

Bayesian Sparse Tucker Models for Dimension Reduction and Tensor Completion

Qibin Zhao, *Member, IEEE*, Liqing Zhang, *Member, IEEE*, and Andrzej Cichocki *Fellow, IEEE*

Abstract—Tucker decomposition is the cornerstone of modern machine learning on tensorial data analysis, which have attracted considerable attention for multiway feature extraction, compressive sensing, and tensor completion. The most challenging problem is related to determination of model complexity (i.e., multilinear rank), especially when noise and missing data are present. In addition, existing methods cannot take into account uncertainty information of latent factors, resulting in low generalization performance. To address these issues, we present a class of probabilistic generative Tucker models for tensor decomposition and completion with structural sparsity over multilinear latent space. To exploit structural sparse modeling, we introduce two group sparsity inducing priors by hierarchical representation of Laplace and Student-t distributions, which facilitates fully posterior inference. For model learning, we derived variational Bayesian inferences over all model (hyper)parameters, and developed efficient and scalable algorithms based on multilinear operations. Our methods can automatically adapt model complexity and infer an optimal multilinear rank by the principle of maximum lower bound of model evidence. Experimental results and comparisons on synthetic, chemometrics and neuroimaging data demonstrate remarkable performance of our models for recovering ground-truth of multilinear rank and missing entries.

Index Terms—Tensor decomposition, tensor completion, Tucker decomposition, multilinear rank, structural sparsity, MRI completion



1 INTRODUCTION

TENSOR decomposition aims to seek a multilinear latent representation of multiway arrays, which is a fundamental technique for modern machine learning on tensorial data mining. In contrast to matrix factorization, tensor decomposition can effectively capture higher order structural properties by modeling multilinear interactions among a group of low-dimensional latent factors, leading to significant advantages for multiway dimension reduction [1], multiway regression/classification [2], [3], and compressive sensing [4] or completion of structured data [5]. In particular, Tucker [6] and Candecomp/Parafac (CP) [7] decompositions have attracted considerable interest in various fields, such as chemometrics, computer vision, social network analysis, and brain imaging [8], [9], [10].

Tucker decomposition represents an N th-order tensor by multilinear operations between N factor matrices and a core tensor. If we view factor matrices as basis of N latent space, then core tensor represents a multilinear coefficient for the whole tensor. An alternative perspective is that core tensor is viewed as basis of a latent manifold space, then factor matrices represent low-dimensional coefficients corresponding to N different profiles for each data point. Tucker decomposition can exactly represent an arbitrary random tensor. In contrast, CP decomposition enforces a strict structure assumption by a super-diagonal core tensor,

which is suitable for highly structured data but with limited generalization ability for various types of data. Tensor decomposition becomes interest only when underlying information has an intrinsic low-rank property, leading to a compact representation in latent space. Hence, one fundamental problem is separating the information and noise space by model selection, i.e., determination of tensor rank, which is challenging when noise variance is comparable to that of information. Note that tensor rank differs from matrix rank in many mathematic properties [11], [12], [13]. Numerous studies proposed tensor decomposition methods by either optimization techniques or probabilistic model learning [8], [9], [14], [15], [16], [17]. However, tensor rank is always required to be specified manually. In addition, general model selection methods need to perform tensor decomposition for each predefined model, which is inapplicable for Tucker model due to that the number of possibilities increases exponentially with the order of tensors. Some preliminary studies on automatic model determination were exploited in [18], [19] which however, may prone to overfitting and fail to recover true tensor rank in the case of highly noisy data, due to point estimation of model parameters.

Tensor completion can be formulated as tensor decomposition with missing values, which seems fairly straightforward but has several essential differences. Tensor decomposition focusing on data representation is referred to unsupervised technique, whereas tensor completion focusing on predictive ability is referred to semi-supervised technique, which aims to predict tensor elements from their N -tuple indices. Another important issue is that the optimal tensor rank for tensor completion is related to missing ratio and may differ from the underlying true rank that is always optimum for tensor decomposition. Hence, the key problem of tensor completion is to choose an appropriate tensor rank (i.e., model complexity) that can obtain optimal generaliza-

- Q. Zhao is with Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan and Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.
- L. Zhang is with MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems and Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.
- A. Cichocki is with Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan and Systems Research Institute in Polish Academy of Science, Warsaw, Poland.

tion performance, whereas this problem is more challenging as compared to tensor decomposition for a fully observed tensor. More specifically, existing model selections including cross-validation are not applicable due to correlations between the optimal tensor rank and number of observed data points. Although numerous tensor decomposition methods for partially observed tensor were developed in [20], [21], [22], [23], [24], [25], [26], the most challenging task of specifying tensor rank was conveyed to user, and thus was done mostly based on an implicit assumption that missing data is known. Several studies have considered automatic rank determination for CP decomposition [5], [27]. In contrast to CP rank, multilinear rank has more degree of freedom. An alternative framework for tensor completion, based on a low-rank assumption, was developed by minimizing the nuclear norm of approximate tensor, which corresponds to a convex relaxation of rank minimization problem [28]. Although this technique successfully avoids specifying tensor rank manually, several tuning parameters are required and sensitive to missing ratio. Hence, it essentially transformed the model selection problem to a parameter selection problem. The nuclear norm based tensor completion was shown to be attractive in recent years [29], [30], [31], [32], [33], [34]. Many variants by imposing additional constraints were also exploited in [35], [36] which have shown advantages for some specific type of data. However, the best performance was mostly obtained by carefully tuning parameters based on implicit assumption that missing data is known. Another issue is that the definition of nuclear norm of tensor corresponds to a (weighted) summation of mode- n rank R_n denoting the dimension of latent factor matrices. However, the dimension of core tensor $\prod_n R_n$ represents the model complexity of whole tensor as described previously. As a result, another possible framework can be introduced by optimization of logarithm transformed nuclear norm.

In this paper, we present a class of generative models for Tucker representation of a complete or incomplete tensor corrupted by noise, which can automatically adapt model complexity and infer an optimal multilinear rank solely from observed data. To achieve automatic model determination, we investigate structural sparse modeling through formulating Laplace as well as Student-t distributions in a hierarchical representation to facilitate full posterior inference, which therefore can be further extended to enforce group sparsity over factor matrices and structural sparsity over core tensor. For model learning, we derive the full posterior inference under a variational Bayesian framework, including remarkable inference for the non-conjugate hierarchical Laplace prior. Finally, all model parameters can be inferred as well as predictive distribution over missing data without needing of tuning parameters. In addition, we introduce several Theorems based on multilinear operations to improve computational efficiency and scalability.

The rest of this paper is organized as follows: Section 2 presents notations and multilinear operations. In Section 3, we present group sparse modeling by hierarchical sparsity inducing priors. Section 4 presents Bayesian Tucker model for tensor decomposition, while Section 5 presents Bayesian Tucker model for tensor completion. The algorithm related issues are discussed in Section 6. Section 7 shows experimental results followed by conclusion in Section 8.

2 PRELIMINARIES AND NOTATIONS

Let $\mathcal{X}, \mathbf{X}, \mathbf{x}$ denote a tensor, matrix, vector respectively. Given an N th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, its (i_1, \dots, i_N) th entry is denoted by $\mathcal{X}_{i_1 \dots i_N}$, where $i_n = 1, \dots, I_n, n = 1, \dots, N$. The standard Tucker decomposition is defined by

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \dots \times_N \mathbf{U}^{(N)}. \quad (1)$$

$\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}\}_{n=1}^N$ are a set of mode- n factor matrices, $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ denotes the core tensor and (R_1, \dots, R_N) denote the dimensions of mode- n latent space, respectively. The overall model complexity can be represented by $\prod_n R_n$ or $\sum_n R_n$, whose minimum associated values $\{R_n\}_{n=1}^N$ is termed as *multilinear rank* of tensor \mathcal{X} [9]. For a specific $\mathbf{U}^{(n)}$, we denote its row vectors by $\{\mathbf{u}_{i_n}^{(n)} | i_n = 1, \dots, I_n\}$ and its column vectors by $\{\mathbf{u}_{r_n}^{(n)} | r_n = 1, \dots, R_n\}$.

Definition 2.1. Let $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}\}_{n=1}^N$ denote a set of matrices, the sequential *Kronecker* products in a reversed order is defined and denoted by

$$\begin{aligned} \bigotimes_n \mathbf{U}^{(n)} &= \mathbf{U}^{(N)} \otimes \mathbf{U}^{(N-1)} \otimes \dots \otimes \mathbf{U}^{(1)}. \\ \bigotimes_{k \neq n} \mathbf{U}^{(k)} &= \mathbf{U}^{(N)} \otimes \dots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \dots \otimes \mathbf{U}^{(1)}. \end{aligned}$$

The symbol \otimes denotes *Kronecker* product. $\bigotimes_n \mathbf{U}^{(n)}$ is a matrix of size $(\prod_n I_n \times \prod_n R_n)$.

The Tucker decomposition (1) can be also represented by using matrix, vector, or element-wise forms, given by

$$\begin{aligned} \mathbf{X}_{(n)} &= \mathbf{U}^{(n)} \mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)T} \right), \\ \text{vec}(\mathcal{X}) &= \left(\bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}), \\ \mathcal{X}_{i_1 \dots i_N} &= \left(\bigotimes_n \mathbf{u}_{i_n}^{(n)T} \right) \text{vec}(\mathcal{G}). \end{aligned} \quad (2)$$

It should be noted that the multilinear operation is significantly efficient for computation. For example, if we compute $\bigotimes_n \mathbf{U}^{(n)}$ firstly and then multiply it with $\text{vec}(\mathcal{G})$, both the computation and memory complexity is $\mathcal{O}(\prod_n I_n R_n)$. In contrast, if we apply a sequence of multilinear operations $(\cdot) \times_n \mathbf{U}^{(n)}$ without explicitly computing $\bigotimes_n \mathbf{U}^{(n)}$, the computational complexity is $\mathcal{O}(\min_n(R_n) \prod_n I_n)$ while the memory cost is $\mathcal{O}(\prod_n I_n)$. In this paper, we use notation $\bigotimes_n (\cdot)$ frequently for clarity, however, the implementation can be performed by using multilinear operations.

3 HIERARCHICAL GROUP SPARSITY PRIORS

The sparsity inducing priors are considerably important and powerful for many machine learning models. The most popular ones are Laplace, Student-t, and Spike and slab priors. However, these priors are often not conjugate with the likelihood distribution, which leads to difficulties for fully Bayesian inference. In contrast, another popular sparsity inducing prior is automatic relevance determination (ARD), which has been widely applied in many powerful

methods such as relevance vector machine and Bayesian principle component analysis [37], [38], [39]. The advantages of ARD prior lie in its conjugacy resulted from a hierarchical structure. Note that ARD prior is essentially a hierarchical Student-t distribution since its marginal distribution is

$$\int_0^\infty \mathcal{N}(x|0, \lambda^{-1}) Ga(\lambda|a, b) d\lambda = \mathcal{T}(x|0, ab^{-1}, 2a). \quad (3)$$

$Ga(x|a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$ denotes a Gamma distribution and $\mathcal{T}(x)$ denotes a Student-t distribution. One can show that when a noninformative prior is specified by $a = b \rightarrow 0$, then $p(x) \propto 1/|x|$, which indicates the sparsity inducing property of a hierarchical Student-t prior.

A straightforward question is that can we represent Laplace prior by the hierarchical distributions which are conjugate? As shown in [40], [41], a hierarchical structure of Gaussian and Exponential distributions yields a Laplace marginal distribution, whereas they are not conjugate priors. In this paper, we present another hierarchical Laplace distribution by employing an Inverse Gamma distribution $IG(x|a, b) = \frac{b^a}{\Gamma(a)} x^{-a-1} e^{-bx^{-1}}$. By a specific setting of $IG(1, \frac{\gamma}{2}) = \frac{\gamma}{2} x^{-2} e^{-\frac{\gamma}{2} x^{-1}}$, we can show that the marginal distribution is

$$\int_0^\infty \mathcal{N}(x|0, \lambda^{-1}) IG(\lambda|1, \frac{\gamma}{2}) d\lambda = Laplace(x|0, \frac{1}{\sqrt{\gamma}}). \quad (4)$$

Note that γ govern the degree of sparsity, for example, if $\gamma = 1$, then $p(x) \propto e^{-|x|}$. To avoid parameter selection manually, we can place a noninformative prior over γ and employ Bayesian inference for posterior estimation. Although inverse Gamma is a non-conjugate prior, we can infer its variational posterior represented by a more generalized distribution.

Let $\mathbf{x} = \{x_1, \dots, x_R\}$ denote a set of random variables, the hierarchical sparsity inducing priors can be specified as $\forall r = 1, \dots, R$,

$$\text{Student-t: } x_r \sim \mathcal{N}(0, \lambda_r^{-1}), \quad \lambda_r \sim Ga(a, b), \quad (5)$$

$$\text{Laplace: } x_r \sim \mathcal{N}(0, \lambda_r^{-1}), \quad \lambda_r \sim IG(1, \frac{\gamma}{2}). \quad (6)$$

Hence, the marginal distributions of \mathbf{x} can be found by (3), (4) as $p(\mathbf{x}) = \prod_{r=1}^R \mathcal{T}(x_r|0, ab^{-1}, 2a)$ and $p(\mathbf{x}) = \prod_{r=1}^R Laplace(x_r|0, \frac{1}{\sqrt{\gamma}})$.

Now we consider to extend the above hierarchical priors for group sparse modeling. Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_R\}$ denote R groups of random variables where $\mathbf{x}_r \in \mathbb{R}^{I_r}$ denote r th group that contains I_r random variables. Note that $I_1 = I_2 \dots = I_R$ is not necessary. The group sparse modeling is to enforce sparsity on groups in contrast to the individual random variables, which effectively take into account the clustering properties of relevant variables. To employ the hierarchical sparsity priors for group sparse model, it can be specified as $\forall r = 1, \dots, R$,

$$\text{Student-t: } \mathbf{x}_r \sim \mathcal{N}(\mathbf{0}, \lambda_r^{-1} \mathbf{I}_{I_r}), \quad \lambda_r \sim Ga(a, b), \quad (7)$$

$$\text{Laplace: } \mathbf{x}_r \sim \mathcal{N}(\mathbf{0}, \lambda_r^{-1} \mathbf{I}_{I_r}), \quad \lambda_r \sim IG(1, \frac{\gamma}{2}). \quad (8)$$

Therefore, the marginal distributions of \mathcal{X} can be derived as $p(\mathcal{X}) = \prod_{r=1}^R \mathcal{T}(\mathbf{x}_r|\mathbf{0}, ab^{-1}, 2a)$ and $p(\mathcal{X}) =$

$\prod_{r=1}^R Laplace(\mathbf{x}_r|\mathbf{0}, \frac{1}{\sqrt{\gamma}})$, where $\mathcal{T}(\mathbf{x})$, $Laplace(\mathbf{x})$ denote a multivariate Student-t distribution and a multivariate Laplace distribution respectively [42]. One can show that when $a = b \rightarrow 0$, then $\forall r$, $p(\mathbf{x}_r) \propto (1/\|\mathbf{x}_r\|_2)^{I_r}$. When $\gamma \rightarrow 1$, then $\forall r$, $p(\mathbf{x}_r) \propto e^{-\|\mathbf{x}_r\|_2}$.

In the following sections, we employ both hierarchical Student-t and hierarchical Laplace priors to impose group sparsity over multi-mode latent factors, yielding the minimum latent dimensions, for tensor decomposition models.

4 BAYESIAN TUCKER MODEL FOR TENSOR DECOMPOSITION

4.1 Model specification

We first consider the Bayesian Tucker model for an N th-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ that is fully observed. We assume that \mathcal{Y} is a measurement of the latent tensor \mathcal{X} corrupted by i.i.d. Gaussian noises, i.e., $\mathcal{Y} = \mathcal{X} + \varepsilon$, where \mathcal{X} is generated exactly by the Tucker representation as shown in (1). Therefore, the observation model can be specified by a vectorized form,

$$\text{vec}(\mathcal{Y}) | \{\mathbf{U}^{(n)}\}, \mathcal{G}, \tau \sim \mathcal{N} \left(\left(\bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}), \tau^{-1} \mathbf{I} \right). \quad (9)$$

Our objective is to infer the model parameters, as well as model complexity, automatically and solely from given data. To this end, we propose the hierarchical prior distributions over all model (hyper)parameters. For noise precision τ , a *Gamma* prior can be simply placed with appropriate hyperparameters, yielding an noninformative prior distribution.

To modeling tensor data by an appropriate model complexity, it is important to design the flexible prior distributions over the group of factor matrices $\{\mathbf{U}^{(n)}\}$, $n = 1, \dots, N$, and the core tensor \mathcal{G} , whose complexity can be adapted to a specific given tensor. Since the model complexity of Tucker decompositions depends on the dimensions of \mathcal{G} , denoted by (R_1, R_2, \dots, R_N) , while $R_n, n = 1, \dots, N$ also corresponds to the number of columns in model- n factor matrix $\mathbf{U}^{(n)}$ and thus represents the dimensions of mode- n latent space. The minimal number of $\{R_n\}_{n=1}^N$ i.e., *multilinear rank* [9] usually need to be given in advance. However, due to the presence of noise, the optimal selection of multilinear rank is quite challenging. Although some model selection criterions can be applied, the accuracy significantly depends on the decomposition algorithms, resulting in less stability and high computational cost. Therefore, we seek an elegant automatic model selection, which can not only infer the multilinear rank, but also effectively avoid overfitting. To achieve this, we employ the proposed group sparsity priors over factor matrices. More specifically, each $\mathbf{U}^{(n)}$ is govern by hyperparameters $\boldsymbol{\lambda}^{(n)} \in \mathbb{R}^{R_n}$, where $\lambda_{r_n}^{(n)}$ controls the precision related to r_n group (i.e., r_n th column). Due to the group sparsity inducing property, the dimensions of latent space will be enforced to be minimal. On the other hand, the core tensor \mathcal{G} also needs to be as sparse as possible. However, if we straightforwardly place an independent sparsity prior, the interactions between \mathcal{G} and $\{\mathbf{U}^{(n)}\}$ cannot be modeled, which may lead to inaccurate estimation of multilinear rank. As $\mathcal{G}_{r_1, \dots, r_N}$ can be considered as the

scalar coefficient of the rank-one tensor $\mathbf{u}_{.r_1}^{(1)} \otimes \cdots \otimes \mathbf{u}_{.r_N}^{(N)}$ that involves r_n th group from each of $\mathbf{U}^{(n)}$, respectively. Taken into account the fact that if $\exists n, \exists r_n, \mathbf{u}_{.r_n}^{(n)} = 0$, then $\mathcal{G}_{r_1, \dots, r_N}$ should be also enforced to be zero, the precision parameter for $\mathcal{G}_{r_1, \dots, r_N}$ can be specified as the product of precisions over $\{\mathbf{u}_{.r_n}^{(n)}\}_{n=1}^N$, which is represented by

$$\mathcal{G}_{r_1, \dots, r_N} \mid \{\boldsymbol{\lambda}^{(n)}\}, \beta \sim \mathcal{N} \left(0, \left(\beta \prod_n \lambda_{r_n}^{(n)} \right)^{-1} \right), \quad (10)$$

where β is a scale parameter related to the magnitude of \mathcal{G} , on which a hyperprior can be placed. The hyperprior for $\boldsymbol{\lambda}^{(n)}$ play a key role for different sparsity inducing priors. We propose two hierarchical priors including Student-t and Laplace for group sparsity. Let $\boldsymbol{\Lambda}^{(n)} = \text{diag}(\boldsymbol{\lambda}^{(n)})$, we can finally specify the hierarchical model priors as

$$\begin{aligned} \tau &\sim Ga(a_0^\tau, b_0^\tau), \\ \text{vec}(\mathcal{G}) \mid \{\boldsymbol{\lambda}^{(n)}\}, \beta &\sim \mathcal{N} \left(\mathbf{0}, \left(\beta \bigotimes_n \boldsymbol{\Lambda}^{(n)} \right)^{-1} \right), \\ \beta &\sim Ga(a_0^\beta, b_0^\beta), \\ \mathbf{u}_{.i_n}^{(n)} \mid \boldsymbol{\lambda}^{(n)} &\sim \mathcal{N} \left(\mathbf{0}, \boldsymbol{\Lambda}^{(n)-1} \right), \quad \forall n, \forall i_n, \\ \text{Student-t: } \lambda_{r_n}^{(n)} &\sim Ga(a_0^\lambda, b_0^\lambda), \quad \forall n, \forall r_n, \\ \text{Laplace: } \lambda_{r_n}^{(n)} &\sim IG(1, \frac{\gamma}{2}), \quad \forall n, \forall r_n, \\ \gamma &\sim Ga(a_0^\gamma, b_0^\gamma). \end{aligned} \quad (11)$$

The prior for the core tensor \mathcal{G} is written as a tensor-variate Gaussian distribution. The observation model in (9) and the hierarchical priors in (11) are integrated, which is termed as Bayesian Tucker Decomposition (BTD) model. Note that there are two proposed hierarchical sparsity priors in BTD model, which are thus denoted respectively by BTD-T (Student-t priors) and BTD-L (Laplace priors).

For simplicity, all unknown (hyper)parameters in BTD model are collected and denoted by $\Theta = \{\mathcal{G}, \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)}, \boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(N)}, \tau, \beta, \gamma\}$. Thus the joint distribution of BTD model is written as

$$\begin{aligned} p(\mathcal{Y}, \Theta) &= p(\mathcal{Y} \mid \{\mathbf{U}^{(n)}\}, \mathcal{G}, \tau) \prod_n p(\mathbf{U}^{(n)} \mid \boldsymbol{\lambda}^{(n)}) \\ &\times p(\mathcal{G} \mid \{\boldsymbol{\Lambda}^{(n)}\}, \beta) \prod_n p(\boldsymbol{\lambda}^{(n)} \mid \gamma) p(\gamma) p(\beta) p(\tau). \end{aligned} \quad (12)$$

In general, maximum a posterior (MAP) of Θ can be estimated by optimizations of logarithm joint distribution w.r.t. each parameters alternately. However, due to the property of point estimation, MAP is still prone to overfitting. In contrast, we aim to infer the posterior distributions of Θ under a fully Bayesian treatment, which is $p(\Theta \mid \mathcal{Y}) = \frac{p(\Theta, \mathcal{Y})}{\int p(\Theta, \mathcal{Y}) d\Theta}$.

4.2 Model inference

The model learning can be performed by the approximate Bayesian inferences when the posterior distributions are analytically intractable. Since the variational Bayesian inference [43] is more efficient and scalable as compared to sampling based inference methods, we employ VB technique to learn both BTD-T and BTD-L models. It should be noted that the hierarchical Laplace priors are not conjugate,

resulting in a challenging problem to be addressed. In this section, we present the main solutions for model inference while the detailed derivations and proofs are provided in the Appendix.

VB inference aims to seek an optimal $q(\Theta)$ to approximate the true posterior distribution in the sense of $\min KL(q(\Theta) \parallel p(\Theta \mid \mathcal{Y}))$. Since $KL(q(\Theta) \parallel p(\Theta \mid \mathcal{Y})) = \ln p(\mathcal{Y}) - L(q)$, the optimum of $q(\Theta)$ can be achieved by maximization of lower bound $L(q)$ that can be computed explicitly. To achieve this, we assume that the variational approximation posteriors can be factorized as

$$q(\Theta) = q(\mathcal{G})q(\beta) \prod_n q(\mathbf{U}^{(n)}) \prod_n q(\boldsymbol{\lambda}^{(n)})q(\gamma)q(\tau). \quad (13)$$

Then, the optimized form of j th parameters based on $\max_{q(\Theta_j)} L(q)$ is given by

$$q_j(\Theta_j) \propto \exp \{ \mathbb{E}_{q(\Theta \setminus \Theta_j)} [\ln p(\mathcal{Y}, \Theta)] \}. \quad (14)$$

$\mathbb{E}_{q(\Theta \setminus \Theta_j)}[\cdot]$ denotes an expectation w.r.t. q distribution over all variables in Θ except Θ_j . In the following, we use $\mathbb{E}[\cdot]$ to denote the expectation w.r.t. $q(\Theta)$ for simplicity.

As can be derived, the variational posterior distribution over the core tensor \mathcal{G} is

$$q(\mathcal{G}) = \mathcal{N} \left(\text{vec}(\mathcal{G}) \mid \text{vec}(\tilde{\mathcal{G}}), \Sigma_G \right), \quad (15)$$

where the posterior parameters can be updated by

$$\text{vec}(\tilde{\mathcal{G}}) = \mathbb{E}[\tau] \Sigma_G \left(\bigotimes_n \mathbb{E} \left[\mathbf{U}^{(n)T} \right] \right) \text{vec}(\mathcal{Y}), \quad (16)$$

$$\Sigma_G = \left\{ \mathbb{E}[\beta] \bigotimes_n \mathbb{E} \left[\boldsymbol{\Lambda}^{(n)} \right] + \mathbb{E}[\tau] \bigotimes_n \mathbb{E} \left[\mathbf{U}^{(n)T} \mathbf{U}^{(n)} \right] \right\}^{-1}. \quad (17)$$

Most of expectation terms in (16), (17) are the functions linearly related to the corresponding Θ_j , which can thus be easily evaluated from their posterior $q(\Theta_j)$. For example, $\mathbb{E}[\mathbf{U}^{(n)T}]$ can be evaluated according to $q(\mathbf{U}^{(n)})$ shown in (18), as can be similarly computed for $\mathbb{E}[\tau]$, $\mathbb{E}[\beta]$, and $\mathbb{E}[\boldsymbol{\Lambda}^{(n)}]$. It should be noted that the expectation involving a quadratic term can be evaluated explicitly by using $\mathbb{E}[\mathbf{U}^{(n)T} \mathbf{U}^{(n)}] = \mathbb{E}[\mathbf{U}^{(n)T}] \mathbb{E}[\mathbf{U}^{(n)}] + I_n \Psi^{(n)}$, which requires the posterior parameters given in (18).

The computational complexity for posterior update of \mathcal{G} is $\mathcal{O}(\prod_n R_n^3 + \prod_n I_n R_n)$ that polynomially scales with model complexity denoted by $\prod_n R_n$ and linearly scales with data size denoted by $\prod_n I_n$. In general, it is dominated by $\mathcal{O}(\prod_n R_n^3)$, which is related to the matrix inverse. The memory cost is $\mathcal{O}(\prod_n R_n^2 + \prod_n I_n R_n)$, dominated by Σ_G and $\bigotimes_n(\cdot)$. It should be noted that multilinear operations $\mathcal{Y} \times_1 \mathbb{E}[\mathbf{U}^{(1)T}] \times \cdots \times_N \mathbb{E}[\mathbf{U}^{(N)T}]$ can be performed without explicitly computing $\bigotimes_n \mathbb{E}[\mathbf{U}^{(n)T}]$, thus reducing the memory cost to $\mathcal{O}(\prod_n R_n^2 + \prod_n I_n)$ and also reducing computation complexity to $\mathcal{O}(\prod_n R_n^3 + \prod_n I_n)$. To further improve computation and memory efficiency, we introduce several important Theorems as follows.

Theorem 4.1. Let $\{\Sigma^{(n)}\}$ be a set of diagonalizable matrices, and c_1, c_2 denote arbitrary scalars. If $\forall n = 1, \dots, N$, the spectral decompositions are represented by $\Sigma^{(n)} = \mathbf{V}^{(n)} \mathbf{D}^{(n)} \mathbf{V}^{(n)T}$, then

$$\left(c_1 \mathbf{I} + c_2 \bigotimes_n \Sigma^{(n)} \right)^{-1} = \left(\bigotimes_n \mathbf{V}^{(n)} \right) \left(c_1 \mathbf{I} + c_2 \bigotimes_n \mathbf{D}^{(n)} \right)^{-1} \left(\bigotimes_n \mathbf{V}^{(n)T} \right).$$

Proof. See Appendix for the detailed proof. \square

Theorem 4.2. Let $\{\Lambda^{(n)}\}$ be a set of diagonal matrices, $\{\Sigma^{(n)}\}$ be a set of diagonalizable matrices, and c_1, c_2 be scalars. If $\forall n = 1, \dots, N$, the spectral decompositions are represented by $\Lambda^{(n)-\frac{1}{2}} \Sigma^{(n)} \Lambda^{(n)-\frac{1}{2}} = \mathbf{V}^{(n)} \mathbf{D}^{(n)} \mathbf{V}^{(n)T}$, then

$$\left(c_1 \bigotimes_n \Lambda^{(n)} + c_2 \bigotimes_n \Sigma^{(n)} \right)^{-1} = \left(\bigotimes_n \Lambda^{(n)-\frac{1}{2}} \mathbf{V}^{(n)} \right) \left(c_1 \mathbf{I} + c_2 \bigotimes_n \mathbf{D}^{(n)} \right)^{-1} \left(\bigotimes_n \mathbf{V}^{(n)T} \Lambda^{(n)-\frac{1}{2}} \right).$$

Proof. See Appendix for the detailed proof. \square

Based on Theorem 4.1 and 4.2, Σ_G can be factorized as the product of sequential Kronecker products, which leads to that matrix inverse operations can be performed by individual eigenvalue decompositions on N small matrices, and inverse operations only on a diagonal matrix. Therefore, the computational complexity for inference of \mathcal{G} can be significantly reduced to $\mathcal{O}(\sum_n R_n^3 + \prod_n I_n)$ while the memory cost can be significantly reduced to $\mathcal{O}(\sum_n R_n^2 + \prod_n I_n)$, if we save Σ_G by a format of sequential Kronecker products.

As can be derived, the variational posterior distribution over the factor matrices $\{\mathbf{U}^{(n)}\}$ is represented by

$$q(\mathbf{U}^{(n)}) = \prod_{i_n=1}^{I_n} \mathcal{N}(\mathbf{u}_{i_n}^{(n)} | \tilde{\mathbf{u}}_{i_n}^{(n)}, \Psi^{(n)}), \quad n = 1, \dots, N, \quad (18)$$

where the posterior parameters can be updated by

$$\tilde{\mathbf{U}}^{(n)} = \mathbb{E}[\tau] \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \mathbb{E}[\mathbf{U}^{(k)}] \right) \mathbb{E}[\mathbf{G}_{(n)}^T] \Psi^{(n)}, \quad (19)$$

$$\Psi^{(n)} = \left\{ \mathbb{E}[\Lambda^{(n)}] + \mathbb{E}[\tau] \mathbb{E} \left[\mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)T} \mathbf{U}^{(k)} \right) \mathbf{G}_{(n)}^T \right] \right\}^{-1}. \quad (20)$$

In (20), the most complex expectation term related to multilinear operations can be computed by

$$\begin{aligned} & \text{vec} \left\{ \mathbb{E} \left[\mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)T} \mathbf{U}^{(k)} \right) \mathbf{G}_{(n)}^T \right] \right\} \\ &= \mathbb{E} [\mathbf{G}_{(n)} \otimes \mathbf{G}_{(n)}] \text{vec} \left(\bigotimes_{k \neq n} \mathbb{E} [\mathbf{U}^{(k)T} \mathbf{U}^{(k)}] \right). \end{aligned} \quad (21)$$

Thus, each posterior expectation term can be evaluated easily according to the corresponding posterior distributions $q(\mathcal{G})$ and $\{q(\mathbf{U}^{(k)})\}$, $k = 1, \dots, N, k \neq n$.

Taken into account the computation and memory efficiency, multilinear operations and sequential Kronecker products format must be employed to avoid explicitly computation of sequential Kronecker products. It should be noted that (21) cannot be factorized into operations on individual Kronecker terms because of $\mathbb{E}[\mathbf{G}_{(n)} \otimes \mathbf{G}_{(n)}]$. To reduce the memory cost, we may approximate it by $\mathbb{E}[\mathbf{G}_{(n)}] \otimes \mathbb{E}[\mathbf{G}_{(n)}]$. Therefore, the computational complexity for inference of $\mathbf{U}^{(n)}$ can be improved to $\mathcal{O}(R_n^3 + \prod_n I_n)$ while the memory cost is $\mathcal{O}(\prod_n R_n + \prod_n I_n)$.

The variational posterior distribution over β can be derived to be a Gamma distribution due to its conjugate prior, which is denoted by $q(\beta) = Ga(a_M^\beta, b_M^\beta)$ where the posterior parameters can be updated by

$$\begin{aligned} a_M^\beta &= a_0^\beta + \frac{1}{2} \prod_n R_n, \\ b_M^\beta &= b_0^\beta + \frac{1}{2} \mathbb{E} [\text{vec}(\mathcal{G}^2)^T] \bigotimes_n \mathbb{E} [\lambda^{(n)}]. \end{aligned} \quad (22)$$

In (22), $\mathbb{E} [\text{vec}(\mathcal{G}^2)^T] = \text{vec}(\mathbb{E}[\mathcal{G}^2])^T + \text{diag}(\Sigma_G)^T$ should be applied for rigorous inference, whereas an alternative approximation is $\mathbb{E} [\text{vec}(\mathcal{G}^2)^T] = \text{vec}(\mathbb{E}[\mathcal{G}^2])^T$ for efficiency. $\{\mathbb{E}[\lambda^{(n)}]\}$ can be easily evaluated according to $\{q(\lambda^{(n)})\}$ described in the following paragraphs. The computational complexity in (22) is $\mathcal{O}(\prod_n R_n)$.

The inference of hyperparameters $\{\lambda^{(n)}\}$ plays a key role for automatic model selection (i.e., determination of multilinear rank). In BTM models, as we proposed two hierarchical sparsity priors, resulting in two different posterior distributions for $\{\lambda^{(n)}\}$.

BTM-T model using hierarchical Student-t priors. As can be derived, the variation posterior distribution over $\{\lambda^{(n)}\}$ is i.i.d. Gamma distributions due to the conjugate priors, which is $\forall n = 1, \dots, N$,

$$q(\lambda^{(n)}) = \prod_{r_n=1}^{R_n} Ga(\lambda_{r_n}^{(n)} | \tilde{a}_{r_n}^{(n)}, \tilde{b}_{r_n}^{(n)}), \quad (23)$$

where the posterior parameters can be updated by

$$\begin{aligned} \tilde{a}_{r_n}^{(n)} &= a_0^\lambda + \frac{1}{2} \left(I_n + \prod_{k \neq n} R_k \right), \\ \tilde{b}_{r_n}^{(n)} &= b_0^\lambda + \frac{1}{2} \mathbb{E} [\mathbf{u}_{r_n}^{(n)T} \mathbf{u}_{r_n}^{(n)}] \\ &\quad + \frac{1}{2} \mathbb{E}[\beta] \mathbb{E} [\text{vec}(\mathcal{G}_{\dots r_n \dots}^2)^T] \bigotimes_{k \neq n} \mathbb{E} [\lambda^{(k)}]. \end{aligned} \quad (24)$$

According to $\{q(\mathbf{U}^{(n)})\}$ described in (18), we obtain that

$$\mathbb{E}[\mathbf{u}_{r_n}^{(n)T} \mathbf{u}_{r_n}^{(n)}] = I_n(\Psi^{(n)})_{r_n r_n} + \tilde{\mathbf{u}}_{r_n}^{(n)T} \tilde{\mathbf{u}}_{r_n}^{(n)}. \quad (25)$$

This represents the posterior expectation of squared L_2 -norm of r th component in mode- n factors, which also takes into account the uncertainty information. $\mathbb{E} [\text{vec}(\mathcal{G}_{\dots r_n \dots}^2)^T]$ represents the posterior expectation of squared L_2 -norm of r_n slice of core tensor \mathcal{G} . Therefore, an intuitive interpretation of automatic model selection is that the smaller of $\|\mathbf{u}_{r_n}^{(n)}\|_F^2$ and $\|\mathcal{G}_{\dots r_n \dots}\|_F^2$ leads to larger $\mathbb{E}[\lambda_{r_n}^{(n)}]$ and updated prior for $p(\mathbf{u}_{r_n}^{(n)} | \lambda_{r_n}^{(n)})$ as well as $p(\mathcal{G}_{\dots r_n \dots} | \lambda_{r_n}^{(n)})$,

which in turn enforces more strongly $\mathbf{u}_{:r_n}^{(n)}$ and $\mathcal{G}_{\dots r_n \dots}$ to be zero. After several iterations, the unnecessary columns in factor matrices and unnecessary slices in core tensor can be reduced to exact zero (i.e., smaller than machine precision). The computational complexity for inference of $q(\boldsymbol{\lambda}^{(n)})$ is $\mathcal{O}(\prod_n R_n + I_n R_n)$. Given updated parameters for $q(\boldsymbol{\lambda}^{(n)})$, we can evaluate the posterior expectations by $\mathbb{E}[\boldsymbol{\lambda}^{(n)}] = [\tilde{a}_1^{(n)}/\tilde{b}_1^{(n)}, \dots, \tilde{a}_{R_n}^{(n)}/\tilde{b}_{R_n}^{(n)}]^T$ and $\mathbb{E}[\boldsymbol{\Lambda}^{(n)}] = \text{diag}(\mathbb{E}[\boldsymbol{\lambda}^{(n)}])$.

BTD-L model using hierarchical Laplace priors. Since the hierarchical Laplace prior is not conjugate, which leads to much difficulties for model inference. To solve this problem, we employ a generalized inverse Gaussian distribution, denoted by $GIG(x|h, a, b)$, which includes Gamma, inverse Gamma, and inverse Gaussian distribution as special cases by an appropriate setting of hyperparameters. For example, one can show that $Ga(a, b) = GIG(a, 2b, 0)$ and $IG(a, b) = GIG(-a, 0, 2b)$. For BTD-L mode, we keep using $\lambda_{r_n}^{(n)} \sim IG(1, \frac{\gamma}{2})$ as the hyper-prior, however the variational posterior $q(\lambda_{r_n}^{(n)})$ cannot be represented as an inverse Gamma distribution.

As can be derived, the variation posterior distribution over $\{\lambda^{(n)}\}$ can be represented as i.i.d. GIG distributions, which is $\forall n = 1, \dots, N$,

$$q(\boldsymbol{\lambda}^{(n)}) = \prod_{r_n=1}^{R_n} GIG(\lambda_{r_n}^{(n)} | h, \tilde{a}_{r_n}^{(n)}, \tilde{b}_{r_n}^{(n)}), \quad (26)$$

where the posterior parameters can be updated by

$$\begin{aligned} h &= \frac{1}{2} \left(I_n + \prod_{k \neq n} R_k \right) - 1, & \tilde{b}_{r_n}^{(n)} &= \mathbb{E}[\gamma], \\ \tilde{a}_{r_n}^{(n)} &= \mathbb{E}[\beta] \mathbb{E} \left[\text{vec}(\mathcal{G}_{\dots r_n \dots}^2)^T \right] \bigotimes_{k \neq n} \mathbb{E} \left[\boldsymbol{\lambda}^{(k)} \right] \\ &+ \mathbb{E} \left[\mathbf{u}_{:r_n}^{(n)T} \mathbf{u}_{:r_n}^{(n)} \right]. \end{aligned} \quad (27)$$

The computational complexity for inference of $q(\boldsymbol{\lambda}^{(n)})$ is also $\mathcal{O}(\prod_n R_n + I_n R_n)$. Given the updated parameters, we can evaluate $\mathbb{E}_{GIG}[\lambda_{r_n}^{(n)}]$ straightforwardly, while an alternative approximation is the posterior mode w.r.t. GIG distribution that can avoid computational instabilities of modified Bessel function.

By comparing (26) with (23), we can investigate the essential difference between Student-t and Laplace priors. One can show that (23) can be rewritten as $GIG(\tilde{a}_{r_n}^{(n)}, 2\tilde{b}_{r_n}^{(n)}, 0)$ with parameters given by (24). Hence, the key difference lies in the setting of γ . If $\gamma = 0$, Student-t and Laplace priors are essentially equivalent. To avoid manually tuning parameters, we also place a hyper-prior over γ and thus derive the variational posterior distribution as $q(\gamma) = Ga(a_M^\gamma, b_M^\gamma)$ whose posterior parameters can be updated by

$$\begin{aligned} a_M^\gamma &= a_0^\gamma + \sum_{n=1}^N R_n, \\ b_M^\gamma &= b_0^\gamma + \frac{1}{2} \sum_{n=1}^N \sum_{r_n=1}^{R_n} \mathbb{E} \left[\lambda_{r_n}^{(n)-1} \right]. \end{aligned} \quad (28)$$

It should be noted that $\mathbb{E}[\lambda_{r_n}^{(n)-1}]$ cannot be computed straightforwardly by $\mathbb{E}[\lambda_{r_n}^{(n)}]^{-1}$. Since $q(\lambda_{r_n}^{(n)})$ is a GIG

distribution as shown in (26), it is not difficult to derive that $q(\lambda_{r_n}^{(n)-1}) = GIG(-h, \tilde{b}_{r_n}^{(n)}, \tilde{a}_{r_n}^{(n)})$, yielding the posterior expectation computed by

$$\mathbb{E}_{GIG} \left[\lambda_{r_n}^{(n)-1} \right] = \frac{\sqrt{\tilde{a}_{r_n}^{(n)}} K_{1-h} \left(\sqrt{\tilde{a}_{r_n}^{(n)} \tilde{b}_{r_n}^{(n)}} \right)}{\sqrt{\tilde{b}_{r_n}^{(n)}} K_{-h} \left(\sqrt{\tilde{a}_{r_n}^{(n)} \tilde{b}_{r_n}^{(n)}} \right)}, \quad (29)$$

and the posterior mode computed by

$$\arg \max_{\lambda_{r_n}^{(n)-1}} GIG \left(\lambda_{r_n}^{(n)-1} \right) = \frac{(-h-1) + \sqrt{(-h-1)^2 + \tilde{a}_{r_n}^{(n)} \tilde{b}_{r_n}^{(n)}}}{\tilde{b}_{r_n}^{(n)}}, \quad (30)$$

where $K_{1-h}(\cdot)$ denotes a modified Bessel function of the second kind.

As can be derived, the variational posterior distribution over the noise hyperparameter is $q(\tau) = Ga(a_M^\tau, b_M^\tau)$ whose parameters can be updated by

$$\begin{aligned} a_M^\tau &= a_0^\tau + \frac{1}{2} \prod_n I_n, \\ b_M^\tau &= b_0^\tau + \frac{1}{2} \mathbb{E} \left[\left\| \text{vec}(\mathcal{Y}) - \left(\bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}) \right\|_F^2 \right], \end{aligned} \quad (31)$$

where the posterior expectation of model residuals can be evaluated by

$$\begin{aligned} &\mathbb{E} \left[\left\| \text{vec}(\mathcal{Y}) - \left(\bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}) \right\|_F^2 \right] = \\ &\|\mathcal{Y}\|_F^2 - 2 \text{vec}(\mathcal{Y})^T \left(\bigotimes_n \mathbb{E}[\mathbf{U}^{(n)}] \right) \mathbb{E}[\text{vec}(\mathcal{G})] \\ &+ \text{Tr} \left(\mathbb{E} \left[\text{vec}(\mathcal{G}) \text{vec}(\mathcal{G})^T \right] \bigotimes_n \mathbb{E} \left[\mathbf{U}^{(n)T} \mathbf{U}^{(n)} \right] \right). \end{aligned} \quad (32)$$

In principle, $\mathbb{E}[\text{vec}(\mathcal{G}) \text{vec}(\mathcal{G})^T] = \text{vec}(\tilde{\mathcal{G}}) \text{vec}(\tilde{\mathcal{G}})^T + \Sigma_G$. However, $\text{vec}(\tilde{\mathcal{G}}) \text{vec}(\tilde{\mathcal{G}})^T$ can be alternatively used as an approximation, which then makes it possible to apply multilinear operations for computing (32) quite efficiently. Hence, the computational complexity can then be reduced to $\mathcal{O}(\prod_n R_n + \prod_n I_n)$.

The inference framework presented in this section can essentially maximize the lower bound of model evidence which is defined by $\mathcal{L}(q) = \mathbb{E}_{q(\Theta)}[\ln p(\mathcal{Y}, \Theta)] + H(q(\Theta))$. The first term denotes the posterior expectation of joint distribution while the second term denotes the entropy of $q(\Theta)$. In principle, $\mathcal{L}(q)$ should increase at each iteration, thus it can be used to test for convergence. We provide the detailed computation forms of $\mathcal{L}(q)$ in the Appendix.

5 BAYESIAN TUCKER MODEL FOR TENSOR COMPLETION

5.1 Model specification

In this section, we consider Bayesian Tucker model for tensor completion. Let \mathcal{Y} denotes an incomplete tensor (i.e., with missing entries), and \mathcal{O} denotes a binary tensor indicating the observation positions, i.e., $\mathcal{O}_{i_1 \dots i_N} = 1$ if $(i_1, \dots, i_N) \in \Omega$ otherwise it is zero. Ω denotes a set of N -tuple indices of observed entries. \mathcal{Y}_Ω denotes only observed

entries. Similar to BTM model, we assume a generative model $\mathcal{Y}_\Omega = \mathcal{X}_\Omega + \varepsilon$ where the latent tensor \mathcal{X} can be represented exactly by a Tucker model with a low multilinear rank and ε denotes i.i.d. Gaussian noise.

Given an incomplete tensor, Bayesian Tucker model only considers the observed entries, yielding a new likelihood function represented by

$$p(\mathcal{Y}_\Omega | \{\mathbf{U}^{(n)}\}, \mathcal{G}, \tau) = \prod_{(i_1, \dots, i_N) \in \Omega} \mathcal{N}(\mathcal{Y}_{i_1 \dots i_N} | \mathcal{X}_{i_1 \dots i_N}, \tau^{-1}).$$

Based on Tucker decomposition framework, we can thus represent the observation model as that $\forall (i_1, \dots, i_N)$,

$$\mathcal{Y}_{i_1 \dots i_N} | \{\mathbf{u}_{i_n}^{(n)}\}, \mathcal{G}, \tau \sim \mathcal{N}\left(\left(\bigotimes_n \mathbf{u}_{i_n}^{(n)T}\right) \text{vec}(\mathcal{G}), \tau^{-1}\right)^{\mathcal{O}_{i_1 \dots i_N}}. \quad (33)$$

For Tucker decomposition of an incomplete tensor, the problem is ill-conditioned and has infinite solutions. The low-rank assumption play an key role for successful tensor completion, which implies that the determination of multilinear rank significantly affects the predictive performance. However, standard model selection strategies, such as cross-validation, cannot be applied for finding the optimal multilinear rank because it varies dramatically with missing ratios. Therefore, the inference of multilinear rank is more challenging when missing values occur.

As described in BTM model, we employ two types of hierarchical group sparsity priors over the factor matrices and core tensor with aim to seek the minimum multilinear rank automatically, which is more efficient and elegant than the standard model selections by repeating many times and selecting one optimum model. Therefore, the model priors for all hidden variables are same with that in BTM model. By combining likelihood model in (33) with the model priors in (11), we propose a Bayesian Tucker Completion (BTC) model, which enable us to infer the minimum multilinear rank as well as the noise level solely from partially observed data without requiring any tuning parameters.

5.2 Model inference

For BTC model, we also employ VB inference framework to learn the model under a fully Bayesian treatment. Since BTC model differs from BTM model in the likelihood function (33), indicating that the inference for factor matrices, core tensor and noise parameter are essentially different, while other hyperparameters can be inferred by the same solutions. In this section, we present only the main solutions while the detailed derivations are provided in the Appendix.

As can be derived, the variational posterior distribution over the core tensor \mathcal{G} is

$$q(\mathcal{G}) = \mathcal{N}\left(\text{vec}(\mathcal{G}) | \text{vec}(\tilde{\mathcal{G}}), \Sigma_G\right) \quad (34)$$

where the posterior parameters can be updated by

$$\text{vec}(\tilde{\mathcal{G}}) = \mathbb{E}[\tau] \Sigma_G \sum_{(i_1, \dots, i_N) \in \Omega} \left(\mathcal{Y}_{i_1 \dots i_N} \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)}] \right) \quad (35)$$

$$\Sigma_G = \left\{ \mathbb{E}[\beta] \bigotimes_n \mathbb{E}[\Lambda^{(n)}] + \mathbb{E}[\tau] \sum_{(i_1, \dots, i_N) \in \Omega} \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)} \mathbf{u}_{i_n}^{(n)T}] \right\}^{-1} \quad (36)$$

It should be noted that Theorems 4.1, 4.2 and multilinear operations cannot be applied to (36) due to the sum of *kronecker* products. Thus the sequential *kronecker* products must be computed explicitly, resulting in the computational cost of $\mathcal{O}(\prod_n R_n^3 + M \prod_n R_n^2)$, where M denotes the number of observed entries (i.e., data size), and memory cost of $\mathcal{O}(\prod_n R_n^2)$. This severely prevents the method from being applied to large-scale datasets. To improve scalability, we propose an alternative solution by optimizing $\arg \min_{\mathcal{G}} \{-\ln q(\mathcal{G})\}$ instead of closed-form update in (35). This can be achieved by employing a nonlinear conjugate gradient method with the gradient given by

$$\begin{aligned} \frac{\partial \{-\ln q(\mathcal{G})\}}{\partial \text{vec}(\mathcal{G})} &= \mathbb{E}[\beta] \left(\bigotimes_n \mathbb{E}[\Lambda^{(n)}] \right) \text{vec}(\mathcal{G}) \\ &+ \mathbb{E}[\tau] \sum_{(i_1, \dots, i_N) \in \Omega} \left\{ \left(\bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)} \mathbf{u}_{i_n}^{(n)T}] \right) \text{vec}(\mathcal{G}) \right\} \\ &- \mathbb{E}[\tau] \sum_{(i_1, \dots, i_N) \in \Omega} \left(\mathcal{Y}_{i_1 \dots i_N} \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)}] \right) \end{aligned} \quad (37)$$

Thus, multilinear operations can be applied without explicitly computation of *kronecker* products, resulting in reduced computational complexity of $\mathcal{O}(M \prod_n R_n)$ and reduced memory cost of $\mathcal{O}(\prod_n R_n)$, which scales linearly with data size and model complexity.

As can be derived, the variational posterior distribution over $\{\mathbf{U}^{(n)}\}$ is factorized as

$$q(\mathbf{U}^{(n)}) = \prod_{i_n} \mathcal{N}\left(\mathbf{u}_{i_n}^{(n)} | \tilde{\mathbf{u}}_{i_n}^{(n)}, \Psi_{i_n}^{(n)}\right), \quad n = 1, \dots, N, \quad (38)$$

where the posterior parameters can be updated by

$$\tilde{\mathbf{u}}_{i_n}^{(n)} = \mathbb{E}[\tau] \Psi_{i_n}^{(n)} \mathbb{E}[\mathbf{G}_{(n)}] \sum_{(i_1, \dots, i_N) \in \Omega} \left(\mathcal{Y}_{i_1 \dots i_N} \bigotimes_{k \neq n} \mathbb{E}[\mathbf{u}_{i_k}^{(k)}] \right) \quad (39)$$

$$\Psi_{i_n}^{(n)} = \left\{ \mathbb{E}[\Lambda^{(n)}] + \mathbb{E}[\tau] \mathbb{E}[\mathbf{G}_{(n)} \Phi_{i_n}^{(n)} \mathbf{G}_{(n)}^T] \right\}^{-1} \quad (40)$$

where $\Phi_{i_n}^{(n)} = \sum_{(i_1, \dots, i_N) \in \Omega} \bigotimes_{k \neq n} \mathbf{u}_{i_k}^{(k)} \mathbf{u}_{i_k}^{(k)T}$, the summation is performed over the observed data locations whose mode- n index is fixed to i_n . In other words, $\Phi_{i_n}^{(n)}$ represents the statistical information of mode- k ($k \neq n$) latent factors that interact with $\mathbf{u}_{i_n}^{(n)}$. In (40), the complex posterior expectation can be computed by

$$\text{vec} \left\{ \mathbb{E}[\mathbf{G}_{(n)} \Phi_{i_n}^{(n)} \mathbf{G}_{(n)}^T] \right\} = \mathbb{E}[\mathbf{G}_{(n)} \otimes \mathbf{G}_{(n)}] \text{vec}(\Phi_{i_n}^{(n)}). \quad (41)$$

The computational complexity for inference of $\mathbf{u}_{i_n}^{(n)}$ is $\mathcal{O}(R_n^3 + M \prod_n R_n)$, while the memory cost is $\mathcal{O}(\prod_n R_n)$.

An intuitive interpretation of (40) is that the posterior covariance $\Psi_{i_n}^{(n)}$ combines prior information denoted by $\mathbb{E}[\Lambda^{(n)}]$ and posterior information of interacted factors in other modes, while the tradeoff between these two terms is controlled by model residual $\mathbb{E}[\tau]$. Hence, if updated $\mathbb{E}[\Lambda_{r_n}^{(n)}]$ is quite large and model is not well fitted, then the posterior variance of r_n th component will be easily forced to zero.

Given updated $q(\mathbf{u}_{i_n}^{(n)})$ from (38), we can easily compute the posterior expectations of some expressions as

$$\begin{aligned}\mathbb{E}[\mathbf{u}_{r_n}^{(n)T} \mathbf{u}_{r_n}^{(n)}] &= \tilde{\mathbf{u}}_{r_n}^{(n)T} \tilde{\mathbf{u}}_{r_n}^{(n)} + \sum_{i_n} (\Psi_{i_n}^{(n)})_{r_n r_n}, \\ \mathbb{E}[\mathbf{u}_{i_n}^{(n)} \mathbf{u}_{i_n}^{(n)T}] &= \tilde{\mathbf{u}}_{i_n}^{(n)} \tilde{\mathbf{u}}_{i_n}^{(n)T} + \Psi_{i_n}^{(n)}.\end{aligned}\quad (42)$$

The variational posterior distribution over noise precision τ can be derived as $q(\tau) = \text{Ga}(\tau | a_M^\tau, b_M^\tau)$ whose parameters are updated by

$$\begin{aligned}a_M^\tau &= a_0^\tau + \frac{1}{2} \sum_{i_1, \dots, i_N} O_{i_1 \dots i_N}, \\ b_M^\tau &= b_0^\tau + \frac{1}{2} \mathbb{E} \left[\sum_{(i_1, \dots, i_N) \in \Omega} \left(\mathcal{Y}_{i_1 \dots i_N} - \left(\bigotimes_n \mathbf{u}_{i_n}^{(n)T} \right) \text{vec}(\mathcal{G}) \right)^2 \right]\end{aligned}\quad (43)$$

The posterior expectation of model residuals over observed entries can be evaluated by

$$\begin{aligned}\mathbb{E} \left[\sum_{(i_1, \dots, i_N) \in \Omega} \left(\mathcal{Y}_{i_1 \dots i_N} - \left(\bigotimes_n \mathbf{u}_{i_n}^{(n)T} \right) \text{vec}(\mathcal{G}) \right)^2 \right] &= \\ \|\mathcal{Y}_\Omega\|_F^2 - 2 \left(\sum_{(i_1, \dots, i_N) \in \Omega} \mathcal{Y}_{i_1 \dots i_N} \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)T}] \right) \mathbb{E}[\text{vec}(\mathcal{G})] & \\ + \text{Tr} \left\{ \mathbb{E}[\text{vec}(\mathcal{G}) \text{vec}(\mathcal{G})^T] \sum_{(i_1, \dots, i_N) \in \Omega} \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)} \mathbf{u}_{i_n}^{(n)T}] \right\}. &\end{aligned}\quad (44)$$

The computational complexity is $\mathcal{O}(M \prod_n R_n)$, if multilinear operations have been applied.

Note that other hidden hyperparameters in Θ except $\{\mathcal{G}, \{\mathbf{U}^{(n)}\}, \tau\}$ can be inferred essentially by the same solutions with BTM model. In addition, the lower bound $\mathcal{L}(q) = \mathbb{E}_{q(\Theta)}[\ln p(\mathcal{Y}_\Omega, \Theta)] + H(q(\Theta))$ can be also evaluated with different expressions related to $\{\mathcal{G}, \{\mathbf{U}^{(n)}\}, \tau\}$ (see Appendix for details).

The predictive distributions over missing entries, given observed entries, can be approximated by using variational posterior distributions $q(\Theta)$ as follows

$$\begin{aligned}p(\mathcal{Y}_{i_1 \dots i_N} | \mathcal{Y}_\Omega) &= \int p(\mathcal{Y}_{i_1 \dots i_N} | \Theta) p(\Theta | \mathcal{Y}_\Omega) d\Theta \\ &\approx \mathcal{N} \left(\mathcal{Y}_{i_1 \dots i_N} \mid \tilde{\mathcal{Y}}_{i_1 \dots i_N}, \mathbb{E}[\tau]^{-1} + \sigma_{i_1 \dots i_N}^2 \right),\end{aligned}\quad (45)$$

where the posterior parameters can be obtained by

$$\begin{aligned}\tilde{\mathcal{Y}}_{i_1 \dots i_N} &= \left(\bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)T}] \right) \mathbb{E}[\text{vec}(\mathcal{G})] \\ \sigma_{i_1 \dots i_N}^2 &= \text{Tr} \left(\mathbb{E}[\text{vec}(\mathcal{G}) \text{vec}(\mathcal{G})^T] \bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)} \mathbf{u}_{i_n}^{(n)T}] \right) \\ &\quad - \mathbb{E}[\text{vec}(\mathcal{G})]^T \left(\bigotimes_n \mathbb{E}[\mathbf{u}_{i_n}^{(n)}] \mathbb{E}[\mathbf{u}_{i_n}^{(n)T}] \right) \mathbb{E}[\text{vec}(\mathcal{G})].\end{aligned}\quad (46)$$

Therefore, our model can provide not only predictions over missing entries, but also the uncertainty of predictions, which is quite important for some specific applications.

6 ALGORITHM RELATED ISSUES

We denote Bayesian sparse Tucker models (i.e., BTM and BTC) when two different group sparsity priors have been employed by BTM-T, BTM-L, BTC-T, BTC-L respectively, where T, L represent a hierarchical Student-t and Laplace priors. The main procedure of model inference is described as follows:

- **Setting of top-level hyperparameters.** For BTM-T and BTC-T models, $\{a_0^\tau, b_0^\tau, a_0^\beta, b_0^\beta, a_0^\lambda, b_0^\lambda\}$ are set to 1e-9. For BTM-L and BTC-L models, $\{a_0^\tau, b_0^\tau, a_0^\beta, b_0^\beta, a_0^\gamma, b_0^\gamma\}$ are set to 1e-9. This setting results in a noninformative prior, which ensures that model inference is solely based on observed data.
- **Initialization of hidden parameters Θ .** For BTM-T and BTC-T, $\Theta = \{\mathcal{G}, \{\mathbf{U}^{(n)}\}, \{\boldsymbol{\lambda}^{(n)}\}, \tau, \beta\}$. For BTM-L and BTC-L, $\Theta = \{\mathcal{G}, \{\mathbf{U}^{(n)}\}, \{\boldsymbol{\lambda}^{(n)}\}, \tau, \beta, \gamma\}$. \mathcal{G} does not need to be initialized due to that it will be updated firstly. The latent dimensions can be initialized by $R_n = I_n, \forall n$ or manually by $R_n < I_n$. $\{\mathbf{U}^{(n)}\}$ can be initialized as left singular vectors by mode- n SVD, while an alternative one is sampling from $\mathcal{N}(0, 1)$. τ is initialized by $1/\sigma_{\mathcal{Y}}^2$ denoting an inverse of data variance. $\{\boldsymbol{\lambda}^{(n)}\}, \beta, \gamma$ are simply initialized to be 1.
- **Variational model inference.** For all the models, the approximate inference can be updated sequentially in the order of $\{\mathcal{G}, \{\mathbf{U}^{(n)}\}, \beta, \{\boldsymbol{\lambda}^{(n)}\}, \gamma, \tau\}$.
- **Model reduction.** Pruning out zero-valued latent components from $\{\mathbf{U}^{(n)}\}$ as well as the associated slices from \mathcal{G} and associated components from $\{\boldsymbol{\lambda}^{(n)}\}$.
- **Lower bound of model evidence.** The lower bound of log-marginal likelihood is evaluated according to specific models, which can be used to test for convergence.
- **Predictive distribution.** For BTM models, the predictive distribution can be inferred for recovering latent tensor without noises. For BTC models, it can be inferred for recovering missing entries.

For BTM-T and BTM-L models, the overall computational complexity is $\mathcal{O}(\sum_n R_n^3 + N \prod_n I_n)$, while the memory cost is $\mathcal{O}(\prod_n R_n + \prod_n I_n)$. For BTC-T and BTC-L models, the overall computational complexity is $\mathcal{O}(\sum_n I_n R_n^3 + M \prod_n R_n \sum_n I_n)$, while the memory cost is $\mathcal{O}(\prod_n R_n + \sum_n I_n R_n)$. Therefore, BTM models are more computational efficient than BTC models, but it requires more memory. The computational complexity of all models scales linearly with data size, but polynomially with model complexity. Hence, our models are suitable for relatively low multilinear rank tensors. It should be noted that, because of automatic model reduction, $\{R_1, \dots, R_N\}$ reduces rapidly in the first few iterations, resulting in that computational complexity will be decreased with number of iterations.

The key advantage of our models is automatic model determination (i.e., learning multilinear rank), which enable us to obtain an optimal low-rank Tucker approximation from a noisy and incomplete tensor. Secondly, taking into account uncertainty information of all model parameters by full posterior inference, our method can effectively pre-

vent overfitting problem and provide predictive uncertainty as well. Thirdly, a deterministic Bayesian inference with closed-form update rules is derived for model learning, which is more efficient and scalable than sampling based inference. Finally, our methods are significantly convenient for practical applications since they do not require any tuning parameters.

7 EXPERIMENTAL RESULTS

We evaluated the proposed models and algorithms by extensive experiments and comparisons with state-of-the-art related works which include HOOI [9], ARD-Tucker [18], iHOOI [31], WTucker [20], HaLRTC [28]. In all our experiments, the parameters of HOOI, WTucker, and HaLRTC were tuned carefully and the best possible performances were reported. ARD-Tucker and iHOOI can automatically adapt tensor rank, which is similar to BTM and BTC methods proposed in this paper.

7.1 Synthetic data

The simulated data was generated according to an orthogonal Tucker model. More specifically, the core tensor was drawn from $\mathcal{G}_{r_1 r_2 r_3} \sim \mathcal{N}(0, 1), \forall r_n = 1, \dots, R_n$. The factor matrices were drawn from $\mathbf{u}_{i_n r_n}^{(n)} \sim \mathcal{N}(0, 1), \forall i_n = 1, \dots, 50, \forall r_n = 1, \dots, R_n, \forall n = 1, 2, 3$, and then orthogonalization was performed such that $\mathbf{U}^{(n)T} \mathbf{U}^{(n)} = \mathbf{I}_{R_n}$. In principle, such data can be perfectly fitted by HOSVD or a more powerful method HOOI.

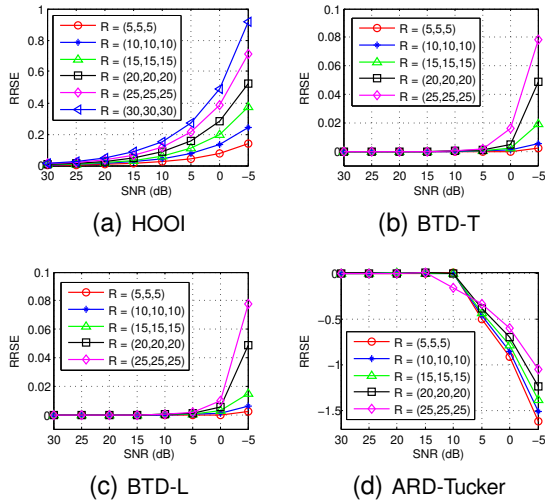


Fig. 1. (a) shows performance baseline using true tensor rank under varying noise levels (SNR) and tensor rank (R). (b)(c)(d) show discrepancy between baseline and three methods which can infer rank automatically.

We evaluated BTM-T/L on complete tensors under varying noise levels with SNR ranging from 30dB to -5dB and true tensor rank (R_1, R_2, R_3) ranging from 5 to 25. The reconstruction performance was evaluated by $RRSE = \|\tilde{\mathcal{X}} - \mathcal{X}\|_F / \|\mathcal{X}\|_F$ where $\tilde{\mathcal{X}}$ denotes the estimation of true latent tensor \mathcal{X} . Theoretically, the best performance should be obtained by HOOI given true tensor rank. As shown in Fig. 1, RRSE of HOOI increases exponentially with

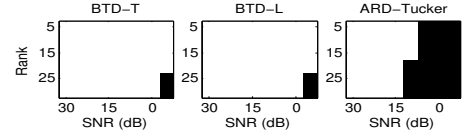


Fig. 2. Accuracy of inferred tensor rank under conditions of varying SNR and R. White area indicates successful estimations and black area indicates failed cases.

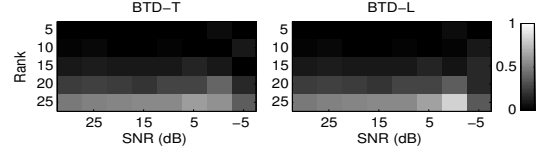


Fig. 3. Absolute error of inferred SNR. The maximum error is close to 1dB when $R=(25, 25, 25)$ and $SNR=-5dB$.

noise levels and larger R leads to a more rapid degeneration of performance. It should be emphasized that BTM-T/L achieve surprisingly improved performances without knowing true tensor rank, and the improvements become more significant for larger R and smaller SNR. The RRSEs obtained by BTM-T and BTM-L are almost equal under all conditions. ARD-Tucker, which can also automatically infer tensor rank, however obtained significantly degenerated performance when $SNR < 10dB$. The accuracy of rank determination is compared in Fig. 2, showing that BTM-T/L are able to exactly infer the tensor rank only except the cases of $SNR=-5dB, R \geq 25$ and outperform ARD-Tucker. In addition, BTM-T/L can infer noise levels accurately as shown in Fig. 3. The averaged runtime of BTM-T/L are 4/3 seconds while ARD-Tucker needs 47 seconds.

We evaluated BTC-T/L on incomplete tensors under varying missing ratios (MR) ranging from 50% to 90% and true rank $\{R_n\}$ ranging from 5 to 15, while SNR was fixed to 30dB. As shown in Fig. 4, BTC-T/L perform similarly under all conditions and significantly outperform iHOOI and HaLRTC. The RRSE increases with MR and larger R leads to a more rapid increase. iHOOI and HaLRTC, which are able to optimize tensor rank base on nuclear norm, perform well on relatively low rank tensors but degenerates rapidly with increasing rank. It should be emphasized that BTC-T/L exactly inferred tensor rank with 100% accuracy under all these conditions. As shown in Fig. 5, BTC-T/L can also accurately infer SNR when either true rank or MR is low, and the runtime depends on true rank. In summary, if both MR and true rank increase, the completion performance will decrease.

7.2 Chemometrics data

We evaluated BTM and BTC models on chemometrics data analysis¹ in terms of recovering the number of latent components which have meaningful interpretation and imputing missing data. Five tested data sets are Amino Acids Fluorescence ($5 \times 201 \times 61$), Flow Injection ($12 \times 100 \times 89$), Electronic Nose ($18 \times 241 \times 12$), Sugar Process ($268 \times 571 \times 7$), and

1. Datasets are available from repository for multi-way data analysis. <http://www.models.kvl.dk/datasets>

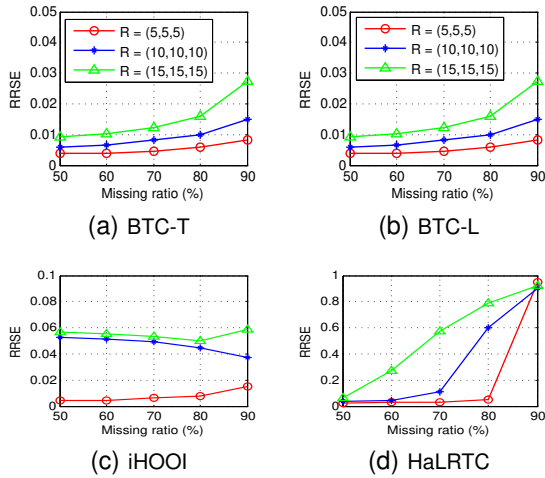


Fig. 4. Tensor completion performances obtained by different methods under varying conditions of MR and true rank. The legend for all figures are same.

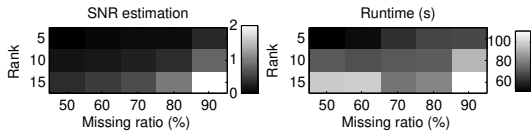


Fig. 5. Absolute error of inferred SNR whose maximum is 2dB when $R = (15, 15, 15)$ and MR = 90%. The runtime is proportional to the true rank.

Enzymatic Activity ($3 \times 3 \times 3 \times 3 \times 5$). For Amino Acids Fluorescence data, each sample contains different amounts of tyrosine, tryptophan and phenylalanine dissolved in phosphate buffered water and measured by fluorescence on a PE LS50B spectrofluorometer. Although each sample is represented as a full rank matrix, leading to a full rank tensor. Ideally it should be describable by three latent components corresponding to individual amino acids.

We tested BTC-T/L methods on the original data as well as noise corrupted data (SNR=0dB). Since the standard Tucker decomposition (HOOI) needs a predefined tensor rank, it can be specified as the inferred rank from BTC-T. As shown in Table 1, tensor rank inferred by BTC-T is (3, 3, 3), indicating that the underlying components associated to three amino acids are successfully captured. In addition, BTC-T yields a stable rank estimation even for highly noisy data. The performances of BTC-T/L are similar and even better than HOOI in most cases, although the same rank was employed. As compared to ARD-Tucker, BTC-T/L always obtain better performance and are more robust to noise. The computation efficiency of BTC-T/L is also significantly higher than ARD-Tucker.

We also tested BTC-T/L methods by randomly selecting 90% data for training, and the remainder was used as test case. We repeated this partition 10 times. Table 2 presents predictive performances and stability of inferred rank. Note that HaLRTC is sensitive to tuning parameters, and the best possible performances by searching tuning parameters in a wide range are reported. iHOOI is able to learn tensor rank automatically, however, the inferred rank is unstable and sensitive to different partitions. In contrast, BTC-

TABLE 4
Global tensor completion for MRI data.

	50%		80%	
	PSNR	RRSE	PSNR	RRSE
BTC-T	26.40	0.12	22.33	0.19
WTucker	19.42	0.27	21.53	0.21
iHOOI	21.57	0.21	19.77	0.26
HaLRTC	23.12	0.18	17.06	0.36

T/L achieved the best predictive performance on all five datasets, while the inferred rank is stable under any specific missing ratio.

7.3 MRI data

In this section, we evaluate BTC-T/L methods by MRI data², particularly for the generalization performance when original data does not possess a global low-rank structure. Since the dimensions and rank of MRI data are generally high, predictions of missing voxels from sparse observations and scalability of methods become more challenging. One simple assumption is that a tensor, which has not a globally low-rank structure, can be separated into smaller block tensors without overlapping and some block tensors may have a low-rank structure (locally low-rank). Hence, we can apply tensor completion to block tensors independently, and this strategy is also useful to handling a large-scale tensor. For HaLRTC, the tuning parameters were carefully selected to yield the best possible performances. BTC methods as well as iHOOI can automatically adapt tensor rank to data. These methods were applied to MRI data completion with varying missing ratios and noise conditions. As shown in Table 3, BTC-T/L achieve the best performance under all different conditions. When missing ratio is low (e.g., 50%), HaLRTC outperforms iHOOI, while iHOOI outperforms HaLRTC when missing ratio is larger than 60%. Since high quality of MRI images is required for medical diagnosis, higher missing ratios (e.g. > 90%) are not tested. In addition, we also applied these methods to MRI data globally for comparisons. WTucker was also used for evaluations, while its tensor rank was specified as the inferred rank by BTC method. As shown in Table 4, BTC always outperforms the other methods under different missing ratios. Note that WTucker can possibly obtain better performance than iHOOI and HaLRTC when tensor rank was specified by inferred rank from BTC under 80% missing ratio. From Table 3, 4, we can observe that when missing ratio is large, global tensor completion yields better performance than local tensor completion strategy. However, global tensor completions are limited by data size. The visual quality is shown in Fig. 6 which was obtained by BTC-T method.

8 CONCLUSION

We have proposed a class of Bayesian Tucker models for both tensor decomposition and completion. To modeling structural sparsity, we introduce two group sparsity inducing priors by the hierarchical representations. The model inferences, especially for the non-conjugate Laplace priors, are

². Dataset is available from <http://brainweb.bic.mni.mcgill.ca/brainweb/>

TABLE 1
Chemometrics data analysis by tensor decompositions. The SNR of noisy data is 0dB. R denotes inferred tensor rank. N/A indicates an inapplicable case. Runtime is measured by seconds.

		Amino		Flow		Electronic		Sugar		Enzymatic	
		Original	Noisy	Original	Noisy	Original	Noisy	Original	Noisy	Original	Noisy
HOOI	RRSE	0.0245	0.0960	0.0321	0.0827	0.0328	0.0664	0.0386	0.0839	0.1677	0.2174
ARD-Tucker	R	(4, 3, 3)	(6, 3, 3)	(5, 4, 3)	(12, 48, 49)	(1, 1, 1)	(3, 2, 3)	(5, 8, 7)	(5, 8, 7)	N/A	N/A
	RRSE	0.0260	0.0961	0.0476	0.4377	0.0328	0.1006	0.0387	0.0788	N/A	N/A
	Runtime	32	38	29	275	22	49	138	145	N/A	N/A
BTD-T	R	(3, 3, 3)	(3, 3, 3)	(3, 5, 3)	(3, 4, 3)	(1, 2, 1)	(1, 46, 1)	(6, 24, 6)	(5, 11, 6)	(1,1,1,1,1)	(1,1,1,1,1)
	RRSE	0.0245	0.0950	0.0321	0.0812	0.0328	0.0664	0.0387	0.0788	0.1677	0.2216
	Runtime	2	2	2	2	2	5	15	15	3	8
BTD-L	R	(3, 6, 3)	(3, 3, 3)	(3, 5, 3)	(3, 4, 3)	(1, 1, 1)	(1, 24, 1)	(7, 25, 6)	(5, 11, 6)	(1,1,1,1,1)	(1,1,1,1,2)
	RRSE	0.0222	0.0950	0.0321	0.0812	0.0328	0.0664	0.0353	0.0789	0.1677	0.2209
	Runtime	5	4	2	4	2	5	15	15	8	7

TABLE 2
Chemometrics data completion with 90% missing ratio. Std(R) denotes standard deviation of inferred tensor rank.

	Amino		Flow		Electronic		Sugar		Enzymatic	
	Std(R)	RRSE	Std(R)	RRSE	Std(R)	RRSE	Std(R)	RRSE	Std(R)	RRSE
HaLRTC	N/A	0.35±0.01	N/A	0.12±0.02	N/A	0.06±0.01	N/A	0.18±0.00	N/A	0.68±0.04
iHOOI	(0,51,17)	0.64±0.08	(0,3,0,3,0,3)	0.44±0.32	(5,38,2)	0.04±0.01	(0,2,0)	0.64±0.00	(0,0,0,0,3,0,5)	0.89±0.49
BTC-T	(0,4,0,4,0)	0.03±0.00	(0,5,0,6,0,4)	0.10±0.04	(0,0,0)	0.03±0	(2,1,5,0)	0.07±0.01	(0,0,0,0,0)	0.21±0.03
BTC-L	(0,5,0,4,0)	0.026±0.00	(0,5,0,5,0,3)	0.11±0.04	(0,0,0)	0.04±0	(1,4,0,7,0)	0.07±0.01	(0,4,0,0,0,0)	0.21±0.03

TABLE 3
The performance of MRI completion evaluated by PSNR and RRSE. For noisy MRI, the standard derivation of Gaussian noise is 3% of brightest tissue. MRI tensor is of size $181 \times 217 \times 165$ and each block tensor is of size $50 \times 50 \times 10$.

	50%		60%		70%		80%									
	Original	Noisy	Original	Noisy	Original	Noisy	Original	Noisy								
BTC-T	27.32	0.11	26.18	0.12	25.30	0.14	24.60	0.15	22.81	0.18	22.35	0.19	20.14	0.25	20.00	0.25
BTC-L	26.91	0.11	25.57	0.13	24.84	0.15	23.95	0.16	22.76	0.19	22.09	0.20	20.12	0.25	19.80	0.26
iHOOI	22.69	0.19	21.45	0.22	22.47	0.19	21.16	0.22	21.63	0.21	20.11	0.25	18.65	0.30	17.89	0.32
HaLRTC	24.84	0.15	23.60	0.17	22.35	0.19	21.65	0.21	19.93	0.26	19.55	0.27	17.37	0.34	17.15	0.35

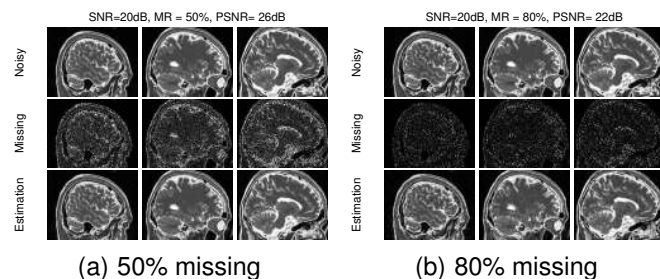


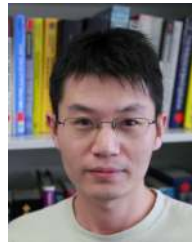
Fig. 6. Visualization of MRI data completion obtained by BTC-T.

derived under variational Bayesian framework. Our models can infer an optimal multilinear rank from whether a fully or partially observed tensor by automatical model reduction, yielding significant advantages for tensor completion. For algorithm implementation, we propose several Theorems related to multilinear operations to improve computational efficiency and scalability. Empirical results on synthetic data as well as chemometrics and neuroscience applications validated the superiority of our models in terms of tensor decomposition and completion.

REFERENCES

- [1] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki, "Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data," *Signal Processing Magazine, IEEE*, vol. 30, no. 4, pp. 137–148, 2013.
- [2] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, "Higher order partial least squares (hopls): A generalized multilinear regression method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 7, pp. 1660–1673, 2013.
- [3] Q. Zhao, L. Zhang, and A. Cichocki, "A tensor-variate gaussian process for classification of multidimensional structured data." in *AAAI*, 2013.
- [4] C. F. Caiafa and A. Cichocki, "Computing sparse representations of multidimensional signals using kronecker bases," *Neural computation*, vol. 25, no. 1, pp. 186–220, 2013.
- [5] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015 (preprint).
- [6] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [7] R. Bro, "PARAFAC. Tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.
- [8] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*. John Wiley & Sons, 2009.
- [9] T. Kolda and B. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [10] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of

- multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.
- [11] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.
- [12] E. S. Allman, P. D. Jarvis, J. A. Rhodes, and J. G. Sumner, "Tensor rank, invariants, inequalities, and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 1014–1045, 2013.
- [13] L.-H. Lim and C. Hillar, "Most tensor problems are NP hard," *University of California, Berkeley*, 2009.
- [14] M. Sørensen, L. De Lathauwer, P. COMON, S. ICART, and L. DENEIRE, "Canonical polyadic decomposition with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, no. accepted, 2012.
- [15] L. Sorber, M. V. Barel, and L. D. Lathauwer, "Tensorlab v1.0." Available online, URL: <http://esat.kuleuven.be/sista/tensorlab>, 2013. [Online]. Available: <http://esat.kuleuven.be/sista/tensorlab/>
- [16] D. Tao, M. Song, X. Li, J. Shen, J. Sun, X. Wu, C. Faloutsos, and S. J. Maybank, "Bayesian tensor approach for 3-d face modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 10, pp. 1397–1410, 2008.
- [17] S. Gao, L. Denoyer, P. Gallinari, and J. GUO, "Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 172–181, 2012.
- [18] M. Mørup and L. K. Hansen, "Automatic relevance determination for multi-way models," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 352–363, 2009.
- [19] T. Yokota and A. Cichocki, "Multilinear tensor rank estimation via sparse Tucker decomposition," in *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*. IEEE, 2014, pp. 478–483.
- [20] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Systems and Signal Processing*, pp. 1–16, 2013.
- [21] W. Chu and Z. Ghahramani, "Probabilistic models for incomplete multi-dimensional arrays," in *JMLR Workshop and Conference Proceedings Volume 5: AISTATS 2009*, vol. 5, 2009, pp. 89–96.
- [22] Z. Xu, F. Yan, and Y. Qi, "Bayesian nonparametric models for multiway data analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 2, pp. 475–487, Feb 2015.
- [23] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proceedings of SIAM Data Mining*, vol. 2010, 2010.
- [24] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.
- [25] K. Hayashi, T. Takenouchi, T. Shibata, Y. Kamiya, D. Kato, K. Kunieda, K. Yamada, and K. Ikeda, "Exponential family tensor factorization for missing-values prediction and anomaly detection," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 216–225.
- [26] P. Jain and S. Oh, "Provable tensor factorization with missing data," in *Advances in Neural Information Processing Systems*, 2014, pp. 1431–1439.
- [27] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable bayesian low-rank decomposition of incomplete multi-way tensors," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1800–1808.
- [28] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [29] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, "Tensor completion via a multi-linear low-n-rank factorization model," *Neurocomputing*, vol. 133, pp. 161–169, 2014.
- [30] H. Wang, F. Nie, and H. Huang, "Low-rank tensor completion with spatio-temporal consistency," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [31] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher-order orthogonal iteration for tensor decomposition and completion," in *Advances in Neural Information Processing Systems*, 2014, pp. 1763–1771.
- [32] Y. Liu, F. Shang, H. Cheng, J. Cheng, and H. Tong, "Factor matrix trace norm minimization for low-rank tensor completion," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, 2014.
- [33] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens, "Learning with tensors: a framework based on convex optimization and spectral regularization," *Machine Learning*, pp. 1–49, 2013.
- [34] D. Kressner, M. Steinlechner, and B. Vandereycken, "Low-rank tensor completion by riemannian optimization," *BIT Numerical Mathematics*, pp. 1–22, 2013.
- [35] Y.-L. Chen, C.-T. Hsu, and H.-Y. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 577–591, 2014.
- [36] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Mining and Knowledge Discovery*, vol. 25, no. 2, pp. 298–324, 2012.
- [37] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
- [38] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.
- [39] C. Bishop, "Variational principal components," in *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, vol. 1. IET, 1999, pp. 509–514.
- [40] M. A. Figueiredo, "Adaptive sparseness for supervised learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1150–1159, 2003.
- [41] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using laplace priors," *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 53–63, 2010.
- [42] S. D. BABACAN, S. NAKAJIMA, and M. N. DO, "Bayesian group-sparse modeling and variational inference," *IEEE transactions on signal processing*, vol. 62, no. 9-12, pp. 2906–2921, 2014.
- [43] J. M. Winn and C. M. Bishop, "Variational message passing," *Journal of Machine Learning Research*, vol. 6, pp. 661–694, 2005.



Qibin Zhao received the Ph.D. degree from Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently a research scientist at Laboratory for Advanced Brain Signal Processing in RIKEN Brain Science Institute, Japan and is also a visiting professor in Saitama Institute of Technology, Japan. His research interests include machine learning, tensor factorization, computer vision and brain computer interface. He has published more than 50 papers in international journals and conferences.



Liqing Zhang received the Ph.D. degree from Zhongshan University, Guangzhou, China, in 1988. He is now a Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests cover computational theory for cortical networks, visual perception and computational cognition, statistical learning and inference. He has published more than 210 papers in international journals and conferences.



Andrzej Cichocki received the Ph.D. and Dr.Sc. (Habilitation) degrees, all in electrical engineering, from Warsaw University of Technology (Poland). He is the senior team leader of the Laboratory for Advanced Brain Signal Processing, at RIKEN BSI (Japan). He is coauthor of more than 400 scientific papers and 4 monographs (two of them translated to Chinese). He served as AE of IEEE Trans. on Signal Processing, TNNLS, Cybernetics and J. of Neuroscience Methods.