichlet}(\frac{1}{k_j}, ..., \frac{1}{k_j}), \; j = 1, ..., p. \end{aligned}$$

These priors impose the non-negativity and sum-to-one constraints naturally and lead to conditional conjugacy in posterior computation. Additionally, [105] assign priors on the hyper-parameters in the Dirichlet priors to promote a fully Bayesian treatment. These priors place most of the probability on few elements to induce sparsity in their vectors.

6.2 Bayesian tensor response regression

Guhaniyogi and Spencer [30] propose a Bayesian regression model with a tensor response and scalar predictors. Let $\mathcal{Y}_t \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$ be a tensor-valued response, and $\boldsymbol{x}_t = (x_{1,t}, ..., x_{m,t}) \in \mathcal{X} \subset \mathbb{R}^m$ be an m-dimensional vector predictor measured at time t. Assuming that both the response $\mathcal{Y}_t$ and the predictors $\boldsymbol{x}_t$ are centered around their respective means, the proposed regression model for $\mathcal{Y}_t$ on $\boldsymbol{x}_t$ is given by

$$(44) \qquad \mathcal{Y}_t = \boldsymbol{\Gamma}_1 x_{1,t} + \cdots + \boldsymbol{\Gamma}_m x_{m,t} + \mathcal{E}_t, \quad i = 1, 2, ..., n,$$

where $\boldsymbol{\Gamma}_k \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}, k = 1, 2, ..., m$ is the tensor coefficient corresponding to the predictor $x_{k,t}$, and $\mathcal{E}_t \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$ represents the error tensor. To account for the temporal correlation in the response tensor, the error tensor $\mathcal{E}_t$ is assumed to follow a component-wise AR(1) structure across t: $\text{vec}(\mathcal{E}_t) = \kappa \text{vec}(\mathcal{E}_{t-1}) + \text{vec}(\boldsymbol{\eta}_t)$, where $\kappa \in (-1, 1)$ is the correlation coefficient, and $\boldsymbol{\eta}_t \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$ is a random tensor, with each entry following a Gaussian distribution $\mathcal{N}(0, \sigma^2/(1-\kappa^2))$.

Next, a CP structure is imposed on each $\boldsymbol{\Gamma}_k$ to reduce the dimensionality of coefficient tensors, i.e., $\boldsymbol{\Gamma}_k = \sum_{r=1}^R \boldsymbol{\gamma}_{1,k}^{(r)} \circ \cdots \circ \boldsymbol{\gamma}_{D,k}^{(r)}$. Although Guhaniyogi et al's previously proposed M-DGDP prior (34)(35) over the latent factors $\boldsymbol{\gamma}_{j,k}^{(r)}$ can promote global and local sparsity, Guhaniyogi and Spencer [30] claim that a direct application of M-DGDP prior leads to inaccurate estimation due to a less desirable tail behavior of the coefficient distributions. Instead, a multiway stick breaking shrinkage prior (M-SB) is assigned to $\boldsymbol{\gamma}_{j,k}^{(r)}$, where the main difference compared to the M-DGDP prior is how shrinkage is achieved across ranks. The construction of the M-SB prior is given as follows. Let $\boldsymbol{W}_{jr,k} = \text{diag}(w_{jr,k,1}, ..., w_{jr,k,p_d})$. Then we set

$$\boldsymbol{\gamma}_{j,k}^{(r)} \sim \mathcal{N}(0, \tau_{r,k} \boldsymbol{W}_{jr,k}).$$

Further set $\tau_{r,k} = \phi_{r,k} \tau_k$ to be scaling specific to rank r ($r = 1, ..., R$). Then effective shrinkage across ranks is achieved by adopting a stick breaking construction for the rank-specific parameter $\phi_{r,k}$:

$$\phi_{r,k} = \xi_{r,k} \prod_{l=1}^{r-1} (1 - \xi_{l,k}), \quad r = 1, ..., R-1,$$

$$\phi_{R,k} = \prod_{l=1}^{R-1} (1 - \xi_{l,k}),$$

where $\xi_{r,k} \sim_{iid} \text{Beta}(1, \alpha_k)$. The Bayesian setting is then completed by specifying

$$\tau_k \sim \text{InvGamma}(a_\tau, b_\tau), \quad w_{jr,k,i} \sim \text{Exp}(\lambda_{jr,k}^2/2),$$
$$\lambda_{jr,k} \sim \text{Gamma}(a_\lambda, b_\lambda),$$

where the hierarchical prior of $w_{jr,k,i}$ allows the local scale parameters $\boldsymbol{W}_{jr,k}$ to achieve individual-level shrinkage.

Based on the regression function (44), Spencer et al. [87, 88] consider a brain imaging application and develop an additive mixed effect model that simultaneously measures the activation due to stimulus at voxels in the gth brain region and connectivity among G brain regions. Let $\mathcal{Y}_{i,g,t} \in \mathbb{R}^{p_{1,g} \times \cdots \times p_{D,g}}$ be the tensor of observed fMRI data in brain region g for the ith subject at the tth time point, and $x_{1,i,t}, ..., x_{m,i,t} \in \mathbb{R}$ be the activation-related predictors. The regression function is given by

$$\mathcal{Y}_{i,g,t} = \boldsymbol{\Gamma}_{1,g} x_{1,i,t} + \cdots \boldsymbol{\Gamma}_{m,g} x_{m,i,t} + d_{i,g} + \mathcal{E}_{i,g,t}$$

for subject $i = 1, 2, ..., n$ in region $g = 1, 2, ..., G$ and time $t = 1, 2, ..., T$. Here $\mathcal{E}_{i,g,t} \in \mathbb{R}^{p_{1,g} \times \cdots \times p_{D,g}}$ is the error tensor, of which the elements are assumed to follow a normal distribution with zero mean and shared variance σ_y^2. $\boldsymbol{\Gamma}_{k,g} \in \mathbb{R}^{p_{1,g} \times \cdots \times p_{D,g}}$ represents activation due to the kth stimulus at gth brain region. Each $\boldsymbol{\Gamma}_{k,g}$ is assumed to follow a CP structure, and an M-SB prior is assigned to the latent factors of the CP decomposition to determine the nature of activation. Also, $d_{i,g} \in \mathbb{R}$ are region- and subject-specific random effects that are jointly modeled to borrow information across regions of interest. Specifically, a Gaussian graphical LASSO prior is imposed on these random effects:

$$\boldsymbol{d}_i = (d_{i,1}, ..., d_{i,G})^\top \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}^{-1}), \quad i = 1, 2, ..., n,$$

$$p(\boldsymbol{\omega}|\zeta) = C^{-1} \prod_{g<g_1} [DE(\omega_{gg_1}|\zeta)] \prod_{g=1}^{G} [\text{Exp}(\omega_{gg}|\frac{\zeta}{2})] \mathbf{1}_{\boldsymbol{\Omega} \in \mathcal{P}^+},$$

where $\mathcal{P}^+$ is the class of all positive definite matrices and C is a normalization constant. The covariance $\boldsymbol{\omega} = (\omega_{gg_1} : g \leq g_1)$ is a vector of upper triangle and diagonal entries of the precision matrix $\boldsymbol{\Omega}$. By properties of the multivariate Gaussian distribution, a small value of ω_{gg_1} stands for weak connectivity between regions of interest (ROIs) g and g_1, given other ROIs. In practice, a double exponential prior is employed on the off-diagonal entries of the precision matrix $\boldsymbol{\Omega}$ to favor shrinkage among these entries. A full Bayesian prior construction is completed by assigning a Gamma prior on ζ and an inverse Gamma prior on the variance parameter σ_y^2.

To study brain connectome datasets acquired using diffusion weighted magnetic resonance imaging (DWI), Guha and Guhaniyogi [26] propose a generalized Bayesian linear model with a symmetric tensor response and scalar predictors. Let $\mathcal{Y}_i \in \mathcal{Y} \subset \mathbb{R}^{p \times \cdots \times p}$ be a symmetric tensor response with diagonal entries being zero, $\boldsymbol{x}_i = (x_{i1}, ..., x_{im})^\top$ be m predictors of interest, and $\boldsymbol{z}_i = (z_{i1}, ..., z_{il})^\top$ be l auxiliary predictors corresponding to the ith individual. Let $\mathcal{J} = \{\boldsymbol{j} = (j_1, ..., j_D) : 1 \leq j_1 < \cdots < j_D \leq p\}$ be a set of indices. Given that $\mathcal{Y}_i$ is symmetric with dummy diagonal entries, it suffices to build a probabilistic generative mechanism for $y_{i,\boldsymbol{j}}$ ($\boldsymbol{j} \in \mathcal{J}$). In practice, a set of conditionally independent generalized linear models are utilized. Let $E(y_{i,\boldsymbol{j}}) = \omega_{i,\boldsymbol{j}}$, for $\boldsymbol{j} \in \mathcal{J}$, we have

$$\omega_{i,\boldsymbol{j}} = \\ H^{-1}(\beta_0 + B_{1,\boldsymbol{j}} x_{i1} + \cdots + B_{m,\boldsymbol{j}} x_{im} + \beta_1 z_{i1} + \cdots + \beta_l z_{il}),$$

where $B_{1,\boldsymbol{j}}, ..., B_{m,\boldsymbol{j}}$ respectively represents the entry $\boldsymbol{j} = (j_1, ..., j_D)$ of the $p \times \cdots \times p$ symmetric coefficient tensors $\mathcal{B}_1, ..., \mathcal{B}_m$ with diagonal entries zero, $\beta_0, \beta_1, ..., \beta_l \in \mathbb{R}$ are the intercept and coefficients corresponding to variables $z_{i1}, ..., z_{il}$, respectively, and $H(\cdot)$ is the link function. The model formulation implies a similar effect of any of the auxiliary variables $(z_{i1}, ..., z_{il})$ on all entries of the response tensor but varying effects of the hth predictor on different entries $\boldsymbol{j} \in \mathcal{J}$ of the response tensor. To account for associations between tensor nodes and predictors and to achieve parsimony in tensor coefficients, a CP-like structure is imposed on symmetric coefficient tensors $\mathcal{B}_1, ..., \mathcal{B}_m$, i.e.,
(45)

$$B_{h,\boldsymbol{j}} = \sum_{r=1}^{R} \lambda_{h,r} u_{h,j_1}^{(r)} \cdots u_{h,j_D}^{(r)}, \quad h = 1, 2, ..., m; \ \boldsymbol{j} \in \mathcal{J},$$

where $\boldsymbol{u}_h^{(r)} = (u_{h,1}^{(r)}, ..., u_{h,p}^{(r)})^\top \in \mathbb{R}^p$ are latent factors and $\lambda_{h,r} \in \{0, 1\}$ is a binary inclusion variable determining if the rth summand in (45) is relevant in model setting. Further let $\tilde{\boldsymbol{u}}_{h,k} = (u_{h,k}^{(1)}, ..., u_{h,k}^{(R)})$, then the hth predictor of interest is considered to have no impact on the kth tensor if $\tilde{\boldsymbol{u}}_{h,k} = 0$. In order to directly study the effect of tensor nodes related to the hth predictor of interest, a spike-and-slab mixture distribution prior is assigned on $\tilde{\boldsymbol{u}}_{h,k}$:

$$\tilde{\boldsymbol{u}}_{h,k} \sim \begin{cases} \mathcal{N}(\boldsymbol{0}, \boldsymbol{M}_h), & \text{if } \eta_{h,k} = 1 \\ \delta_{\boldsymbol{0}}, & \text{if } \eta_{h,k} = 0 \end{cases}, \quad \eta_{h,k} \sim \text{Bern}(\xi_h),$$
$$\boldsymbol{M}_h \sim IW(\boldsymbol{S}, \nu), \quad \xi_h \sim U(0, 1),$$

where $\delta_{\boldsymbol{0}}$ is the Dirac function at $\boldsymbol{0}$ and $\boldsymbol{M}_h$ is a covariance matrix of order $R \times R$. Here $IW(\boldsymbol{S}, \nu)$ denotes an Inverse-Wishart distribution with an $R \times R$ positive definite scale matrix $\boldsymbol{S}$ and ν degrees of freedom. The parameter ξ_h corresponds to the probability of the nonzero mixture component and $\eta_{h,k}$ is a binary indicator that equals 0 if $\tilde{\boldsymbol{u}}_{h,k} = \delta_{\boldsymbol{0}}$. Thus, the posterior distributions of $\eta_{h,k}$'s can help identify nodes related to a chosen predictor.

To impart increasing shrinkage on $\lambda_{h,r}$ as r grows, a hierarchical prior is imposed on $\lambda_{h,r}$:

$$\lambda_{h,r} \sim \text{Bern}(\nu_{h,r}), \ \nu_{h,r} \sim \text{Beta}(1, r^\zeta), \zeta > 1.$$

In addition, a Gaussian prior $\mathcal{N}(a_\beta, b_\beta)$ is placed on $\beta_0, \beta_1, ..., \beta_l$.

Recently, Lee et al. [52] develop a Bayesian skewed tensor normal (BSTN) regression, which addresses the problem of considerable skewness in the tensor response in a study of periodontal disease (PD). For an order-K tensor response $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ with a vector of covariates $\boldsymbol{x}_i \in \mathbb{R}^p$, the regression model is given by

$$\mathcal{Y}_i = \mathcal{B} \bar{\times}_{(K+1)} \boldsymbol{x}_i + \mathcal{E}_i, \quad \text{for } i = 1, 2, ..., n,$$

where $\mathcal{B} \in \mathbb{R}^{d_1 \times \cdots \times d_K \times p}$ is an order-$(K+1)$ coefficient tensor, $\bar{\times}_{(K+1)}$ is the $(K+1)$th mode vector product, and $\mathcal{E}_i \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ is the error tensor. The skewness in the distribution of $\mathcal{Y}$ is modeled by

$$\mathcal{E}_i = |\mathcal{Z}_{2i}| \times_K \boldsymbol{\Lambda} + \mathcal{Z}_{1i},$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_{d_K}) \in \mathbb{R}^{d_K \times d_K}$ is a digonal matrix with skewness parameters $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{d_K})$, $|\boldsymbol{M}|$ denotes a matrix whose elements are absolute values of the corresponding elements in matrix $\boldsymbol{M}$, and $\times_K$ is the mode-K tensor matrix product. The tensor $\mathcal{Z}_{2i} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ follows a tensor normal distribution $\mathcal{Z}_{2i} \sim \text{TN}(\boldsymbol{0}; \boldsymbol{I}_{d_1}, ..., \boldsymbol{I}_{d_{K-1}}, \boldsymbol{D}_{\boldsymbol{\sigma}}^2)$, and is assumed to be independent of $\mathcal{Z}_{1i} \sim \text{TN}(\boldsymbol{0}; \boldsymbol{R}_1, ..., \boldsymbol{R}_{K-1}, \boldsymbol{D}_{\boldsymbol{\sigma}} \boldsymbol{R}_K \boldsymbol{D}_{\boldsymbol{\sigma}})$, where $\boldsymbol{R}_1, ..., \boldsymbol{R}_K$ are positive-definite correlation matrices, and $\boldsymbol{D}_{\boldsymbol{\sigma}} = \text{diag}(\sigma_1, ..., \sigma_{d_K})$ is a diagonal matrix of positive scale parameters $\sigma_1, ..., \sigma_{d_K}$. The parameterization for the tensor normal $\mathcal{Z}_{1i}$ via correlation matrices $\boldsymbol{R}_1, ..., \boldsymbol{R}_K$ avoids the common identifiability issue. Only the Kth mode of $\mathcal{Z}_{2i}$ is multiplied by a skewness matrix $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_{d_K})$ because the skewness level is assumed to be the same in all combinations of the first $(K-1)$ modes in the PD dataset. When λ_j is positive (or negative), the corresponding marginal density of $y_{i_1, ..., i_{K-1}, j}$ of tensor response $\mathcal{Y}$ is skewed to the right (left).

Various prior distributions can be put on the parameters. For example, an independent zero-mean normal density with a pre-specified variance is utilized as the common prior for $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_{d_K})$, and common independent inverse-gamma distributions $IG(g_1, g_2)$ with pre-specified shape $g_1 > 0$ and scale $g_2 > 0$ are imposed on $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_{d_K})$. The parametric correlation matrices $\boldsymbol{R}_1, ..., \boldsymbol{R}_K$ are assumed to be equicorrelation matrices with independent uniform priors $Unif(-1, 1)$ for unknown off-diagonal elements. A tensor normal distribution $\text{TN}(\boldsymbol{0}; \boldsymbol{C}_1, ..., \boldsymbol{C}_{K+1})$ with zero mean and known covariance matrices $\boldsymbol{C}_1, ..., \boldsymbol{C}_{K+1}$ is put on the tensor coefficient $\mathcal{B}$. Lee et al. [52] also propose an alternative prior distribution for $\mathcal{B}$, where a spike-and-slab prior is employed to introduce sparsity.

Similar to the tensor predictor regression, Gaussian Process (GP) based nonparametric models are also studied for regression problems with tensor responses. Li et al. [58] propose a method based on the Gaussian process regression networks (GPRN), where no special kernel structure is pre-assumed. Tensor/matrix-normal variational posteriors are introduced to improve the inference performance.

The aforementioned methods assume a low-dimensional structure of the predictors (either in the form of a vector or a matrix), and are generally incapable of modeling high-dimensional tensor predictors. Under such circumstances, various tensor-on-tensor methods are proposed to deal with regression problems with both tensor-valued responses and predictors, and some are analyzed under the Bayesian framework. Given a tensor response $\mathcal{Y}_i \in \mathbb{R}^{p_1\times\ldots\times p_K}$ and tensor predictors $\mathcal{X}_i \in \mathbb{R}^{m_1\times\ldots\times m_K}$, Hoff [37] associate $\mathcal{Y}_i$ and $\mathcal{X}_i$ through a Tucker structure (6)

$$\mathcal{Y}_i = \mathcal{X}_i \times_1 \boldsymbol{B}_1 \times_2 \boldsymbol{B}_2 \times_3 \cdots \times_K \boldsymbol{B}_K + \mathcal{E}_i, \tag{46}$$

where $\boldsymbol{B}_1, ..., \boldsymbol{B}_K$ are matrices of dimension $p_1 \times m_1, ..., p_K \times m_K$ respectively. The error tensors $\mathcal{E}_i$ are i.i.d with dimension $p_1 \times \cdots \times p_D$, and are assumed to follow a tensor normal distribution

$$\mathcal{E}_i \sim \mathrm{TN}(\mathbf{0}; \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K).$$

Under the Bayesian framework, matrix normal priors are assigned to $\boldsymbol{B}_k|\boldsymbol{\Sigma}_k$, and inverse Wishart priors are imposed on $\boldsymbol{\Sigma}_k$ $(k = 1, 2, ..., K)$ to deliver efficient posterior computation.

Hoff [37] require that the responses and predictors have the same number of modes. Lock [63] circumvent this restriction by employing a regression structure based on the tensor contraction product in (14). Utilizing the same structure, Billio et al. [10] develop a Bayesian dynamic regression model that allows tensor-valued predictors and responses to be of arbitrary dimension. Specifically, denote the tensor response by $\mathcal{Y}_t \in \mathbb{R}^{p_1\times\ldots\times p_{D_1}}$ and the tensor predictor measured at time t by $\mathcal{X}_t \in \mathbb{R}^{q_1\times\ldots\times q_{D_2}}$. Billio et al. [10] propose the following dynamic regression model:

$$\mathcal{Y}_t = \sum_{j=1}^{q} \mathcal{B}_j * \mathcal{Y}_{t-j} + \mathcal{A} * \mathcal{X}_t + \mathcal{E}_t,$$

where $\mathcal{B}_j$ and $\mathcal{A}$ are coefficient tensors of dimension $p_1 \times \cdots \times p_{D_1} \times p_1 \times \cdots \times p_{D_1}$ and $p_1 \times \cdots \times p_{D_1} \times q_1 \times \cdots \times q_{D_2}$, respectively, and $*$ is the tensor contraction product (4). The random error tensor $\mathcal{E}_t$ follows a tensor normal distribution, $\mathcal{E}_t \sim \mathrm{TN}(\mathbf{0}; \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_{D_1})$. The parsimony of coefficients is achieved by CP structures on the tensor coefficients, and an M-DGDP prior is assigned to the latent factors to promote shrinkage across tensor coefficients and improve computational scalability in high-dimensional settings.

6.3 Theoretical properties of Bayesian tensor regression

In this section, we discuss the theoretical properties for several Bayesian tensor regression methods.

In [91], the in-sample predictive accuracy of an estimator coefficient tensor $\hat{\mathcal{W}}$ in (32) is defined by

$$\|\hat{\mathcal{W}} - \mathcal{W}^*\|_n^2 := \frac{1}{n}\sum_{i=1}^{n} \langle X_i, \hat{\mathcal{W}} - \mathcal{W}^*\rangle^2,$$

where $\mathcal{W}^*$ is the true coefficient tensor, $\{X_i\}_{i=1}^n$ are the observed input samples. Here $\|\cdot\|_n$ is not the usual l_2-norm. The out-of-sample predictive accuracy is defined by

$$\|\hat{\mathcal{W}} - \mathcal{W}^*\|_{L_2(P(X))}^2 := E_{X\sim P(X)}[\langle X, \hat{\mathcal{W}} - \mathcal{W}^*\rangle^2],$$

where $P(X)$ is the distribution of X that generates the observed samples $\{X_i\}_{i=1}^n$ and the expectation is taken with respect to $P(X)$.

Assume that the l_1-norm of X_i is bounded by 1, the convergence rate of the expected in-sample predictive accuracy of the posterior mean estimator $\int \mathcal{W} d\Pi(\mathcal{W}|Y_{1:n})$,

$$E\left[\left\| \int \mathcal{W} d\Pi(\mathcal{W}|Y_{1:n}) - \mathcal{W}^*\right\|_n^2\right],$$

is characterized by the actual degree of freedom up to a log term. Specifically, let d^* be the CP-rank of the true tensor $\mathcal{W}^*$, and $M_1, ..., M_K$ be the dimensions for each order of $\mathcal{W}^*$, the rate is essentially

$$O\left(\frac{\text{degree of freedom}}{n}\right) = O\left(\frac{d^*(M_1 + \cdots + M_K)}{n}\right)$$

up to a log term and is optimal. Although the true rank d^* is unknown, by placing a prior distribution on the rank, the Bayes estimator can appropriately estimate the rank and give an almost optimal rate depending on the true rank. In this sense, the Bayes estimator is adaptive to the true rank. Additionally, frequentist methods often assume a variant of strong convexity (e.g., a restricted eigenvalue condition [9] and the restricted strong convexity [70]) to derive a fast convergence rate of sparse estimators such as Lasso and the trace-norm regularization estimator. In contrast, the convergence rate in [91] does not require the strong-convexity assumption in the model.

In terms of the out-of-sample predictive accuracy, the convergence rate achieved is also optimal up to a log term under the infinity norm thresholding assumption ($\|\mathcal{W}^*\|_\infty < R$, where $R > 0$). Specifically, the rate is

$$O\left(\frac{d^*(M_1 + \cdots + M_K)}{n}(R^2 \vee 1)\right)$$

up to a log factor.

Based on equation (33), Guhaniyogi et al. [29] prove the posterior consistency of the estimated coefficient tensor $\mathcal{B}$. Define a Kulback-Leibler (KL) neighborhood around the true tensor $\mathcal{B}_n^0$ as

$$\mathbb{B}_n = \left\{\mathcal{B}_n : \frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}\left(f(y_i|\mathcal{B}_n^0), f(y_i|\mathcal{B}_n)\right) < \epsilon\right\},$$

where $f(\cdot)$ is the glm density in (33). Let Π_n be the posterior probability given n observations, Guhnaiyogi et al. [29] establish the posterior consistency by showing that

$$\Pi_n(\mathbb{B}_n^c) \to 0 \quad a.s. \text{ as } n \to \infty$$

under the probability measure induced by the $\mathcal{B}_n^0$ when the prior $\pi_n(\mathcal{B}_n)$ satisfies a concentration condition. Based on this result, Guhaniyogi et al. further establish the posterior consistency for the M-DGDP prior in their study.

In a subsequent work [27], the authors relax the key assumption in [29] which requires that both the true and fitted tensor coefficients have the same rank in CP decomposition. Instead, the theoretical properties are obtained based on a more realistic assumption that the rank of the fitted tensor coefficient is merely greater than the rank of the true tensor coefficients. Under additional assumptions, the authors prove that the in-sample predictive accuracy is upper bounded by a quantity given below:

$$E_{\mathcal{B}_n^0} \int \|\mathcal{B}_n - \mathcal{B}_n^0\|_n^2 \Pi(\mathcal{B}_n | y_{1:n}, X_{1:n}) \leq AH_n/n,$$

where $H_n = o\{\log(n)^d\}$ and A are positive constants depending on the other parameters. By applying Jensen's inequality

$$\begin{aligned} &E_{\mathcal{B}_n^0}[\|E(\mathcal{B}_n|Y_{1:n}, \mathcal{X}_{1:n}) - \mathcal{B}_n^0\|_n^2] \\ &\quad \leq E_{\mathcal{B}_n^0} \int \|\mathcal{B}_n - \mathcal{B}_n^0\|_n^2 \Pi(\mathcal{B}_n|Y_{1:n}, X_{1:n}), \end{aligned}$$

the posterior mean of the tensor coefficient, $E(\mathcal{B}_n|Y_{1:n}, X_{1:n})$, converges to the truth with a rate of order $n^{-1/2}$ up to a $\log(n)$ factor, which is near-optimal. Similar to Suzuki [91], this result on convergence rate does not require a strong convexity assumption on the model.

For the AMNR function defined in equation (40), Imaizumi and Hayashi [42] establish an asymptotic property of the distance between the true function and its estimator. Let $f^* \in \mathcal{W}^\beta(\mathcal{X})$ ($\mathcal{W}^\beta(\mathcal{X})$ is the Sobolev space) be the true function and $\hat{f}_n$ be their estimator for f^*. Let M^* be the rank of the true function. Then the behavior of the distance $\|f^* - \hat{f}_n\|$ strongly depends on M^*. Let $\|f\|_n$ be the empirical norm satisfying

$$\|f\|_n^2 := \frac{1}{n}\sum_{i=1}^n f(x_i)^2.$$

When M^* is finite, under certain assumptions and for some finite constant $C > 0$, by [42], it follows that

$$E\|\hat{f}_n - f^*\|_n^2 \leq Cn^{-2\beta/(2\beta+\max_k I_k)},$$

where $\max_k I_k$ is the maximum dimension of the tensor predictor $\mathcal{X}$. This property indicates that the convergence rate of the estimator achieves the minimax optimal rate of estimating a function in $\mathcal{W}^\beta$ on a compact support in $\mathbb{R}^{I_k}$. The convergence rate of AMNR depends only on the largest dimension of $\mathcal{X}$.

When M^* is infinite, by truncating M^* at a finite value M, the convergence rate is nearly the same as the case of finite M^*, which is slightly worsened by a factor $\gamma/(1+\gamma)$ [42]:

$$E\|\hat{f}_n - f^*\|_n^2 \leq C(n^{-2\beta/(2\beta+\max_k I_k)})^{\gamma/(1+\gamma)}.$$

For the CATCH model in (41)-(43), Pan et al. [72] establish the asymptotic properties for a simplified model, where only the tensor predictor $\mathcal{X}$ is collected (the covariates $\boldsymbol{U}$ are not included). They define the classification error rate of the CATCH estimator and that of the Bayes rule as

$$\begin{aligned} R_n &= \Pr(\hat{Y}(\mathcal{X}^{\text{new}}|\hat{\mathcal{B}}_k, \hat{\pi}_k, \hat{\boldsymbol{\mu}}_k) \neq Y^{\text{new}}), \\ R &= \Pr(\hat{Y}(\mathcal{X}^{\text{new}}|\mathcal{B}_k, \pi_k, \boldsymbol{\mu}_k) \neq Y^{\text{new}}), \end{aligned}$$

where $\hat{\mathcal{B}}_k, \hat{\pi}_k$ and $\hat{\boldsymbol{\mu}}_k$ are the estimated coefficients, and $\mathcal{B}_k, \pi_k$ and $\boldsymbol{\mu}_k$ are true coefficients. Under certain conditions, $R_n \to R$ with probability tending to 1. In other words, CATCH can asymptotically achieve the optimal classification accuracy.

In [105], Yang and Dunson establish the posterior contraction rate of their proposed classification model. Suppose that the data are obtained for n observations $y^n = (y_1, ..., y_n)^\top$ ($y_i \in \{1, 2, ..., d_0\}$), which are conditionally independent given $\boldsymbol{X}^n = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^\top$ with $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip_n})^\top$, $x_{ij} \in \{1, ..., d\}$ and $p_n \gg n$. Assume that the design points $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are independent observations from an unknown probability distribution G_n on $\{1, 2, ..., d\}^{p_n}$. Denote

$$d(P, P_0) = \int \sum_{y=1}^{d_0} |P(y|x_1, ..., x_p) - P_0(y|x_1, ..., x_p)| G_n(dx_1, ..., dx_p),$$

where P_0 is the true distribution, and P is the estimated distribution. Then under the given prior and other assumptions, it follows that

$$\Pi_n\{P : d(P, P_0) \geq M\epsilon_n | y^n, \boldsymbol{X}^n\} \to 0 \ \ a.s.,$$

where $\epsilon_n \to 0$ ($n\epsilon_n^2 \to \infty, \sum_n \exp(-n\epsilon_n^2) < \infty$), M is a constant, and $\Pi_n(A|y^n, \boldsymbol{X}^n)$ is the posterior distribution of A given the observations. Based on this result, Yang and Dunson [105] further prove that the posterior convergence of the model can be very close to $n^{-1/2}$ under some near low rankness conditions.

Among tensor response regression problems, Guha and Guhaniyogi [26] establish the convergence rate for predictive densities of their proposed SGTM model. Specifically, let $f^*(\mathcal{Y}|\boldsymbol{x})$ be the true conditional density of $\mathcal{Y}$ given $\boldsymbol{x}$ and $f(\mathcal{Y}|\boldsymbol{x})$ be the random predictive density for which a posterior is obtained. Define an integrated Hellinger distance between f^* and f as

$$\mathcal{D}_H(f, f^*) = \sqrt{\int \int (\sqrt{f(\mathcal{Y}|\boldsymbol{x})} - \sqrt{f^*(\mathcal{Y}|\boldsymbol{x})})^2 \nu_{\mathcal{Y}}(d\mathcal{Y}) \nu_{\boldsymbol{x}}(d\boldsymbol{x})},$$

where $\nu_{\boldsymbol{x}}$ is the unknown probability measure for $\boldsymbol{x}$ and $\nu_{\mathcal{Y}}$ is the dominating measure for f and f^*. For a sequence ϵ_n satisfying $0 < \epsilon_n < 1, \epsilon_n \to 0$, and $n\epsilon_n^2 \to \infty$, under certain conditions it satisfies

$$E_{f^*}\Pi_n\{\mathcal{D}_H(f, f^*) > 4\epsilon_n | \{\mathcal{Y}_i, \boldsymbol{x}_i\}_{i=1}^n\} < 4e^{-n\epsilon_n^2}$$

for all large n, where Π_n is the posterior density. This result implies that the posterior probability outside a shrinking neighborhood around the true predictive density f^* converges to 0 as $n \to \infty$. Under further assumptions, the convergence rate ϵ_n can have an order close to the parametric optimal rate of $n^{-1/2}$ up to a $\log(n)$ factor.

6.4 Posterior computation

In terms of posterior inference methods, sampling methods such as MCMC and variational methods (e.g., Variational Expectation Maximization, Variational Inference, and Variational Bayes) are the two popular choices for Bayesian tensor analysis. MCMC is utilized in a majority of Bayesian tensor regression and some Bayesian tensor completion (decomposition) problems. The ergodic theory of MCMC guarantees that the sampled chain converges to the desired posterior distribution, and sometimes the MAP result is utilized to initialize the MCMC sampling for accelerating the convergence [103, 84]. In order to reduce the computational cost and adapt to different situations, batch MCMC and online MCMC are also used for posterior sampling [41, 40].

As an alternative strategy to approximate posterior densities for Bayesian models, variational inference is very frequently employed in Bayesian tensor completion methods. These methods do not guarantee producing samples from the exact target density, but they are in general faster and more scalable to large datasets than MCMC are. In this category, Variational Expectation Maximization (VEM) [104, 117, 118, 116], Variational Inference (VI) [119, 101, 39, 58], and Variational Bayes (VB) [82, 41, 112, 113, 115, 94, 64] are the classical choices, and the recently developed auto-encoding VB algorithm is employed to deal with intractable distributions [62, 36]. Various studies have also adopted specific frameworks to reduce computational complexity (e.g., batch VB [41], variational sparse Gaussian Processes [94, 116, 119, 101]) and accommodate online or streaming data (e.g., online VB-EM [117], streaming VB [18, 108], and Assumed Density Filtering/Expectation Propagation [19, 20, 73, 21]). Additionally, Bayesian tensor completion (regression) methods also utilize other methods including MLE [72], MAP [114] and EM [77, 35].

7. CONCLUSION

In Bayesian tensor analysis, the unique data structure and its high dimensionality create challenges in both computation and theory. Bayesian methods impose different decomposition structures on the tensor-valued data or coefficients to reduce the number of free parameters. While CP, Tucker and non-parametric decompositions are the most commonly used decomposition structures, other decompositions have received some attention under the Bayesian framework in recent years (e.g., tensor ring [64], tensor train [39], neural [36]).

A full Bayesian model requires the complete specification of a probabilistic model and priors over model parameters, both of which depends on the data type. For example, in tensor completion, when the tensor is continuous, the elements are usually assumed to follow a Gaussian distribution with the tensor mean following a decomposition structure [103, 62, 104]. The Gaussian distribution can be extended to model the binary data through a link function [84]. In terms of count data, an element-wise Poisson distribution is often utilized to relate the decomposition structure to the tensor-valued data, and a Dirichlet or Gamma prior can be applied to latent factors or the core tensor to enforce the non-negativity in coefficients [82, 41, 83]. For tensor regression problems, multivariate normal priors are placed over latent factors in the CP decomposition, with a Gaussian-Wishart prior on the hyper-parameters of the normal distribution to achieve conjugacy [103, 14, 84]. Specific priors on core tensor (e.g., the MGP prior [76, 77], the Gamma-Beta hierarchical prior [41]) or latent factors [113] in CP/Tucker structure can promote automatic rank inference by letting the posterior decide the optimal rank. Sparsity priors such as the M-DGDP prior [29, 10] and the M-SB prior [30] are also popular choices for latent factors in the CP structure to promote low rankness, and local/global sparsity. Integrating robust, interpretable and computationally scalable Bayesian tensor methods with complex models (e.g., nonlinear machine learning, reinforcement learning, causal inference, and dynamic models) remains an interesting future direction.

Bayesian tensor regression has been widely used in applications, especially in medical imaging analysis (e.g., MRI and EGG), where high resolution spatially correlated data are produced. For both tensor-predictor and tensor-response regressions, there is a need to model tensor-valued coefficients, which is achieved by using CP/Tucker decomposition or nonparametric models that utilize Gaussian processes to model the non-linear relationship in the coefficient tensor. Posterior inference is conducted by Markov Chain Monte Carlo (MCMC) with Gibbs sampling, optimization based methods (e.g., variational Bayes), and streaming methods (e.g., expectation propagation). It is still of interest to develop scalable algorithms that accommodate challenging settings such as streaming data analysis.

In terms of theoretical studies, most of the existing work focus on (near-)optimal convergence rates for posterior distributions of the tensor coefficients in regression-related problems [91, 27, 42, 72, 105, 26]. There are still many open problems such as theoretical analysis for Bayesian tensor completion (and other tensor problems that we did not cover in this review) and convergence analysis of computational algorithms.

Received 12 February 2023

REFERENCES

[1] Acar, E., Dunlavy, D. M., Kolda, T. G. and Mørup, M. (2011). Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* **106** 41–56.

[2] Adomavicius, G. and Tuzhilin, A. (2010). Context-aware recommender systems. In *Recommender Systems Handbook* 217–253. Springer.

[3] Andersson, C. A. and Bro, R. (1998). Improving the speed of multi-way algorithms:: Part I. Tucker3. *Chemometrics and Intelligent Laboratory Systems* **42** 93–103.

[4] Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics* 1152–1174. MR0365969

[5] Bazerque, J. A., Mateos, G. and Giannakis, G. B. (2013). Rank regularization and Bayesian inference for tensor completion and extrapolation. *IEEE Transactions on Signal Processing* **61** 5689–5703. MR3130035

[6] Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika* 291–306. MR2806429

[7] Bi, X., Qu, A. and Shen, X. (2018). Multilayer tensor factorization with applications to recommender systems. MR3852653

[8] Bi, X., Tang, X., Yuan, Y., Zhang, Y. and Qu, A. (2021). Tensors in statistics. *Annual Review of Statistics and its Application* **8** 345–368. MR4243551

[9] Bickel, P. J., Ritov, Y. and Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. MR2533469

[10] Billio, M., Casarin, R., Iacopini, M. and Kaufmann, S. (2022). Bayesian dynamic tensor regression. *Journal of Business & Economic Statistics* 1–11. MR4568033

[11] Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in Artificial Intelligence* 33–42.

[12] Bro, R. (1997). PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems* **38** 149–171.

[13] Bro, R. (1998). Multi-way analysis in the food industry-models, algorithms, and applications. In *MRI, EPG and EMA," Proc ICSLP 2000*. Citeseer.

[14] Chen, X., He, Z. and Sun, L. (2019). A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies* **98** 73–84.

[15] Chen, H., Raskutti, G. and Yuan, M. (2019). Non-convex projected gradient descent for generalized low-rank tensor regression. *The Journal of Machine Learning Research* **20** 172–208. MR3911412

[16] Chu, W. and Ghahramani, Z. (2009). Probabilistic models for incomplete multi-dimensional arrays. In *Artificial Intelligence and Statistics* 89–96. PMLR.

[17] Da Silva, C. and Herrmann, F. (2013). Hierarchical tucker tensor optimization-applications to tensor completion. SampTA 2013. In *10th International Conference on Sampling Theory and Application, Jacobs University Bremen*.

[18] Du, Y., Zheng, Y., Lee, K.-c. and Zhe, S. (2018). Probabilistic streaming tensor decomposition. In *2018 IEEE International Conference on Data Mining (ICDM)* 99–108. IEEE.

[19] Fang, S., Kirby, R. M. and Zhe, S. (2021). Bayesian streaming sparse Tucker decomposition. In *Uncertainty in Artificial Intelligence* 558–567. PMLR.

[20] Fang, S., Wang, Z., Pan, Z., Liu, J. and Zhe, S. (2021). Streaming Bayesian Deep Tensor Factorization. In *International Conference on Machine Learning* 3133–3142. PMLR.

[21] Fang, S., Narayan, A., Kirby, R. and Zhe, S. (2022). Bayesian Continuous-Time Tucker Decomposition. In *International Conference on Machine Learning* 6235–6245. PMLR.

[22] Gahrooei, M. R., Yan, H., Paynabar, K. and Shi, J. (2021). Multiple tensor-on-tensor regression: An approach for modeling processes with heterogeneous sources of data. *Technometrics* **63** 147–159. MR4251490

[23] Gandy, S., Recht, B. and Yamada, I. (2011). Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* **27** 025010. MR2765628

[24] Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85** 398–409. MR1141740

[25] Geng, X., Smith-Miles, K., Zhou, Z.-H. and Wang, L. (2009). Face image modeling by multilinear subspace analysis with missing values. In *Proceedings of the 17th ACM International Conference on Multimedia* 629–632.

[26] Guha, S. and Guhaniyogi, R. (2021). Bayesian generalized sparse symmetric tensor-on-vector regression. *Technometrics* **63** 160–170. MR4251491

[27] Guhaniyogi, R. (2017). Convergence rate of Bayesian supervised tensor modeling with multiway shrinkage priors. *Journal of Multivariate Analysis* **160** 157–168. MR3688696

[28] Guhaniyogi, R. (2020). Bayesian methods for tensor regression. *Wiley StatsRef: Statistics Reference Online* 1–18. MR3714242

[29] Guhaniyogi, R., Qamar, S. and Dunson, D. B. (2017). Bayesian tensor regression. *The Journal of Machine Learning Research* **18** 2733–2763. MR3714242

[30] Guhaniyogi, R. and Spencer, D. (2021). Bayesian tensor response regression with an application to brain activation studies. *Bayesian Analysis* **16** 1221–1249. MR4381133

[31] Guo, W., Kotsia, I. and Patras, I. (2011). Tensor learning for regression. *IEEE Transactions on Image Processing* **21** 816–827. MR2932176

[32] Håstad, J. (1989). Tensor rank is NP-complete. In *International Colloquium on Automata, Languages, and Programming* 451–460. Springer.

[33] Hackbusch, W. (2012). *Tensor Spaces and Numerical Tensor Calculus* **42**. Springer. MR3236394

[34] Hao, B., Wang, B., Wang, P., Zhang, J., Yang, J. and Sun, W. W. (2021). Sparse tensor additive regression. *Journal of Machine Learning Research* **22**. MR4253757

[35] Hayashi, K., Takenouchi, T., Shibata, T., Kamiya, Y., Kato, D., Kunieda, K., Yamada, K. and Ikeda, K. (2010). Exponential family tensor factorization for missing-values prediction and anomaly detection. In *2010 IEEE International Conference on Data Mining* 216–225. IEEE.

[36] He, L., Liu, B., Li, G., Sheng, Y., Wang, Y. and Xu, Z. (2018). Knowledge base completion by variational bayesian neural tensor decomposition. *Cognitive Computation* **10** 1075–1084.

[37] Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics* **9** 1169. MR3418719

[38] Hou, M., Wang, Y. and Chaib-draa, B. (2015). Online local gaussian process for tensor-variate regression: Application to fast reconstruction of limb movements from brain signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 5490–5494. IEEE.

[39] Hu, R., Nicholls, G. K. and Sejdinovic, D. (2021). Large scale tensor regression using kernels and variational inference. *Machine Learning* 1–51. MR4449005

[40] Hu, C., Rai, P. and Carin, L. (2015). Zero-truncated poisson tensor factorization for massive binary tensors. arXiv preprint 1508.04210.

[41] Hu, C., Rai, P., Chen, C., Harding, M. and Carin, L. (2015). Scalable bayesian non-negative tensor factorization for massive count data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* 53–70. Springer.

[42] Imaizumi, M. and Hayashi, K. (2016). Doubly decomposing nonparametric tensor regression. In *International Conference on Machine Learning* 727–736. PMLR.

[43] Karatzoglou, A., Amatriain, X., Baltrunas, L. and

Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender systems* 79–86.

[44] Kasai, H. and Mishra, B. (2016). Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *International Conference on Machine Learning* 1012–1021. PMLR.

[45] Kiers, H. A. (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society* **14** 105–122.

[46] Kiers, H. A., Ten Berge, J. M. and Bro, R. (1999). PARAFAC2–part I. A direct fitting algorithm for the PARAFAC2 model. *Journal of Chemometrics: A Journal of the Chemometrics Society* **13** 275–294.

[47] Kilmer, M. E., Braman, K., Hao, N. and Hoover, R. C. (2013). Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications* **34** 148–172. MR3032996

[48] Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

[49] Kolda, T. G. and Bader, B. W. (2009). Tensor Decompositions and Applications. *SIAM review* **51** 455–500. MR2535056

[50] Kressner, D., Steinlechner, M. and Vandereycken, B. (2014). Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics* **54** 447–468. MR3223510

[51] Kroonenberg, P. M. (1983). *Three-Mode Principal Component Analysis: Theory and Applications* **2**. DSWO press.

[52] Lee, I., Sinha, D., Mai, Q., Zhang, X. and Bandyopadhyay, D. (2022). Bayesian Regression Analysis of Skewed Tensor Responses. *Biometrics*.

[53] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics* **2** 164–168. MR0010666

[54] Li, L. and Zhang, X. (2017). Parsimonious Tensor Response Regression. *Journal of the American Statistical Association* **112** 1131–1146. MR3735365

[55] Li, W., Liu, C.-C., Zhang, T., Li, H., Waterman, M. S. and Zhou, X. J. (2011). Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS Computational Biology* **7** e1001106. MR2832106

[56] Li, Z., Suk, H.-I., Shen, D. and Li, L. (2016). Sparse multi-response tensor regression for Alzheimer's disease study with multivariate clinical assessments. *IEEE Transactions on Medical Imaging* **35** 1927–1936.

[57] Li, X., Xu, D., Zhou, H. and Li, L. (2018). Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences* **10** 520–545.

[58] Li, S., Xing, W., Kirby, M. and Zhe, S. (2020). Scalable Variational Gaussian Process Regression Networks. arXiv preprint arXiv:2003.11489.

[59] Liu, J., Musialski, P., Wonka, P. and Ye, J. (2009). Tensor Completion for Estimating Missing Values in Visual Data. In *ICCV*.

[60] Liu, J., Musialski, P., Wonka, P. and Ye, J. (2013). Tensor Completion for Estimating Missing Values in Visual Data. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **35** 208–220.

[61] Liu, Y., Shang, F., Cheng, H., Cheng, J. and Tong, H. (2014). Factor matrix trace norm minimization for low-rank tensor completion. In *Proceedings of the 2014 SIAM International Conference on Data Mining* 866–874. SIAM.

[62] Liu, B., He, L., Li, Y., Zhe, S. and Xu, Z. (2018). Neuralcp: Bayesian multiway data analysis with neural tensor decomposition. *Cognitive Computation* **10** 1051–1061.

[63] Lock, E. F. (2018). Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics* **27** 638–647. MR3863764

[64] Long, Z., Zhu, C., Liu, J. and Liu, Y. (2020). Bayesian low rank tensor ring model for image completion. arXiv preprint arXiv:2007.01055. MR4231918

[65] Mardani, M., Mateos, G. and Giannakis, G. B. (2015). Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing* **63** 2663–2677. MR3341765

[66] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* **11** 431–441. MR0153071

[67] Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence* 362–369.

[68] Mu, C., Huang, B., Wright, J. and Goldfarb, D. (2014). Square deal: Lower bounds and improved relaxations for tensor recovery. In *International Conference on Machine Learning* 73–81. PMLR.

[69] Narita, A., Hayashi, K., Tomioka, R. and Kashima, H. (2012). Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery* **25** 298–324. MR2951039

[70] Negahban, S. and Wainwright, M. J. (2012). Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research* **13** 1665–1697. MR2930649

[71] Nguyen-Tuong, D., Peters, J. and Seeger, M. (2008). Local Gaussian process regression for real time online model learning. *Advances in Neural Information Processing Systems* **21**.

[72] Pan, Y., Mai, Q. and Zhang, X. (2018). Covariate-adjusted tensor classification in high dimensions. *Journal of the American Statistical Association*. MR4011781

[73] Pan, Z., Wang, Z. and Zhe, S. (2020). Streaming nonlinear Bayesian tensor decomposition. In *Conference on Uncertainty in Artificial Intelligence* 490–499. PMLR.

[74] Rabanser, S., Shchur, O. and Günnemann, S. (2017). Introduction to tensor decompositions and their applications in machine learning. arXiv preprint arXiv:1711.10781.

[75] Rabusseau, G. and Kadri, H. (2016). Low-rank regression with tensor responses. *Advances in Neural Information Processing Systems* **29**.

[76] Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D. and Carin, L. (2014). Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *International Conference on Machine Learning* 1800–1808. PMLR.

[77] Rai, P., Hu, C., Harding, M. and Carin, L. (2015). Scalable probabilistic tensor factorization for binary and count data. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[78] Raskutti, G., Yuan, M. and Chen, H. (2019). Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statistics* **47** 1554–1584. MR3911122

[79] Rauhut, H., Schneider, R. and Stojanac, Ž. (2015). Tensor completion in hierarchical tensor representations. In *Compressed sensing and its applications* 419–450. Springer. MR3382114

[80] Rauhut, H., Schneider, R. and Stojanac, Ž. (2017). Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications* **523** 220–262. MR3624675

[81] Rendle, S. and Schmidt-Thieme, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining* 81–90.

[82] Schein, A., Paisley, J., Blei, D. M. and Wallach, H. (2014). Inferring polyadic events with Poisson tensor factorization. In *Proceedings of the NIPS 2014 Workshop on Networks: From Graphs to Rich Data*.

[83] Schein, A., Zhou, M., Blei, D. and Wallach, H. (2016). Bayesian Poisson Tucker decomposition for learning the structure of international relations. In *International Conference on Machine Learning* 2810–2819. PMLR.

[84] Sheng, G., Denoyer, L., Gallinari, P. and Jun, G. (2012). Probabilistic latent tensor factorization model for link pattern

prediction in multi-relational networks. *The Journal of China Universities of Posts and Telecommunications* **19** 172–181.

[85] Signoretto, M., De Lathauwer, L. and Suykens, J. A. (2010). Nuclear norms for tensors and their use for convex multilinear estimation. *Submitted to Linear Algebra and Its Applications* **43**.

[86] Song, Q., Ge, H., Caverlee, J. and Hu, X. (2019). Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **13** 1–48.

[87] Spencer, D., Guhaniyogi, R. and Prado, R. (2019). Bayesian mixed effect sparse tensor response regression model with joint estimation of activation and connectivity. arXiv preprint arXiv:1904.00148.

[88] Spencer, D., Guhaniyogi, R. and Prado, R. (2020). Joint Bayesian estimation of voxel activation and inter-regional connectivity in fMRI experiments. *Psychometrika* **85** 845–869. MR4204740

[89] Sun, W. W., Hao, B. and Li, L. (2014). Tensors in modern statistical learning. *Wiley StatsRef: Statistics Reference Online* 1–25.

[90] Sun, W. W. and Li, L. (2017). STORE: sparse tensor response regression and neuroimaging analysis. *The Journal of Machine Learning Research* **18** 4908–4944. MR3763769

[91] Suzuki, T. (2015). Convergence rate of Bayesian tensor estimator and its minimax optimality. In *International Conference on Machine Learning* 1273–1282. PMLR.

[92] Tarzanagh, D. A. and Michailidis, G. (2022). Regularized and smooth double core tensor factorization for heterogeneous data. *The Journal of Machine Learning Research* **23** 13162–13210. MR4577729

[93] Thanh, L. T., Linh-Trung, N. et al. (2022). A Contemporary and Comprehensive Survey on Streaming Tensor Decomposition.

[94] Tillinghast, C., Fang, S., Zhang, K. and Zhe, S. (2020). Probabilistic neural-kernel tensor decomposition. In *2020 IEEE International Conference on Data Mining (ICDM)* 531–540. IEEE.

[95] Tomasi, G. and Bro, R. (2005). PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems* **75** 163–180.

[96] Tomioka, R., Hayashi, K. and Kashima, H. (2010). Estimation of low-rank tensors via convex optimization. arXiv preprint arXiv:1010.0789.

[97] Trouillon, T., Dance, C. R., Welbl, J., Riedel, S., Gaussier, É. and Bouchard, G. (2017). Knowledge graph completion via complex tensor factorization. arXiv preprint arXiv:1702.06879. MR3763764

[98] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika* **31** 279–311. MR0205395

[99] Urtasun, R. and Darrell, T. (2008). Sparse probabilistic regression for activity-independent human pose inference. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* 1–8. IEEE.

[100] Walczak, B. and Massart, D. L. (2001). Dealing with missing data: Part I. *Chemometrics and Intelligent Laboratory Systems* **58** 15–27.

[101] Wang, Z., Chu, X. and Zhe, S. (2020). Self-modulating nonparametric event-tensor factorization. In *International Conference on Machine Learning* 9857–9867. PMLR.

[102] Wimalawarne, K., Tomioka, R. and Sugiyama, M. (2016). Theoretical and experimental analyses of tensor-based regression and classification. *Neural Computation* **28** 686–715. MR3867768

[103] Xiong, L., Chen, X., Huang, T.-K., Schneider, J. and Carbonell, J. G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining* 211–222. SIAM.

[104] Xu, Z., Yan, F. et al. (2011). Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis. arXiv preprint arXiv:1108.6296.

[105] Yang, Y. and Dunson, D. B. (2016). Bayesian conditional tensor factorizations for high-dimensional classification. *Journal of the American Statistical Association* **111** 656–669. MR3538695

[106] Ying, J., Lu, H., Wei, Q., Cai, J.-F., Guo, D., Wu, J., Chen, Z. and Qu, X. (2017). Hankel matrix nuclear norm regularized tensor completion for n-dimensional exponential signals. *IEEE Transactions on Signal Processing* **65** 3702–3717. MR3669920

[107] Yuan, M. and Zhang, C.-H. (2017). Incoherent tensor norms and their applications in higher order tensor completion. *IEEE Transactions on Information Theory* **63** 6753–6766. MR3707566

[108] Zhang, Z. and Hawkins, C. (2018). Variational bayesian inference for robust streaming tensor factorization and completion. In *2018 IEEE International Conference on Data Mining (ICDM)* 1446–1451. IEEE.

[109] Zhang, Z., Ely, G., Aeron, S., Hao, N. and Kilmer, M. (2014). Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3842–3849.

[110] Zhang, X., Li, L., Zhou, H., Zhou, Y., Shen, D. et al. (2019). Tensor generalized estimating equations for longitudinal imaging analysis. *Statistica Sinica* **29** 1977. MR3970344

[111] Zhang, A. R., Luo, Y., Raskutti, G. and Yuan, M. (2020). ISLET: Fast and optimal low-rank tensor regression via importance sketching. *SIAM Journal on Mathematics of Data Science* **2** 444–479. MR4106613

[112] Zhao, Q., Zhang, L. and Cichocki, A. (2015a). Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37** 1751–1763.

[113] Zhao, Q., Zhang, L. and Cichocki, A. (2015b). Bayesian sparse tucker models for dimension reduction and tensor completion. arXiv preprint arXiv:1505.02343.

[114] Zhao, Q., Zhou, G., Zhang, L. and Cichocki, A. (2014). Tensor-variate gaussian processes regression and its application to video surveillance. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1265–1269. IEEE. MR3406097

[115] Zhao, Q., Zhou, G., Zhang, L., Cichocki, A. and Amari, S.-I. (2015). Bayesian robust tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems* **27** 736–748. MR3476834

[116] Zhe, S. and Du, Y. (2018). Stochastic nonparametric event-tensor decomposition. *Advances in Neural Information Processing Systems* **31**.

[117] Zhe, S., Xu, Z., Chu, X., Qi, Y. and Park, Y. (2015). Scalable nonparametric multiway data analysis. In *Artificial Intelligence and Statistics* 1125–1134. PMLR.

[118] Zhe, S., Qi, Y., Park, Y., Xu, Z., Molloy, I. and Chari, S. (2016a). Dintucker: Scaling up gaussian process models on large multidimensional arrays. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[119] Zhe, S., Zhang, K., Wang, P., Lee, K.-c., Xu, Z., Qi, Y. and Ghahramani, Z. (2016b). Distributed flexible nonlinear tensor factorization. *Advances in Neural Information Processing Systems* **29**.

[120] 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. MR3174640

[121] Zhou, J., Sun, W. W., Zhang, J. and Li, L. (2021). Partially observed dynamic tensor response regression. *Journal of the American Statistical Association* 1–16. MR4571132

Yiyao Shi
Department of Statistics
University of California, Irvine
2209 Bren Hall
Irvine, CA 92697-1250
E-mail address: yiyaos@uci.edu

Weining Shen
Department of Statistics
University of California
Irvine
2206 Bren Hall
Irvine, CA 92697-1250
E-mail address: weinings@uci.edu