Bayesian Tensor Ring Decomposition for Low Rank Tensor Completion

Zerui Tao[1] , **Qibin Zhao**[2*]
[1]School of Mathematics and Statistics, Lanzhou University, Lanzhou, China
[2]Riken Center for Advanced Intelligence Project, Tokyo, Japan
taozr17@lzu.edu.cn, qibin.zhao@riken.jp

Abstract

Recently, tensor network (TN) becomes an attractive topic in the cross discipline of physics and machine learning. By factorizing a higher-order tensor into small tensors, TNs are able to capture complex multi-linear relations within the data. However, in most of the applications, the structures of the TNs are predefined, which greatly limit the flexibility the models in diverse situations. In this paper, we aim to bring the great power of Bayesian learning into tensor ring (TR) decomposition, which is one of the most popular TN structures. The advantages of Bayesian TR model are two-folds: 1). Under the full Bayesian framework, the estimator is probabilistic and robust; 2). With the help of sparse priors, the proposed model can automatically prune redundant factors and infer the underlying structures of the data. To approximate the posterior, we establish two inference algorithms, including Gibbs sampler and variational inference (VI). Also, we conduct experiments on simulation data and image inpainting tasks to show the effectiveness of the proposed model.

1 Introduction

Tensors are natural representations for high dimensional arrays. Contemporarily, many fields encounter tensor representations, as in neuroimaging [Zhou *et al.*, 2013], video processing [Lu *et al.*, 2020], recommender systems [Romera-Paredes *et al.*, 2013], among many others. Despite the great success of traditional tensor decomposition models like Tucker decomposition and CP decomposition [Kolda and Bader, 2009], tensor network (TN) [Cichocki *et al.*, 2016] becomes a powerful tool to tackle with higher-order tensors, due to its powerful ability to capture complex correlations in high dimensional data.

The basic idea of TNs is to factorize large tensors into contraction forms of small core tensors. Among many sophisticated structures of TNs, tensor train (TT) [Oseledets, 2011] and tensor ring (TR) [Zhao *et al.*, 2016] format are proved to be highly expressive in many machine learning applications, e.g., tensor completion [Wang *et al.*, 2017], deep learning [Novikov *et al.*, 2015], Gaussian process [Izmailov *et al.*, 2018] and so on. The main difference between TT and TR is that the first and last TT-rank must be 1, while there is no such restriction in TR. Hence, TR can be regarded as an extension of TT. While factorizing the large tensor data, one critical issue that affects the performance of TNs is the tensor rank, e.g., TT/TR-ranks. Most of the existence literature regards the tensor ranks as a prior knowledge. Nevertheless, in real applications, the underlying signals are unknown and the tensor ranks should be carefully tuned using techniques like cross-validation. Moreover, factorizing an order-D tensor requires to specify a TT/TR-rank of length D, which is hard to search in the parameter space. To alleviate the heavy computation, many works assume the TT/TR ranks are the same in all the D modes, e.g., [Wang *et al.*, 2017; Yuan *et al.*, 2018]. This compromise seems to constrain the expressive ability of TNs, since the data structure may be different varying the modes.

One important application of tensor decomposition models is the low rank tensor completion problem. It is believed that the underlying true signal has low rank and can be approximated by tensor decompositions. To get the low rank estimators, there are two strategies. The first is to apply tensor nuclear norm regularizations [Liu *et al.*, 2012]. However, such methods usually suffer from heavy computation. Another is to use tensor decompositions. Traditional algorithms include CP-WOPT [Acar *et al.*, 2011], Tucker-WOPT [Filipović and Jukić, 2015]. Recently, TR-based methods like TR-ALS [Wang *et al.*, 2017] and TR-WOPT [Yuan *et al.*, 2018] show their expressive power in completion problems. However, all the model above need to specify a tensor rank in advance. To alleviate the problem, one possible choice is rank adaption techniques [Grasedyck and Krämer, 2019]. However, such methods did not consider the distributions of the tensors and the noises, which may fail in low signal-to-noise ratio cases.

In this paper, we aim to enhance the traditional TR decomposition models with Bayesian techniques. To address the rank selection issue, we construct full Bayesian version of TR decomposition. By adopting shrinkage priors, the proposed model can automatically shrink the redundant factors to zeros and select the optimal TR ranks for better perfor-

*Contact Author

mance. Sparse Bayesian learning has been widely studied in factor analysis, e.g., ARD model [Tipping, 2001], sparse factor analysis [Bhattacharya and Dunson, 2011]. In the tensor discipline, the Bayesian CP factorization (BCPF) [Zhao *et al.*, 2015] applied ARD prior on CP decomposition. To inference the posterior, we develop efficient MCMC and VI algorithms.

The rest of the paper is summarized as follows. In Section 2, we provide notations and some basic preliminaries. In Section 3, we present the details of our model and develop Gibbs sampler to inference the posterior. Section 4 gives our algorithms. Experiment results are shown in Section 5.

2 Notations and Preliminaries

2.1 Notations

Tensors are denoted by bold calligraphic letters, e.g., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$. Matrices are denoted by bold capital letters, e.g., $\boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2}$. Vectors are denoted by bold letters, e.g., $\boldsymbol{x} \in \mathbb{R}^I$. For indexing, we denote $\mathbf{i} = [i_1, \ldots, i_D]$ and $\mathbf{i}_{-d} = [i_1, \ldots, i_{d-1}, i_{d+1}, \ldots, i_D]$. We use lowercase letters to denote scalars and subscripts to denote the elements, e.g., $x_{\mathbf{i}}$ is the $\mathbf{i}$-th element of $\boldsymbol{\mathcal{X}}$. Capital calligraphic letters represent distributions. In specific, we denote normal distribution, matrix normal distribution, Gamma distribution with rate parameter as $\mathcal{N}, \mathcal{MN}, \Gamma$, respectively. Notation $\otimes$ denotes Kronecker product and $*$ denotes Hadamard product.

2.2 Tensor Ring Format

Given an order-D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$, the TR format is

$$\boldsymbol{\mathcal{X}} = \ll \boldsymbol{\mathcal{G}}^{(1)}, \ldots, \boldsymbol{\mathcal{G}}^{(D)} \gg,$$

where $\boldsymbol{\mathcal{G}}^{(d)} \in \mathbb{R}^{R_d \times I_d \times R_{d+1}}, \forall d = 1, \ldots, D$ are core tensors and $R_{D+1} = R_1$. Each element of the full tensor $\boldsymbol{\mathcal{X}}$ can be expressed as matrix product of the core tensors, namly,

$$x_{\mathbf{i}} = \mathrm{tr}(\boldsymbol{G}^{(1)}[i_d] \ldots \boldsymbol{G}^{(D)}[i_D]),$$

where $\boldsymbol{G}^{(d)}[i_d] \in \mathbb{R}^{R_d \times R_{d+1}}$ are lateral slices of the core tensors. Hence, TR is also referred to as matrix product state (MPS) with periodic boundary conditions, in physics community [Perez-Garcia *et al.*, 2007].

The left subchain $\boldsymbol{\mathcal{G}}^{<d} \in \mathbb{R}^{R_1 \times \prod_{j=1}^{d-1} I_j \times R_d}$ of TR is defined by tensor contraction among a subsequence of core tensors, whose lateral slices are defined as

$$\boldsymbol{G}^{<d}[\overline{i_1 \ldots i_{d-1}}] = \prod_{j=1}^{d-1} \boldsymbol{G}^{(j)}[i_j].$$

Similarly, we denote the right subchain $\boldsymbol{\mathcal{G}}^{>d} \in \mathbb{R}^{R_{d+1} \times \prod_{j=d+1}^{D} I_j \times R_1}$ and $\boldsymbol{\mathcal{G}}^{\neq d} \in \mathbb{R}^{R_{d+1} \times \prod_{j=1, j\neq d}^{D} I_j \times R_d}$. For more details about the TR format, we refer to [Zhao *et al.*, 2016].

2.3 Multi-way Shrinkage Prior

Shrinkage priors are widely used to induce sparse factors in Bayesian learning. Popular sparse inducing priors include the automatic relevance determination (ARD) [Tipping, 2001], the horseshoe [Carvalho *et al.*, 2010] and so on. These traditional shrinkage priors are designed for vector factors. However, in the TR format, factors admit matrix form, i.e., $\boldsymbol{G}^{(d)}[i_d]$. Hence, we extend the ARD prior to multi-way scenario.

Firstly, we introduce the matrix normal distribution.

Definition 1. *For a matrix $\boldsymbol{X}$ of shape $n \times p$, the matrix normal distribution is denoted as $\boldsymbol{X} \sim \mathcal{MN}(\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$. And the PDF is*

$$\begin{aligned} &p(\boldsymbol{X}|\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}) \\ =&\frac{\exp(-\frac{1}{2}\mathrm{tr}(\boldsymbol{V}^{-1}(\boldsymbol{X}-\boldsymbol{M})^\intercal \boldsymbol{U}^{-1}(\boldsymbol{X}-\boldsymbol{M})))}{(2\pi)^{np/2}|\boldsymbol{V}|^{n/2}|\boldsymbol{U}|^{p/2}}. \end{aligned}$$

If $\boldsymbol{X} \sim \mathcal{MN}(\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$, we have $\mathrm{vec}(\boldsymbol{X}) \sim \mathcal{N}(\mathrm{vec}(\boldsymbol{M}), \boldsymbol{V} \otimes \boldsymbol{U})$. Hence, covariance matrix $\boldsymbol{V}$ and $\boldsymbol{U}$ control the variance of columns and rows, respectively. To apply the ARD prior to matrix form, we assume $\boldsymbol{U}$ and $\boldsymbol{V}$ are diagonal and the elements follow Gamma distribution, namly,

$$\begin{aligned} \boldsymbol{U} &= \mathrm{diag}(\boldsymbol{u}), \quad u_i \sim \Gamma(a_0^u, b_0^u), \\ \boldsymbol{V} &= \mathrm{diag}(\boldsymbol{v}), \quad v_j \sim \Gamma(a_0^v, b_0^v), \end{aligned}$$

for $i = 1, \ldots, n, j = 1, \ldots, p$.

3 Bayesian Tensor Ring Decomposition

In this section, we introduce the basic setting of the Bayesian tensor ring decomposition (BTRD) model.

Suppose we have an order-D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ and partially observed tensor $\boldsymbol{\mathcal{Y}}$, corrupted by noise tensor $\boldsymbol{\mathcal{W}}$, namely,

$$\boldsymbol{\mathcal{Y}}_\Omega = \boldsymbol{\mathcal{X}}_\Omega + \boldsymbol{\mathcal{W}}_\Omega,$$

where Ω denotes observed indexes. To tackle internal relationships, we assume that the underlying tensor admits TR format,

$$\boldsymbol{\mathcal{X}} = \ll \boldsymbol{\mathcal{G}}^{(1)}, \ldots, \boldsymbol{\mathcal{G}}^{(D)} \gg .$$

Firstly, to explore the Bayesian framework for tensor ring, we assume the conditional distribution of the observations is Gaussian, namely,

$$\begin{aligned} &p(\boldsymbol{\mathcal{Y}}_\Omega | \{\boldsymbol{\mathcal{G}}^{(i)}\}_i, \tau) = \\ &\prod_{i_1=1}^{I_1} \cdots \prod_{i_D=1}^{I_D} \mathcal{N}(y_{\mathbf{i}} | \mathrm{tr}(\boldsymbol{G}^{(1)}[i_1] \cdots \boldsymbol{G}^{(D)}[i_D]), \tau^{-1})^{o_{\mathbf{i}}}, \end{aligned}$$

where $o_{\mathbf{i}}$ is the $\mathbf{i}$-th element of the mask tensor $\boldsymbol{\mathcal{O}}$. The mask tensor $\boldsymbol{\mathcal{O}}$ has elements of 1 if observed, and 0 otherwise. If $o_{\mathbf{i}} = 0$, the corresponding term has no influence on the conditional distribution.

Then we apply matrix normal distribution prior on the latent factors, in order to induce sparsity in the TR-ranks, namely,

$$\begin{aligned} &p(\boldsymbol{G}^{(d)}[i_d] | (\boldsymbol{U}^{(d)})^{-1}, (\boldsymbol{U}^{(d+1)})^{-1}) = \\ &\mathcal{MN}(0, (\boldsymbol{U}^{(d)})^{-1}, (\boldsymbol{U}^{(d+1)})^{-1}), \end{aligned} \tag{1}$$

where $\boldsymbol{U}^{(d)} = \mathrm{diag}(\boldsymbol{u}^{(d)})$ and $\boldsymbol{U}^{(D+1)} = \boldsymbol{U}^{(1)}$.

According to the ARD prior introduced in Sec. 2.3, we adopt the following prior on the factor variance

$$u_i^{(d)} \sim \Gamma(a_0, b_0),$$

for $i = 1, \dots, R_d$. Finally, we suppose τ follows Gamma distribution, i.e., $\tau \sim \Gamma(c_0, d_0)$. Notice that all the distributions in our model are conjugate exponential families. Hence it is convenient for us to establish effective inference algorithms.

We summarize the BTRD model as follows

$$\begin{aligned}
\boldsymbol{\mathcal{Y}}_\Omega|\{\boldsymbol{\mathcal{G}}^{(i)}\}_i, \tau \sim \\
\prod_{i_1=1}^{I_1} \cdots \prod_{i_D=1}^{I_D} \mathcal{N}(y_{\mathbf{i}}|\mathrm{tr}(\boldsymbol{G}^{(1)}[i_1] \cdots \boldsymbol{G}^{(D)}[i_D]), \tau^{-1})^{\mathcal{O}_{\mathbf{i}}}, \\
\boldsymbol{G}^{(d)}[i_d]|\boldsymbol{U}^{(d)}, \boldsymbol{U}^{(d+1)} \sim \mathcal{MN}(0, \boldsymbol{U}^{(d)}, \boldsymbol{U}^{(d+1)}), \\
\boldsymbol{u}_i^{(d)} \sim \Gamma(a_0, b_0), \\
\tau \sim \Gamma(c_0, d_0),
\end{aligned}$$

for $d = 1, \dots, D$.

4 Approximate the Posterior

The largest challenge for Bayesian models is to compute the posterior. In most of the models, the posteriors are intractable. To this end, many approximation inference algorithms have been proposed, e.g., approximate message passing (AMP), Markov chain Monte Carlo (MCMC) and variational inference (VI). Benefiting from the conjugacy of our model, we can develop inference algorithms with high efficiency, based on MCMC and VI.

Here, we denote the whole parameters set as $\Theta = \{\boldsymbol{\mathcal{G}}^{(1)}, \dots, \boldsymbol{\mathcal{G}}^{(D)}, \boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(D)}, \tau\}$. Then the joint distribution is

$$\begin{aligned}
p(\boldsymbol{\mathcal{Y}}_\Omega, \Theta) = p(\boldsymbol{\mathcal{Y}}_\Omega|\{\boldsymbol{\mathcal{G}}^{(d)}\}_{d=1}^D, \tau^{-1}) \cdot \prod_{d=1}^D p(\boldsymbol{u}^{(d)}) \cdot p(\tau) \\
\cdot \prod_{d=1}^D \prod_{i_d=1}^{I_d} p(\boldsymbol{G}^{(d)}[i_d]|(\boldsymbol{U}^{(d)})^{-1}, (\boldsymbol{U}^{(d+1)})^{-1}).
\end{aligned} \tag{2}$$

4.1 Gibbs Sampler

In this section, we briefly introduce the Gibbs sampler to estimate the posterior.

Gibbs sampler is one of the most popular MCMC methods, due to its simplicity. It is designed for scenarios where the posterior is intractable but the conditional posteriors have simple forms. The basic idea is to sample the parameters from its conditional posteriors circularly, which is analogous to the alternating least square algorithms. For the BTRD model, the conditional posteriors are summarized as follows.

Sample the factor variance $\{\boldsymbol{u}^{(d)}\}_{d=1}^D$. The conditional distribution is

$$\begin{aligned}
\log p(\boldsymbol{u}^{(d)}|-) &\propto \log p(\boldsymbol{\mathcal{Y}}_\Omega, \Theta) \\
&= \prod_{r_d=1}^{R_d} \Gamma(a_{r_d}^d, b_{r_d}^d),
\end{aligned}$$

where

$$\begin{aligned}
a_{r_d}^d =& a_0 + \frac{I_d R_{d+1} + I_{d-1} R_{d-1}}{2}, \\
b_{r_d}^d =& b_0 + \frac{1}{2} \boldsymbol{u}^{(d-1),\intercal} \sum_{i=1}^{I_{d-1}} \boldsymbol{g}_{\cdot r_d}^{(d-1)}[i] * \boldsymbol{g}_{\cdot r_d}^{(d-1)}[i] \\
&+ \frac{1}{2} \boldsymbol{u}^{(d+1),\intercal} \sum_{i=1}^{I_d} \boldsymbol{g}_{r_d \cdot}^{(d)}[i] * \boldsymbol{g}_{r_d \cdot}^{(d)}[i].
\end{aligned}$$

Sample the core tensors $\{\boldsymbol{G}^{(d)}\}_{d=1}^D$. The conditional distribution is

$$\begin{aligned}
&\log p(\boldsymbol{G}^{(d)}[i_d]|-) \\
\propto& \log p(\boldsymbol{\mathcal{Y}}_\Omega, \Theta), \\
=& \mathcal{MN}(\mathrm{vec}(\boldsymbol{G}^{(d)}[i_d])|\mathrm{vec}(\tilde{\boldsymbol{G}})^{(d)}[i_d], \tilde{\boldsymbol{V}}^{(d)}[i_d]),
\end{aligned}$$

where

$$\begin{aligned}
\mathrm{vec}(\tilde{\boldsymbol{G}})^{(d)}[i_d] =& \tau \tilde{\boldsymbol{V}}^{(d)}[i_d] \sum_{\mathbf{i}_{-d} \in \Omega_d[i_d]} y_{\mathbf{i}_{-d}} \mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal), \\
(\tilde{\boldsymbol{V}}^{(d)}[i_d])^{-1} =& \tau \sum_{\mathbf{i}_{-d} \in \Omega_d[i_d]} \mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal) \mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)^\intercal \\
&+ \boldsymbol{U}^{(d+1)} \otimes \boldsymbol{U}^{(d)},
\end{aligned}$$

where $\Omega_d[i_d]$ is the set of all observed index $[\dots, i_d, \dots]$.

Sample the noise level τ. The posterior follows $p(\tau|-) = \Gamma(c^\tau, d^\tau)$, where $c^\tau = c_0 + \frac{1}{2}\sum_{\mathbf{i} \in \Omega} o_{\mathbf{i}}$ and $d^\tau = d_0 + \frac{1}{2}\sum_{\mathbf{i} \in \Omega}(y_{\mathbf{i}} - \hat{y}_{\mathbf{i}})^2$.

4.2 Variational Inference

In real applications, MCMC algorithms usually take long time to converge and is computationally infeasible. VI provides another solution for the posterior and has much faster convergence rate. In this subsection, we study the VI procedure for the BTRD model. The VI uses a family of variational distributions to approximate the true posterior, denoted as $q(\Theta)$. Then the objective is to find the optimal variational distribution $q^*(\Theta)$, which has the minimum Kullback–Leibler divergence with the true posterior. However, the variational distribution is still very hard to optimize. The solution is the elegant mean-field approximation. Adopting the mean-field approximation, the variational distribution can be factorized as follows,

$$q(\Theta) = q(\tau) \prod_{d=1}^D q(\boldsymbol{u}^{(d)}) \prod_{d=1}^D \prod_{i_d=1}^{I_D} q(\boldsymbol{G}^{(d)}[i_d]).$$

According to the coordinate ascent variational inference (CAVI) algorithm [Bishop, 2006], the optimal solution is

$$\ln q_j^*(\Theta_j) = \langle \ln p(\boldsymbol{\mathcal{Y}}_\Omega, \Theta) \rangle_{q(\Theta \setminus \Theta_j)} + const, \tag{3}$$

where $const$ is some normalization constant and $\langle \cdot \rangle_{q(\Theta \setminus \Theta_j)}$ represents expectation w.r.t. distribution $q(\Theta \setminus \Theta_j)$. Due to the conjugacy of our model, all the variational distributions are tractable and we summarize as follows.

Variational posterior of the core tensors $\{\boldsymbol{G}^{(d)}\}_{d=1}^{D}$. To inference the core tensor, we circlically inference the factors, namely,

$$\ln q^*(\boldsymbol{G}^{(d)}[i_d]) = \langle \ln p(\boldsymbol{\mathcal{Y}}_\Omega, \{\boldsymbol{\mathcal{G}}^{(d)}\}, \{\boldsymbol{U}^{(d)}\}, \tau)\rangle_{\Theta\setminus G^{(d)}[i_d]}.$$

Hence, the posterior is multivariate normal distribution, namely,

$$q(\boldsymbol{G}^{(d)}[i_d]) \sim \mathcal{N}(\mathrm{vec}(\boldsymbol{G}^{(d)}[i_d])|\mathrm{vec}(\tilde{\boldsymbol{G}}^{d}[i_d]), \tilde{\boldsymbol{V}}^{(d)}[i_d]), \quad (4)$$

where

$$\begin{aligned}
&\mathrm{vec}(\tilde{\boldsymbol{G}}^{(d)}[i_d]) = \\
&\langle\tau\rangle\tilde{\boldsymbol{V}}^{(d)}[i_d]\left\langle \sum_{\mathbf{i}_{-d}\in\Omega_d[i_d]} y_{\mathbf{i}_{-d}}\mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)\right\rangle, \\
&(\tilde{\boldsymbol{V}}^{(d)}[i_d])^{-1} = \\
&\langle\tau\rangle\left\langle \sum_{\mathbf{i}_{-d}\in\Omega} \mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)\mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)^\intercal\right\rangle \\
&+\langle \boldsymbol{U}^{(d+1)}\otimes\boldsymbol{U}^{(d)}\rangle.
\end{aligned}$$

Without inducing ambiguities, we dismiss the subscripts of the expectations terms $\langle\cdot\rangle$ for simplicity.

Thanks for the independence assumption in mean-field approximation, most of the expectation terms above are easy to compute. The most difficult part is to compute the expectation of the outer-product of the subchains. it involves square terms which are influenced by the variance $\tilde{\boldsymbol{V}}$.

Consider one core tensor $\boldsymbol{\mathcal{G}}^{(d)}$ which follows distribution given in Eq. (4). We denote its outer product as

$$\boldsymbol{\mathcal{A}}^{(d)} = \boldsymbol{\mathcal{G}}^{(d)}\circ\boldsymbol{\mathcal{G}}^{(d)} \in \mathbb{R}^{I_d\times R_d\times R_{(d+1)}\times R_d\times R_{(d+1)}},$$

for $d = 1,\ldots,D$, where each element of $\boldsymbol{\mathcal{A}}^{(d)}$ is defined as

$$a^{(d)}_{i_d lkmn} = g^{(d)}_{lk}[i_d]\cdot g^{(d)}_{mn}[i_d],$$

for $i_d = 1,\ldots,I_d$, $l,m = 1,\ldots,R_d$ and $k,n = 1,\ldots,R_{d+1}$. Moreover, we reshape the covariance matrices into tensors, like,

$$\tilde{\boldsymbol{\mathcal{V}}}^{(d)}[i_d] = \mathrm{reshape}(\tilde{\boldsymbol{V}}^{(d)}[i_d], R_d, R_{d+1}, R_d, R_{d+1}).$$

Then we can compute the expectation of $\boldsymbol{\mathcal{A}}^{(d)}$ as

$$\langle\boldsymbol{\mathcal{A}}^{(d)}[i_d]\rangle = \tilde{\boldsymbol{G}}^{(d)}[i_d]\circ\tilde{\boldsymbol{G}}^{(d)}[i_d] + \tilde{\boldsymbol{\mathcal{V}}}^{(d)}[i_d], \quad (5)$$

where $\tilde{\boldsymbol{G}}^{(d)}[i_d]$ is defined in Eq. (4).

Now we denote

$$\boldsymbol{B}^{(d)}[\overline{\mathbf{i}_{-d}}] = \mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)\mathrm{vec}((\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal)^\intercal,$$

and its tensorized form

$$\boldsymbol{\mathcal{B}}^{(d)}[\overline{\mathbf{i}_{-d}}] = \mathrm{reshape}(\boldsymbol{B}^{(d)}[\overline{\mathbf{i}_{-d}}], R_d, R_{d+1}, R_d, R_{d+1}).$$

We have the following relationship

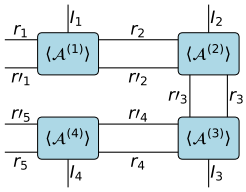
$$\boldsymbol{\mathcal{B}}^{(d)}[\overline{\mathbf{i}_{-d}}] = (\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal\circ(\boldsymbol{G}^{\neq d}[\overline{\mathbf{i}_{-d}}])^\intercal.$$

Using the Einstein's summation notation, we can simplify the expectation as

$$\begin{aligned}
&\langle\boldsymbol{\mathcal{B}}^{(d)}_{\mathbf{i}_{-d}lkmn}\rangle \\
=&\langle\boldsymbol{\mathcal{A}}^{(d-1)}_{i_{d-1}r_{d-1}lr'_{d-1}m}\rangle\langle\boldsymbol{\mathcal{A}}^{(d-2)}_{i_{d-2}r_{d-2}r_{d-1}r'_{d-2}r'_{d-1}}\rangle\cdots \\
&\langle\boldsymbol{\mathcal{A}}^{(1)}_{i_1r_1r_2r'_1r'_2}\rangle\langle\boldsymbol{\mathcal{A}}^{(D)}_{i_Dr_Dr_1r'_Dr'_1}\rangle\cdots\langle\boldsymbol{\mathcal{A}}^{(d+1)}_{i_{d+1}kr_{d+2}nr'_{d+2}}\rangle,
\end{aligned} \quad (6)$$

where the same subscripts represent the contraction indexes. The expectation of $\boldsymbol{\mathcal{A}}$ can be computed according to Eq. (5).

To illustrate Eq. (6), we take an order-5 tensor as an example. Using the tensor network diagrams [Cichocki *et al.*, 2016], when $d = 5$, Eq. (6) can be represented as

$\langle\boldsymbol{\mathcal{B}}^{(5)}\rangle =$  . (7)

Variational posterior of the factor variance $\{\boldsymbol{u}^{(d)}\}_{d=1}^{D}$. The variational posterior of $\boldsymbol{u}^{(d)}$ is

$$\begin{aligned}
\ln q(\boldsymbol{u}^{(d)}) =&\langle \ln p(\boldsymbol{\mathcal{Y}}_\Omega, \{\boldsymbol{\mathcal{G}}^{(d)}\}, \{\boldsymbol{U}^{(d)}\}, \tau)\rangle_{\Theta\setminus\boldsymbol{u}^{(d)}} + const, \\
=&\prod_{r_d=1}^{R_d}\Gamma(u^{(d)}_{r_d}|a^d_{r_d}, b^d_{r_d}),
\end{aligned}$$

where

$$\begin{aligned}
a^d_{r_d} &= a_0 + \frac{I_dR_{d+1} + I_{d-1}R_{d-1}}{2}, \\
b^d_{r_d} &= b_0 + \frac{1}{2}\langle(\boldsymbol{u}^{(d-1)})^\intercal\rangle\left\langle\sum_{i=1}^{I_{d-1}}\boldsymbol{g}^{(d-1)}_{\cdot r_d}[i]*\boldsymbol{g}^{(d-1)}_{\cdot r_d}[i]\right\rangle \\
&+\frac{1}{2}\langle(\boldsymbol{u}^{(d+1)})^\intercal\rangle\left\langle\sum_{i=1}^{I_d}\boldsymbol{g}^{(d)}_{r_d\cdot}[i]*\boldsymbol{g}^{(d)}_{r_d\cdot}[i]\right\rangle.
\end{aligned}$$

The expectations can be computed as follow

$$\begin{aligned}
&\left\langle\sum_{i=1}^{I_{d-1}}\boldsymbol{g}^{(d-1)}_{\cdot r_d}[i]*g^{(d-1)}_{\cdot r_d}[i]\right\rangle \\
=&\sum_{i=1}^{I_{d-1}}\left[\tilde{\boldsymbol{g}}^{(d-1)}_{\cdot r_d}[i]*\tilde{\boldsymbol{g}}^{(d-1)}_{\cdot r_d}[i] + \mathrm{diag}(\tilde{\boldsymbol{V}}^{(d)}_{r_d\cdot r_d\cdot}[i_d])\right].
\end{aligned}$$

Variational posterior of the noise level τ. Similarly, the variational posterior of the noise level τ is

$$\begin{aligned}
\ln q(\tau) =&\langle \ln p(\boldsymbol{\mathcal{Y}}_\Omega, \{\boldsymbol{\mathcal{G}}^{(d)}\}, \{\boldsymbol{U}^{(d)}\}, \tau)\rangle_{q(\Theta\setminus\tau)} + const \\
=&\Gamma(\tau|c^\tau, d^\tau),
\end{aligned}$$

where

$$\begin{aligned}
c^\tau &= c_0 + \frac{1}{2}\sum_{\mathbf{i}\in\Omega}\boldsymbol{\mathcal{O}}_{\mathbf{i}}, \\
d^\tau &= d_0 + \frac{1}{2}\left\langle||\boldsymbol{\mathcal{O}}*(\boldsymbol{\mathcal{Y}} - \ll\boldsymbol{\mathcal{G}}^{(1)},\ldots,\boldsymbol{\mathcal{G}}^{(D)}\gg)||_F^2\right\rangle.
\end{aligned}$$

Again the difficulty is to compute the expectation term. We have

$$\begin{aligned}&\langle||\boldsymbol{\mathcal{O}} * (\boldsymbol{\mathcal{Y}} - \ll \boldsymbol{\mathcal{G}}^{(1)}, \ldots, \boldsymbol{\mathcal{G}}^{(D)} \gg)||_F^2\rangle \\ =&||\boldsymbol{\mathcal{Y}}_\Omega||_F^2 - 2\text{vec}(\boldsymbol{\mathcal{Y}}_\Omega)^T \text{vec}(\ll \tilde{\boldsymbol{\mathcal{G}}}^{(1)}, \ldots, \tilde{\boldsymbol{\mathcal{G}}}^{(D)} \gg_\Omega) \\ &+ \langle|| \ll \boldsymbol{\mathcal{G}}^{(1)}, \ldots, \boldsymbol{\mathcal{G}}^{(D)} \gg_\Omega ||_F^2\rangle.\end{aligned}$$

Similar with the inference of core tensor, we use Einstein's notation to simplify the equations. For each element $\hat{y}_{\mathbf{i}} = \text{tr}(\boldsymbol{G}^{(1)}[i_1] \cdots \boldsymbol{G}^{(D)}[i_D])$, we have

$$\langle \hat{y}_{\mathbf{i}} \rangle = \langle \boldsymbol{\mathcal{A}}^{(1)}_{r_1 r_2 r_1' r_2'}[i_1] \rangle \cdots \langle \boldsymbol{\mathcal{A}}^{(D)}_{r_D r_1 r_D' r_1'}[i_D] \rangle, \tag{8}$$

where the expectation of $\boldsymbol{\mathcal{A}}$ is again computed through Eq. (5).

4.3 Some Details

Hyperparameters. For most of applications, it is likely that we do not have any information about the data. Hence, we adopt the non-informative prior. Specifically, we set all hyperparameters a_0, b_0, c_0, d_0 as a very small number, e.g., $1\text{e}{-6}$, to induce as less influence on the posterior as possible.

Initialization. In TN-based model, a good initialization point usually results in much faster convergence [Stoudenmire and Schwab, 2016]. For the core tensors, we use the tensor ring approximation (TRA) initialization method described in [Wang *et al.*, 2017], which factorizes the observed tensor as initializations. For $\boldsymbol{u}$ and τ, we simply set them as 1.

Pruning Factors. One main advantage of our model is to inference the true rank of underlying signals. The basic idea is setting a relatively large initialization rank. Due to the sparsity inducing priors, many factors becomes zero during the inference procedure and can be discarded. However, large ranks may cause heavy computational burden. In practice, we truncate a factor if its Frobenius norm is smaller than a small constant, e.g., $1\text{e}{-3}$. We also observed that it's possible to set a much smaller truncation level and the results are usually similar.

5 Experiments

In this section, we give the experimental results, including simulation study and image inpainting experiments. For the simulation study, we generate synthetic data and test the ability of our model to infer true underlying structures. For the image inpainting part, we compare the performance of our model with several tensor completion models. We test both the Gibbs sampler and VI algorithms, denoted as BTRD-GS and BTRD-VI respectively. All of our experiments are performed on a GNU/Linux workstation with Intel Xeon E5-2690 3.50GHz CPU and 64GB memory.

5.1 Simulation Study

For the simulation study, we artificially generate synthetic data and test the efficiency of the BTRD model.

To illustrate our model, we conduct experiments on an order-4 tensor of shape $10 \times 10 \times 10 \times 10$ with TR-rank $[3, 3, 3, 3]$. To be specific, we independently generate 4 core tensors of shape $3 \times 10 \times 3$ from standard Gaussian distribution, and then compute the true signal $\boldsymbol{\mathcal{X}} = \ll \boldsymbol{\mathcal{G}}^{(1)}, \ldots, \boldsymbol{\mathcal{G}}^{(4)} \gg$. Then we add i.i.d. Gaussian noise $\boldsymbol{\mathcal{W}}$ on the true signal to get the observed signal, i.e., $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{W}}$. To test the robustness of our model under different circumstances, we adopt different missing rate ranging from 0.1 to 0.7 and different signal-to-noise (SNR) levels ranging from -5 to 30. The SNR is defined as $\text{SNR} = ||\boldsymbol{\mathcal{X}}||_F / \sqrt{\prod_d I_d}\sigma$, where σ^2 is the noise variance. We repeat the experiments 20 times and take the average result.

For the synthetic data, we mainly test the ability of our model to estimate the true underlying TR-rank and the noise variance. We set the initialization TR-rank as $[20, 20, 20, 20]$ and calculate the relative error of the true value and the estimated value. The rank estimation results are shown in Fig. 1 and the noise variance estimation results are shown in Fig. 2. These experiments manifest that the BTRD model can effectively inference the underlying data structures.

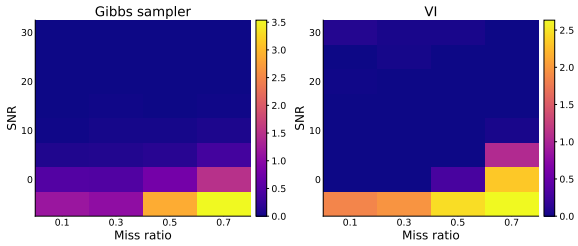


Figure 1: Rank estimation error.

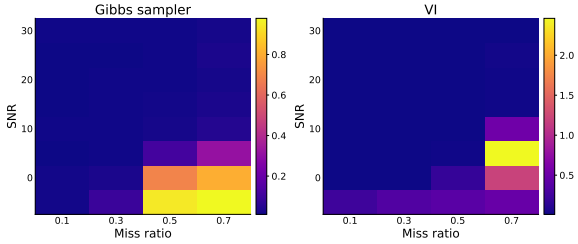


Figure 2: Noise variance estimation error.

5.2 Image Inpainting

In the image inpainting experiments, we compare our proposed model with some state-of-the-art tensor completion algorithms, including BCPF [Zhao *et al.*, 2015], TRALS [Wang *et al.*, 2017], TTWOPT [Yuan *et al.*, 2019b], TRLRF [Yuan *et al.*, 2019a]. The BCPF model also uses Bayesian frame work and requires no hyperparameters. However, for the rest of models, we have to carefully tune the hyperparameters, e.g., the TT/TR-ranks. Here we compute those models under several groups of TT/TR-ranks and select the best performance. However, it should be noted that this is not realistic in many applications where the true signals are unknown.

For our model, we set the initialization TR-rank as $[20, 20, 20]$. For BTRD-GS algorithm, we set 1000 iterations with 200 burn-in. We illustrate the convergence process for

Lena image with missing rate 0.9 in Fig. 3. It shows that the VI algorithm converges much faster than Gibbs sampler.

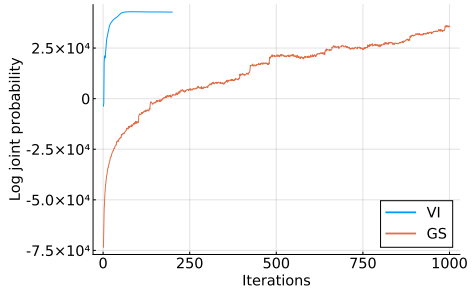


Figure 3: Convergence process for Lena image with missing rate 0.9.

The evaluate the performance, we adopt the relative standard error (RSE) and peak signal-to-noise ratio (PSNR), defined as follows, $\text{RSE} = ||\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}||_F / ||\boldsymbol{\mathcal{X}}||_F$ and $\text{PSNR} = 10\log_{10}(\text{numel}(\boldsymbol{\mathcal{X}})\,\|\boldsymbol{\mathcal{X}}\|_\infty^2 / ||\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}||_F^2)$, where $\text{numel}(\boldsymbol{\mathcal{X}})$ represents the number of elements in $\boldsymbol{\mathcal{X}}$ and $||\boldsymbol{\mathcal{X}}||_\infty$ is the maximum element of $\boldsymbol{\mathcal{X}}$.

We choose 8 pictures to test the performance, shown in Fig. 4. For the completion problem, we randomly generate masks of missing rate 0.5, 0.7 and 0.9.

Figure 4: Original benchmark images.

Fig. 5 illustrates the completion results of the Lena image under different missing rate. The full quantitative results are show in Tab. 1. It shows that our proposed model outperforms others when the missing rate becomes high. However, when the missing rate is 0.5, it results that the BTRD-VI performs better than BTRD-GS. This may due to the slow convergence rate of Gibbs sampler and we do not set enough iterations.

6 Discussions

In this paper, we study the Bayesian tensor ring decomposition and its application in tensor completion. By adopting the sparsity inducing prior, our model can automatically infer the underlying true structures. We develop both Gibbs sampler and VI algorithms to approximate the true posterior. Experiments shows that the proposed VI algorithm has very fast convergence. To test the effectiveness of our model, we conduct both synthetic data experiments and image inpainting experiments. The results confirm the advantages of our model, in inference of the true signals and the completion performance.

References

[Acar *et al.*, 2011] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[Bhattacharya and Dunson, 2011] A. Bhattacharya and D. B. Dunson. Sparse bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[Carvalho *et al.*, 2010] Carlos Marinho Carvalho, Nicholas G. Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[Cichocki *et al.*, 2016] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[Filipović and Jukić, 2015] Marko Filipović and Ante Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing*, 26(3):677–692, Jul 2015.

[Grasedyck and Krämer, 2019] Lars Grasedyck and Sebastian Krämer. Stable als approximation in the tt-format for rank-adaptive tensor completion. *Numerische Mathematik*, 143(4):855–904, 2019.

[Izmailov *et al.*, 2018] Pavel Izmailov, Alexander Novikov, and Dmitry Kropotov. Scalable gaussian processes with billions of inducing inputs via tensor train decomposition. In *International Conference on Artificial Intelligence and Statistics*, pages 726–735, 2018.

[Kolda and Bader, 2009] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[Liu *et al.*, 2012] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2012.

[Lu *et al.*, 2020] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, April 2020.

[Novikov *et al.*, 2015] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pages 442–450, 2015.

(a) Missing (b) BTRD-GS (c) BTRD-VI (d) TRALS (e) TRLRF (f) TTWOPT (g) BCPF

Figure 5: Illustration of Lena image completion. Each row from the above to the bottom according to missing rate 0.5, 0.7 and 0.9 respectively.

Missing	Metric	BTRD-GS	BTRD-VI	TRALS	TRLRF	TTWOPT	BCPF
50%	RSE↓	0.0925	0.0608	0.0738	**0.0555**	0.0783	0.0833
	PSNR↑	26.82	29.8856	28.41	**31.03**	27.70	27.18
70%	RSE↓	**0.0892**	0.0909	0.1133	0.1000	0.1272	0.1113
	PSNR↑	**26.36**	26.1957	24.46	25.81	23.31	24.58
90%	RSE↓	**0.1587**	0.1658	0.5340	0.2286	0.2321	0.1813
	PSNR↑	**21.11**	20.7802	11.17	18.33	17.98	20.24

Table 1: Results for image inpainting.

[Oseledets, 2011] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[Perez-Garcia *et al.*, 2007] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Information & Computation*, 7(5):401–430, 2007.

[Romera-Paredes *et al.*, 2013] Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil. Multilinear multitask learning. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1444–1452, 2013.

[Stoudenmire and Schwab, 2016] Edwin Miles Stoudenmire and David J. Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[Tipping, 2001] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[Wang *et al.*, 2017] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *IEEE International Conference on Computer Vision*, pages 5697–5705, 2017.

[Yuan *et al.*, 2018] Longhao Yuan, Jianting Cao, Xuyang Zhao, Qiang Wu, and Qibin Zhao. Higher-dimension tensor completion via low-rank tensor ring decomposition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1071–1076, 2018.

[Yuan *et al.*, 2019a] Longhao Yuan, Chao Li, Danilo P. Mandic, jianting cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. *AAAI 2019 : Thirty-Third AAAI Conference on Artificial Intelligence*, 33(1):9151–9158, 2019.

[Yuan *et al.*, 2019b] Longhao Yuan, Qibin Zhao, Lihua Gui, and Jianting Cao. High-order tensor completion via gradient-based optimization under tensor train format. *Signal Processing: Image Communication*, 73:53–61, 2019.

[Zhao *et al.*, 2015] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.

[Zhao *et al.*, 2016] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[Zhou *et al.*, 2013] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.