Bayesian CP Factorization of Incomplete Tensors with Automatic Rank Determination

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

Abstract—CANDECOMP/PARAFAC (CP) tensor factorization of incomplete data is a powerful technique for tensor completion through explicitly capturing the multilinear latent factors. The existing CP algorithms require the tensor rank to be manually specified, however, the determination of tensor rank remains a challenging problem especially for *CP rank*. In addition, existing approaches do not take into account uncertainty information of latent factors, as well as missing entries. To address these issues, we formulate CP factorization using a hierarchical probabilistic model and employ a fully Bayesian treatment by incorporating a sparsity-inducing prior over multiple latent factors and the appropriate hyperpriors over all hyperparameters, resulting in automatic rank determination. To learn the model, we develop an efficient deterministic Bayesian inference algorithm, which scales linearly with data size. Our method is characterized as a tuning parameter-free approach, which can effectively infer underlying multilinear factors with a low-rank constraint, while also providing predictive distributions over missing entries. Extensive simulations on synthetic data illustrate the intrinsic capability of our method to recover the ground-truth of *CP rank* and prevent the overfitting problem, even when a large amount of entries are missing. Moreover, the results from real-world applications, including image inpainting and facial image synthesis, demonstrate that our method outperforms state-of-the-art approaches for both tensor factorization and tensor completion in terms of predictive performance.

Index Terms—Tensor factorizations, tensor completion, rank determination, Bayesian inference, image synthesis, inpainting

1 Introduction

Tensors (i.e., multiway arrays) provide an effective and faithful representation of the structural properties of data, in particular, when multidimensional data are involved. For instance, an image ensemble measured under multiple conditions can be represented by a higher order tensor with dimensionality of $pixel \times person \times pose \times illumination$. Tensor factorization enables us to explicitly take into account the structure information by effectively capturing the multilinear interactions among multiple latent factors. Therefore, its theory and algorithms have been an active area of study during the past decade (see e.g., [1], [2]), and have been successfully applied to various application fields, such as face recognition [3], [4], social network analysis, image compression [5], and brain signal processing. The two most popular tensor factorization frameworks are Tucker [6] and CANDECOMP/PARAFAC (CP), also known as canonical polyadic decomposition (CPD) [7], [8], [9].

The problem of missing data can arise in a variety of real-world applications, which has attracted a great deal of research interest in tensor completion in recent years. It can be achieved by either factorization or completion based schemes. Tensor factorization based completion is to infer the underlying factors from partially observed entries based on a multilinear generative model assumption with a fixed rank, which can thus predict missing data. In [10], CP factorization with missing data was formulated as a weighted least squares problem, termed CP weighted optimization (CPWOPT). Some other related methods were also investigated such as structured CPD using nonlinear least squares (CPNLS) [11] and geometric nonlinear conjugate gradient (geomCG) [12]. However, tensor factorization scheme is prone to overfitting due to an incorrect tensor rank and point estimations of latent factors, resulting in severe deterioration of predictive performance. In contrast, completion based scheme exploits an automatic rank optimization and does not make model assumptions, where the rank minimization is formulated as a convex optimization on the matrix nuclear norm. This technique has been extended to tensor completion by defining the nuclear norm of a tensor [13]. Some variants were also proposed under this framework, such as an inexact splitting method [14] and fast composite splitting algorithms (FCSA) [15]. To improve the efficiency, Douglas-Rachford splitting technique [16], nonlinear Gauss-Seidal method [17] were also investigated. Recently, the nuclear norm based optimization was also applied to a supervised tensor dimensionality reduction method [18]. The theoretical bound on the number of observations was studied in [19]. Since completion-based methods cannot explicitly capture the underlying factors, a simultaneous tensor decomposition and completion (STDC) [20] method was introduced in which rank minimization

- *Q. Zhao is with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan and the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. E-mail: qbzhao@brain.riken.jp.*
- *L. Zhang is with the Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: zhang-lq@cs.sjtu.edu.cn.*
- *A. Cichocki is with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan and the Systems Research Institute at the Polish Academy of Science, Warsaw, Poland. E-mail: a.cichocki@riken.jp.*

Manuscript received 28 Aug. 2014; revised 12 Nov. 2104; accepted 29 Dec. 2014. Date of publication 14 Jan. 2015; date of current version 7 Aug. 2015. Recommended for acceptance by D. Xu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2392756

was combined with Tucker decomposition. Furthermore, auxiliary information was also exploited in [20], [21], resulting in a significant improvement on some specific applications. However, nuclear norm of a tensor, defined straightforwardly as a weighted sum of nuclear norm of mode-n matricizations, is related to *multilinear rank* rather than *CP rank*. It is also noteworthy that the rank minimization based on nuclear norm is sensitive to tuning parameters, which may tend to over- or under-estimate the true tensor rank.

It is important to emphasize that our knowledge about the properties of *CP rank*, defined by the minimum number of rank-one terms in CP decomposition, is surprisingly limited. There is no straightforward algorithm to compute the rank even for a given specific tensor, and the problem has been shown to be NP-complete [22]. The lower and upper bound of tensor rank was studied in [23], [24]. The ill-posedness of the best low-rank approximation of a tensor was investigated in [25]. In fact, determining or even bounding the rank of an arbitrary tensor is quite difficult in contrast to the matrix rank [26], and this difficulty would be significantly exacerbated in the presence of missing data.

Probabilistic models for matrix/tensor factorization have attracted much interest in collaborative filtering and matrix/tensor completion. Probabilistic matrix factorization was proposed in [27], and its fully Bayesian treatment using Markow chain Monte Carlo (MCMC) inference was shown in [28] and using variational Bayesian inference in [29], [30]. Further extensions of nonparametric and robust variants were presented in [31], [32]. The probabilistic frameworks of tensor factorization were presented in [33], [34], [35]. Other variants include extensions of the exponential family model [36] and the nonparametric Bayesian model [37]. However, the tensor rank needs to be predefined by a tuning parameter selected by either maximum likelihood or cross-validations, which are computationally expensive and inaccurate. Another important issue is that the inference of factor matrices is performed by either point estimation, which is prone to overfitting, or MCMC inference, which tends to converge very slowly.

To address these issues, we propose a fully Bayesian probabilistic CP factorization model. Our objective is to infer the multilinear factors and the predictive distribution over missing entries given a noisy incomplete tensor, while CP rank of the underlying true tensor can be determined automatically and implicitly. To achieve this, we specify a sparsity-inducing hierarchical prior over multiple factor matrices with individual hyperparameters associated to each latent dimension, such that the number of components in factor matrices is constrained to be minimum. All the model parameters, including noise precision, are considered to be latent variables over which the corresponding priors are placed. Due to complex interactions among multiple factors, full Bayesian inference is analytically intractable. Thus, we derive a deterministic solution to approximate the posteriors of unknowns under the framework of variational Bayesian inference. Our method is characterized as a tuning parameter-free approach that can effectively avoid parameter selections. The extensive experiments and comparisons on synthetic data illustrate the advantages of our approach in terms of rank determination, predictive capability, and robustness to overfitting. Moreover, several real-word applications, including image completion, restoration, and synthesis, demonstrate that our method outperforms state-of-the-art approaches, including both tensor factorization and tensor completion, in terms of the predictive performance.

The rest of this paper is organized as follows. In Section 2, preliminary multilinear operations and notations are presented. In Section 3, we introduce our model specification and its Bayesian inference. An extension of our method using mixture priors is proposed in Section 4. In Section 5, we present the comprehensive experimental results for both synthetic data and real-world applications, followed by our conclusion in Section 6.

2 Preliminaries and Notations

The order of a tensor is the number of dimensions, also known as ways or modes. Vector, matrix and higher-order ($N \geq 3$) tensor are denoted by $\mathbf{a}$, $\mathbf{A}$ and $\mathcal{A}$ respectively. Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its $(i_1, i_2, \ldots, i_N)$th entry is denoted by $\mathcal{X}_{i_1 i_2 \ldots i_N}$, where $i_n = 1, 2, \ldots, I_n, \forall n \in [1, N]$.

The *inner product of two tensors* is defined by $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \ldots, i_N} \mathcal{A}_{i_1 i_2 \ldots i_N} \mathcal{B}_{i_1 i_2 \ldots i_N}$, and the squared Frobenius norm by $\|\mathcal{A}\|_F^2 = \langle \mathcal{A}, \mathcal{A} \rangle$. As an extension to N variables, the *generalized inner product* of a set of vectors, matrices, or tensors is defined as a sum of element-wise products. For example, given $\{\mathbf{A}^{(n)} | n = 1, \ldots, N\}$, we define

$$\left\langle \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \right\rangle = \sum_{i,j} \prod_n A_{ij}^{(n)}. \tag{1}$$

The *Hadamard product* is an entrywise product of two vectors, matrices, or tensors which are of the same sizes. Given $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$, their Hadamard product is $\mathbf{A} \circledast \mathbf{B} \in \mathbb{R}^{I \times J}$. The Hadamard product of a set of matrices can be simply denoted by

$$\circledast_n \mathbf{A}^{(n)} = \mathbf{A}^{(1)} \circledast \mathbf{A}^{(2)} \circledast \cdots \circledast \mathbf{A}^{(N)}. \tag{2}$$

The *Kronecker product* [1] of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is a matrix of size $IK \times JL$, denoted by $\mathbf{A} \otimes \mathbf{B}$. The *Khatri-Rao product* of matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$ is a matrix of size $IJ \times K$, defined by a columnwise Kronecker product and denoted by $\mathbf{A} \odot \mathbf{B}$. In particular, the Khatri-Rao product of a set of matrices in a reverse order is defined by

$$\bigodot_n \mathbf{A}^{(n)} = \mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \cdots \odot \mathbf{A}^{(1)}, \tag{3}$$

while the Khatri-Rao product of a set of matrices, except the nth matrix, denoted by $\mathbf{A}^{(\backslash n)}$, is

$$\bigodot_{k \neq n} \mathbf{A}^{(k)} = \mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}. \tag{4}$$

3 Bayesian Tensor Factorization

3.1 Probabilistic Model and Priors

Let $\mathcal{Y}$ be an incomplete Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$. $\mathcal{Y}_{i_1 i_2 \ldots i_N}$ is observed if $(i_1, i_2, \cdots, i_N) \in \Omega$, where Ω denotes a set of N-tuple indices. For simplicity, we also define a binary tensor $\mathcal{O}$ of the same size as $\mathcal{Y}$ as an indicator of observed entries. We assume $\mathcal{Y}$ is a noisy observation of true latent tensor $\mathcal{X}$, that is, $\mathcal{Y} = \mathcal{X} + \varepsilon$, where the noise term is assumed to be an i.i.d. Gaussian distribution, i.e., $\varepsilon \sim \prod_{i_1,\ldots,i_N} \mathcal{N}(0, \tau^{-1})$, and the latent tensor $\mathcal{X}$ is generated by a CP model, given by

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)} = [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!], \tag{5}$$

where $\circ$ denotes the outer product of vectors and $[\![\cdots]\!]$ is a shorthand notation, also termed as the Kruskal operator. CP factorization can be interpreted as a sum of R rank-one tensors, while the smallest integer R is defined as *CP rank* [1]. $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$ are a set of factor matrices where mode-n factor matrix $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ can be denoted by row-wise or column-wise vectors

$$\mathbf{A}^{(n)} = \left[\mathbf{a}_1^{(n)}, \ldots, \mathbf{a}_{i_n}^{(n)}, \ldots, \mathbf{a}_{I_n}^{(n)}\right]^T = \left[\mathbf{a}_{\cdot 1}^{(n)}, \ldots, \mathbf{a}_{\cdot r}^{(n)}, \ldots, \mathbf{a}_{\cdot R}^{(n)}\right].$$

The CP generative model, together with noise assumption, directly give rise to the observation model, which is factorized over observed tensor elements

$$p\left(\mathcal{Y}_\Omega \middle| \{\mathbf{A}^{(n)}\}_{n=1}^{N}, \tau\right) = \prod_{i_1=1}^{I_1} \cdots \prod_{i_N=1}^{I_N} \mathcal{N}\left(\mathcal{Y}_{i_1 i_2 \ldots i_N} \middle| \left\langle \mathbf{a}_{i_1}^{(1)}, \mathbf{a}_{i_2}^{(2)}, \ldots, \mathbf{a}_{i_N}^{(N)} \right\rangle, \tau^{-1}\right)^{\mathcal{O}_{i_1 \ldots i_N}}, \tag{6}$$

where τ denotes the noise precision, and $\left\langle \mathbf{a}_{i_1}^{(1)}, \mathbf{a}_{i_2}^{(2)}, \ldots, \mathbf{a}_{i_N}^{(N)} \right\rangle$ denotes a generalized inner-product of N vectors. The likelihood model in (6) indicates that $\mathcal{Y}_{i_1 \cdots i_N}$ is generated from multiple R-dimensional latent vectors $\left\{\mathbf{a}_{i_n}^{(n)} \middle| n = 1, \ldots, N\right\}$, where each latent vector $\mathbf{a}_{i_n}^{(n)}$ contributes to a set of observations, i.e., a subtensor whose mode-n index is i_n, such that the multilinear interactions are taken into account by the likelihood function. The essential difference between matrix and tensor factorization is that the inner product of $N \geq 3$ vectors allows us to model the multilinear interaction structure.

In general, the effective dimensionality of the latent space, i.e., $Rank_{CP}(\mathcal{X}) = R$, is a tuning parameter whose selection is quite challenging and computational costly. Therefore, we seek an elegant automatic model selection, which can not only infer the rank of the latent tensor $\mathcal{X}$, but also effectively avoid overfitting. To achieve this, a set of continuous hyperparameters are employed to control the variance related to each dimensionality of the latent space, respectively. Since the minimum R is desired in the sense of low rank approximation, a sparsity-inducing prior is specified over these hyperparameters, resulting in it being possible to achieve automatic rank determination as a part of the Baybesian inference process. This technique is related to automatic relevance determination (ARD) [38] or sparse Bayesian learning [39]. However, unlike the traditional methods that place the ARD prior over either latent variables or weight parameters, such as Bayesian principle component analysis [40], our method considers all model parameters as latent variables over which a sparsity-inducing prior is placed with shared hyperparameters.

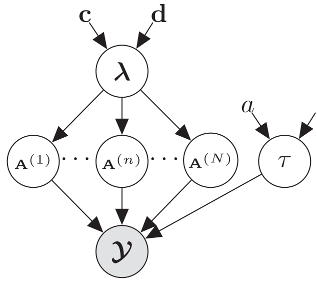


Fig. 1. Probabilistic graphical model of Bayesian CP factorization of an Nth-order tensor.

More specifically, we place a prior distribution over the latent factors, governed by hyperparameters $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R]$ where each λ_r controls rth component in $\mathbf{A}^{(n)}$, which is

$$p\left(\mathbf{A}^{(n)} \middle| \boldsymbol{\lambda}\right) = \prod_{i_n=1}^{I_n} \mathcal{N}\left(\mathbf{a}_{i_n}^{(n)} \middle| \mathbf{0}, \boldsymbol{\Lambda}^{-1}\right), \forall n \in [1, N], \tag{7}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$ denotes the inverse covariance matrix, also known as the precision matrix, and is shared by latent factor matrices in all modes. We can further define a hyperprior over $\boldsymbol{\lambda}$, which is factorized over latent dimensions

$$p(\boldsymbol{\lambda}) = \prod_{r=1}^{R} \mathrm{Ga}(\lambda_r | c_0^r, d_0^r), \tag{8}$$

where $\mathrm{Ga}(x|a,b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)}$ denotes a Gamma distribution.

The initialization point of the dimensionality of latent space (i.e., R) is usually set to its maximum possible value, while the effective dimensionality can be inferred automatically under a Bayesian inference framework. It should be noted that since the priors are shared across N latent matrices, the same sparsity pattern can be obtained, yielding the minimum number of rank-one terms. Therefore, our model can effectively infer the rank of tensor while performing tensor factorization, which can be treated as a *Bayesian low-rank tensor factorization*.

To complete the model, we also place a hyperprior over the noise precision τ, that is,

$$p(\tau) = \mathrm{Ga}(\tau | a_0, b_0). \tag{9}$$

For simplicity of notation, all unknowns including latent variables and hyperparameters are collected and denoted together by $\Theta = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}, \boldsymbol{\lambda}, \tau\}$. The probabilistic graph model is illustrated in Fig. 1, from which we can easily write the joint distribution of the model as

$$p(\mathcal{Y}_\Omega, \Theta) = p\left(\mathcal{Y}_\Omega \middle| \{\mathbf{A}^{(n)}\}_{n=1}^{N}, \tau\right) \prod_{n=1}^{N} p\left(\mathbf{A}^{(n)} \middle| \boldsymbol{\lambda}\right) p(\boldsymbol{\lambda}) p(\tau).$$

By combining the likelihood in (6), the priors of model parameters in (7), and the hyperpriors in (8) and (9), the logarithm of the joint distribution is given by (see Section 1 of Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2392756 available online, for details)

$$\begin{aligned}\ell(\Theta) = &-\frac{\tau}{2}\left\|\mathcal{O}\circledast\left(\mathcal{Y}-[\![\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}]\!]\right)\right\|_F^2\\ &-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Lambda}\sum_n\mathbf{A}^{(n)T}\mathbf{A}^{(n)}\right)+\left(\frac{M}{2}+a_0-1\right)\ln\tau\\ &+\sum_r\left[\left(\frac{\sum_n I_n}{2}+\left(c_0^r-1\right)\right)\ln\lambda_r\right]\\ &-\sum_r d_0^r\,\lambda_r-b_0\tau+\mathrm{const},\end{aligned}\tag{10}$$

where $M=\sum_{i_1,\ldots,i_N}\mathcal{O}_{i_1\ldots i_N}$ denotes the total number of observations. Without loss of generality, we can perform maximum a posteriori (MAP) estimation of Θ by maximizing (10), which is, to some extent, equivalent to optimizing a squared error function with regularizations imposed on the factor matrices and additional constraints imposed on the regularization parameters.

However, our objective is to develop a method that, in contrast to the point estimation, computes the full posterior distribution of all variables in Θ given the observed data, that is,

$$p(\Theta|\mathcal{Y}_\Omega)=\frac{p(\Theta,\mathcal{Y}_\Omega)}{\int p(\Theta,\mathcal{Y}_\Omega)\,d\Theta}.\tag{11}$$

Based on the posterior distribution of Θ, the predictive distribution over missing entries, denoted by $\mathcal{Y}_{\backslash\Omega}$, can be inferred by

$$p(\mathcal{Y}_{\backslash\Omega}|\mathcal{Y}_\Omega)=\int p(\mathcal{Y}_{\backslash\Omega}|\Theta)p(\Theta|\mathcal{Y}_\Omega)\,\mathrm{d}\Theta.\tag{12}$$

3.2 Model Learning via Bayesian Inference

An exact Bayesian inference in (11) and (12) would integrate over all latent variables as well as hyperparameters, which is obviously analytically intractable. In this section, we describe the development of a deterministic approximate inference under variational Bayesian (VB) framework [41] to learn the probabilistic CP factorization model.

We therefore seek a distribution $q(\Theta)$ to approximate the true posterior distribution $p(\Theta|\mathcal{Y}_\Omega)$ by minimizing the KL divergence, that is,

$$\begin{aligned}\mathrm{KL}\big(q(\Theta)\big|\big|p(\Theta|\mathcal{Y}_\Omega)\big)&=\int q(\Theta)\ln\left\{\frac{q(\Theta)}{p(\Theta|\mathcal{Y}_\Omega)}\right\}d\Theta\\ &=\ln p(\mathcal{Y}_\Omega)-\int q(\Theta)\ln\left\{\frac{p(\mathcal{Y}_\Omega,\Theta)}{q(\Theta)}\right\}d\Theta,\end{aligned}\tag{13}$$

where $\ln p(\mathcal{Y}_\Omega)$ represents the model evidence, and its *lower bound* is defined by $\mathcal{L}(q)=\int q(\Theta)\ln\left\{\frac{p(\mathcal{Y}_\Omega,\Theta)}{q(\Theta)}\right\}d\Theta$. Since the model evidence is a constant, the maximum of the lower bound occurs when the KL divergence vanishes, which implies that $q(\Theta)=p(\Theta|\mathcal{Y}_\Omega)$.

Based on mean-field approximation, it will be assumed that the variational distribution is factorized w.r.t. each variable Θ_j such that

$$q(\Theta)=q_\lambda(\boldsymbol{\lambda})q_\tau(\tau)\prod_{n=1}^{N}q_n\big(\mathbf{A}^{(n)}\big).\tag{14}$$

It should be noted that this is the only assumption about the distribution, while the particular functional forms of $q_j(\Theta_j)$ can be explicitly derived in turn. The optimised form of the jth factor based on the maximization of $\mathcal{L}(q)$ is given by

$$\ln q_j(\Theta_j)=\mathbb{E}_{q(\Theta\backslash\Theta_j)}[\ln p(\mathcal{Y}_\Omega,\Theta)]+\mathrm{const},\tag{15}$$

where $\mathbb{E}_{q(\Theta\backslash\Theta_j)}[\cdot]$ denotes an expectation w.r.t. the q distributions over all variables except Θ_j. Since the distributions of all parameters are drawn from the exponential family and are conjugate w.r.t. the distributions of their parents (see Fig. 1), we can derive the closed-form posterior update rules for Θ.

3.2.1 Posterior Distribution of Factor Matrices

As can be seen from the graphical model shown in Fig. 1, the inference of mode-n factor matrix $\mathbf{A}^{(n)}$ can be performed by receiving the messages from observed data and its co-parents, including other factors $\mathbf{A}^{(k)}, k\neq n$ and the hyperparameter τ, which are expressed by the likelihood term (6), and incorporating the messages from its parents, which are expressed by the prior term (7). By applying (15), it has been shown that their posteriors can be factorized as independent distributions of their rows, which are also Gaussian (see Section 2 of Appendix, available in the online supplemental material, for details), given by

$$q_n(\mathbf{A}^{(n)})=\prod_{i_n=1}^{I_n}\mathcal{N}\big(\mathbf{a}_{i_n}^{(n)}|\tilde{\mathbf{a}}_{i_n}^{(n)},\mathbf{V}_{i_n}^{(n)}\big),\ \forall n\in[1,N]\tag{16}$$

where the posterior parameters can be updated by

$$\begin{aligned}\tilde{\mathbf{a}}_{i_n}^{(n)}&=\mathbb{E}_q[\tau]\mathbf{V}_{i_n}^{(n)}\mathbb{E}_q\big[\mathbf{A}_{i_n}^{(\backslash n)T}\big]\mathrm{vec}\big(\mathcal{Y}_{\mathbb{I}(\mathcal{O}_{i_n}=1)}\big)\\ \mathbf{V}_{i_n}^{(n)}&=\big(\mathbb{E}_q[\tau]\mathbb{E}_q\big[\mathbf{A}_{i_n}^{(\backslash n)T}\mathbf{A}_{i_n}^{(\backslash n)}\big]+\mathbb{E}_q[\boldsymbol{\Lambda}]\big)^{-1},\end{aligned}\tag{17}$$

where $\mathcal{Y}_{\mathbb{I}(\mathcal{O}_{i_n}=1)}$ denotes a subset of the observed entries $\mathcal{Y}_\Omega$, whose mode-n index is i_n, i.e., the observed entries associated to the latent factor $\mathbf{a}_{i_n}^{(n)}$. The most complex term in (17) is related to

$$\mathbf{A}_{i_n}^{(\backslash n)T}=\Big(\bigodot_{k\neq n}\mathbf{A}^{(k)}\Big)^T_{\mathbb{I}(\mathcal{O}_{i_n}=1)},\tag{18}$$

where $(\odot_{k\neq n}\mathbf{A}^{(k)})^T$ is of size $R\times\prod_{k\neq n}I_k$, and each column is computed by $\circledast_{k\neq n}\mathbf{a}_{i_k}^{(k)}$ with varying mode-k index i_k. The symbol $(\cdot)_{\mathbb{I}(\mathcal{O}_{i_n}=1)}$ denotes a subset of columns sampled according to the subtensor $\mathrm{vec}(\mathcal{O}_{\cdots i_n\cdots})=1$. Hence, $\mathbb{E}_q[\mathbf{A}_{i_n}^{(\backslash n)T}\mathbf{A}_{i_n}^{(\backslash n)}]$ denotes the posterior covariance matrix of the Khatri-Rao product of latent factors in all modes except the nth-mode, and is computed by only the columns corresponding to the observed entries whose mode-n index is i_n. In order to evaluate this posterior covariance matrix, we need to introduce the following results.

Theorem 3.1. *Given a set of independent random matrices* $\{\mathbf{A}^{(n)}\in\mathbb{R}^{I_n\times R}|n=1,\ldots,N\}$*, we assume that* $\forall n,\forall i_n$*, the row vectors* $\{\mathbf{a}_{i_n}^{(n)}\}$ *are independent, then*

$$\mathbb{E}\Big[\Big(\bigodot_n\mathbf{A}^{(n)}\Big)^T\Big(\bigodot_n\mathbf{A}^{(n)}\Big)\Big]=\sum_{i_1,\ldots,i_N}\circledast_n\Big(\mathbb{E}\Big[\mathbf{a}_{i_n}^{(n)}\mathbf{a}_{i_n}^{(n)T}\Big]\Big)\tag{19}$$

where $\mathbb{E}[\mathbf{a}_{i_n}^{(n)}\mathbf{a}_{i_n}^{(n)T}]=\mathbb{E}[\mathbf{a}_{i_n}^{(n)}]\mathbb{E}[\mathbf{a}_{i_n}^{(n)T}]+\mathrm{Var}(\mathbf{a}_{i_n}^{(n)})$.

Proof. See Section 3 of Appendix, available in the online supplemental material, for details. □

For simplicity, we attempt to compute (19) by multilinear operations. Let $\forall n$, $\mathbf{B}^{(n)}$ of size $I_n \times R^2$ denote an expectation of a quadratic form related to $\mathbf{A}^{(n)}$ by defining the i_nth-row vector as

$$\mathbf{b}_{i_n}^{(n)} = \mathrm{vec}\big(\mathbb{E}_q\big[\mathbf{a}_{i_n}^{(n)}\mathbf{a}_{i_n}^{(n)T}\big]\big) = \mathrm{vec}\big(\tilde{\mathbf{a}}_{i_n}^{(n)}\tilde{\mathbf{a}}_{i_n}^{(n)T} + \mathbf{V}_{i_n}^{(n)}\big), \tag{20}$$

then we have

$$\mathrm{vec}\left(\sum_{i_1,\dots,i_N} \mathop{\circledast}_{n} \big(\mathbb{E}\big[\mathbf{a}_{i_n}^{(n)}\mathbf{a}_{i_n}^{(n)T}\big]\big)\right) = \Big(\bigodot_{n} \mathbf{B}^{(n)}\Big)^T \mathbf{1}_{\prod_n I_n}, \tag{21}$$

where $\mathbf{1}_{\prod_n I_n}$ denotes a vector of length $\prod_n I_n$ and all elements are equal to one.

According to Theorem 3.1 and the computation form in (21), the term $\mathbb{E}_q\big[\mathbf{A}_{i_n}^{(\backslash n)T}\mathbf{A}_{i_n}^{(\backslash n)}\big]$ in (17) can be evaluated efficiently by

$$\mathrm{vec}\Big(\mathbb{E}_q\big[\mathbf{A}_{i_n}^{(\backslash n)T}\mathbf{A}_{i_n}^{(\backslash n)}\big]\Big) = \Big(\bigodot_{k\neq n} \mathbf{B}^{(k)}\Big)^T \mathrm{vec}(\boldsymbol{\mathcal{O}}_{\cdots i_n\cdots}), \tag{22}$$

where $\boldsymbol{\mathcal{O}}_{\cdots i_n\cdots}$ denotes a subtensor by fixing model-n index to i_n. It should be noted that the Khatri-Rao product is computed by all mode factors except the nth mode, while the sum is performed according to the indices of observations, implying that only factors that interact with $\mathbf{a}_{i_n}^{(n)}$ are taken into account. Another complex term in (17) can also be simplified by multilinear operations, i.e.,

$$\begin{aligned}&\mathbb{E}_q\big[\mathbf{A}_{i_n}^{(\backslash n)T}\big]\mathrm{vec}\big(\boldsymbol{\mathcal{Y}}_{\mathbb{I}(\mathcal{O}_{i_n}=1)}\big)\\ &\quad= \Big(\bigodot_{k\neq n} \mathbb{E}_q[\mathbf{A}^{(k)}]\Big)^T \mathrm{vec}\{(\boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}})_{\cdots i_n\cdots}\}.\end{aligned} \tag{23}$$

Finally, the variational posterior approximation of factor matrices can be updated by (17) and the posterior moments, including $\forall n, \forall i_n$, $\mathbb{E}_q[\mathbf{a}_{i_n}^{(n)}]$, $\mathrm{Var}(\mathbf{a}_{i_n}^{(n)})$, $\mathbb{E}_q[\mathbf{A}^{(n)}]$, and $\mathbb{E}_q[\mathbf{a}_{i_n}^{(n)}\mathbf{a}_{i_n}^{(n)T}], \mathbb{E}_q[\mathbf{a}_{i_n}^{(n)T}\mathbf{a}_{i_n}^{(n)}]$, can be easily evaluated, which are required by the inference of other hyperparameters in $\boldsymbol{\Theta}$.

An intuitive interpretation of (17) is given as follows. The posterior covariance $\mathbf{V}_{i_n}^{(n)}$ is updated by combining the prior information $\mathbb{E}_q[\boldsymbol{\Lambda}]$ and the posterior information from other factor matrices computed by (22), while the tradeoff between these two terms is controlled by $\mathbb{E}_q[\tau]$ that is related to the quality of model fitting. In other words, the better fitness of the current model leads to more information from other factors than from prior information. The posterior mean $\tilde{\mathbf{a}}_{i_n}^{(n)}$ is updated first by linear combination of all other factors, expressed by (23), where the coefficients are observed values. This implies that the larger observation leads to more similarity of its corresponding latent factors. Then, $\tilde{\mathbf{a}}_{i_n}^{(n)}$ is rotated by $\mathbf{V}_{i_n}^{(n)}$ to obtain the property of sparsity and is scaled according to the model fitness $\mathbb{E}_q[\tau]$.

3.2.2 Posterior Distribution of Hyperparameters $\boldsymbol{\lambda}$

It should be noted that, instead of point estimation via optimizations, learning the posterior of $\boldsymbol{\lambda}$ is crucial for automatic rank determination. As seen in Fig. 1, the inference of $\boldsymbol{\lambda}$ can be performed by receiving messages from N factor matrices and incorporating the messages from its hyperprior. By applying (15), we can identify the posteriors of $\lambda_r, \forall r \in [1, R]$ as an independent Gamma distribution (see Section 4 of Appendix, available in the online supplemental material, for details),

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) = \prod_{r=1}^{R} \mathrm{Ga}\big(\lambda_r | c_M^r, d_M^r\big), \tag{24}$$

where c_M^r, d_M^r denote the posterior parameters learned from M observations and can be updated by

$$\begin{aligned} c_M^r &= c_0^r + \frac{1}{2}\sum_{n=1}^{N} I_n,\\ d_M^r &= d_0^r + \frac{1}{2}\sum_{n=1}^{N} \mathbb{E}_q\big[\mathbf{a}_{\cdot r}^{(n)T}\mathbf{a}_{\cdot r}^{(n)}\big].\end{aligned} \tag{25}$$

The expectation of the inner product of the rth component in mode-n matrix w.r.t. q distribution can be evaluated using the posterior parameters in (16), i.e.,

$$\mathbb{E}_q\big[\mathbf{a}_{\cdot r}^{(n)T}\mathbf{a}_{\cdot r}^{(n)}\big] = \tilde{\mathbf{a}}_{\cdot r}^{(n)T}\tilde{\mathbf{a}}_{\cdot r}^{(n)} + \sum_{i_n}\big(\mathbf{V}_{i_n}^{(n)}\big)_{rr}. \tag{26}$$

By combining (25) and (26), we can further simplify the computation of $\mathbf{d}_M = [d_M^1, \dots d_M^R]^T$ as

$$\mathbf{d}_M = \sum_{n=1}^{N}\left\{\mathrm{diag}\left(\tilde{\mathbf{A}}^{(n)T}\tilde{\mathbf{A}}^{(n)} + \sum_{i_n}\mathbf{V}_{i_n}^{(n)}\right)\right\}, \tag{27}$$

where $\tilde{\mathbf{A}} = \mathbb{E}_q[\mathbf{A}^{(n)}]$. Hence, the posterior expectation can be obtained by $\mathbb{E}_q[\boldsymbol{\lambda}] = [c_M^1/d_M^1, \dots, c_M^R/d_M^R]^T$, and thus, $\mathbb{E}_q[\boldsymbol{\Lambda}] = \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}])$.

An intuitive interpretation of (25) is that λ_r is updated by the sum of squared L_2-norm of rth component, expressed by (26), from N factor matrices. Therefore, the smaller of $\|\mathbf{a}_{\cdot r}\|_2^2$ leads to larger $\mathbb{E}_q[\lambda_r]$ and updated priors of factor matrices, which in turn enforces more strongly the rth component to be zero.

3.2.3 Posterior Distribution of Hyperparameter τ

The inference of the noise precision τ can be performed by receiving the messages from observed data and its co-parents, including N factor matrices, and incorporating the messages from its hyperprior. By applying (15), the variational posterior is a Gamma distribution (see Section 5 of Appendix, available in the online supplemental material, for details), given by

$$q_\tau(\tau) = \mathrm{Ga}(\tau | a_M, b_M), \tag{28}$$

where the posterior parameters can be updated by

$$\begin{aligned} a_M &= a_0 + \frac{1}{2}\sum_{i_1,\dots,i_N} \mathcal{O}_{i_1\dots i_N}\\ b_M &= b_0 + \frac{1}{2}\mathbb{E}_q\left[\left\|\boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - [\![\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}]\!]\right)\right\|_F^2\right].\end{aligned} \tag{29}$$

However, the posterior expectation of model error in the above expression cannot be computed straightforwardly, and therefore, we need to introduce the following results.

Theorem 3.2. *Assume a set of independent R-dimensional random vectors $\{\mathbf{x}^{(n)}|n=1,\dots,N\}$, then*

$$\mathbb{E}\big[\langle \mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\rangle^2\big] = \big\langle \mathbb{E}\big[\mathbf{x}^{(1)}\mathbf{x}^{(1)T}\big],\dots,\mathbb{E}\big[\mathbf{x}^{(N)}\mathbf{x}^{(N)T}\big]\big\rangle, \quad (30)$$

where the left term denotes the expectation of the squared inner product of N vectors, and the right term denotes the inner product of N matrices, where each matrix of size $R\times R$ denotes an expectation of the outer product of the nth vector, respectively.

Proof. See Section 6 of Appendix, available in the online supplemental material, for details. □

Theorem 3.3. *Given a set of independent random matrices $\{\mathbf{A}^{(n)}|n=1,\dots,N\}$, we assume that $\forall n, \forall i_n$, the row vectors $\{\mathbf{a}^{(n)}_{i_n}\}$ are independent, then*

$$\begin{aligned}&\mathbb{E}\big[\big\|[\![\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}]\!]\big\|_F^2\big]\\ &\quad=\sum_{i_1,\dots,i_N}\big\langle \mathbb{E}\big[\mathbf{a}^{(1)}_{i_1}\mathbf{a}^{(1)T}_{i_1}\big],\dots,\mathbb{E}\big[\mathbf{a}^{(N)}_{i_N}\mathbf{a}^{(N)T}_{i_N}\big]\big\rangle.\end{aligned} \quad (31)$$

Let $\mathbf{B}^{(n)}$ denote the expectation of a quadratic form related to $\mathbf{A}^{(n)}$ with i_nth-row vector $\mathbf{b}^{(n)}_{i_n}=\mathrm{vec}(\mathbb{E}[\mathbf{a}^{(n)}_{i_n}\mathbf{a}^{(n)T}_{i_n}])$; thus, (31) can be computed by

$$\mathbb{E}[\big\|[\![\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}]\!]\big\|_F^2] = \mathbf{1}^T_{\prod_n I_n}\Big(\bigodot_n \mathbf{B}^{(n)}\Big)\mathbf{1}_{R^2}.$$

Proof. See Section 7 of Appendix, available in the online supplemental material, for details. □

From Theorems 3.2 and 3.3, the posterior expectation term in (29) can be evaluated explicitly. Due to the missing entries in $\mathcal{Y}$, the evaluation form is finally written as (see Section 8 of Appendix, available in the online supplemental material, for details)

$$\begin{aligned}&\mathbb{E}_q\Big[\Big\|\mathcal{O}\circledast\Big(\mathcal{Y}-[\![\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}]\!]\Big)\Big\|_F^2\Big]\\ &\quad=\|\mathcal{Y}_\Omega\|_F^2-2\mathrm{vec}^T(\mathcal{Y}_\Omega)\mathrm{vec}\big([\![\tilde{\mathbf{A}}^{(1)},\dots,\tilde{\mathbf{A}}^{(N)}]\!]_\Omega\big)\\ &\qquad+\mathrm{vec}^T(\mathcal{O})\Big(\bigodot_n \mathbf{B}^{(n)}\Big)\mathbf{1}_{R^2},\end{aligned} \quad (32)$$

where $\tilde{\mathbf{A}}^{(n)}=\mathbb{E}_q\big[\mathbf{A}^{(n)}\big]$ and $\mathbf{B}^{(n)}$ is computed by (20). Hence, the posterior approximation of τ can be obtained by (29) together with $\mathbb{E}_q[\tau]=a_M/b_M$.

An intuitive interpretation of (29) is straightforward. a_M is related to the number of observations and b_M is related to the residual of model fitting measured by the squared Frobenius norm on observed entries.

3.2.4 Lower Bound of Model Evidence

The inference framework presented in the previous section can essentially maximize the lower bound of model evidence that is defined in (13). Since the lower bound should not decrease at each iteration, it can be used to test for convergence. The lower bound of the log-marginal likelihood is computed by

$$\mathcal{L}(q)=\mathbb{E}_{q(\Theta)}[\ln p(\mathcal{Y}_\Omega,\Theta)]+H(q(\Theta)), \quad (33)$$

where the first term denotes the posterior expectation of joint distribution, and the second term denotes the entropy of posterior distributions.

Various terms in the lower bound are evaluated and derived by taking parametric forms of q distribution, giving the following results (see Section 9 of Appendix, available in the online supplemental material, for details)

$$\begin{aligned}\mathcal{L}(q)=&-\frac{a_M}{2b_M}\mathbb{E}_q\Big[\Big\|\mathcal{O}\circledast\Big(\mathcal{Y}-[\![\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}]\!]\Big)\Big\|_F^2\Big]\\ &-\frac{1}{2}\mathrm{Tr}\Big\{\tilde{\mathbf{\Lambda}}\sum_n\Big(\tilde{\mathbf{A}}^{(n)T}\tilde{\mathbf{A}}^{(n)}+\sum_{i_n}\mathbf{V}^{(n)}_{i_n}\Big)\Big\}\\ &+\frac{1}{2}\sum_n\sum_{i_n}\big\{\ln\big|\mathbf{V}^{(n)}_{i_n}\big|\big\}+\sum_r\big\{\ln\Gamma\big(c^r_M\big)\big\}\\ &+\sum_r\Big\{c^r_M\Big(1-\ln d^r_M-\frac{d^r_0}{d^r_M}\Big)\Big\}+\ln\Gamma(a_M)\\ &+a_M\Big(1-\ln b_M-\frac{b_0}{b_M}\Big)+\mathrm{const}.\end{aligned} \quad (34)$$

An intuitive interpretation of (34) is as follows. The first term is related to model residual; the second term is related to the weighted sum of squared L_2-norm of each component in factor matrices, while the uncertainty information is also considered; the rest terms are related to negative KL divergence between the posterior and prior distributions of hyperparameters.

3.2.5 Initialization of Model Parameters

The variational Bayesian inference is guaranteed to converge only to a local minimum. To avoid getting stuck in poor local solutions, it is important to choose an initialization point. In our model, the top level hyperparameters including $\mathbf{c}_0,\mathbf{d}_0,a_0,b_0$ are set to 10^{-6}, resulting in a noninformative prior. Thus, we have $\mathbb{E}[\mathbf{\Lambda}]=\mathbf{I}$ and $\mathbb{E}[\tau]=1$. For the factor matrices, $\{\mathbb{E}[\mathbf{A}^{(n)}]\}_{n=1}^N$ can be initialized by two different strategies, one is randomly drawn from $\mathcal{N}(\mathbf{0},\mathbf{I})$ for $\mathbf{a}^{(n)}_{i_n}$, $\forall i_n\in[1,I_n]$, $\forall n\in[1,N]$. The other is set to $\mathbf{A}^{(n)}=\mathbf{U}^{(n)}\mathbf{\Sigma}^{(n)\frac{1}{2}}$, where $\mathbf{U}^{(n)}$ denotes the left singular vectors and $\mathbf{\Sigma}^{(n)}$ denotes the diagonal singular values matrix, obtained by SVD of mode-n matricization of tensor $\mathcal{Y}$. The covariance matrix $\mathbf{V}^{(n)}$ is simply set to $\mathbf{I}$. The tensor rank R is usually initialized by the weak upper bound on its maximum rank, i.e., $R\le\min_n P_n$, where $P_n=\prod_{i\neq n}I_i$. In practice, we can also manually define the initialization value of R for computational efficiency. These settings using noninformative priors can generally result in a solution that solely depends on the observed data.

3.2.6 Interpretaion of Automatic Rank Determination

The entire procedure of model inference is summarized in Algorithm 1. It should be noted that tensor rank is determined automatically and implicitly. More specifically, updating $\boldsymbol{\lambda}$ in each iteration results in a new prior over $\{\mathbf{A}^{(n)}\}$, and then, $\{\mathbf{A}^{(n)}\}$ can be updated using this new prior in the subsequent iteration, which in turn affects $\boldsymbol{\lambda}$. Hence, if the posterior mean of λ_r becomes very large, the rth components in $\{\mathbf{A}^{(n)}\}, \forall n \in [1, N]$ are forced to be zero because of their prior information, and the tensor rank can be obtained by simply counting the number of non-zero components in the factor matrices. For implementation of the algorithm, we can keep the size of $\{\mathbf{A}^{(n)}\}$ unchanged during iterations; an alternative method is to eliminate the zero-components of $\{\mathbf{A}^{(n)}\}$ after each iteration.

Algorithm 1. Fully Bayesian CP Factorization (FBCP)

```
**Input:** an $N$th-order incomplete tensor $\mathcal{Y}_\Omega$ and an indicator tensor $\mathcal{O}$.
**Initialization:** $\tilde{\mathbf{A}}^{(n)}, \mathbf{V}^{(n)}_{i_n}, \forall i_n \in [1, I_n], \forall n \in [1, N]$, $a_0, b_0, \mathbf{c}_0, \mathbf{d}_0$, and $\tau = a_0/b_0$, $\lambda_r = c_0^r/d_0^r, \forall r \in [1, R]$.
**repeat**
  **for** $n = 1$ **to** $N$ **do**
    Update the posterior $q(\mathbf{A}^{(n)})$ using (17);
  **end for**
  Update the posterior $q(\boldsymbol{\lambda})$ using (25);
  Update the posterior $q(\tau)$ using (29);
  Evaluate the lower bound using (34);
  Reduce rank $R$ by eliminating zero-components of $\{\mathbf{A}^{(n)}\}$ (an optional procedure);
**until** convergence.
Computation of predictive distributions using (35).
```

3.3 Predictive Distribution

The predictive distributions over missing entries, given observed entries, can be approximated by using variational posterior distribution, that is,

$$\begin{aligned} p(\mathcal{Y}_{i_1\ldots i_N}|\mathcal{Y}_\Omega) &= \int p(\mathcal{Y}_{i_1\ldots i_N}|\Theta)p(\Theta|\mathcal{Y}_\Omega)\,\mathrm{d}\Theta \\ &\simeq \iint p\big(\mathcal{Y}_{i_1\ldots i_N}|\big\{\mathbf{a}^{(n)}_{i_n}\big\}, \tau^{-1}\big)q\big(\big\{\mathbf{a}^{(n)}_{i_n}\big\}\big)q(\tau)\,\mathrm{d}\big\{\mathbf{a}^{(n)}_{i_n}\big\}\,\mathrm{d}\tau. \end{aligned} \tag{35}$$

We can now approximate these integrations, yielding a Student's t-distribution $\mathcal{Y}_{i_1\ldots i_N}|\mathcal{Y}_\Omega \sim \mathcal{T}(\tilde{\mathcal{Y}}_{i_1\ldots i_N}, \mathcal{S}_{i_1\ldots i_N}, \nu_y)$ (see Section 10 of Appendix, available in the online supplemental material, for details) with its parameters given by

$$\begin{aligned} \tilde{\mathcal{Y}}_{i_1\ldots i_N} &= \big\langle \tilde{\mathbf{a}}^{(1)}_{i_1}, \cdots, \tilde{\mathbf{a}}^{(n)}_{i_N} \big\rangle, \quad \nu_y = 2a_M, \\ \mathcal{S}_{i_1\ldots i_N} &= \left\{ \frac{b_M}{a_M} + \sum_n \left\{ \left(\circledast_{k\neq n} \tilde{\mathbf{a}}^{(k)}_{i_k} \right)^T \mathbf{V}^{(n)}_{i_n} \left(\circledast_{k\neq n} \tilde{\mathbf{a}}^{(k)}_{i_k} \right) \right\} \right\}^{-1}. \end{aligned}$$

Thus, the predictive variance can be obtained by $\mathrm{Var}(\mathcal{Y}_{i_1\ldots i_N}) = \frac{\nu_y}{\nu_y - 2}\mathcal{S}^{-1}_{i_1\ldots i_N}$.

3.4 Computational Complexity

The computation cost of the N factor matrices in (17) is $\mathcal{O}(NR^2M + R^3\sum_n I_n)$, where N is the order of the tensor, M denotes the number of observations, i.e., the input data size. R is the number of latent components in each $\mathbf{A}^{(n)}$, i.e., model complexity or tensor rank, and is generally much smaller than the data size, i.e., $R \ll M$. Hence, it has linear complexity w.r.t. the data size and polynomial complexity w.r.t. the model complexity. It should be noted that, because of the automatic model selection, the excessive latent components are pruned out in the first few iterations such that R reduces rapidly in practice. The computation cost of the hyperparameter $\boldsymbol{\lambda}$ in (25) is $\mathcal{O}(R^2\sum_n I_n)$, which is dominated by the model complexity, while the computation cost of noise precision τ in (29) is $\mathcal{O}(R^2M)$. Therefore, the overall complexity of our algorithm is $\mathcal{O}(NR^2M + R^3)$, which scales linearly with the data size but polynomially with the model complexity.

3.5 Discussion of Advantages

- The *automatic determination of* CP *rank* enables us to obtain an optimal low-rank tensor approximation, even from a highly noisy and incomplete tensor.
- Our method is characterized as a *tuning parameter-free* approach and all model parameters can be inferred from the observed data, which avoids the computational expensive parameter selection procedure. In contrast, the existing tensor factorization methods require a predefined rank, while the tensor completion methods based on nuclear norm require several tuning parameters.
- The *uncertainty information* over both latent factors and predictions of missing entries can be inferred by our method, while most existing tensor factorization and completion methods provide only the point estimations.
- An efficient and deterministic Bayesian inference is developed for model learning, which empirically shows a *fast convergence*.

4 Mixture Factor Priors

The low-rank assumption is powerful in general cases, however if the tensor data does not satisfy an intrinsic low-rank structure and a large amount of entries are missing, it may yield an oversimplified model. In this section, we present a variant of Bayesian CP factorization model which can take into account the local similarity in addition to the low-rank assumption.

We specify a Gaussian mixture prior over factor matrices such that the prior distribution in (7) can be rewritten as $\forall i_n \in [1, I_n], \forall n \in [1, N]$,

$$\mathbf{a}^{(n)}_{i_n}|\boldsymbol{\lambda}, \big\{\mathbf{a}^{(n)}_k\big\} \sim w_{i_n,i_n}\mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^{-1}) + \sum_{k\neq i_n} w_{i_n,k}\mathcal{N}\big(\mathbf{a}^{(n)}_k, \beta_k\mathbf{I}\big),$$

where $\sum_k w_{i_n,k} = 1$. This indicates that the i_nth row vector $\mathbf{a}^{(n)}_{i_n}$ is similar to kth row vectors with the probability of $w_{i_n,k}$. Based on our assumption that the adjacent rows are highly correlated, we can define the mixture coefficients by $w_{i,j} = z_i\exp(-|i-j|^2)$ where $z_i = 1/\sum_j \exp(-|i-j|^2)$ is used to ensure the sum of mixture coefficients to be 1. For model learning, we can easily verify that the posterior distribution is also a mixture distribution. For simplicity, we set $\forall k, \beta_k = 0$, thus the posterior mean of factor matrix can be updated first by (17) and followed by $\mathbb{E}_q[\mathbf{A}^{(n)}] \leftarrow \mathbf{W}\mathbb{E}_q[\mathbf{A}^{(n)}]$, while the posterior covariance $\{\mathbf{V}^{(n)}_{i_n}\}_{i_n=1}^{I_n}$ keep unchanged. Furthermore, the inference of all other variables do not need any changes.

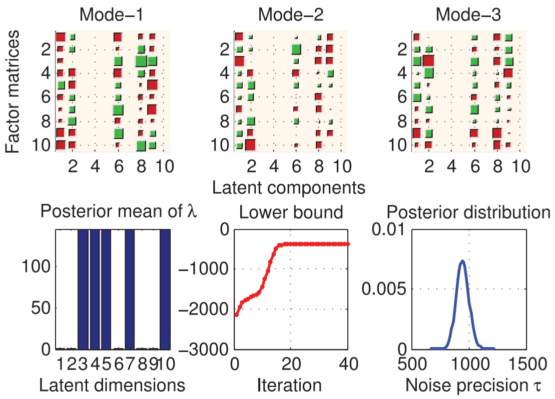


Fig. 2. The top row shows Hinton diagram of factor matrices, where the color and size of each square represent the sign and magnitude of the value, respectively. The bottom row shows the posterior of $\boldsymbol{\lambda}$, the lower bound of model evidence, and the posterior of τ from left to right.

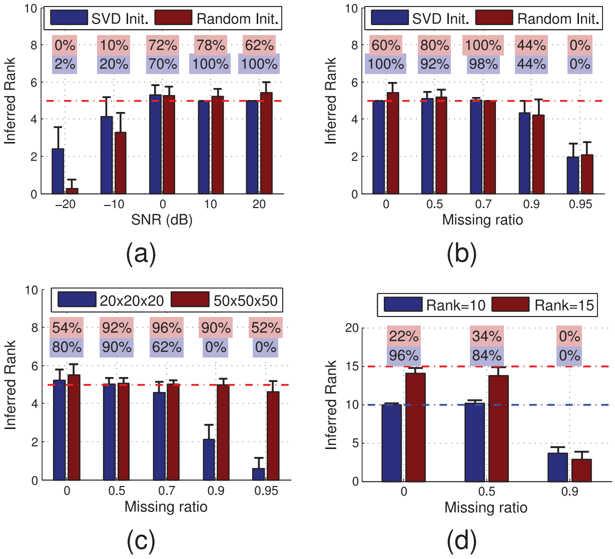


Fig. 3. Determination of tensor rank under varying conditions. Each vertical bar shows the mean and standard deviation of estimations from 50 MC runs, while the accuracy of detections is shown on the top of the corresponding bar. The red and blue horizontal dash dotted lines indicate the true tensor rank.

5 Experimental Results

We conducted extensive experiments using both synthetic data and real-world applications, and compared our fully Bayesian CP factorization (FBCP)[1] and its extension using mixture prior (FBCP-MP) with several state-of-the-art methods. Tensor factorization based scheme includes CPWOPT [10], CPNLS [11], [42], and KTD [43], while the completion based scheme includes HaLRTC and FaLRTC [13], FCSA [15], hard-completion (HardC.) [14], geomCG [12], and STDC [20]. Our objective when using synthetic data was to validate our method from several aspects: i) capability of rank determination; ii) reconstruction performance given a complete tensor; iii) predictive performance over missing entries given an incomplete tensor. Two real-world applications including image inpainting and facial image synthesis were used for evaluating the completion performance. All experiments were performed by a PC (Intel Xeon(R) 3.3 GHz, 64 GB memory).

5.1 Validation on Synthetic Data

The synthetic tensor data is generated by the following procedure. N factor matrices $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$ are drawn from a standard normal distribution, i.e., $\forall n, \forall i_n, \mathbf{a}_{i_n}^{(n)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R)$. Then, the true tensor is constructed by $\mathcal{X} = [\![\mathbf{A}^1, \ldots, \mathbf{A}^{(N)}]\!]$, and an observed tensor by $\mathcal{Y} = \mathcal{X} + \varepsilon$, where $\varepsilon \sim \prod_{i_1,\ldots,i_N} \mathcal{N}(0, \sigma_\varepsilon^2)$ denotes i.i.d. additive noise. The missing entries, chosen uniformly, are marked by an indicator tensor $\mathcal{O}$.

5.1.1 A Toy Example

To illustrate our model, we provide two demo videos in the supplemental materials, available online. A true latent tensor $\mathcal{X}$ is of size $10 \times 10 \times 10$ with *CP* rank $R = 5$, the noise parameter was $\sigma_\varepsilon^2 = 0.001$, and 40 percent of entries were missing. Then, we applied our method with the initial rank being set to 10. As shown in Fig. 2, three factor matrices are inferred in which five components are effectively pruned out, resulting in correct estimation of tensor rank. The lower bound of model evidence increases monotonically, which indicates the effectiveness and convergence of our algorithm. Finally, the posterior of noise precision $\tau \approx 1{,}000$ implies the method's capability of denoising and the estimation of $\sigma_\varepsilon^2 \approx 0.001$, $\mathrm{SNR} = 10\log(\sigma_{\mathcal{X}}^2 \tau^{-1})$.

5.1.2 Automatic Determination of Tensor Rank

To evaluate the automatic determination of tensor rank (i.e., *CP rank*), extensive simulations were performed under varying experimental conditions related to tensor size, tensor rank, noise level, missing ratio, and the initialization method of factor matrices (e.g., SVD or random sample). Each result is evaluated by 50 Monte Carlo (MC) runs performed on different tensors generated by the same criterion. There are four groups of experiments. (A) Given complete tensors of size $20 \times 20 \times 20$ with $R = 5$, the evaluations were performed under varying noise levels and by two different initializations (see Fig. 3a). (B) Given incomplete tensors of size $20 \times 20 \times 20$ with $R = 5$ and $\mathrm{SNR} = 20$ dB, the evaluations were performed under five different missing ratios, and by different initializations (see Fig. 3b). (C) Given incomplete tensors with $R = 5$ and $\mathrm{SNR} = 0$ dB, the evaluations were performed under varying missing ratios and two different tensor sizes (see Fig. 3c). (D) Given incomplete tensors of size $20 \times 20 \times 20$ with $\mathrm{SNR} = 20$ dB, the evaluations were performed under varying missing ratios and two different true ranks (see Fig. 3d).

From Fig. 3, we observe that SVD initialization is slightly better than random initialization in terms of the determination of tensor rank. If the tensor is complete, our model can detect the true tensor rank with 100 percent accuracy when $\mathrm{SNR} \geq 10$ dB. Although the accuracy decreased to 70 percent under a high noise level of 0 dB, the error deviation is only ± 1. On the other hand, if the tensor is incomplete with slight noises, the detection rate is 100 percent, when missing ratio is 0.7, and is 44 percent with an error deviation of only ± 1, even under a high missing ratio of 0.9. As both missing data and high noise level are presented, our model can achieve 90 percent accuracy under the condition

[1] The Matlab codes are provided in the supplemental materials, available online.

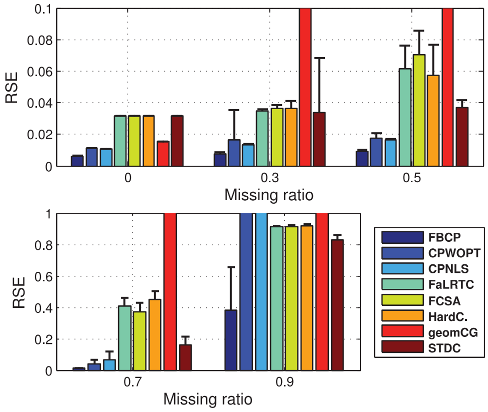


Fig. 4. Predictive performance when SNR = 30 dB.

of SNR = 0 dB and 0.5 missing ratio. It should be noted that, when the data size is larger, such as $50 \times 50 \times 50$, our model can achieve 90 percent accuracy, even when SNR = 0 dB and the missing ratio is 0.9. If the true rank is larger, such as $R = 15$, the model can correctly recover the rank from a complete tensor, but fails to do so when the missing ratio is larger than 0.5.

We can conclude from these results that the determination of the tensor rank depends primarily on the number of observed entries and the true tensor rank. In general, more observations are necessary if the tensor rank is larger; however, when high-level noise occurs, the excessive number of observations may not be helpful for rank determination.

5.1.3 Predictive Performance

In this experiment, we considered incomplete tensors of size $20 \times 20 \times 20$ generated by the true rank $R = 5$ and SNR = 30 dB under varying missing ratios. The initial rank was set to 10. The relative standard error $RSE = \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|_F}{\|\mathcal{X}\|_F}$, where $\hat{\mathcal{X}}$ denotes the estimation of the true tensor $\mathcal{X}$, was used to evaluate the performance. To ensure statistically consistent results, the performance is evaluated by 50 MC runs for each condition. As shown in Fig. 4, our method significantly outperforms other algorithms under all missing ratios. Factorization-based methods, including CPWOPT, and CPNLS show a better performance than completion-based methods when the missing ratio is relatively small, while they perform worse than completion methods when the missing ratio is large, e.g., 0.9. FaLRTC, FCSA, and HardC. achieve similar performances, because they are all based on nuclear norm optimization. geomCG achieves a performance comparable with that of CWOPT and CPNLS when data is complete, while it fails as the missing ratio becomes high. These results demonstrate that FBCP, as a tensor factorization method, can be also effective for tensor completion, even when an extremely sparse tensor is presented. In addition, we also conducted two additional experiments that are the reconstruction of a complete tensor and tensor completion when SNR = 0 dB (see Sections 11, 12 in Appendix, available in the online supplemental material).

5.2 Image Inpainting

In this section, image inpainting based on several benchmark images, shown in Fig. 5, are used to evaluate the performance of different methods. The colorful image can be represented by a third-order tensor of size $200 \times 200 \times 3$. We considered four groups of experiments. (A) *Structural image with uniformly random missing pixels.* A building facade image with 95 percent missing pixels under two noise conditions, i.e., noise free and SNR = 5 dB, were considered as observations. (B) *Natural image with uniformly random missing pixels.* The Lena image of size 300×300 with 90 percent missing pixels under two noise conditions, i.e., noise free and SNR = 10 dB, were considered. (C) *Non-random missing pixels.* We conducted two experiments for image restoration from a corrupted image. 1) The Lena image corrupted by superimposed text.[2] Since the location of text pixels are difficult to detect exactly, we can simply indicate missing entries by values larger than 200 to ensure that the text pixels are completely missing. 2) The scrabbled Lena image was used as an observed image and pixels with values larger than 200 can be marked as missing. (D) *Object removal*. Given an image and a mask covering the object area, our goal was to complete the image without that object. The algorithm settings of compared methods are described as follows. For factorization-based methods, the initial rank was set to 50 in cases of (A) and (B) due to the high missing ratios, and 100 in cases of (C) and (D). For completion-based methods, the tuning parameters were chosen by RSE evaluated on the ground-truth image, which gave the best possible performances.

Fig. 5. Ground-truth of eight benchmark images.

The visual effects of image inpainting are shown in Fig. 6, and the predictive performances are shown in Table 1 where case (D) is not available due to the lack of ground-truth. In case (A), we observe that FBCP and FBCP-MP outperform other methods for a structural image under an extremely high missing ratio and the superiority is more significant when an additive noise is involved. In case (B), FBCP-MP obtains the best performance followed by STDC. However, STDC is severely degraded when noise is involved, and obtains the same performance as FBCP, while its visual quality is still much better than others. These indicate that the local similarity in FBCP-MP is suitable for natural image. In case (C), FBCP and FBCP-MP are superior to all other methods, followed by STDC. The completion-based methods obtain relatively smoother effects than factorization-based methods, but the global color of the image is not recovered well, resulting in a poor predictive performance. In case (D), FBCP obtains the most clean image by removing the object completely while the ghost effects appear in all other methods. HaLRTC, FaLRTC and FCSA outperform FBCP-MP, CPWOPT, CPNLS and STDC.

From these results we can conclude that the completion-based methods generally outperform factorization-based methods for image completion. However, FBCP significantly improves ability of factorization-based scheme by automatic model selection and robustness to overfitting.

[2] A demo video is available in the supplemental materials, available online.

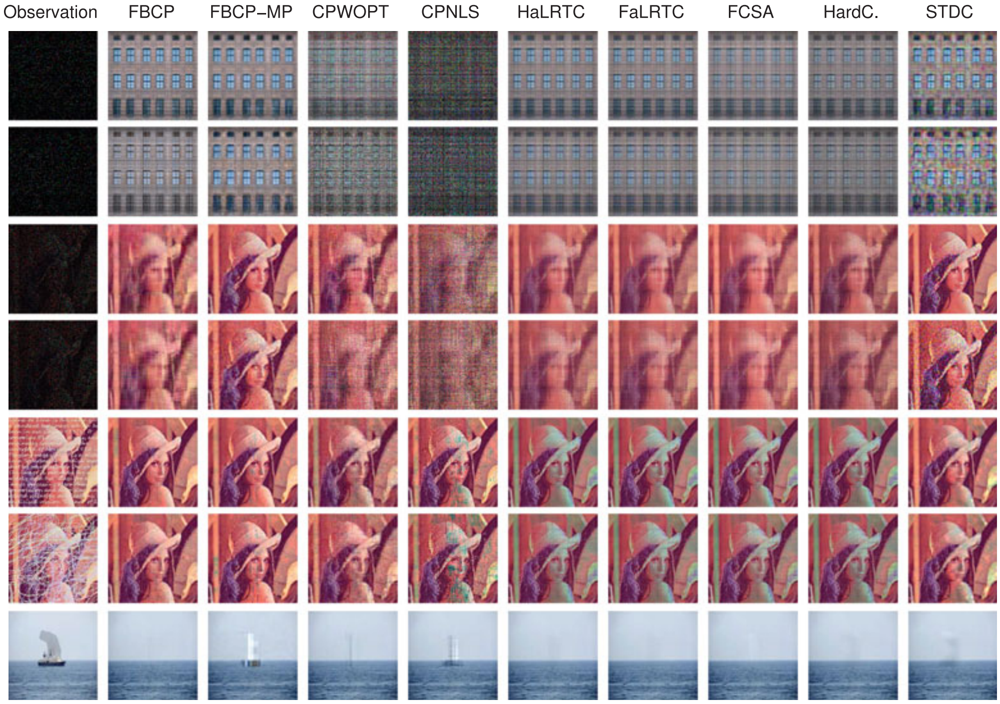


Fig. 6. Visual effects of image inpainting. Seven examples shown from top to bottom are (1) facade image with 95 percent missing; (2) facade image with 95 percent missing and an additive noise; (3) Lena image with 90 percent missing; (4) Lena image with 90 percent missing and an additive noise; (5) Lena image with superimposed text; (6) scribbled Lena image; (7) an image of ocean with an object.

The necessary number of observed entries mainly depends on the rank of true image. For instance, a structural image with an intrinsic low-rank need very fewer observations than a natural image. Due to the local similarity constrains, FBCP-MP and STDC can further reduce the necessary number of observations, which has been shown for Lena image. However, STDC degrades in presence of the non-random missing pixels or noise. Moreover, the performance of HaLRTC and STDC are sensitive to tuning parameters that must be carefully chosen for each specific condition.

Next, we conducted extensive experiments on eight images of size 256×256 shown in Fig. 5 with randomly missing pixels. Since most of these images are natural images on which the low-rank approximations require relatively large number of observations, we compared FBCP-MP with FBCP and other related methods. For FBCP and FBCP-MP, the same initialization of $R = 100$ was applied, while CPWOPT was performed by using the optimal ranks obtained from FBCP and FBCP-MP and the best performance was reported. The parameter selection for other methods was same with previous experiments. For KTD, due to large number of tuning parameters, we can only chose the optimal settings empirically. Fig. 7 shows an example for visual quality and Table 2 shows averaged quantitative results in terms of recovery performance and runtime. Observe that FBCP-MP improves the performance of FBCP significantly and achieves the best recovery performance, especially in the case of high missing rate. The time costs of FBCP and FBCP-MP are comparable with completion-based methods and significantly lower than CPWOPT. STDC obtains the comparable performance with FBCP-MP, however the parameters must be manually tuned for the specific condition. More detailed results on each image are shown visually and quantitatively in the supplemental materials, available online. These results demonstrate the effectiveness of mixture priors and the advantages when the local similarity is taken into account.

TABLE 1
Performance (RSEs) Evaluated on Missing Pixels

Method	Facade		Lena		Non-random	
	NF	N	NF	N	T	S
FBCP	**0.13**	0.17	0.17	0.20	**0.13**	**0.14**
FBCP-MP	**0.13**	**0.16**	**0.10**	**0.12**	**0.13**	**0.14**
CPWOPT	0.33	0.41	0.25	0.41	0.18	0.18
CPNLS	0.84	0.84	0.62	0.73	0.22	0.30
HaLRTC	0.15	0.21	0.19	0.21	0.29	0.28
FaLRTC	0.16	0.21	0.19	0.22	0.29	0.29
FCSA	0.19	0.21	0.19	0.21	0.28	0.28
HardC.	0.19	0.25	0.20	0.23	0.31	0.30
STDC	0.14	0.22	0.11	0.20	0.15	0.16

"NF", "N" indicate noise free or noisy image. "T", "S" indicate the text corruption or scrabbled image.

5.3 Facial Image Synthesis

For recognition of face images captured from surveillance videos, the ideal solution is to create a robust classifier that is invariant to some factors, such as pose and illumination. Hence, there arises the question whether we can generate novel facial images under multiple conditions given images under other conditions. Tensors are highly suitable for modeling a multifactor image ensemble, and therefore, we introduce a novel application of tensor factorization approaches for facial image synthesis.

We used the data set of 3D Basel Face Model [44], which contains an ensemble of facial images of 10 people, each rendered in nine different poses under three different

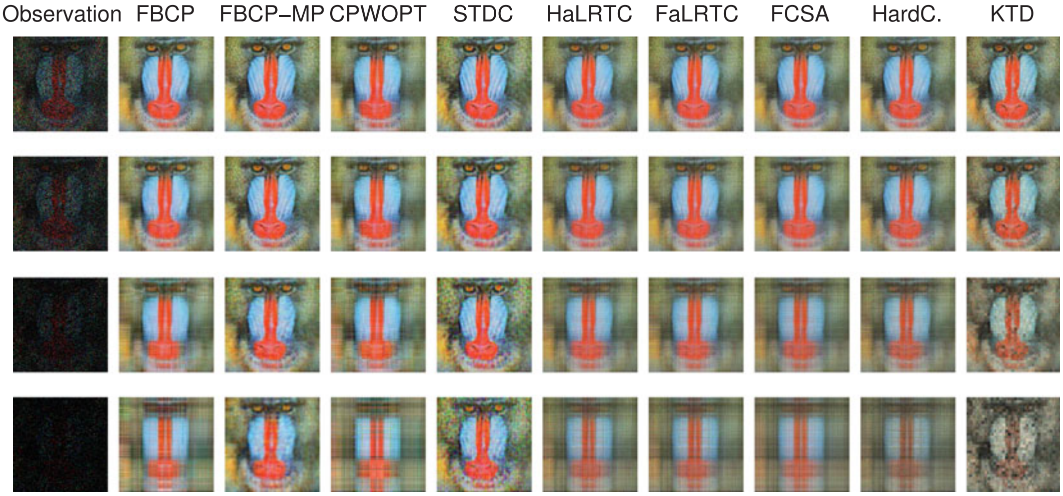


Fig. 7. Tensor completion for Baboon image under missing rates of 70, 80, 90 and 95 percent are shown from top to bottom. The left column shows observed images with randomly missing pixels, while the recovered images by nine different methods are shown from left to right.

TABLE 2
The Averaged Recovery Performance (RSE, PSNR, SSIM) and Runtime (Seconds) on Eight Images with Missing Rates of 70, 80, 90 and 95 Percent

		FBCP	FBCP-MP	CPWOPT	STDC	HaLRTC	FaLRTC	FCSA	HardC.	KTD
70%	RSE	0.1209	**0.0986**	0.1493	0.1003	0.1205	0.1205	0.1406	0.1254	0.1744
	PSNR	25.13	26.78	23.35	**26.79**	25.12	25.12	23.64	24.71	21.72
	SSIM	0.7546	**0.8531**	0.6417	0.8245	0.7830	0.7831	0.7437	0.7730	0.6579
	Runtime	83	251	1,807/2,908	32/292	18/139	46/232	**9**	36	169
80%	RSE	0.1423	**0.1084**	0.1700	0.1095	0.1479	0.1479	0.1675	0.1548	0.2030
	PSNR	23.09	25.36	21.52	**25.40**	22.72	22.72	21.53	22.32	19.80
	SSIM	0.6515	**0.7941**	0.5567	0.7781	0.6716	0.6716	0.6410	0.6579	0.5241
	Runtime	76	196	590/2316	30/328	25/122	57/282	**9**	39	129
90%	RSE	0.1878	**0.1295**	0.2372	0.1316	0.1992	0.1995	0.2342	0.2121	0.2407
	PSNR	20.12	**23.26**	18.08	23.21	19.62	19.61	18.09	19.11	17.80
	SSIM	0.4842	**0.6956**	0.3628	0.6950	0.5005	0.4998	0.4477	0.4790	0.3951
	Runtime	69	169	390/1475	32/378	21/127	61/307	**9**	32	82
95%	RSE	0.2420	**0.1566**	0.3231	0.1600	0.2549	0.2564	0.2777	0.2903	0.3091
	PSNR	17.76	**21.34**	15.35	21.18	17.25	17.19	16.39	16.12	15.41
	SSIM	0.3455	**0.6031**	0.2539	0.5810	0.3676	0.3649	0.3535	0.3369	0.2967
	Runtime	66	133	201/881	35/400	24/137	63/313	**8**	32	58

For methods that need to tune parameters, both the runtime with the best tuning parameter and the overall runtime are reported.

illuminations. As shown in Fig. 8, some images were fully missing. All 270 facial images were decimated and cropped to 68×68 pixels, and were then represented by a fourth-order tensor of size $4624 \times 10 \times 9 \times 3$. Since facial image does not possess an intrinsic low-rank structure, the vectorization operation was performed such that the whole ensemble satisfies the low-rank assumption. Since some methods are either computationally intractable or not applicable to $N \geq 4$ order tensor, five algorithms were finally applied on this data set under different missing ratios. The initial rank was set to 100 in factorization based methods, while the parameters of completion based methods were well tuned based on the ground-truth.

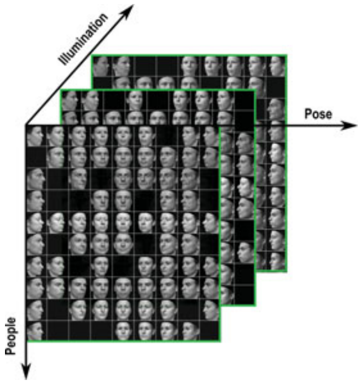
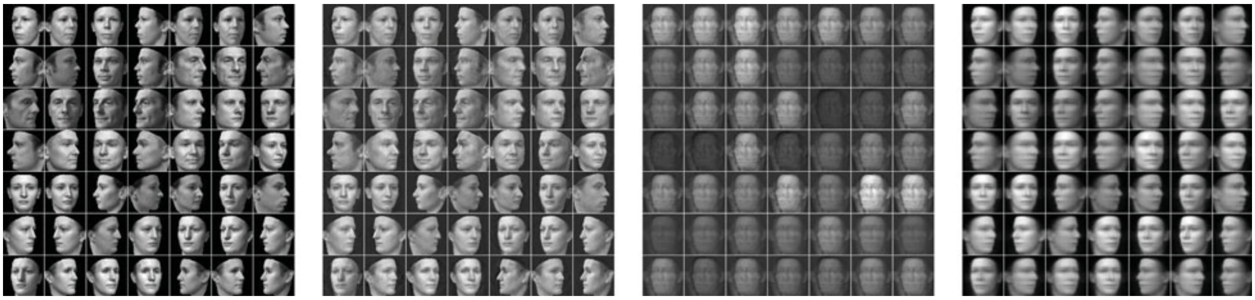


Fig. 8. Facial images under multiple conditions including people, poses and illuminations. Some of the images chosen randomly are fully missing.

As shown in Fig. 9, the visual quality of image synthesis obtained by FBCP is significantly superior to those by other methods. Although both CPWOPT and HaLRTC produce images that are smooth and blurred, HaLRTC obtains much better visual quality than CPWOPT. The detailed performances are compared in Table 3, where RSE w.r.t. observed entries reflects the performance of model fitting, and RSE w.r.t. missing entries particularly reflects the predictive ability. Note that RSE $=$ N/A implies that



(a) Missing faces (b) FBCP (c) CPWOPT (d) HaLRTC

Fig. 9. The ground-truth of 49 missing facial images and the synthetic images obtained by different methods.

HaLRTC and FaLRTC do not model the observed entries. The inferred rank by FBCP is within the range of [98, 100]. Observe that completion based methods including HaLRTC, FaLRTC and HardC. achieve better performance than CPWOPT. FBCP demonstrates the possibility that factorization-based scheme can significantly outperform completion-based methods. An interpretation is that HaLRTC based on rank minimization of unfolding matrices is prone to underestimate the tensor rank, while CPWOPT is prone to overfitting due to the point estimation based on maximum likelihood. By contrast, FBCP can infer CP rank more accurately and its predictive distribution obtained by integrating over latent factors can effectively prevent overfitting.

6 Conclusion

We proposed a fully Bayesian CP factorization which can naturally handle incomplete and noisy tensor data. By employing hierarchical priors over all unknown parameters, we derived a deterministic model inference under a fully Bayesian treatment. The most significant advantage is automatic determination of *CP* rank. Moreover, as a tuning parameter-free approach, our method avoids the parameter selection problem and can also effectively prevent overfitting. In addition, we proposed a variant of our method by using mixture priors, which shows advantages on natural images with a highly missing rate. Empirical results validated the effectiveness in terms of discovering the ground-truth of tensor rank and imputing missing values for an extremely sparse tensor. Several real-world applications, such as image completion and image synthesis, demonstrated the superiority of our methods over state-of-the-art techniques. Due to several interesting properties, our methods would be attractive for many potential applications.

TABLE 3
RSEs on Observed Images (O) and Missing Images (M)

Method	36/270		49/270		64/270		81/270	
	O	M	O	M	O	M	O	M
FBCP	0.05	**0.10**	0.05	**0.10**	0.05	**0.15**	0.05	**0.20**
CPWOPT	0.50	0.65	0.55	0.61	0.57	0.59	0.62	0.73
HaLRTC	N/A	0.28	N/A	0.30	N/A	0.31	N/A	0.34
FaLRTC	N/A	0.28	N/A	0.30	N/A	0.31	N/A	0.34
HardC.	0.37	0.37	0.36	0.40	0.36	0.40	0.36	0.40

The cases of 36, 49, 64 and 81 missing images were tested.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant No. 61202155). The work of Liqing Zhang was partially supported by the National Natural Science Foundation of China (Grant Nos 61272251, 91120305) and the National Basic Research Program of China (Grant No 2015CB856004). Q. Zhao is the corresponding author.

References

[1] T. Kolda and B. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[2] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*. New York, NY, USA: Wiley, 2009.

[3] D. Xu and S. Yan, “Semi-supervised bilinear subspace learning,” *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1671–1676, Jul. 2009.

[4] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, “Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 36–47, Jan. 2008.

[5] D. Xu, S. Yan, S. Lin, T. S. Huang, and S.-F. Chang, “Enhancing bilinear subspace learning by element rearrangement,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1913–1920, Oct. 2009.

[6] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[7] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an ”explanatory” multimodal factor analysis,” *UCLA Working Papers Phonetics*, vol. 16, pp. 1–84, 1970.

[8] R. Bro, “PARAFAC. Tutorial and applications,” *Chemom. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, 1997.

[9] M. Sørensen, L. De Lathauwer, P. Comon, S. Icart, and L. Deneire, “Canonical polyadic decomposition with orthogonality constraints,” *SIAM J. Matrix Anal. Appl.*, vol. 33, pp. 1190–1213, 2012.

[10] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations for incomplete data,” *Chemom. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.

[11] L. Sorber, M. Van Barel, and L. De Lathauwer, “Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(l_r,l_r,1) terms, and a new generalization,” *SIAM J. Opt.*, vol. 23, no. 2, pp. 695–720, 2013.

[12] D. Kressner, M. Steinlechner, and B. Vandereycken, “Low-rank tensor completion by Riemannian optimization,” *BIT Numer. Math.*, vol. 54, no. 2, pp. 447–468, Jun. 2014.

[13] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[14] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens, “Learning with tensors: A framework based on convex optimization and spectral regularization,” *Mach. Learn.*, pp. 1–49, 2013.

[15] J. Huang, S. Zhang, H. Li, and D. Metaxas, “Composite splitting algorithms for convex optimization,” *Comput. Vis. Image Understanding*, vol. 115, no. 12, pp. 1610–1622, 2011.

[16] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Probl.*, vol. 27, no. 2, p. 025010, 2011.

[17] 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.

[18] G. Zhong and M. Cheriet, "Large margin low rank tensor analysis," *Neural Comput.*, vol. 26, no. 4, pp. 761–780, 2014.

[19] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proc. Adv. Neural Inform. Process. Syst.*, 2013, pp. 836–844.

[20] Y.-L. Chen, C.-T. Hsu, and H.-Y. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 577–591, Mar. 2014.

[21] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Min. Knowl. Discov.*, vol. 25, no. 2, pp. 298–324, 2012.

[22] J. Hå stad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[23] B. Alexeev, M. A. Forbes, and J. Tsimerman, "Tensor rank: Some lower and upper bounds," in *Proc. IEEE 26th Annu. Conf. Comput. Complexity*, 2011, pp. 283–291.

[24] P. Bürgisser, and C. Ikenmeyer, "Geometric complexity theory and tensor rank," in *Proc. 43rd Annu. ACM Symp. Theory Comput.*, 2011, pp. 509–518.

[25] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, 2008.

[26] E. S. Allman, P. D. Jarvis, J. A. Rhodes, and J. G. Sumner, "Tensor rank, invariants, inequalities, and applications," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 1014–1045, 2013.

[27] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inform. Process. Syst.*, 2008, vol. 20, pp. 1257–1264.

[28] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 880–887.

[29] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3964–3977, Aug. 2012.

[30] Y. J. Lim and Y. W. Teh, "Variational Bayesian approach to movie rating prediction," in *Proc. KDD Cup Workshop*, 2007, pp. 15–21.

[31] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 601–608.

[32] B. Lakshminarayanan, G. Bouchard, and C. Archambeau, "Robust Bayesian matrix factorisation," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2011, pp. 425–433.

[33] W. Chu and Z. Ghahramani, "Probabilistic models for incomplete multi-dimensional arrays," in *JMLR Workshop Conf. Proc.*, 2009, vol. 5, pp. 89–96.

[34] S. Gao, L. Denoyer, P. Gallinari, and J. GUO, "Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks," *J. China Univ. Posts Telecommun.*, vol. 19, pp. 172–181, 2012.

[35] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Data Min.*, 2010, vol. 2010.

[36] 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 *Proc. IEEE 10th Int. Conf. Data Min.*, 2010, pp. 216–225.

[37] Z. Xu, F. Yan, and A. Qi, "Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis," in *Proc. 29th Int. Conf. Mach. Lear.*, 2012, pp. 1023–1030.

[38] D. J. MacKay, "Bayesian methods for backpropagation networks," in *Models of Neural Networks III*. New York, NY, USA: Springer, 1996, pp. 211–254.

[39] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

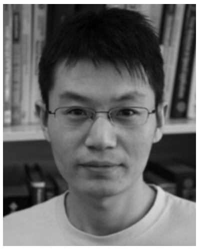
[40] C. M. Bishop, "Bayesian PCA," in *Proc. Adv. Neural Inform. Process. Syst.*, 1999, pp. 382–388.

[41] J. M. Winn and C. M. Bishop, "Variational message passing," *J. Mach. Learn. Res.*, vol. 6, pp. 661–694, 2005.

[42] L. Sorber, M. V. Barel, and L. D. Lathauwer. (2013). Tensorlab v1.0 [Online]. Available: http://esat.kuleuven.be/sista/tensorlab/

[43] A. H. Phan, A. Cichocki, P. Tichavsky, G. Luta, and A. J. Brockmeier, "Tensor completion through multiple Kronecker product decomposition." in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 3233–3237.

[44] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2009, pp. 296–301.



Qibin Zhao received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently a research scientist at the Laboratory for Advanced Brain Signal Processing in RIKEN Brain Science Institute, Japan, and is also a visiting professor in the 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. He is a member of the IEEE.

Liqing Zhang received the PhD degree from Zhongshan University, Guangzhou, China, in 1988. He is now a professor with the 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. He is a member of the IEEE.

Andrzej Cichocki received the PhD and DrSc (Habilitation) degrees, all in electrical engineering, from the 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 the coauthor of more than 400 scientific papers and four monographs (two of them translated to Chinese). He served as AE of *IEEE Transaction on Signal Processing*, TNNLS, *Cybernetics* and *Journal of Neuroscience Methods*. He is a fellow of the IEEE.