Scalable Bayesian Low-Rank Decomposition of Incomplete Multiway Tensors

Piyush Rai[*†] PIYUSH.RAI@DUKE.EDU
Yingjian Wang[*†] YW65@DUKE.EDU
Shengbo Guo[‡] S.GUO@SAMSUNG.COM
Gary Chen[‡] GARY.CHEN@SAMSUNG.COM
David Dunson[§] DUNSON@DUKE.EDU
Lawrence Carin[†] LCARIN@DUKE.EDU

[†]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
[‡]Samsung Research USA
[§]Department of Statistical Science, Duke University, Durham, NC 27708, USA

Abstract

We present a scalable Bayesian framework for low-rank decomposition of multiway tensor data with missing observations. The key issue of pre-specifying the rank of the decomposition is sidestepped in a principled manner using a multiplicative gamma process prior. Both continuous and binary data can be analyzed under the framework, in a coherent way using fully conjugate Bayesian analysis. In particular, the analysis in the non-conjugate binary case is facilitated via the use of the Pólya-Gamma sampling strategy which elicits closed-form Gibbs sampling updates. The resulting samplers are efficient and enable us to apply our framework to large-scale problems, with time-complexity that is *linear* in the number of observed entries in the tensor. This is especially attractive in analyzing very large but sparsely observed tensors with very few known entries. Our method outperforms several state-of-the-art tensor decomposition methods on various synthetic and benchmark real-world datasets, achieving excellent scalability.

1. Introduction

Sparsely observed multiway data are now routinely collected in various application domains such as multirelational social networks (Nickel et al., 2011), recommender systems (Xiong et al., 2010), contingency table analyses (Zhou et al., 2013), brain-computer imaging (Cichocki, 2013), and chemometrics (Mørup & Hansen, 2009), among many others. The prevalence of such data necessitates developing flexible and scalable methods to analyze them. Multiway tensor decomposition methods based on the low-rank tensor approximation (Kolda & Bader, 2009) offer an attractive way to extract useful information from such data, by providing a concise representation that captures the salient characteristics of the data. Of particular interest are methods that can analyze large-scale but incomplete, sparsely observed data where a significant fraction of the entries is missing (Acar et al., 2011).

Probabilistic tensor decomposition methods (Chu & Ghahramani, 2009), in particular Bayesian tensor decomposition methods (Xu et al., 2013; Xiong et al., 2010) are naturally appealing since they provide a principled mechanism for dealing with missing data, allow analysis of diverse data types (continuous, binary, ordinal, etc.) using suitable likelihood models, and make it possible to quantify the uncertainty in the parameter estimates and the predictions (when dealing with missing data). Unfortunately, these methods require that the rank of the decomposition is specified prior to the analysis. The rank-estimation problem is further confounded in the case of tensor data for which rank determination is known to be an NP-hard problem (Hastad, 1990). Finally, scalability is another concern when applying these methods. Inference via MCMC or variational methods can be slow as the tensor size becomes large (in the number of observed entries, in the number/dimensions of tensor modes, or in all of these).

Motivated by these, we present a flexible and scalable nonparametric Bayesian tensor decomposition method for analyzing multiway tensor data. Our method has the following key properties: (1) The tensor rank does not have to be specified beforehand and is learned *adaptively* from the data in a principled way using the theoretically motivated multiplicative gamma process prior (Bhattacharya & Dunson, 2011) on the elements of the core diagonal tensor in

[*]contributed equally

Proceedings of the 31[st] International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

the CANDECOMP/PARAFAC (CP) low-rank decomposition of tensors (Kolda & Bader, 2009); (2) Both continuous and binary datasets can be analyzed using a fully Bayesian framework, via simple closed-form Gibbs sampling updates; (3) Inference scales linearly with the number of observed entries in the tensor, which makes inference highly scalable for large but sparsely observed multiway datasets, commonly encountered in application domains such as multirelational networks, recommender systems, etc. Even on *non-sparse* tensors, our framework, capable of dealing with large amounts of missing data, allows us to use a very small fraction of the entire data while achieving reconstruction quality that is close to using the complete data (our experimental results on tasks such as image inpainting corroborate this). Our framework is therefore also scalable for analyzing large-scale *dense* tensors.

2. Low-Rank Tensor Decomposition

In this section, we present our framework for low-rank tensor decomposition based on the CP decomposition (Kolda & Bader, 2009). We infer the rank by placing a shrinkage prior, the multiplicative gamma process (MGP) (Bhattacharya & Dunson, 2011), over the superdiagonal elements of the *core tensor* (Λ in Figure 1) in the CP decomposition. The MGP prior adaptively learns the appropriate number of component tensors, and leads to an efficient low-rank approximation of the tensor.

2.1. CP Decomposition of Tensor

The CANDECOMP/PARAFAC (CP) decomposition decomposes a tensor into a sum of rank-1 *component tensors* (Kolda & Bader, 2009). A K-way (or K-mode) tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$, with the integer n_k being the dimension of $\mathcal{X}$ along the k^{th} way, can be represented in its CP decomposition form:

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

where the vector $\boldsymbol{u}_r^{(k)} \in \mathbb{R}^{n_k}$ and '$\circ$' denotes the vector outer product. Here R is referred to as the *rank* of the tensor $\mathcal{X}$. With the CP decomposition as given in (1), the tensor element $x_{\boldsymbol{i}}$, with $\boldsymbol{i} = [i_1, i_2, \cdots, i_K]$ its K-dimensional index vector, can be concisely represented by:

$$x_{\boldsymbol{i}} = \sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)} \tag{2}$$

Denote by $U^{(k)} = [\boldsymbol{u}_1^{(k)}, \boldsymbol{u}_2^{(k)}, \cdots, \boldsymbol{u}_R^{(k)}]$, $k = 1, 2, \cdots, K$, the $n_k \times R$ *factor matrix* of the k-th mode of the tensor. When a vector form of the tensor $\mathcal{X}$ is desired, the above CP decomposition can be written as:

$$vec(\mathcal{X}) = U^{(1)} \odot U^{(2)} \odot \cdots \odot U^{(K)} \cdot \boldsymbol{\lambda} \tag{3}$$

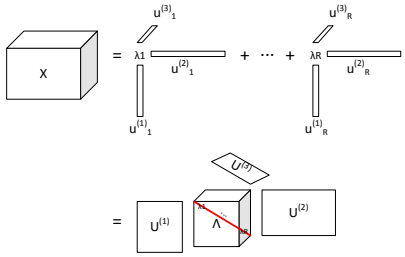

Figure 1. The CP decomposition of tensors (a three-mode tensor shown for illustration).

where $\odot$ denotes the Khatri-Rao product and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_R]$ denotes the vector along the superdiagonal of the core tensor.

2.2. Rank Specification

The CP decomposition yields a concise representation of tensors. However, the tradeoff of the conciseness is that in the CP decomposition the rank of the tensor being decomposed needs to be pre-specified. However, rank estimation for tensors is in general an NP hard problem (Hastad, 1990). To avoid the burdensome rank estimation task, a reasonable solution is to express the original tensor with a "good-enough" low-rank approximation; for example, in the sense of the Frobenius norm. But unfortunately, unlike the 2-way (matrix) cases, where the low-rank approximation is completely solved with the Eckart-Young theorem (Eckart & Young, 1936), for tensors the low-rank approximation can often be an ill-posed problem as discussed in (de Silva & Lim, 2008).

Such theoretical dilemma inspires alternative solutions for low-rank approximation of tensors. Instead of relying on ad-hoc or cumbersome model selection methods such as AIC, BIC, or the marginal likelihood, we turn to the nonparametric Bayesian modeling paradigm to adaptively infer the rank of the tensor being decomposed (or a close approximation of the rank necessary to obtain a sufficiently good low-rank approximation for a given dataset). In particular, we propose a nonparametric Bayesian low-rank CP decomposition for tensors based on the theoretically well motivated multiplicative gamma process (Bhattacharya & Dunson, 2011) prior (MGP) construction to infer the rank, as opposed to priors such as the Indian Buffet Process (Griffiths & Ghahramani, 2011) for which inference can be complicated/slow. As we show subsequently, the shrinkage property of the MGP leads to fully conjugate models in both continuous and binary data cases and allows us to derive simple, closed-form Gibbs sampling updates for all model parameters. For the binary case in particular, the conjugacy is achieved via the Pólya-Gamma sampling strategy (Polson et al., 2012) which elicits a closed-form Gibbs sampler.

2.3. CP Decomposition with MGP

Our low-rank tensor decomposition model construction is based on the multiplicative gamma process (MGP), originally proposed in the context of factor analysis of matrix data (Bhattacharya & Dunson, 2011). In (Bhattacharya & Dunson, 2011), this prior was employed on the columns of the factor loading matrix, such that the columns increasingly shrink to zero as the column index increases. We generalize this construction for the multi-way tensor case. Crucially, different from the construction used in (Bhattacharya & Dunson, 2011), for the low-rank decomposition of tensors we put the MGP prior on the superdiagonal elements $\boldsymbol{\lambda}$ of the core tensor Λ. This greatly reduces the number of parameters to be estimated in the tensor case. We denote this CP decomposition driven by the MGP as MGP-CP. The MGP prior is represented by:

$$\begin{aligned} &\lambda_r \sim \mathcal{N}(0, \tau_r^{-1}), \quad 1 \leq r \leq R \\ &\tau_r = \prod_{l=1}^{r} \delta_l, \quad \delta_l \sim Ga(a_c, 1) \quad a_c > 1 \end{aligned} \tag{4}$$

The multiplicative gamma process prior described in (4) on the precision of the Gaussian distribution for λ_r will shrink the λ_r towards zero as r increases. The appropriate rank R under our MGP based tensor decomposition model can be inferred two ways: (i) using a reasonably large truncation level, or (ii) using an adaptation strategy (discussed in Section 3.2) which allows growing or shrinking ranks as inference progresses. We refer to the truncation-based version as MGP-CP^t and the adaptation-based version as MGP-CP^a.

We assume that for each mode of the tensor, the R columns $\boldsymbol{u}_r^{(k)}$ of the factor matrix $U^{(k)}$, are drawn from a Gaussian distribution:

$$\boldsymbol{u}_r^{(k)} \sim N(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}), \; 1 < r \leq R, \; 1 < k \leq K \tag{5}$$

where $\boldsymbol{\mu}^{(k)}$ and $\Sigma^{(k)}$ are the mean vector and covariance matrix of the Gaussian distribution of the k^{th} tensor mode. Then the covariance between any two elements in the tensor $\mathcal{X}$, $x_{\boldsymbol{i}}$ and $x_{\boldsymbol{j}}$ conditioned on the factor matrices is:

$$\text{Cov}(x_{\boldsymbol{i}}, x_{\boldsymbol{j}} | \{U^{(k)}\}_{k=1}^K) = \sum_{r=1}^{R} \tau_r^{-1} \prod_{k=1}^{K} u_{i_k r}^{(k)} u_{j_k r}^{(k)} \tag{6}$$

In (6) we observe that the covariance is structured as the sum of the covariances associated with each rank-one component tensor, indicating that each component tensor stands for an independent significant factor constituting the data.

We now state some results characterizing the bound on the approximation error for the truncation based MGP-CP which show that the approximation error decreases exponentially fast as the truncation level tends to infinity (proofs are given in the supplementary material).

Theorem 1. *With $a_c > 1$, the sequence $\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}$ converges in ℓ_2 as $R \to \infty$.*

Theorem 2. *Denote the residual by $M_{i_1 i_2 \ldots i_k}^R = \sum_{r=R+1}^{\infty} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}$. Then $\forall \epsilon > 0$ we have $P\{(M_{i_1 i_2 \ldots i_k}^R)^2 > \epsilon\} < \frac{1}{\epsilon a_c^R (a_c - 1)} \prod_{k=1}^{K} \zeta_k$, where ζ_k is defined such that $\mathbb{E}(u_{i_k r}^{(k)})^2$ is bounded by $\zeta_k < \infty$.*

3. Model and Inference

3.1. Model Description

The goal of inference in our model is to infer the parameters of the CP decomposition, Λ, $U^{(1)}, U^{(2)}, \cdots, U^{(K)}$, based on potentially a very limited (sparse) set of observations $\mathcal{Y} = \{y_{\boldsymbol{i}}\}_{\boldsymbol{i} \in I}$, where I is the index set of all the observations, and $N = |\mathcal{Y}|$ the number of these observations. Following the MGP-CP model given in (4) and (5), the prior, $p(\Lambda, \{U^{(k)}\}_{k=1}^K)$, is given below

$$\prod_{r=1}^{R} \mathcal{N}(\lambda_r | 0, \tau_r^{-1}) \text{Ga}(\delta_r | a_r, 1) \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{u}_r^{(k)} | \boldsymbol{\mu}_r^{(k)}, \Sigma_r^{(k)}) \tag{7}$$

We further assume that the covariance matrices $\Sigma_r^{(k)}$'s are diagonal, which amounts to assuming that the entities in each tensor mode are *a priori* independent of each other.

Two types of likelihood models are considered based on different types of real-world data: continuous and binary data. The observations $\mathcal{Y}$ are assumed to be i.i.d. For the continuous observations with Gaussian noise, where τ_ϵ is the precision, the model likelihood is given by

$$p(\mathcal{Y} | \mathcal{X}) = \prod_{\boldsymbol{i}} \mathcal{N}(y_{\boldsymbol{i}} | x_{\boldsymbol{i}}, \tau_\epsilon^{-1}) \tag{8}$$

For binary-valued data ($e.g.$, relational data) the logistic link function is applied:

$$p(\mathcal{Y} | \mathcal{X}) = \prod_{\boldsymbol{i}} \left(\frac{1}{1 + \mathrm{e}^{-x_{\boldsymbol{i}}}}\right)^{y_{\boldsymbol{i}}} \left(\frac{\mathrm{e}^{-x_{\boldsymbol{i}}}}{1 + \mathrm{e}^{-x_{\boldsymbol{i}}}}\right)^{1 - y_{\boldsymbol{i}}} \tag{9}$$

3.2. Inference via Gibbs Sampling

For continuous data, our model construction with prior in (7) and likelihood model in (8) is locally conjugate and a Gibbs sampler can easily be derived for all the model parameters. For the binary case, the logistic likelihood in (9) is not conjugate to the prior in (7). To achieve conjugacy in the binary case, we use the Pólya-Gamma sampling strategy (Polson et al., 2012), which allows us to derive a fully analytic Gibbs sampler in the binary case as well.

The Gibbs sampling update equations for the various model parameters $\{\delta_r\}_{r=1}^R$, $\{\lambda_r\}_{r=1}^R$, and $\{U^{(k)}\}_{k=1}^K$,

given continuous or binary observations $\mathcal{Y}$, are as follows:

(i) For the update of the MGP, for $\delta_r, 1 \le r \le R$:

$$\delta_r \sim \text{Ga}(a_c + \frac{1}{2}(R-r+1), 1 + \frac{1}{2}\sum_{h=r}^{R}\lambda_h^2 \prod_{l=1, l\neq r}^{h}\delta_l) \quad (10)$$

(ii) When the observations $\mathcal{Y}$ are real and the likelihood is given as in (8), for the update of $\lambda_r, 1 \le r \le R$:

$$x_{\boldsymbol{i}} = (\prod_{k=1}^{K} u^{(k)}_{i_k r})\lambda_r + (\sum_{r'\neq r}\lambda_{r'}\prod_{k=1}^{K} u^{(k)}_{i_k r'}) = a^r_{\boldsymbol{i}}\lambda_r + b^r_{\boldsymbol{i}} \quad (11)$$

$$\begin{aligned} &\lambda_r \sim \mathcal{N}(\hat{\mu}_r, \hat{\tau}_r^{-1}), \quad \hat{\tau}_r = \tau_r + \tau_\epsilon \sum_{\boldsymbol{i}} {a^r_{\boldsymbol{i}}}^2 \\ &\hat{\mu}_r = \hat{\tau}_r^{-1}\tau_\epsilon \sum_{\boldsymbol{i}} a^r_{\boldsymbol{i}}(y_{\boldsymbol{i}} - b^r_{\boldsymbol{i}}) \end{aligned} \quad (12)$$

For the update of $\boldsymbol{u}^{(k)}_r, 1 \le r \le R, 1 \le k \le K$, denote:

$$\begin{aligned} x_{\boldsymbol{i}} &= (\lambda_r \prod_{k'\neq k} u^{(k)}_{i_{k'} r})u^{(k)}_{i_k r} + (\sum_{r'\neq r}\lambda_{r'}\prod_{k=1}^{K} u^{(k)}_{i_k r'}) \\ &= c^{(k)}_{i_k r}u^{(k)}_{i_k r} + d^{(k)}_{i_k r} \end{aligned} \quad (13)$$

Then $\boldsymbol{u}^{(k)}_r \sim \mathcal{N}(\hat{\boldsymbol{\mu}}^{(k)}_r, \hat{\Sigma}^{(k)}_r)$ with:

$$\begin{aligned} \hat{\Sigma}^{(k)}_r &= ({\Sigma^{(k)}}^{-1} + T^{(k)}_r)^{-1} \\ T^{(k)}_r &= \text{diag}(\tau^{(k)}_{1r}, \tau^{(k)}_{2r}, \cdots, \tau^{(k)}_{n_k r}) \\ \tau^{(k)}_{nr} &= \tau_\epsilon \sum_{\boldsymbol{i}, i_k = n} {c^{(k)}_{i_k r}}^2, \quad 1 \le n \le n_k \end{aligned} \quad (14)$$

$$\begin{aligned} &\hat{\boldsymbol{\mu}}^{(k)}_r = \hat{\Sigma}^{(k)}_r({\Sigma^{(k)}}^{-1}\boldsymbol{\mu}^{(k)} + T^{(k)}_r\boldsymbol{\alpha}^{(k)}_r) \\ &\boldsymbol{\alpha}^{(k)}_r = [\alpha^{(k)}_{1r}, \alpha^{(k)}_{2r}, \cdots, \alpha^{(k)}_{n_k r}]^\top, \text{for } 1 \le n \le n_k: \\ &\alpha^{(k)}_{nr} = (\tau^{(k)}_{nr})^{-1}\tau_\epsilon \sum_{\boldsymbol{i}, i_k = n} c^{(k)}_{i_k r}(y_{\boldsymbol{i}} - d^{(k)}_{i_k r}) \end{aligned} \quad (15)$$

Additionally we put a gamma prior on the noise precision $\tau_\epsilon \sim Ga(a_0, b_0)$, with the posterior $\hat{\tau}_\epsilon \sim Ga(a_0 + \frac{1}{2}N, b_0 + \frac{1}{2}\sum_{\boldsymbol{i}}(x_{\boldsymbol{i}} - \hat{x}_{\boldsymbol{i}})^2)$, with $\hat{x}_{\boldsymbol{i}}$ is the estimation of $x_{\boldsymbol{i}}$ reconstructed by following (2).

(iii) When the observations $\mathcal{Y}$ are binary and the likelihood is given as in (9), the model is a latent Gaussian model (LGM) with Logit likelihood. We apply the recent result of (Polson et al., 2012) which elicits a conjugate Gibbs sampler. For the update of $\lambda_r, 1 \le r \le R$, with (11) we have, $\lambda_r = \frac{1}{a^r_{\boldsymbol{i}}}x_{\boldsymbol{i}} - \frac{b^r_{\boldsymbol{i}}}{a^r_{\boldsymbol{i}}}$. Then the augment random variable $\phi_{\boldsymbol{i}}$ are drawn independently from the Pólya-Gamma distribution:

$$\phi_{\boldsymbol{i}} \sim \text{PG}(1, x_{\boldsymbol{i}}) \quad (16)$$

where $\text{PG}(\cdot, \cdot)$ represents the Pólya-Gamma distribution. Then $\lambda_r, 1 \le r \le R$ is drawn from Gaussian:

$$\begin{aligned} &\lambda_r \sim \mathcal{N}(\hat{\mu}_r, \hat{\tau}_r^{-1}), \quad \hat{\tau}_r = \tau_r + \sum_{\boldsymbol{i}} {a^r_{\boldsymbol{i}}}^2\phi_{\boldsymbol{i}} \\ &\hat{\mu}_r = \hat{\tau}_r^{-1}\sum_{\boldsymbol{i}} a^r_{\boldsymbol{i}}(y_{\boldsymbol{i}} - 0.5 - \phi_{\boldsymbol{i}}b^r_{\boldsymbol{i}}) \end{aligned} \quad (17)$$

For the update of $\boldsymbol{u}^{(k)}_r, 1 \le r \le R, 1 \le k \le K$, with (13) we have: $u^{(k)}_{i_k r} = \frac{1}{c^{(k)}_{i_k r}}x_{\boldsymbol{i}} - \frac{d^{(k)}_{i_k r}}{c^{(k)}_{i_k r}}$. Then $\boldsymbol{u}^{(k)}_r \sim \mathcal{N}(\hat{\boldsymbol{\mu}}^{(k)}_r, \hat{\Sigma}^{(k)}_r)$ with the $\hat{\boldsymbol{\mu}}^{(k)}_r, \hat{\Sigma}^{(k)}_r$ given as same as in (14) and (15), but $\tau^{(k)}_{nr}$, $\alpha^{(k)}_{nr}$ changed. For $1 \le n \le n_k$:

$$\begin{aligned} \tau^{(k)}_{nr} &= \sum_{\boldsymbol{i}, i_k = n} {c^{(k)}_{i_k r}}^2\phi_{\boldsymbol{i}}, \quad 1 \le n \le n_k \\ \alpha^{(k)}_{nr} &= (\tau^{(k)}_{nr})^{-1}\sum_{\boldsymbol{i}, i_k = n} c^{(k)}_{i_k r}(y_{\boldsymbol{i}} - 0.5 - \phi_{\boldsymbol{i}}d^{(k)}_{i_k r}) \end{aligned} \quad (18)$$

Adaptation Strategy: In our truncation based variant, MGP-CP^t, we run the Gibbs sampler using a reasonably large truncation level R and, as inference progresses, only relevant components have significant contribution to the model, with the λ_r for the rest shrinking to values close to zero. In our adaptive variant, MGP-CP^a, whenever λ_r becomes smaller than a predefined threshold ϵ (say 0.001), the component tensors with $|\lambda_r| < \epsilon$ are removed from the model; otherwise if all the $|\lambda_r| > th$, a new component tensor is added. Such adaptation occurs with probability $p(t) = \exp(\beta_0 + \beta_1 t)$ at the t^{th} iteration, with β_0, β_1 chosen so that adaptation occurs around every 10 iterations at the beginning of the chain but decreases in frequency exponentially fast (Bhattacharya & Dunson, 2011). The simple strategy of thresholding based on the absolute values of λ_r worked well in all of our experiments. Other criteria can also be used to decide whether to discard a rank-1 component or to add a new component. For example, in the continuous -data case, one possible strategy would be to monitor the explained variances by each rank-1 component on some held-out data and if the contribution of certain rank-1 components to the total explained variance drops below a very small value (say <1% of the explained variance), we drop them (otherwise we add a new component based on the adaptation probability $p(t)$). In the binary data case, we can likewise monitor the contributions by each rank-1 component to the predictive probabilities of all the observations and drop components which are *non-informative*, *e.g.*, if the empirical distribution estimated using the predictive probabilities of all the observations is close to a uniform distribution *and* if the mean of the empirical distribution is close to 0.5 (otherwise we add a new component based on $p(t)$).

3.3. Computational Complexity

The per-iteration computational cost of our inference algorithm is *linear* in the number of observation N. For sparse tensors N is considerably smaller than the tensor size $L = \prod_{k=1}^{K} n_k$. The individual contributions to the overall time complexity are as follows: (i) sampling each δ_r, $r = \{1, \ldots, R\}$ takes $O(R^2)$ time leading to a time-complexity $O(R^3)$; (ii) sampling each λ_r, $r = \{1, \ldots, R\}$ takes $O(NK)$ time leading to a time-complexity $O(NRK)$; (iii) sampling each $\boldsymbol{u}_r^{(k)}$, $r = \{1, \ldots, R\}, k = \{1, \ldots, K\}$ takes $O(NRK)$ time leading to a time-complexity $O(NK^2R^2)$. Note that no explicit matrix inversions are involved in our inference procedure since the covariance of the prior on $\boldsymbol{u}_r^{(k)}$ is assumed to be diagonal and therefore the computations in (14) can be performed in $O(n_k)$ time. The overall time-complexity is dominated by the third term $O(NR^2K^2)$ which is linear in the number of observations N. This is especially encouraging for a sampling based inference method.

The linear scalability of our method in N is appealing since real-world tensor datasets tend to be extremely sparse ($N \ll L$). For example, in most social network datasets, there are less than 0.1% observed interactions. Our experiments corroborate the linear scalability behavior (Section 5.5, Figure 4) on a sparsely observed tensor of size $1000 \times 1000 \times 1000$ for which L is 1 billion.

4. Related Work

With the advent of social networks and multirelational/multiway data observed in many application domains, tensor decomposition methods have gained much attention recently. Although a number of tensor decomposition methods have been proposed, many state-of-the-art methods (Nickel et al., 2011; Bordes et al., 2012; Jenatton et al., 2012) are specialized for analyzing three-mode tensor data and do not generalize to higher-order tensors.

Tensor decomposition methods that can infer the rank are relatively few. Among the probabilistic approaches, one option is to use the Automatic Relevance Determination (ARD) method (Mørup & Hansen, 2009; Zhao et al., 2014). We use this method as a baseline in our experiments. Some non-probabilistic methods for tensor decomposition employ trace-norm regularization (Tomioka et al., 2010) to get an approximation of the tensor rank. In another recent work, nuclear-norm based rank-regularization (Bazerque et al., 2013) is used to infer the rank in a probabilistic tensor factorization model and inference is based on MAP estimation. All of these methods assume that the observations are real-valued unlike our method which can deal with both real and binary data.

A nonparametric Bayesian method similar in spirit to ours is presented in (Dunson & Xing, 2012), where a stick-breaking prior is put on the superdiagonal of Λ in the CP decomposition of a probability tensor (a special type of tensor whose entries sum to 1), to nonparametrically learn a low-rank representation through inference. The main consideration for applying the stick-breaking process prior in (Dunson & Xing, 2012) comes from its statistically decreasing property and the norm one requirement of the specific type of tensors (three-mode probability tensor). In another recent work, (Yoshii et al., 2013) proposed a positive semidefinite tensor factorization (PSDTF) which corresponds to the CP decomposition where the rank is learned with a truncation level put on the gamma random variables along the superdiagonal of Λ. However, this method is also limited to special types of 3-way tensors where each slice is a positive semidefinite matrix. A potential alternative is to put a gamma process along the superdiagonal is the construction of the gamma process discussed in (Wang & Carin, 2012), where the statistically decreasing samples facilitate the nonparametric learning of the required rank. However, the resulting model will not conjugate with this choice of the prior.

Tensor decomposition methods that explicitly model binary data are also relatively few. The recently proposed Infinite Tucker Decomposition method (Xu et al., 2013) uses a probit model for binary data. We compare with this method in our experiments. Another recent work on modeling binary tensor data is a logistic loss based extension of the non-probabilistic RESCAL model (Nickel & Tresp, 2013). This is, however, limited to three-mode tensor data.

5. Experiments

We perform experiments with our model on both synthetic and real-world tensor datasets, and compare it with several baselines. The datasets used in our experiments span a wide range of application domains, such as chemometrics, multirelational social networks, brain-signal analysis (EEG), and image analysis. We experiment with both variants of our model: truncated MGP (referred to as MGP-CP^t) and the adaptive MGP (referred to as MGP-CP^a). For both methods, we use the Gaussian likelihood model for continuous data and the logistic model (with Pólya Gamma sampling during inference) for binary data.

The following baselines were used for comparisons. (i) Bayesian CP (BCP), a fully Bayesian version of the standard probabilistic CP decomposition (Xiong et al., 2010). It assumes that the rank is known. (ii) ARD based CP (ARD-CP), a method that uses automatic relevance determination (ARD) (Mørup & Hansen, 2009; Zhao et al., 2014) to determine the rank of a tensor by inferring the relevant columns in the factor matrix of each mode. (iii) An Infinite Tucker Decomposition based on t process (InfTucker^{tp}),

	Synthetic Data (R=10)	Amino Acid	Flow-injection	EEG Data
Bayesian CP	0.1231 (±0.0278)	**0.0004 (±0.0001)**	0.0012 (±0.0002)	0.1760 (±0.0032)
ARD-CP	0.0921 (±0.0006)	0.0350 (±0.0535)	0.0196 (±0.0133)	0.1860 (±0.0050)
InfTucker^{tp}	0.6644 (±0.0136)	0.8478 (±0.0103)	0.8382 (±0.0080)	0.5394 (±0.0823)
MGP-CP^{t}	**0.0922 (±0.0011)**	0.0006 (±0.0001)	0.0007 (±0.0003)	0.1622 (±0.0016)
MGP-CP^{a}	0.0935 (±0.0007)	0.0005 (±0.0001)	**0.0005 (±0.0003)**	**0.1608 (±0.0033)**

Table 1. Continuous Data: MSE

which is a kernel-based nonparametric Bayesian generalization of the low-rank Tucker decomposition(Xu et al., 2013), and is based on an *implicit* mapping of the component tensors to a higher (potentially infinite) dimensional space and performing a low-rank Tucker decomposition in that space. This method requires that the rank is given.

We evaluate our model and the various baselines on the following experiments: (i) tensor completion for continuous data, (ii) tensor completion for binary data, (iii) SVM based classification for EEG data using factors learned by different tensor decomposition methods, and (iv) image inpainting for color images by posing it as tensor completion problem for continuous data.

We initialize the MGP-CP^{a} using an initial rank = 1 and allow the rank to grow/shrink using our adaptation strategy discussed in Section 2.3. For MGP-CP^{t} and the ARD-CP baseline, we set the truncation level to a sufficiently large value. We run the sampling based methods for 1500 iterations with 1000 burn-in iterations, collect samples every 5 iterations after the burn-in phase, and report all results using the posterior sample based averages. For Bayesian CP and InfTucker^{tp}, which require the rank to be specified, we vary the ranks over a range and report the results using the rank that gave the best held-out data predictions.

5.1. Low-rank Tensor Completion: Continuous Data

We first experiment on the tensor completion task for continuous data. For this experiment, we use four datasets: (i) Synthetic data of size $20 \times 20 \times 20 \times 20$, generated as an equally-weighted sum of 10 rank-1 tensors of the same size (so the ground-truth rank is 10). (ii) Amino Acid data (Xu et al., 2013; Chu & Ghahramani, 2009) of size $5 \times 61 \times 201$, consisting of five laboratory-made amino acid samples. (iii) Flow Injection data (Xu et al., 2013; Chu & Ghahramani, 2009) of size $12 \times 100 \times 89$ obtained from a flow injection analysis (FIA) system. (iv) EEG data of size $15 \times 16 \times 560$ consisting of EEG measurements of 560 subjects. For this task, we treat 50% of the data as missing and reconstruct it using the model learned on the remaining 50% data. We report the results in terms of the mean-squared-error (MSE) on the reconstruction task. Each experiment is repeated 10 times with different splits of observed and missing data.

The results are shown in Table 1. Both our models achieve reconstruction accuracies comparable to or better than the gold-standard Bayesian CP (which was given the ground-truth rank for synthetic data, and best rank chosen via held-out error on real-world datasets - 5 for Amino Acid, 6 for Flow-injection, 30 for EEG data). Moreover, on all datasets, both our models perform better than ARD-CP and InfTucker^{tp}.

To see whether our method can recover the true underlying rank, we run MGP-CP^{a} on the $20 \times 20 \times 20 \times 20$ synthetic data having a ground-truth rank 10, first with 90% missing data and then with 50% missing data. Figure 2 shows the posterior distribution of the inferred rank (based on the estimated empirical distribution of the ranks using posterior samples after the burn-in phase). As shown in the figure, in both cases, the posterior is concentrated at rank 10 and as the amount of training data increases from 90% missing to 50% missing, the posterior peaks further at rank 10. On real-world datasets, our method discovers ranks that are consistent with what is known from domain knowledge in the chemometrics literature on analyzing these datasets (our method infers the rank to be 3-4 on average on Amino Acid data and 6-7 on average on Flow Injection data).

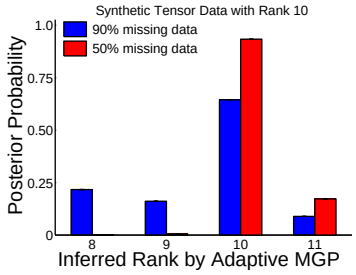

Figure 2. Empirical distribution of the inferred rank by MGP-CP^{a} run with 90% and 50% missing data (starting with $R = 1$)

5.2. Low-rank Tensor Completion: Binary Data

We next experiment with tensor completion for binary tensor data. We use four binary datasets for this experiment: (i) Synthetic data of size $20 \times 20 \times 20 \times 20$ having a ground-truth rank 10 (about 1.6% non-zero entries). (ii) Lazega-Lawyers multirelational social network data (Lazega, 2001) given in the form of a tensor of size

	Synthetic Data (R=10)	Lazega Lawyers	Kinship	Nation
Bayesian CP	0.6997 ($\pm$0.0434)	0.5671 ($\pm$0.0243)	0.9754 ($\pm$0.0022)	0.7230 ($\pm$0.0344)
ARD-CP	0.6045 ($\pm$0.0461)	0.5542 ($\pm$0.0378)	0.9842 ($\pm$0.0019)	0.6698 ($\pm$0.0527)
InfTucker^{tp}	0.8759 ($\pm$0.0143)	0.5982 ($\pm$0.0179)	0.9825 ($\pm$0.0022)	0.7981 ($\pm$0.0133)
MGP-CP^{t}	**0.9288 ($\pm$0.0140)**	0.6412 ($\pm$0.0101)	0.9896 ($\pm$0.0014)	**0.8105 ($\pm$0.0083)**
MGP-CP^{a}	0.9283 ($\pm$0.0109)	**0.6448 ($\pm$0.0139)**	**0.9909 ($\pm$0.0015)**	0.8096 ($\pm$0.0082)

Table 2. Binary Data: AUC Scores

$71 \times 71 \times 3$ (about 15% non-zero entries) containing three types of social networks (friendship, coworker, and advisory relationships) between 71 partners and associates in several New England law firms. (iii) Kinship multirelational data (Nickel et al., 2011) of size $104 \times 104 \times 26$ (about 3.84% non-zero entries) containing 26 types of kinship relations within the Alwayarra tribe. (iv) Nation multirelational data (Nickel et al., 2011) given in the form of a tensor of size $14 \times 14 \times 56$ (about 19% non-zero entries) containing 57 types of relationships ($e.g.$, export, protests, economic aid, etc.) among 14 countries. For each dataset, except Kinship, we treat 90% of the entries as missing and predict them using the rest 10% data. For Kinship data we use the experimental setting of 90% training and 10% test data as done in other recent works (Nickel et al., 2011; Jenatton et al., 2012). We use the area under the receiver-operating characteristic curve (AUC) score to compare the different methods in terms of their predictive ability. Each experiment is repeated 10 times with different splits of observed and missing data.

As the results in Table 2 show, both our methods outperform the other baselines in terms of the AUC scores. It is noteworthy to see that on binary data the improvements of our methods over Bayesian CP are much more significant than the continuous-data case (even though the Bayesian CP baseline is provided with the ground-truth rank for the synthetic data and the best chosen rank based on held-out error for the real-world data). This can be attributed to the fact that Bayesian CP uses least-square minimization whereas our methods use the logistic loss. Because of this, for datasets having a significant number of zero entries (like the ones used in the experiments here), the Bayesian CP will tend to be biased towards predicting zeros. The ARD-CP baseline, although in principle able to infer the rank, suffers due to squared-loss minimization like Bayesian CP. InfTucker^{tp}, the next-best performing method, uses the logistic loss like our method; however, it relies on variational EM for inference and is prone to local-optimal issues (besides having to select the rank via cross-validation).

5.3. Binary Classification with Extracted Factors

We now experiment on an extrinsic evaluation task: binary classification using the factors learned via tensor decomposition. On the EEG data used in Section 5.1 for tensor completion experiments, we also have binary labels for each of the 560 subjects. We conduct an SVM based classification experiment where the factors extracted by various tensor decomposition methods are used to train an SVM. The tensor decomposition step uses only 50% of the total data and the 3^{rd} mode factors (the 3^{rd} mode represents the subjects) are used to train an SVM. After the tensor decomposition step, we use 10% of the subjects to train the SVM (using the extracted factors) and test on the remaining 90% subjects (to simulate a small sample size setting where a naïve approach of flattening the $15 \times 16 \times 560$ tensor a *matrix* could overfit). Because the tensor methods use only 50% data in the factor extraction stage, in the SVM experiment with the flattened tensor which resulted in a matrix of size 560×240, we hide 50% of the features and impute them using the respective feature means.

We repeat the classification experiment 20 times with different splits of training and test data. As Table 3 shows, the tensor decomposition methods perform better than SVM on flattened tensor which seems to overfit due to small sample size. Among the various tensor-decomposition-based methods, MGP-CP^{a} yields the best classification accuracy.

	Classification Accuracy
SVM (on flattened tensor)	65.02% ($\pm$3.10%)
Bayesian CP + SVM	67.95% ($\pm$4.39%)
ARD-CP + SVM	72.26% ($\pm$3.03%)
Infinite Tucker + SVM	66.53% ($\pm$3.32%)
MGP-CP^{t} + SVM	72.32% ($\pm$3.54%)
MGP-CP^{a} + SVM	**74.57% ($\pm$2.42%)**

Table 3. Binary classification using factors learned from tensor decomposition

5.4. Image Inpainting

Image inpainting is the task of completing an image with missing pixels. A two-dimensional RGB image can be treated as a three-dimensional tensor and the image inpainting task can be formulated as a tensor completion problem where the goal is to predict the values of the missing pixels using the observed pixel values. We apply our methods and the other baselines on this task using the benchmark Lena image of size $256 \times 256 \times 3$ for various fraction

Figure 3. **Top row:** 90% missing. left: original, right: reconstructed. **Middle row:** 80% missing. left: original, right: reconstructed. **Bottom row:** 50% missing. left: original, right: reconstructed.

of missing pixels (90% missing, 80% missing, and 50% missing). Bayesian CP and InfTucker^{tp} were run with R ranging from 5 to 50 and we report the result with the best reconstruction error. For ARD-CP and MGP-CP^t, the truncation level was set to 50. The MGP-CP^a was initialized with $R = 1$. In Table 4, the reconstruction accuracies for each case are shown, and the reconstructed images for each case using our MGP-CP^t model are shown in Figure 3. As shown in Table 4, both MGP-CP^t and MGP-CP^a outperform the other baselines on this task in all the three cases. As Figure 3 shows, our method can recover the underlying ground-truth image up to a very reasonable quality even when the percentage of missingness is very high.

	90%	**80%**	**50%**
Bayesian CP	0.0146	0.0099	0.0088
ARD-CP	0.0203	0.0197	0.0193
Infinite Tucker	0.2563	0.2106	0.1056
MGP-CP^t	0.0125	**0.0049**	**0.0023**
MGP-CP^a	**0.0102**	0.0057	0.0031

Table 4. Image Inpainting: Reconstruction errors (MSE) on different amounts (90%, 80%, and 50%) of missing pixels

5.5. Scalability

To assess the scalability of our method, we run an experiment on a large but sparsely observed synthetic tensor dataset of size $1000 \times 1000 \times 1000$ having 1 billion cells but sparsely observed such that only 1 million entries are known. For this dataset, we vary the number of observations from 0.2 million to 1 million and run the MGP-CP^t with a fixed truncation level (so R and K stay fixed and only N varies) for 200 iterations in each case. Even when using an unoptimized MATLAB implementation, using our method we are able to deal with datasets of this scale in a reasonable amount of time. As shown in Figure 4, our method scales linearly with the number of observations.

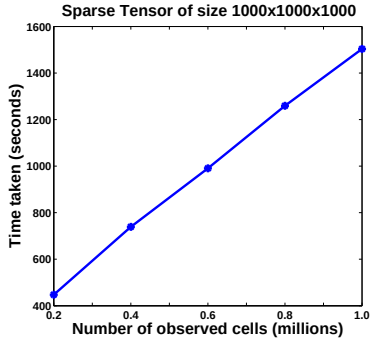

Figure 4. Linear scalability on a large-scale but sparse tensor

6. Conclusion

We have developed a flexible and scalable nonparametric Bayesian framework for analyzing multiway tensor data. Our framework is flexible as it does not require the tensor rank to be specified beforehand. The model can adapt its complexity (the rank of the decomposition which can grow or shrink as inference progresses) as apppropriate for the data under consideration. Our framework can naturally handle both continuous and binary datasets using suitable likelihood models. Bayesian inference can be efficiently done in both cases using closed-form Gibbs sampling which scales linearly in the number of observations in the tensor. The 2-way version of our model with binary observations can also be a scalable alternative to other state-of-the-art nonparametric Bayesian methods (Miller et al., 2009) for link-prediction in *single*-relational networks. Although in this work, we considered the CP decomposition with two specific likelihood models, integrating our tensor decomposition framework with other task-specific objectives (e.g., supervised classification or ranking for multiway data) could be another future avenue of work.

Acknowledgements: The research reported here was funded in part by ARO, DARPA, DOE, NGA and ONR. The authors would like to thank Feng Yan, Zenglin Xu, and Alan Qi for helpful discussions and sharing their code.

References

Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 2011.

Bazerque, J., Mateos, G., and Giannakis, G. Rank regularization and bayesian inference for tensor completion and extrapolation. *arXiv preprint arXiv:1301.7619*, 2013.

Bhattacharya, A. and Dunson, D. Sparse bayesian infinite factor models. *Biometrika*, 2011.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 2012.

Chu, W. and Ghahramani, Z. Probabilistic models for incomplete multi-dimensional arrays. In *AISTATS*, 2009.

Cichocki, A. Tensor decompositions: A new concept in brain data analysis? *arXiv preprint arXiv:1305.0395*, 2013.

de Silva, V. and Lim, L. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Analysis Applications*, 2008.

Dunson, D. and Xing, C. Nonparametric bayes modeling of multivariate categorical data. *JASA*, 2012.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936.

Griffiths, Thomas L and Ghahramani, Zoubin. The indian buffet process: An introduction and review. *JMLR*, 2011.

Hastad, J. Tensor rank is NP-complete. *Journal of Algorithms*, 1990.

Jenatton, R., Le Roux, N., Bordes, A., and Obozinski, G. A latent factor model for highly multi-relational data. In *NIPS*, 2012.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM review*, 2009.

Lazega, E. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

Miller, K., Jordan, M. I., and Griffiths, T. L. Nonparametric latent feature models for link prediction. In *NIPS*, 2009.

Mørup, M. and Hansen, L. K. Automatic relevance determination for multi-way models. *Journal of Chemometrics*, 2009.

Nickel, M. and Tresp, V. Logistic tensor factorization for multi-relational data. *arXiv preprint arXiv:1306.2084*, 2013.

Nickel, M., Tresp, V., and Kriegel, H. A three-way model for collective learning on multi-relational data. In *ICML*, 2011.

Polson, N., Scott, J., and Windle, J. Bayesian inference for logistic models using Polya-Gamma latent variables, http://arxiv.org/abs/1205.0310, 2012. URL http://arxiv.org/abs/1205.0310.

Tomioka, R., Hayashi, K., and Kashima, H. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

Wang, Y. and Carin, L. Levy measure decompositions for the beta and gamma processes. In *ICML*, 2012.

Xiong, L., Chen, X., Huang, T., Schneider, J. G., and Carbonell, J. G. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.

Xu, Z., Yan, F., and Qi, Y. Bayesian nonparametric models for multiway data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

Yoshii, K., Tomioka, R., Mochihashi, D., and Goto, M. Infinite positive semidefinite tensor factorization for source separation of mixture signals. In *ICML*, 2013.

Zhao, Q., Zhang, L., and Cichocki, A. Bayesian cp factorization of incomplete tensors with automatic rank determination. *arXiv preprint arXiv:1401.6497*, 2014.

Zhou, J., Bhattacharya, A., Herring, A., and Dunson, D. Bayesian factorizations of big sparse tensors. *arXiv preprint arXiv:1306.1598*, 2013.