# Scalable Bayesian Non-negative Tensor Factorization for Massive Count Data

Changwei Hu[1], Piyush Rai[1(✉)], Changyou Chen[1], Matthew Harding[2], and Lawrence Carin[1]

[1] Department of Electrical and Computer Engineering, Duke University, Durham, USA
{ch237,piyush.rai,cc448,lcarin}@duke.edu
[2] Sanford School of Public Policy and Department of Economics, Duke University, Durham, USA
matthew.harding@duke.edu

**Abstract.** We present a Bayesian non-negative tensor factorization model for count-valued tensor data, and develop scalable inference algorithms (both batch and online) for dealing with massive tensors. Our generative model can handle overdispersed counts as well as infer the rank of the decomposition. Moreover, leveraging a reparameterization of the Poisson distribution as a multinomial facilitates conjugacy in the model and enables simple and efficient Gibbs sampling and variational Bayes (VB) inference updates, with a computational cost that only depends on the number of nonzeros in the tensor. The model also provides a nice interpretability for the factors; in our model, each factor corresponds to a "topic". We develop a set of online inference algorithms that allow further scaling up the model to massive tensors, for which batch inference methods may be infeasible. We apply our framework on diverse real-world applications, such as *multiway* topic modeling on a scientific publications database, analyzing a political science data set, and analyzing a massive household transactions data set.

**Keywords:** Tensor factorization · Bayesian learning · Latent factor models · Count data · Online bayesian inference

## 1 Introduction

Discovering interpretable latent structures in complex multiway (tensor) data is an important problem when learning from polyadic relationships among multiple sets of objects. Tensor factorization [5,14] offers a promising way of extracting such latent structures. The inferred factors can be used to analyze objects in each mode of the tensor (e.g., via classification or clustering using the factors), or to do tensor completion.

Of particular interest, in the context of such data, are sparsely-observed *count-valued* tensors. Tensors are routinely encountered in many applications. For example, in analyzing a database of scientific publications, the data may be in form of a sparse four-way count-valued tensor (authors × words × journals

© Springer International Publishing Switzerland 2015
A. Appice et al. (Eds.): ECML PKDD 2015, Part II, LNAI 9285, pp. 53–70, 2015.
DOI: 10.1007/978-3-319-23525-7_4

× years). Another application where multiway count data is routinely encountered is the analysis of contingency tables [11] which represent the co-occurrence statistics of multiple sets of objects.

We present a scalable Bayesian model for analyzing such sparsely-observed tensor data. Our framework is based on a beta-negative binomial construction, which provides a principled generative model for tensors with sparse and potentially overdispersed count data, and produces a non-negative tensor factorization. In addition to performing non-negative tensor factorization and tensor completion for count-valued tensors, our model has the property that each latent factor inferred for a tensor mode also represents a *distribution* (or "topic", as in topic models) over the objects of that tensor mode; our model naturally accomplishes this by placing a Dirichlet prior over the columns of the factor matrix of each tensor mode. In addition to providing an expressive and interpretable model for analyzing sparse count-valued tensors, the model automatically infers the rank of the decomposition, which side-steps the crucial issue of pre-specifying the rank of the decomposition [14,18,22].

Our framework also consists of a set of batch and scalable online inference methods. Using a reparameterization of the Poisson distribution as a multinomial allows us to achieve conjugacy, which facilitates closed-form Gibbs sampling as well as variational Bayes (VB) inference. Moreover, we also develop two *online* inference algorithms - one based on online MCMC [7] and the other based on stochastic variational inference [9]. These inference algorithms enable scaling up the model to massive-sized tensor data.

One of the motivations behind our work is analyzing massive multiway data for tasks such as understanding thematic structures in scholarly databases (e.g., to design better recommender systems for scholars), understanding consumer behavior from shopping patterns of large demographies (e.g., to design better marketing and supply strategies), and understanding international relations in political science studies. In our experiments, we provide qualitative analyses for such applications on large-scale real-world data sets, and the scalability behavior of our model.

## 2 Canonical PARAFAC Decomposition

Given a tensor $\mathcal{Y}$ of size $n_1 \times n_2 \times \cdots \times n_K$, with $n_k$ denoting the size of $\mathcal{Y}$ along the $k^{th}$ mode (or "way") of the tensor, the goal in a Canonical PARAFAC (CP) decomposition [14] is to decompose $\mathcal{Y}$ into a set of $K$ factor matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(K)}$ where $\mathbf{U}^{(k)} = [\boldsymbol{u}_1^{(k)}, \ldots, \boldsymbol{u}_R^{(k)}]$, $k = \{1, \ldots, K\}$, denotes the $n_k \times R$ factor matrix associated with mode $k$. In its most general form, CP decomposition expresses the tensor $\mathcal{Y}$ via a weighted sum of $R$ rank-1 tensors as $\mathcal{Y} \sim f(\sum_{r=1}^{R} \lambda_r . \boldsymbol{u}_r^{(1)} \odot \ldots \odot \boldsymbol{u}_r^{(K)})$. The form of $f$ depends on the type of data being modeled (e.g., $f$ can be Gaussian for real-valued, Bernoulli-logistic for binary-valued, Poisson for count-valued tensors). Here $\lambda_r$ is the weight associated with the $r^{th}$ rank-1 component, the $n_k \times 1$ column vector $\boldsymbol{u}_r^{(k)}$ represents the $r^{th}$ latent factor of mode $k$, and $\odot$ denotes vector outer product.

## 3 Beta-Negative Binomial CP Decomposition

We focus on modeling count-valued tensor data [4] and assume the following generative model for the tensor $\mathcal{Y}$

$$\mathcal{Y} \sim \text{Pois}(\sum_{r=1}^{R} \lambda_r . \boldsymbol{u}_r^{(1)} \odot \ldots \odot \boldsymbol{u}_r^{(K)}) \tag{1}$$

$$\boldsymbol{u}_r^{(k)} \sim \text{Dir}(a^{(k)}, \ldots, a^{(k)}) \tag{2}$$

$$\lambda_r \sim \text{Gamma}(g_r, \frac{p_r}{1-p_r}) \tag{3}$$

$$p_r \sim \text{Beta}(c\epsilon, c(1-\epsilon)) \tag{4}$$

We use subscript $\boldsymbol{i} = \{i_1, \ldots, i_K\}$ to denote the index of the $\boldsymbol{i}$-th entry in $\mathcal{Y}$. Using this notation, the $\boldsymbol{i}$-th entry of the tensor can be written as $y_{\boldsymbol{i}} \sim \text{Pois}(\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)})$. We assume that we are given $N$ observations $\{y_{\boldsymbol{i}}\}_{\boldsymbol{i}=1}^{N}$ from the tensor $\mathcal{Y}$.

Since the gamma-Poisson mixture distribution is equivalent to a negative binomial distribution [15], (1) and (3), coupled with the beta prior (Eq 4) on $p_r$, lead to what we will call the beta-negative binomial CP (BNBCP) decomposition model. A few things worth noting about our model are

- The Dirichlet prior on the factors $\boldsymbol{u}_r^{(k)}$ naturally imposes non-negativity constraints [4] on the factor matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(K)}$. Moreover, since each column $\boldsymbol{u}_r^{(k)}$ of these factor matrices sums to 1, $\boldsymbol{u}_r^{(k)}$ can also be thought of a distribution (e.g., a "topic") over the $n_k$ entities in mode $k$.
- The gamma-beta hierarchical construction of $\lambda_r$ (Eq 3 and 4) allows inferring the rank of the tensor by setting an upper bound $R$ on the number of factors and letting the inference procedure infer the appropriate number of factors by shrinking the coefficients $\lambda_r$'s to close to zero for the irrelevant factors.
- The resulting negative binomial model is useful for modeling *overdispersed* count data in cases where the Poisson likelihood may not be suitable.
- Using alternate parameterizations (Section 3.1) of the Poisson distribution in (1) leads to a fully conjugate model and facilitates efficient Gibbs sampling and variational Bayes (VB) inference, in both batch as well as online settings.

### 3.1 Reparametrizing the Poisson Distribution

The generative model described in Eq (1)-(4) is not conjugate. We now describe two equivalent parametrizations [6,24] of (1), which transform (1)-(4) into a fully conjugate model and facilitate easy-to-derive and scalable inference procedures. These parameterizations are based on a data augmentation scheme described below.

The first parametrization expresses the $\boldsymbol{i}$-th count-valued entry $y_{\boldsymbol{i}}$ of the tensor $\mathcal{Y}$ as a sum of $R$ *latent* counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^{R}$

$$y_{\boldsymbol{i}} = \sum_{r=1}^{R} \tilde{y}_{\boldsymbol{i}r}, \quad \tilde{y}_{\boldsymbol{i}r} \sim \text{Pois}(\lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}) \tag{5}$$

The second parametrization assumes the vector $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^{R}$ of latent counts is drawn from a multinomial as

$$\begin{aligned} \tilde{y}_{\boldsymbol{i}1}, \ldots, \tilde{y}_{\boldsymbol{i}R} &\sim \text{Mult}(y_{\boldsymbol{i}}; \zeta_{\boldsymbol{i}1}, \ldots, \zeta_{\boldsymbol{i}R}) \\ \zeta_{\boldsymbol{i}r} &= \frac{\lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}}{\sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)}} \end{aligned} \tag{6}$$

The above parameterizations follows from the following lemma [6,24]:

**Lemma 1.** *Suppose that $x_1, \ldots, x_R$ are independent random variables with $x_r \sim Pois(\theta_r)$ and $x = \sum_{r=1}^{R} x_r$. Set $\theta = \sum_{r=1}^{R} \theta_r$; let $(z, z_1, \ldots, z_R)$ be another set of random variables such that $z \sim Pois(\theta)$, and $(z_1, \ldots, z_R)|z \sim Mult(z; \frac{\theta_1}{\theta}, \ldots, \frac{\theta_R}{\theta})$. Then the distribution of $\boldsymbol{x} = (x, x_1, \ldots, x_R)$ is the same as the distribution of $\boldsymbol{z} = (z, z_1, \ldots, z_R)$.*

These parameterizations, along with the fact that the columns $\boldsymbol{u}_r^{(k)}$ of each factor matrix are drawn from a Dirichlet, allows us to leverage the Dirichlet-multinomial conjugacy and derive simple Gibbs sampling and variational Bayes (VB) inference update equations, as described in Section 4.

## 4 Inference

We first present the update equations for batch Gibbs sampling (Section 4.1) and batch VB inference (Section 4.2). We then present two online inference algorithms, based on: (*i*) conditional density filtering [7], which provides an efficient way to perform online MCMC sampling using conditional sufficient statistics of the model parameters; and (*ii*) stochastic variational inference [9], which will allow scaling up VB inference by processing data in small minibatches.

We also define two quantities $s_{j,r}^{(k)} = \sum_{\boldsymbol{i}:i_k=j} \tilde{y}_{\boldsymbol{i}r}$ and $s_r = \sum_{\boldsymbol{i}} \tilde{y}_{\boldsymbol{i},r}$ which denote aggregates (sufficient statistics) computed using the latent counts $\tilde{y}_{\boldsymbol{i}r}$. These quantities appear at various places in the description of the inference algorithms we develop.

### 4.1 Gibbs Sampling

– **Sampling $\tilde{y}_{\boldsymbol{i}r}$:** The latent counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^{R}$ are sampled from a multinomial (6).

– **Sampling $\boldsymbol{u}_r^{(k)}$:** Due to the Dirichlet-multinomial conjugacy, the columns of each factor matrix have Dirichlet posterior and are sampled as

$$\boldsymbol{u}_r^{(k)} \sim \text{Dir}(a^{(k)} + s_{1,r}^{(k)}, a^{(k)} + s_{2,r}^{(k)}, \ldots, a^{(k)} + s_{n_k,r}^{(k)}) \tag{7}$$

– **Sampling $p_r$:** Using the fact that $s_r = \sum_{\boldsymbol{i}} \tilde{y}_{\boldsymbol{i},r}$ and marginalizing over the $u_{i_k r}^{(k)}$'s in (5), we have $s_r \sim \text{Pois}(\lambda_r)$. Using this, along with (3), we can express

$s_r$ using a negative binomial distribution, i.e., $s_r \sim \text{NB}(g_r, p_r)$. Then, due to the conjugacy between negative binomial and beta, we can sample $p_r$ as

$$p_r \sim \text{Beta}(c\epsilon + s_r, c(1-\epsilon) + g_r) \tag{8}$$

– **Sampling $\lambda_r$:** Again using the fact that $s_r \sim \text{Pois}(\lambda_r)$, and due to the gamma-Poisson conjugacy, we have

$$\lambda_r \sim \text{Gamma}(g_r + s_r, p_r) \tag{9}$$

**Computational Complexity:** Sampling the latent counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^R$ for each nonzero observation $y_{\boldsymbol{i}}$ (note that for $y_{\boldsymbol{i}} = 0$, the latent counts are trivially zero) requires computing $\{\zeta_{\boldsymbol{i}r}\}_{r=1}^R$, and computing each $\zeta_{\boldsymbol{i}r}$ requires $O(K)$ time (Eq 6). Therefore, sampling all the latent counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^R$ requires $O(NRK)$ time. Sampling the latent factors $\{\boldsymbol{u}_r^{(k)}\}_{r=1}^R$ for the $K$ tensor modes requires $O(RK)$ time. Sampling $\{p_r\}_{r=1}^R$ and $\{\lambda_r\}_{r=1}^R$ requires $O(R)$ time each. Of all these steps, sampling the latent counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^R$ (which are also used to compute the sufficient statistics $s_{j,r}^{(k)}$ and $s_r$) is the most dominant step, leading to an overall time-complexity of $O(NRK)$ for the Gibbs sampling procedure.

The linear dependence on $N$ (number of nonzeros) is especially appealing because most real-world count-valued tensors are extremely sparse (have much less than even 1% nonzeros. In contrast to the standard negative-binomial models for count data, for which the inference complexity also depends on the zeros whose number may be massive (and therefore heuristics, such as subsampling the zeros, are needed), the reparametrizations (Section 3.1) used by our model allow us to ignore the zeros in the multinomial sampling step (the sufficient statistics do not depend on the zero entries in the tensor), thereby significantly speeding up the inference.

### 4.2 Variational Bayes Inference

Using the mean-field assumption [12], we approximate the target posterior distribution by $Q = \prod_{\boldsymbol{i},r} q(\tilde{y}_{\boldsymbol{i}r}) \prod_{k,r} q(\boldsymbol{u}_r^{(k)}) \prod_r q(\lambda_r) \prod_r q(p_r)$. Our fully conjugate model enables closed-form variational Bayes (VB) inference updates, with the distribution $q(\tilde{y}_{\boldsymbol{i}r})$, $q(\boldsymbol{u}_r^{(k)})$, $q(\lambda_r)$, and $q(p_r)$ being multinomial, Dirichlet, beta, and gamma, respectively. We summarize the update equations for the variational parameters of each of these distributions, below:

– **Updating $\tilde{y}_{\boldsymbol{i}r}$:** Using (6), the updates for $y_{\boldsymbol{i}r}$ are given by $\mathbb{E}[y_{\boldsymbol{i}r}] = y_{\boldsymbol{i}}\zeta_{\boldsymbol{i}r}$ where $\zeta_{\boldsymbol{i}r}$ is defined as $\zeta_{\boldsymbol{i}r} = \frac{\tilde{\zeta}_{\boldsymbol{i}r}}{\sum_{r=1}^R \tilde{\zeta}_{\boldsymbol{i}r}}$ and $\tilde{\zeta}_{\boldsymbol{i}r}$ can be computed as

$$\tilde{\zeta}_{\boldsymbol{i}r} = \exp\{\Psi(s_r + g_r) + \ln(p_r) + \sum_{k=1}^K \Psi(s_{i_k,r}^{(k)} + a^{(k)}) - \Psi[\sum_{k=1}^K (s_{i_k,r}^{(k)} + a^{(k)})]\} \tag{10}$$

where $\Psi(.)$ is the digamma function, which is the first derivative of the logarithm of the gamma function.

– **Updating $\boldsymbol{u}_{i_k r}^{(k)}$:** The mean-field posterior $q(\boldsymbol{u}_r^{(k)})$ is Dirichlet with each of the component means given by $\mathbb{E}[\boldsymbol{u}_{i_k r}^{(k)}] = \frac{\rho_{i_k r}^{(k)}}{\sum_{i_k=1}^{n_k} \rho_{i_k r}^{(k)}}$ where $\rho_{i_k r}^{(k)} = a^{(k)} + s_{i_k,r}^{(k)}$.

– **Updating $p_r$:** The mean-field posterior $q(p_r)$ is beta with mean given by $\mathbb{E}[p_r] = \frac{p_{ra}}{p_{ra}+p_{rb}}$ where $p_{ra} = c\epsilon + s_r$, $p_{rb} = c(1-\epsilon) + g_r$.

– **Updating $\lambda_r$:** The mean-field posterior $q(\lambda_r)$ is gamma with mean given by $\mathbb{E}[\lambda_r] = \lambda_{ra}\lambda_{rb}$, where $\lambda_{ra} = (g_r + s_r)$ and $\lambda_{rb} = p_r$.

**A note on Gibbs vs VB:** The per-iteration time-complexity of the VB inference procedure is also $O(NRK)$. It is to be noted however that, in practice, one iteration of VB in this model is a bit more expensive than one iteration of Gibbs, due to the digamma function evaluation for the $\tilde{\zeta}_{\boldsymbol{i}r}$ which is needed in VB when updating the $\tilde{y}_{\boldsymbol{i}r}$'s. Prior works on Bayesian inference for topic models [8] also support this observation.

### 4.3 Online Inference

Batch Gibbs (Section 4.1) and VB (Section 4.2) inference algorithms are simple to implement and efficient to run on moderately large-sized problems. These algorithms can however be slow to run for massive data sets (e.g., where the number of tensor entries $N$ and/or the dimension of the tensor is massive). The Gibbs sampler may exhibit slow mixing and the batch VB may be slow to converge. To handle such massive tensor data, we develop two online inference algorithms. The first is online MCMC based conditional density filtering [7], while the second is based on stochastic variational inference [9]. Both these inference algorithms allow processing data in small minibatches and enable our model to analyze massive and/or streaming tensor data.

**Conditional Density Filtering:** The conditional density filtering (CDF) algorithm [7] for our model selects a minibatch of tensor entries at each iteration, samples the latent counts $\{\tilde{y}_{\boldsymbol{i}r}\}_{r=1}^R$ for these entries conditiond on the previous estimates of the model parameters, updates the sufficient statistics $s_{j,r}^{(k)}$ and $s_r$ using these latent counts (as described below), and resamples the model parameters conditioned on these sufficient statistics. Denoting $I_t$ as data indices in minibatch at round $t$, the algorithm proceeds as

– **Sampling $\tilde{y}_{\boldsymbol{i}r}$:** For all $\boldsymbol{i} \in I_t$, sample the latent counts $\tilde{y}_{\boldsymbol{i}r(\boldsymbol{i}\in I_t)}$ using (6).

– **Updating the conditional sufficient statistics:** Using data from the current minibatch, update the conditional sufficient statistics as:

$$s_{j,r}^{(k,t)} = (1-\gamma_t) s_{j,r}^{(k,t-1)} + \gamma_t \frac{N}{B} \sum_{\boldsymbol{i}\in I_t : i_k = j} \tilde{y}_{\boldsymbol{i}r} \tag{11}$$

$$s_r^{(t)} = (1-\gamma_t) s_r^{(t-1)} + \gamma_t \frac{N}{B} \sum_{\boldsymbol{i}\in I_t} \tilde{y}_{\boldsymbol{i},r} \tag{12}$$

Note that the updated conditional sufficient statistics (CSS), indexed by superscript $t$, is a weighted average of the old CSS, indexed by superscript $t-1$, and of that computing only using the current minibatch (of size $B$). In addition, the latter term is further weighted by $N/B$ so as to represent the *average* CSS over the *entire* data. In the above, $\gamma_t$ is defined as $\gamma_t = (t_0+t)^{-\kappa}$, $t_0 \geq 0$, and $\kappa \in (0.5, 1]$ is needed to guarantee convergence [3].

– **Updating $\boldsymbol{u}_r^{(k)}, p_r, \lambda_r$:** Using the updated CSS, draw $M$ samples for each of the model parameters $\{\boldsymbol{u}_r^{(k,m)}, p_r^{(m)}, \lambda_r^{(m)}\}_{m=1}^M$, from the following conditionals:

$$\boldsymbol{u}_r^{(k)} \sim \text{Dir}(a^{(k)} + s_{1,r}^{(k,t)}, \ldots, a^{(k)} + s_{n_k,r}^{(k,t)}) \tag{13}$$

$$p_r \sim \text{Beta}(c\epsilon + s_r^{(t)}, c(1-\epsilon) + g_r) \tag{14}$$

$$\lambda_r \sim \text{Gamma}(g_r + s_r^{(t)}, p_r) \tag{15}$$

and either store the sample averages of $\boldsymbol{u}_r^{(k)}, p_r$, and $\lambda_r$, or their analytic means to use for the next CDF iteration [7]. Since the analytic means of the model parameters are available in closed-form in this case, we use the latter option, which obviates the need to draw $M$ samples, thereby also speeding up the inference significantly.

We next describe the stochastic (online) VB inference for our model.

**Stochastic Variational Inference:** The batch VB inference (Section 4.2) requires using the entire data for the parameter updates in each iteration, which can be computationally expensive and can also result in slow convergence. Stochastic variational inference (SVI), on the other hand, leverages ideas from stochastic optimization [9] and, in each iteration, uses a small randomly chosen minibatch of the data to updates the parameters. Data from the current minibatch is used to compute stochastic gradients of the variational objective w.r.t. each of the parameters and these gradients are subsequently used in the parameter updates. For our model, the stochastic gradients depend on the sufficient statistics computed using the current minibatch $I_t$: $s_{j,r}^{(k,t)} = \sum_{\boldsymbol{i} \in I_t : i_k = j} \tilde{y}_{\boldsymbol{i}r}$ and $s_r^{(t)} = \sum_{\boldsymbol{i} \in I_t} \tilde{y}_{\boldsymbol{i},r}$, where $\tilde{y}_{\boldsymbol{i}r}$ is computed using Eq 10. Denoting $B$ as the minibatch size, we reweight these statistics by $N/B$ to compute the *average* sufficient statistics over the entire data [9] and update the other variational parameters as follows:

$$\rho_{i_k r}^{(k,t)} = (1-\gamma_t)\rho_{i_k r}^{(k,t-1)} + \gamma_t(a^{(k)} + (N/B)s_{i_k,r}^{(k,t)}) \tag{16}$$

$$p_{ra}^{(t)} = (1-\gamma_t)p_{ra}^{(t-1)} + \gamma_t(c\epsilon + (N/B)s_r^{(t)}) \tag{17}$$

$$p_{rb}^{(t)} = (1-\gamma_t)p_{rb}^{(t-1)} + \gamma_t(c(1-\epsilon) + g_r) \tag{18}$$

$$\lambda_{ra}^{(t)} = (1-\gamma_t)\lambda_{ra}^{(t-1)} + \gamma_t(g_r + (N/B)s_r^{(t)}) \tag{19}$$

$$\lambda_{rb}^{(t)} = (1-\gamma_t)\lambda_{ra}^{(t-1)} + \gamma_t p_r \tag{20}$$

where $\gamma_t$ is defined as $\gamma_t = (t_0 + t)^{-\kappa}$, $t_0 \geq 0$, and $\kappa \in (0.5, 1]$ is needed to guarantee convergence [9].

**Computational Complexity:** In contrast to the batch Gibbs and batch VB, both of which have $O(NRK)$ cost per-iteration, the per-iteration cost of the online inference algorithms (CDF and SVI) is $O(|I_t|RK)$ where $|I_t|$ is the mini-batch size at round $t$. We use a fixed minibatch size $B$ for each minibatch, so the per-iteration cost is $O(BRK)$.

## 5 Related Work

Although tensor factorization methods have received considerable attention recently, relatively little work exists on scalable analysis of massive count-valued tensor data. Most of the recently proposed methods for scalable tensor decomposition [2,10,13,17] are based on minimizing the Frobenious norm of the tensor reconstruction error, which may not be suitable for count or overdispersed count data. The rank of decomposition also needs to be pre-specified, or chosen via cross-validation. Moreover, these methods assume the tensor to be fully observed and thus cannot be used for tensor completion tasks. Another key difference between these methods and ours is that scaling up these methods requires parallel or distributed computing infrastructure, whereas our fully Bayesian method exhibits excellent scalability on a single machine. At the same time, the simplicity of the inference update equations would allow our model to be easily parallelized or distributed. We leave this possibility to future work.

One of the first attempts to explicitly handle count data in the context of non-negative tensor factorization includes the work of [4], which is now part of the Tensor Toolbox [1]. This method optimizes the Poisson likelihood, using an alternating Poisson regression sub-routine, with non-negative constraints on the factor matrices. However, this method requires the rank of the decomposition to be specified, and cannot handle missing data. Due to its inability in handling missing data, for our experiments (Section 6), as a baseline, we implement and use a Bayesian version of this model which *can* handle missing data.

Among other works of tensor factorization for count data, the method in [1] can deal with missing values, though the rank still needs to be specified, and moreover the factor matrices are assumed to be real-valued, which makes it unsuitable for interpretability of the inferred factor matrices.

In addition to the Poisson non-negative tensor factorization method of [4], some other non-negative tensor factorization methods [5,20,21] also provide interpretability for the factor matrices. However, these methods usually have one or more of the following limitations: (1) there is no explicit generative model for the count data, (2) the rank needs to be specified, and (3) the methods do not scale to the massive tensor data sets of scales considered in this work.

Methods that facilitate a full Bayesian analysis for massive count-valued tensors, which are becoming increasingly prevalent nowadays, are even fewer. A recent attempt on Bayesian analysis of count data using Poisson likelihood is considered in [19]; however, unlike our model, their method cannot infer the rank and relies on batch VB inference, limiting its scaling behavior. Moreover, the Poisson likelihood may not be suitable for overdispersed counts.

[1] http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html

Finally, inferring the rank of the tensor, which is NP-complete in general [14], is another problem for which relatively little work exists. Recent attempts at inferring the rank of the tensor in the context of CP decomposition include [18,22]; however (1) these methods are not applicable for count data, and (2) the inferred factor matrices are real-valued, lacking the type of interpretability needed in many applications.

Our framework is similar in spirit to the matrix factorization setting proposed in [24] which turns out to be a special case of our framework. In addition, while [24] only developed (batch) Gibbs sampling based inference, we present both Gibbs sampling as well as variational Bayesian inference, and design efficient *online* Bayesian inference methods to scale up our framework for handling massive real-world tensor data.

To summarize, in contrast to the existing methods for analyzing tensors, our fully Bayesian framework, based on a proper generative model, provides a flexible method for analyzing massive count-valued tensors, side-stepping crucial issues such as rank-specification, providing good interpretability of the latent factors, while still being scalable for analyzing massive real-world tensors via online Bayesian inference.

## 6 Experiments

We apply the proposed model on a synthetic and three real-world data sets that range in their sizes from moderate to medium to massive. The real-world tensor data sets we use in our experiments are from diverse application domains, such as analyzing country-country interaction data in political science, topic modeling on *multiway* publications data (with entities being authors, words, and publication venues), and analysis of massive household transactions data. These data sets include:

- **Synthetic Data:** This is a tensor of size $300 \times 300 \times 300$ generated using our model by setting an upper bound $R = 50$ over the number of factors; only 20 factors were significant (based on the values of $\lambda_r$), resulting in an effective rank 20.

- **Political Science Data (GDELT):** This is a real-world four-way tensor data of country-country interactions. The data consists of 220 countries, 20 action types, and the interactions date back to 1979 [16]. We focus on a subset of this data collected during the year 2011, resulting in a tensor of size $220 \times 220 \times 20 \times 52$. Section 6.4 provides further details.

- **Publications Data:** This is a $2425 \times 9088 \times 4068$ count-valued tensor, constructed from a database of research papers published by researchers at Duke University[2]; the three tensor modes correspond to authors, words, and venues. Section 6.3 provides further details.

[2] Obtained from https://scholars.duke.edu/

– **Transactions (Food) Data:** This is a $117054 \times 438 \times 67095$ count-valued tensor, constructed from a database of transactions data of food item purchases at various stores in the US [3]; the three tensor modes correspond to households, stores, and items. Section 6.5 provides further details.

We compared our model with the following baselines: (*i*) Bayesian Poisson Tensor Factorization (BayesPTF), which is fully Bayesian version of the Poisson Tensor Factorization model proposed in [4], and (*ii*) Non-negative Tensor Decomposition based on Low-rank Approximation (lraNTD) proposed in [23]. All experiments are done on a standard desktop computer with Intel i7 3.4GHz processor and 24GB RAM.

### 6.1 Inferring the Rank

To begin with, as a sanity check for our model, we first perform an experiment on the synthetic data described above to see how well the model can recover the true rank (tensor completion results are presented separately in Section 6.2). For this experiment, we run the batch Gibbs sampler (the other inference methods also yield similar results) with 1000 burn-ins, and 1000 collection samples. We experiment with three settings: using 20%, 50% and 80% data for training. The empirical distribution (estimated using the collected MCMC samples) of the effective inferred rank for each of these settings is shown in Figure 1 (left). In each collection iteration, the effective rank is computed after a simple thresholding on the $\lambda_r$'s where components with very small $\lambda_r$ are not counted (also see Figure 1

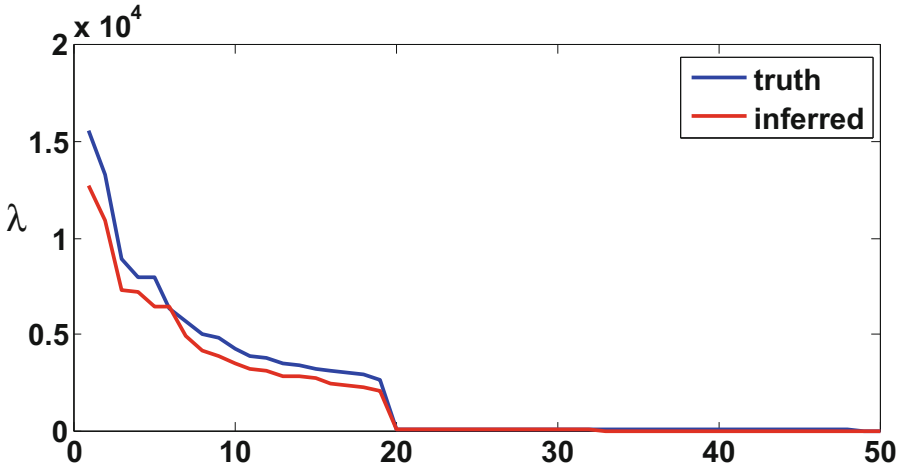


**Fig. 1.** Distribution over inferred ranks for syntheric data (left), and $\boldsymbol{\lambda}$ inferred using 80% training data (right).

[3] Data provided by United States Department of Agriculture (USDA) under a Third Party Agreement with Information Resources, Inc. (IRI).

(right)). With 80% training data, the distribution shows a distinct peak at 20 and even with smaller amounts of training data (20% and 50%), the inferred rank is fairly close to the ground truth of 20. In Figure 1 (right), we show the spectrum of all the $\lambda_r$'s comparing the ground truth vs the inferred values;

### 6.2 Tensor Completion Results

We next experiment on the task of tensor completion, where for each method 95% of the data are used for training and the remaining 5% data is used as the heldout set (note that the data sets we use are extremely sparse in nature, with considerably less than 1% entries of the tensor being actually observed). The results are reported in Table 1 where we show the log likelihood and the mean-absolute error (MAE) in predicting the heldout data. Timing-comparison for the various batch and online inference methods is presented separately in Section 6.6.

For this experiment, we compare our BNBCP model (using the various inference methods) with (1) BayesPTF - a fully Bayesian variant (we implented it ourselves) of a state-of-the-art Poisson Tensor Factorization model originally proposed in [4] (which cannot however handle missing data), and (2) lraNTD [23] which is an optimization based non-negative tensor decomposition method. As Table 1 shows, our methods achieve better log-likelihood and MAE as compared to these baselines. Moreover, among our batch and online Bayesian inference methods, the online inference methods give competitive or better results as compared to their batch counterparts. In particular, the online MCMC method based on conditional density filtering (BNBCP-CDF) works the best across all the methods (please see Section 6.6 for a timing comparison).

**Table 1.** Loglikelihood and MAE comparison for different methods (the two baselines, our model with batch inference, and our model with online inference) on four datasets. Note: lraNTD gave out-of-memory error on publications and food transactions data sets so we are unable to report its results on these data sets. We also only report the MAE for lraNTD, and not the log-likelihood, because it uses a Gaussian likelihood model for the data.

| Datasets | Toy data | GDELT | Publication | Food | Toy data | GDELT | Publication | Food |
|---|---|---|---|---|---|---|---|---|
| BayesPTF | -107563 | -4425695 | -860808 | -2425433 | 1.012 | 55.478 | 1.636 | 1.468 |
| lraNTD | N/A | N/A | N/A | N/A | 1.019 | 65.049 | N/A | N/A |
| BNBCP-Gibbs | -97580 | -3079883 | -619258 | -2512112 | 0.989 | 45.436 | 1.565 | 1.459 |
| BNBCP-VB | -99381 | -2971769 | -632224 | -2533086 | 0.993 | **43.485** | 1.574 | 1.472 |
| BNBCP-CDF | -95472 | **-2947309** | **-597817** | **-2403094** | 0.985 | 44.243 | **1.555** | **1.423** |
| BNBCP-OnlineVB | -98446 | -3169335 | -660068 | -2518996 | 0.989 | 46.188 | 1.601 | 1.461 |

### 6.3 Analyzing Publications Database

The next experiment is on a three-way tensor constructed from a scientific publications database. The data consist of abstracts from papers published by various researchers at Duke University [4]. In addition to the paper abstract, the venue

[4] Data crawled from https://scholars.duke.edu/

information for each paper is also available. The data collection contains 2425 authors, 9088 words (after removing stop-words), and 4068 venues which results in a $2425 \times 9088 \times 4068$ word-counts tensor, on which we run our model. As the output of the tensor decomposition, we get three factor matrices. Since the latent factors in our model are non-negative and sum to one, each latent factor can also be interpreted as a *distribution* over authors/words/venues, and consequently represents a "topic". Therefore the three factor matrices inferred by our model for this data correspond to authors $\times$ topics, words $\times$ topics, and venue $\times$ topics, which we use to further analyze the data.

We apply the model BNBCP-CDF on this data (with $R = 200$) and using the inferred words $\times$ topics matrix, in Table 2 (left) we show the list of 10 most probable words in four factors/topics that seem to correspond to **optics**, **genomics**, **machine learning & signal processing**, and **statistics**. To show the topic representation across different departments, we present a histogram of *departmental affiliations* for 20 authors with highest probabilities in these four factors. We find that, for the genomics factor, the top authors (based on their topic scores) have affiliations related to biology which makes intuitive sense. Likewise, for the statistics factor, most of the top authors are from statistics and biostatistics departments. The top 20 authors in factors that correspond to optics and machine learning & signal processing, on the other hand, are from departments of electrical and computer engineering and/or computer science, etc.

**Table 2.** Most probable words in topics related to optics, genomics, machine learning/signal processing(ML/SP) and statistics (Stats), and top ranked venues in ML/SP community.

| Optics | Genomics | ML/SP | Stats | Top Venues in ML/SP |
|---|---|---|---|---|
| gigapixel | gene | dictionary | model | ICASSP |
| microcamera | chromatin | sparsity | priors | IEEE trans. sig. proc. |
| cameras | occupancy | model | bayesian | ICML |
| aperture | centromere | bayesian | lasso | Siam J. img. sci. |
| lens | transcription | compressed | latent | IEEE trans. img. proc. |
| multiscale | genome | compressive | inference | IEEE int. symp. biomed. img. |
| optical | sites | matrix | regression | NIPS |
| system | expression | denoising | sampler | IEEE trans. wireless comm. |
| nanoprobes | sequence | gibbs | semiparametric | IEEE workshop stat. sig. proc. |
| metamaterial | vegfa | noise | nonparametric | IEEE trans. inf. theory |

Similarly, using the inferred venues $\times$ topics matrix, we list the most likely venues for each topic. Due to space-limitations, here we only present the most likely venues in machine learning & signal processing factor/topic; the result is shown in Table 2 (right-most column). The result shows that venues like ICASSP, IEEE Trans. Signal Proc., ICML, and NIPS all rank at the top in the machine learning & signal processing factor, which again makes intuitive sense.

### 6.4 Analyzing Political Science Data

We use the model to analyze a real-world political science data set consisting of country-country interactions. Such analyses are typically done by political

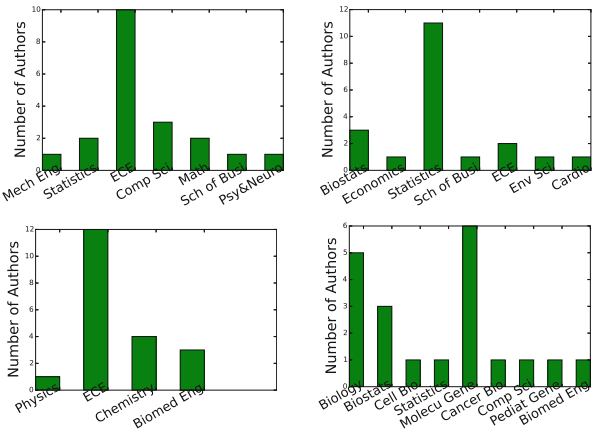


**Fig. 2.** Histogram of affiliations for top 20 authors in factors related to machine learning/signal processing (top left) and statistics (top right), optics (bottom left), and genomics(bottom right)

scientists to study, analyze and understand complex international multilateral relations among countries. The data set is from the Global Database of Events, Location, and Tone (GDELT) [16]. GDELT records the dyadic interactions between countries in the form of "Country A did something to Country B". In our experiments, we consider 220 countries ("actors") and 20 unique high-level action types in 52 weeks of year 2012. After preprocessing, we have a four-way (country-country-action-time) action counts tensor of size $220 \times 220 \times 20 \times 52$. Note that both first and second tensor mode represents countries; first mode as "sender" and the second mode as "receiver" of a particular action. In this analysis, we set $R$ to be large enough (200) and the model discovered roughly about 120 active components (i.e., components with significant value of $\lambda_r$).

We apply the model (BNBCP-CDF; other methods yield similar results) and examine each of the time dimension factors, specifically looking for the

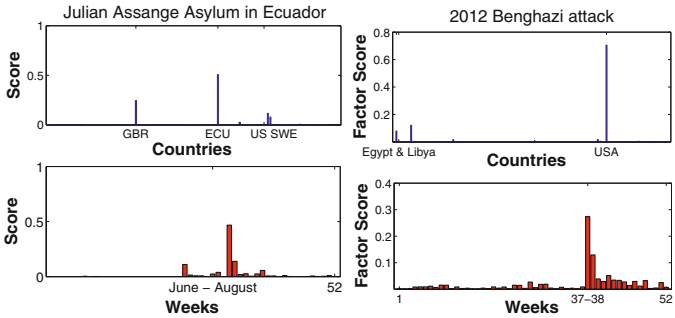


**Fig. 3.** Country factors (top row) and time factors (bottom row) for Julian Assange asylum in Ecuador (left column) and 2012 Benghazi attack (right column).

significant components (based on the magnitude of $\lambda_r$) in which the time dimension factor also peaks during certain time(s) of the year. We show results with two such factors in Figure 3. In Figure 3 (column 1), the time and country (actor) factors seems to suggest that this factor/topic corresponds to the event "Julian Assange". The actor subplot shows spikes at Ecuador, United Kingdom, United States, and Sweden whereas the time factor in the bottom left subplot shows spikes between June and August. The time and countries involved are consistent with the public knowledge of the event of Julian Assange seeking refuge in Ecuador.

Likewise, in Figure 3 (column 2), the time and country (actor) factors seems to suggest that this factor corresponds to the event "Benghazi Attack" which took place on Sept. 12 (week 37) of 2012, in which Islamic militants attacked American diplomatic compound in Benghazi, Libya. The attack killed an US Ambassador. As the Figure shows, the top actors identified are US, Libya and Egypt, and spikes are found at around week 37 and 38, which are consistent with the public knowledge of this event.

The results of these analyses demonstrate that the interpretability of our model can be useful for identifying events or topics in such multiway interaction data.

### 6.5 Analyzing Transactions Data

We next apply our model (BNBCP-CDF; other methods yield similar results) for analyzing transactions data for food item purchases made at stores. Our data is collected for a demographically representative sample of US consumers who reside in large urban and suburban areas and purchase food in supermarkets and grocery stores. The data were provided by the USDA under a Third Party Agreement with IRI. Each transaction is identified by a unique Universal Product Code (UPC) barcode and the store where the transaction occurred. Some items such as fresh produce do not have UPCs and are identified separately. The households are observed over a four year period, during which they are provided with a technology that allows them to scan each purchase and record additional information such as the store where the purchase was made (and other economic data). Participating households are provided with incentives designed to encourage compliance. For each household-product-store combination we record the number of unique purchases over the sampling period. The database has a total of 117,054 unique households, 438 stores, and 67,095 unique items and we construct a 3-way count tensor of size $117054 \times 438 \times 67095$ with about 6.2 million nonzero entries.

We apply the proposed model on this data by setting $R = 100$ (out of which about 60 components were inferred to have a significant value of $\lambda_r$) and looked at the stores factor matrix. Since each column (which sums to 1) of the store factor matrix can be thought of as a *distribution* over the stores, we look at three of the factors from the store factor matrix and tried to identify the stores that rank at the top in that factor. In Table 3, we show results from each of these factors. Factor 1 seems to suggest that it is about the most popular stores

(included Walmart, for example), Factor 2 has stores that primarily deal in wholesale (e.g., Costco, Sam's Wholesale Club), and Factor 3 contains stores that sell none or very few food items (e.g., Mobil, Petco). Note that the Walmart Super Center figures prominently in both Factor 1 and Factor 2.

**Table 3.** Three of the store factors inferred from the transaction data (top-5 stores shown for each)

| Factor 1 | Factor 2 | Factor 3 |
|---|---|---|
| Walmart Sup. Center | Sam's Club | Dick's Sporting |
| Walmart Traders | Meijer | Mobil |
| Walmart Neighb. | Costco | Petco |
| Walmart | B J'S Wholesale | Sally Beauty |
| Kroger | Walmart Sup. Center | GNC All |

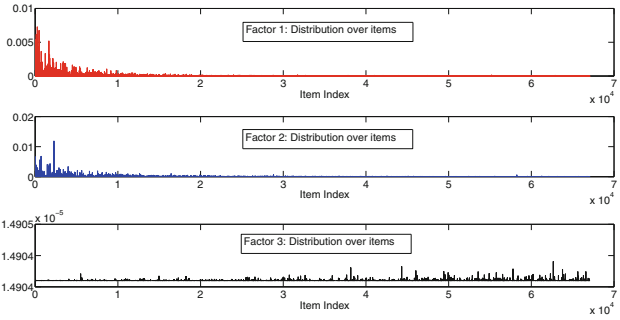


**Fig. 4.** Distributions over items for three factors (each factor corresponds to a cluster).

We next look at the items factor matrix. In Figure 2, we plot the inferred distribution over items in each of the three clusters described above. For factors 1 and 2 (which correspond to the most popular stores and wholesale stores respectively), the distribution over the items (top and bottom panel in Figure 2) have a reasonably significant mass over a certain range of items (for the items indexed towards the left side in the plots of factors 1 and 2). On the other hand, for factor 3 which corresponds to stores that sell no or very few types of food items, the distribution over the items is rather flat and diffuse with very weak intensities (looking at the scale on the y axis). From the Figure 2, it is also interesting to observe that the set of active items in factors (1 & 2) vs factor 3 seem to be mostly disjoint.

This analysis provides a first attempt to analyze food shopping patterns for American consumers on a large scale. As the world, at large, struggles with a combination of increasing obesity rates and food insecurity, this analysis shows that consumer preferences are densely clustered across both stores and items. This indicates that household tend to have fairly rigid preferences over the stores where they shop. Furthermore, they tend to consume a relatively small number of products from the universe of available products. The concentration in both stores and products is indicative of limited search behavior and substantial behavioral rigidity which may be associated with suboptimal outcomes in terms of nutrition and health.

### 6.6 Scalability

We now perform an experiment comparing the proposed inference methods (batch and online) to assess their scalability (Figure 5). We first use the Transactions data ($117054 \times 438 \times 67095$) for this experiment. We would like to note that the state-of-the-art methods for count-valued tensor, such as the Poisson Tensor Factorization (PTF) method from the Tensor Toolbox [4], are simply infeasible to run on this data because of storage explosion issue (the method requires expensive flattening operations of the tensor). The other baseline LRANTD [23] we used in our experiments was also infeasible to run on this data. We set $R = 100$ for each method (about 60 factors were found to be significant, based on the inferred values of the $\lambda_r$'s) and use a minibatch size of 100000 for all the online inference methods. For the conditional density filtering as well as stochastic variational inference, we set the learning rate as $t_0 = 0$ and $\kappa = 0.5$. Figure 5 shows that online inference methods (conditional density filtering and stochastic variational inference) converge much faster to a good solution than batch methods. This experiment shows that our online inference methods can be computationally viable alternatives if their batch counterparts are slow/infeasible to run on such data.

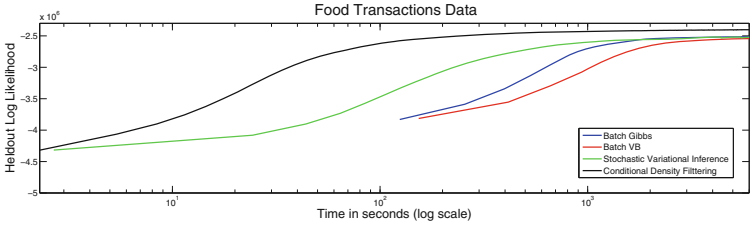


**Fig. 5.** Time vs heldout log likelihoods with various methods on transactions data

We then perform another experiment on the Scholars data, on which the PTF method of [4] was feasible to run and compare its per-iteration running time with our model (using both batch as well as online inference). Since PTF cannot handle missing data, for this experiment, each method was run with all the data. As Fig 6 shows, our methods have running times that are considerably smaller than that of PTF.

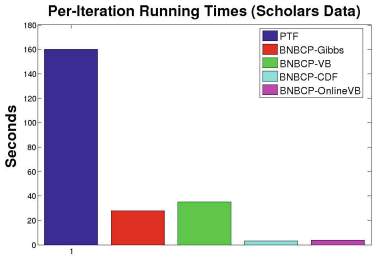


**Fig. 6.** Timing comparison of various methods on Scholars data

## 7 Conclusion

We have presented a fully Bayesian framework for analyzing massive tensors with count data, and have designed a suite of scalable inference algorithms for

handling massive tensor data. In addition to giving interpretable results and inferring the rank from the data, the proposed model can infer the distribution over objects in each of the tensor modes which can be useful for understanding groups of similar objects, and also for doing other types of qualitative analyses on such data, as shown by our various experiments on real-world data sets. Simplicity of the inference procedure also makes the proposed model amenable for parallel and distributed implementations. e.g., using MapReduce or Hadoop. The model can be a useful tool for analyzing data from diverse applications and scalability of the model opens door to the application of scalable Bayesian methods for analyzing massive multiway count data.

**Acknowledgments.** The research reported here was supported in part by ARO, DARPA, DOE, NGA and ONR. Any opinions, findings, recommendations, or conclusions are those of the authors and do not necessarily reflect the views of the Economic Research Service, U.S. Department of Agriculture. The analysis, findings, and conclusions expressed in this paper also should not be attributed to either Nielsen or Information Resources, Inc. (IRI). This research was conducted in collaboration with USDA under a Third Party Agreement with IRI.

## References

1. Bazerque, J.A., Mateos, G., Giannakis, G.B.: Inference of poisson count processes using low-rank tensor data. In: ICASSP (2013)
2. Beutel, A., Kumar, A., Papalexakis, E.E., Talukdar, P.P., Faloutsos, C., Xing, E.P.: Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In: SDM (2014)
3. Cappé, O., Moulines, E.: On-line expectation-maximization algorithm for latent data models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **71**(3), 593–613 (2009)
4. Chi, E.C., Kolda, T.G.: On tensors, sparsity, and nonnegative factorizations. SIAM Journal on Matrix Analysis and Applications **33**(4), 1272–1299 (2012)
5. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons (2009)
6. Dunson, D.B., Herring, A.H.: Bayesian latent variable models for mixed discrete outcomes. Biostatistics **6**(1), 11–25 (2005)
7. Guhaniyogi, R., Qamar, S., Dunson, D.B.: Bayesian conditional density filtering. arXiv preprint arXiv:1401.3632 (2014)
8. Heinrich, G., Goesele, M.: Variational Bayes for Generic Topic Models. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS, vol. 5803, pp. 161–168. Springer, Heidelberg (2009)
9. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
10. Inah, J., Papalexakis, E.E., Kang, U., Faloutsos, C.: Haten2: Billion-scale tensor decompositions. In: ICDE (2015)
11. Johndrow, J.E., Battacharya, A., Dunson, D.B.: Tensor decompositions and sparse log-linear models. arXiv preprint arXiv:1404.0396 (2014)
12. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2), 183–233 (1999)

13. Kang, U., Papalexakis, E., Harpale, A., Faloutsos, C.: Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In: KDD (2012)
14. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review **51**(3), 455–500 (2009)
15. Kozubowski, T.J., Podgórski, K.: Distributional properties of the negative binomial Lévy process. Centre for Mathematical Sciences, Faculty of Engineering, Lund University, Mathematical Statistics (2008)
16. Leetaru, K., Schrodt, P.A.: Gdelt: Global data on events, location, and tone, 1979–2012. ISA Annual Convention **2**, 4 (2013)
17. Papalexakis, E., Faloutsos, C., Sidiropoulos, N.: Parcube: Sparse parallelizable candecomp-parafac tensor decompositions. ACM Transactions on Knowledge Discovery from Data (2015)
18. Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., Carin, L.: Scalable bayesian low-rank decomposition of incomplete multiway tensors. In: ICML (2014)
19. Schein, A., Paisley, J., Blei, D.M., Wallach, H.: Inferring polyadic events with poisson tensor factorization. In: NIPS Workshop (2014)
20. Schmidt, M., Mohamed, S.: Probabilistic non-negative tensor factorisation using markov chain monte carlo. In: 17th European Signal Processing Conference (2009)
21. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: ICML (2005)
22. Zhao, Q., Zhang, L., Cichocki, A.: Bayesian cp factorization of incomplete tensors with automatic rank determination
23. Zhou, G., Cichocki, A., Xie, S.: Fast nonnegative matrix/tensor factorization based on low-rank approximation. IEEE Transactions on Signal Processing 60(6), 2928–2940 (2012)
24. Zhou, M., Hannah, L.A., Dunson, D., Carin, L.: Beta-negative binomial process and poisson factor analysis. In: AISTATS (2012)