Nonparametric Sparse Tensor Factorization with Hierarchical Gamma Processes

Conor Tillinghast [1] **Zheng Wang** [2] **Shandian Zhe** [2]

Abstract

We propose a nonparametric factorization approach for sparsely observed tensors. The sparsity does not mean zero-valued entries are massive or dominated. Rather, it implies the observed entries are very few, and even fewer with the growth of the tensor; this is ubiquitous in practice. Compared with the existent works, our model not only leverages the structural information underlying the observed entry indices, but also provides extra interpretability and flexibility — it can simultaneously estimate a set of location factors about the intrinsic properties of the tensor nodes, and another set of sociability factors reflecting their extrovert activity in interacting with others; users are free to choose a trade-off between the two types of factors. Specifically, we use hierarchical Gamma processes and Poisson random measures to construct a tensor-valued process, which can freely sample the two types of factors to generate tensors and always guarantees an asymptotic sparsity. We then normalize the tensor process to obtain hierarchical Dirichlet processes to sample each observed entry index, and use a Gaussian process to sample the entry value as a nonlinear function of the factors, so as to capture both the sparse structure properties and complex node relationships. For efficient inference, we use Dirichlet process properties over finite sample partitions, density transformations, and random features to develop a stochastic variational estimation algorithm. We demonstrate the advantage of our method in several benchmark datasets.

[1]Department of Mathematics, University of Utah [2]School of Computing, University of Utah. Correspondence to: Shandian Zhe <zhe@cs.utah.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

1 Introduction

Tensor factorization is a popular tool to analyze multiway data, such as purchase records among *(users, products, sellers)* in online shopping. While numerous algorithms have been proposed, *e.g.*, (Tucker, 1966; Harshman, 1970; Chu & Ghahramani, 2009; Kang et al., 2012), most of these methods are essentially modeling dense tensors. For instance, many of them (Kolda & Bader, 2009; Kang et al., 2012; Choi & Vishwanathan, 2014; Xu et al., 2012) use tensor and matrix algebras (Kolda, 2006) to update the latent factors and hence demand all the tensor entries be observed (although the majority are often assumed to be zero-valued). While other methods (Rai et al., 2014; Zhe et al., 2016b; Du et al., 2018), especially with a Bayesian formulation, can perform element-wise factorization to handle a small subset of observed entries, from modeling perspective, they are equivalent to sampling a full tensor first and then marginalizing out the unobserved entry values. One might consider adding a Bernoulli likelihood (or a thresholding function) to model the presence of each entry. However, this still leads to dense tensors. In theory, these models are random function prior models (Lloyd et al., 2012), and the generated mask tensors are exchangeable arrays. According to the Aldous-Hoover theorem (Aldous, 1981; Hoover, 1979), such exchangeability can make the sampled tensors "either trivially empty or dense". That is, the present entries (*i.e.*, with mask one) increases linearly with the size of the tensor.

However, in practice, tensors are often sparsely observed. The ratio of observed entries is very small, *e.g.*, 0.01%. This ratio can even decrease with more nodes added. For example, in online shopping, with the growth of users, products and sellers, the number of actual purchases — *i.e.*, the present entries — while growing, can take an even smaller percentage of all possible purchases (entries), *i.e.*, all (user, product, seller) combinations, because the latter grows much faster. Hence, the (asymptotic) sparse nature in these real-world applications implies that the existent dense models are misspecified. Note that the sparsity here does not refer to that the zero-valued entries are massive or dominated (Choi & Vishwanathan, 2014; Hu et al., 2015)

To address the model misspecification, Tillinghast & Zhe (2021) recently used Dirichlet processes (DPs) (normalized

Gamma processes) and GEM distributions to build two asymptotically sparse tensor models, which have shown significant advantages over the popular dense models in entry value and index (link) prediction. However, their model structures severely restrict the type of factors to be learned — either all the factors are sociability factors to account for the generation of the interactions, or only one factor is the sociability factor and the remaining are location factors about the intrinsic properties of the tensor nodes. The former misses the chance to capture the intrinsic node properties while the latter can be insufficient to estimate the sociabilities. Accordingly, both models can limit the flexibly of the factor estimation and hence lower the interpretability.

To address these limitations, we propose a new nonparametric factorization model, which not only guarantees an asymptotic sparsity, but also provides more flexibility and interpretability. Our model can estimate an arbitrary number of sociability and location factors so as to fully capture the two types of tensor-node properties. Specifically, we use hierarchical Gamma processes (HΓP) to construct a summation of product measures, with which we construct a tensor-variate process. We show that the proportion of the sampled entries is asymptotically tending to zero with the growth of tensor size. More important, the ΓPs in the top level enable the sharing of locations across the low-level ΓPs. As a result, for each node, our model can sample not only an arbitrary number of location factors representing the node's intrinsic properties, but also an arbitrary number of sociability factors accounting for its extrovert interaction activity in different communities. Hence, the results can provide more interpretability and users are flexible to select a trade-off. Next, for convenient modeling and inference, we normalize our tensor process to obtain hierarchical Dirichlet processes so as to sample the observed entry indices; we then use a Gaussian process to sample the entry value as a nonlinear function of the latent factors of the associated nodes so as to embed both the sparse structural knowledge and complex relationships between the tensor nodes. For efficient inference, we use the Dirichlet process property over finite, non-overlapping sample space partitions to aggregate the sociabilities of non-active nodes, probability density transformations, and random Fourier features to develop a scalable stochastic variational estimation algorithm.

For evaluation, we first ran simulations to confirm that our method can indeed generate increasingly sparser tensors. In three real-world applications, we examined the performance in predicting missing entry indices and entry values. In both tasks, our approach obtains the best performance, as compared with the state-of-the-art dense models and (Tillinghast & Zhe, 2021). The factors estimated by our method reveal more clear and interesting clustering patterns than those from the prior sparse models in (Tillinghast & Zhe, 2021), while the factors estimated from the popular dense models do not exhibit any patterns.

2 Background

Notation and Settings. We denote a full K-mode tensor by $\mathcal{Y} \in \mathbb{R}^{D_1 \times \ldots \times D_K}$, where mode k includes D_k nodes. Each entry is indexed by $\mathbf{i} = (i_1, \ldots, i_K)$ and the entry value by $y_{\mathbf{i}}$. Practical tensors are often very sparse. That means most entries of $\mathcal{Y}$ are nonexistent. Given a small set of present entries, $\mathcal{D} = \{(\mathbf{i}_1, y_{\mathbf{i}_1}), \ldots, (\mathbf{i}_N, y_{\mathbf{i}_N})\}$ where $N \ll \prod_{k=1}^K D_k$, we aim to estimate a set of factors $\mathbf{u}_j^k$ to represent each node j in mode k. We denote by $\mathcal{U} = \{\mathbf{U}^1, \ldots, \mathbf{U}^K\}$ all the factors, where each $\mathbf{U}^k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_{D_k}^k]^\top$.

Dense Tensor Models. The classical Tucker (Tucker, 1966) and CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970) require $\mathcal{Y}$ to be fully observed. The Tucker decomposition assumes $\mathcal{Y} = \mathcal{W} \times_1 \mathbf{U}^1 \times_2 \ldots \times_K \mathbf{U}^K$ where $\mathcal{W} \in \mathbb{R}^{r_1 \times \ldots \times r_K}$ is a parametric core tensor and $\times_k$ is the tensor-matrix product at mode k (Kolda, 2006). CP decomposition is a special case of Tucker decomposition where $\mathcal{W}$ is restricted to be diagonal. While many methods can perform element-wise factorization over a subset of observations, *e.g.*, (Zhe et al., 2016b; Rai et al., 2015; Schein et al., 2015; 2016b; Zhe & Du, 2018), from the Bayesian perspective, these methods can be viewed as first sampling all the entry values of $\mathcal{Y}$ given the factors $\mathcal{U}$, and then marginalize out the unobserved ones. In other words, they never consider the sparsity of the observed entry indices (links). Therefore, these methods are in essence still modeling a dense tensor. In theory, this has been justified by the random function prior models (Lloyd et al., 2012), where a Bernoulli distribution is used to sample the presence (or existence) of each entry given the factors, $p(\mathcal{Z}|\mathcal{U}) = \prod_{\mathbf{i}} p(z_{\mathbf{i}}|\mathcal{U}) = \prod_{\mathbf{i}} \text{Bern}(z_{\mathbf{i}}|f(\mathbf{x}_{\mathbf{i}}))$, where $\mathcal{Z}$ is a mask tensor, $z_{\mathbf{i}} = 1$ means the entry $\mathbf{i}$ is generated, and $z_{\mathbf{i}} = 0$ means $\mathbf{i}$ is nonexistent, f is a function of latent factors, *e.g.*, element-wise CP/Tucker form or a Gaussian process (GP). The mask tensor $\mathcal{Z}$ is known to be exchangeable (Lloyd et al., 2012). According to the Aldous-Hoover theorem (Aldous, 1981; Hoover, 1979), the asymptotic[1] proportion of the present entries (*i.e.*, the entries with mask 1) is $\eta = \int \text{Bern}(z_{\mathbf{i}}|f(\mathbf{x}_{\mathbf{i}}))\text{d}p(\mathcal{U})$. This implies that the sampled tensors are either empty ($\eta = 0$) or dense ($\eta > 0$) almost surely, *i.e.*, the number of sampled entries grows linearly with the tensor size ($\Theta(\eta \prod_{k=1}^K D_K)$).

Sparse Tensor Model. Recently, Tillinghast & Zhe (2021) proposed nonparametric models that guarantee to generate asymptotically sparse tensors, *i.e.*, the proportion of sampled entries tend to zero with the growth of the tensor size ($o(\prod_{k=1}^K D_K)$). The idea is to use Gamma processes (ΓPs)

[1] by asymptotic, it means when we sample a sequence of $\mathcal{Z}$'s, with increasing sizes (till infinity).

to construct a tensor-variate random process. This is equivalent to sampling a Dirichlet process (Ferguson, 1973) (normalized ΓP) for each mode k, and combining the weights of each node to sample the present entries,

$$\mathrm{DP}^k = \sum\nolimits_j w_j^k \delta_{\boldsymbol{\theta}_j^k}, \quad p(\mathbf{i}_n) = \prod\nolimits_k w_{i_{nk}}^k, \tag{1}$$

where $1 \le k \le K$, $1 \le n \le N$, w_j^k and $\boldsymbol{\theta}_j^k$ are the DP weights and locations for node j in mode k, which naturally represent the sociability and location factors. The sociability factor w_j^k represents the activity of the node in interacting with others to generate the entry index, *i.e.*, link. The location factors naturally represent the intrinsic properties of the node. Then w_j^k and $\boldsymbol{\theta}_j^k$ are jointly used to sample the value of the observed entry $y_{\mathbf{i}_n}$. To incorporate multiple sociability factors, Tillinghast & Zhe (2021) proposed a second model that draws multiple weights for each node via the GEM distribution (Griffiths, 1980; Engen, 1975; McCloskey, 1965). Note that GEM essentially samples DP weights (no locations). These weights are integrated to sample both the observed entry indices and entry values.

3 Model

While successful, the popular dense tensor models can be misspecified for the sparse data that are ubiquitous in practice. Although the work of (Tillinghast & Zhe, 2021) have overcome this issue, their model structures severely restrict the type of the factors to be estimated. From (1), we can only estimate one sociability factor for each node to account for their complex interactions (*i.e.*, links), which can be quite insufficient. However, to incorporate multiple sociability factors, we have to use GEM distributions and drop the location factors (the second model). Accordingly, we cannot capture any intrinsic properties of the nodes. Note that we cannot simply draw multiple DPs, because the indices of the nodes in different DPs are not necessarily the same and we are unable to align their location factors.

To address these limitations, we propose a new sparse tensor factorization model, which not only guarantees the asymptotic sparsity in tensor generation, but also is free to estimate an arbitrary number of sociability and location factors so as to fully capture the extrovert and intrinsic properties of the tensor nodes. Accordingly, our model can provide more interpretability and additional flexibly to choose the trade-off between the two types of the factors (*e.g.*, via cross-validation).

3.1 HΓP Based Sparse Tensor-Variate Process

Specifically, we use hierarchical Gamma processes (HΓPs) to construct a sparse tensor-variate process. At the top level, we sample a Gamma process (ΓP) (Hougaard, 1986) in each mode k,

$$L_k^\alpha \sim \Gamma\mathrm{P}(\lambda_\alpha) \ \ (1 \le k \le K), \tag{2}$$

where λ_α is a Lebesgue base measure restricted to $[0,\alpha]^{R_1} = \underbrace{[0,\alpha] \times \ldots \times [0,\alpha]}_{R_1 \text{times}} (\alpha > 0)$. Accordingly, L_k^α is a discrete measure,

$$L_k^\alpha = \sum\nolimits_{j=1}^{\infty} w_{kj}^\alpha \cdot \delta_{\boldsymbol{\theta}_j^k} \tag{3}$$

where $\delta_{[\cdot]}$ is the Dirac Delta measure, $\{w_{kj}^\alpha > 0\}$ and $\{\boldsymbol{\theta}_j^k \in [0,\alpha]^{R_1}\}$ are the weights and locations, which correspond to an infinite number of tensor nodes in mode k. We refer to each $\boldsymbol{\theta}_j^k$ as R_1 location factors to represent the intrinsic properties of node j in mode k. In the second level, we use L_k^α as the base measure to sample R_2 Gamma processes in each mode k,

$$W_{k,r}^\alpha \mid L_k^\alpha \sim \Gamma\mathrm{P}(L_k^\alpha), \ \ W_{k,r}^\alpha = \sum\nolimits_{j=1}^{\infty} v_{krj}^\alpha \cdot \delta_{\boldsymbol{\theta}_j^k}, \tag{4}$$

where $1 \le r \le R_2$. Note that due to the common base measure L_k^α, the locations are shared across all $\{W_{k,r}^\alpha\}_{r=1}^{R_2}$. Hence, for each node j in mode k, we not only generate R_1 location factors $\boldsymbol{\theta}_j^k$, but also R_2 weights, $\{v_{krj}^\alpha\}_{r=1}^{R_2}$. We refer to the normalized weights as sociability factors, since we will use them to sample the observed entry indices (links). More detailed discussions will be given in Sec. 3.2. Next, we use the HΓPs to construct a product measure sum as the rate (or mean) measure to sample a Poisson random measure (PRM) (Kingman, 1992), which represents the sampled tensor entries,

$$\begin{aligned} &A = \sum\nolimits_{r=1}^{R_2} W_{1,r}^\alpha \times \ldots \times W_{K,r}^\alpha, \\ &T \mid \{W_{k,r}^\alpha\}_{1\le k\le K, 1\le r\le R_2} \sim \mathrm{PRM}(A). \end{aligned} \tag{5}$$

Accordingly, T has the following form, $T = \sum_{\mathbf{i}\in\mathcal{S}} c_{\mathbf{i}} \cdot \delta_{(\boldsymbol{\theta}_{i_1}^1, \ldots, \boldsymbol{\theta}_{i_K}^K)}$, where each point represents an entry index, $\mathcal{S}$ is the set of all the sampled points in the PRM, *i.e.*, entry indices, $c_{\mathbf{i}} > 0$ is the count of the point $\mathbf{i}$, and $(\boldsymbol{\theta}_{i_1}^1, \ldots, \boldsymbol{\theta}_{i_K}^K)$ is the location of that point.

In a nut shell, the HΓPs sample an infinite number of nodes in each mode, then the PRM picks the nodes from each mode to generate the entries (*i.e.*, links).

Lemma 3.1. *For any fixed $\alpha > 0$, the number of sampled entries $N^\alpha = |\mathcal{S}|$ via (2)(4)(5) is finite almost surely. When $\alpha \to \infty$, $N^\alpha \to \infty$ a.s.*

We leave the proof in Appendix. Note that this conclusion also applies to the sparse tensor models of Tillinghast & Zhe (2021), but they did not provide a proof. Our proof can be slightly modified to apply to their models. Given the sampled tensor, we are interested in the active nodes that

show up in the sampled entry indices $\mathcal{S}$. For example, if an entry $(1, 3, 5)$ is sampled, then the nodes 1, 3, and 5 in mode 1, 2, and 3, respectively are active nodes. Denote by D_k^α the number of (distinct) active nodes in mode k. If we use these active nodes to construct a full tensor, which we refer to as the active tensor, the size will be $\prod_{k=1}^K D_k^\alpha$. The asymptotic sparsity means the proportion of the present entries in the active tensor is small, and the increase of present entries is much slower than the growth of the active tensor size. That is,

Lemma 3.2. *$N^\alpha = o(\prod_{k=1}^K D_k^\alpha)$ almost surely as $\alpha \to \infty$, i.e., $\lim_{\alpha\to\infty} \frac{N^\alpha}{\prod_{k=1}^K D_k^\alpha} = 0 \quad a.s.$*

Due to one more level of ΓPs, the proof of the sparsity becomes more challenging. We used Campbell's Theorem, ΓP properties, independence of CRMs on disjoint sets, and the strong law of large numbers to adjust our HΓPs to the proof framework in (Tillinghast & Zhe, 2021). We leave the details in Appendix. We refer to the model defined by (2)(4)(5) as our Sparse Tensor-variate Process (STP).

3.2 Projection on Finite Data

Directly using the STP for modeling and inference is inconvenient, because the infinite discrete measures of ΓPs and PRMs are computationally challenging. On the other hand, in practice, we always observe a finite number of entries, $\mathcal{D} = \{(\mathbf{i}_1, y_1), \ldots, (\mathbf{i}_N, y_N)\}$. Hence, we can use the standard PRM construction (Kingman, 1992) to equivalently model the sampling procedure. That is, we normalize the rate measure $A = \sum_{r=1}^{R_2} W_{1,r}^\alpha \times \ldots \times W_{K,r}^\alpha$ in (5) to obtain a probability measure, and use it to sample N entry indices (points) independently. To normalize A, we need to first normalize each ΓP $W_{k,r}^\alpha (1 \le k \le K, 1 \le r \le R_2)$, which gives a Dirichlet process (DP) (Ferguson, 1973) with the base measure as the normalized base measure of $W_{k,r}^\alpha$. Since the base measure of $W_{k,r}^\alpha$ is another ΓP, L_k^α (see (2)), its normalization is a DP again. Hence, we obtain a set of hierarchical Dirichlet processes (HDPs) (Teh et al., 2006),

$$\begin{aligned} G_k &\sim \mathrm{DP}\big(\widetilde{\alpha}, \mathrm{Uniform}([0,\alpha]^{R_1})\big), \\ H_r^k &\sim \mathrm{DP}(\gamma_r^k, G_k) \end{aligned} \tag{6}$$

where $1 \le k \le K$, $1 \le r \le R_2$, $\widetilde{\alpha} = \alpha^{R_1}$, and $\gamma_r^k \sim \mathrm{Gamma}(\cdot|1, \widetilde{\alpha})$. Then the normalized A is $\widehat{A} = \frac{1}{R_2}\sum_{r=1}^{R_2} H_r^1 \times \ldots \times H_r^K$. The sampling of the N observed entry indices is as follows. We first sample each G_k and H_r^k from (6), which gives

$$G_k = \sum\nolimits_{j=1}^\infty \beta_j^k \cdot \delta_{\boldsymbol{\theta}_j^k}, \quad H_r^k = \sum\nolimits_{j=1}^\infty \omega_{rj}^k \cdot \delta_{\boldsymbol{\theta}_j^k}, \tag{7}$$

where each location $\boldsymbol{\theta}_j^k$ is independently sampled from $\mathrm{Uniform}([0,\alpha]^{R_1})$. Then we obtain the sample of the normalized rate measure as

$$\widehat{A} = \sum\nolimits_{\mathbf{i}=(1,\ldots,1)}^{(\infty,\ldots,\infty)} w_{\mathbf{i}} \cdot \delta_{(\boldsymbol{\theta}_{i_1}^1, \ldots, \boldsymbol{\theta}_{i_K}^K)}, \tag{8}$$

where

$$w_{\mathbf{i}} = \frac{1}{R_2} \sum_{r=1}^{R_2} \prod_{k=1}^K \omega_{r i_k}^k.$$

We can see $\widehat{A}$ is essentially a probability measure over all possible tensor entries, *i.e.*, $\sum_{\mathbf{i}} w_{\mathbf{i}} = 1$. We then sample each observed entry $\mathbf{i}_n \sim \widehat{A}$, and hence the probability is

$$p(\mathcal{S}) = \prod\nolimits_{n=1}^N p(\mathbf{i}_n) = \prod\nolimits_{n=1}^N w_{\mathbf{i}_n}. \tag{9}$$

Now, it can be seen clearly that in (6) and (7), for each node j in mode k, we sample the location $\boldsymbol{\theta}_j^k$ and a set of weights $\boldsymbol{\nu}_j^k = [\omega_{1j}^k, \ldots, \omega_{R_2 j}^k]$ from R_2 DPs. From (8), we can see that these weights represent the activity of the node interacting with other nodes. Each weight naturally corresponds to the activity in one community/group. Hence, we refer to $\boldsymbol{\nu}_j^k$ as the sociability factors of the node in R_2 overlapping communities. The location factors $\boldsymbol{\theta}_j^k$ naturally represent the node's intrinsic properties. Thanks to the HDP (HΓP) structure, the locations can be shared in an arbitrary number of low-level DPs (ΓPs). That means, we are free to choose the number of sociability factors and location factors. Hence, it provides not only more interpretability but also an extra flexibility to select their trade-off.

Given the sampled entries $\mathcal{S}$, we then sample the entry values $\mathbf{y} = [y_{\mathbf{i}_1}, \ldots, y_{\mathbf{i}_N}]$. In this work, we mainly consider continuous values. It is straightforward to extend our model for other value types. We sample each value from a Gaussian noise model,

$$p(y_{\mathbf{i}_n}|\mathcal{S}) = \mathcal{N}(y_{\mathbf{i}_n}|f(\mathbf{x}_{\mathbf{i}_n}), \sigma^2), \tag{10}$$

where $\mathbf{x}_{\mathbf{i}_n} = [\mathbf{u}_{i_{n1}}^1; \ldots; \mathbf{u}_{i_{nK}}^K]$ are the factors of all the nodes associated with entry $\mathbf{i}_n$, each $\mathbf{u}_{i_{nk}}^k = [\boldsymbol{\theta}_{i_{nk}}^k; \boldsymbol{\nu}_{i_{nk}}^k]$, *i.e.*, including R_1 location and R_2 sociability factors, and $f(\cdot)$ is a latent factorization function. To flexibly estimate $f(\cdot)$ so as to capture the complex relationships between the tensor nodes (in terms of their factor representations), we assign a Gaussian process (Rasmussen & Williams, 2006) prior over $f(\cdot)$. Accordingly, the function values $\mathbf{f} = [f(\mathbf{x}_{\mathbf{i}_1}), \ldots, f(\mathbf{x}_{\mathbf{i}_N})]$ follow a joint Gaussian distribution, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is an $N \times N$ kernel (covariance) matrix, each element $[\mathbf{K}]_{st} = \kappa(\mathbf{x}_{\mathbf{i}_s}, \mathbf{x}_{\mathbf{i}_t})$, and $\kappa(\cdot,\cdot)$ is a kernel function.

From (9) and (10), we can see via coupling the HDPs and GP, both the structural knowledge in sparse entry indices and complex relationships of the tensor nodes can be captured and encoded into the sociability and location factors.

4 Algorithm

The estimation of our model is still challenging. First, we have to deal with infinite discrete measures (see (7)) from HDPs. Second, the exact GP prior includes an $N \times N$ covariance matrix. When N is large, the computation is very costly or even infeasible. To address these issues, we use the DP property over sample partitions (Ferguson, 1973), probability density transformations, and random Fourier features (Rahimi et al., 2007; Lázaro-Gredilla et al., 2010) to develop an efficient, scalable variational learning algorithm.

Specifically, since we only need to estimate the latent factors for finite active nodes, *i.e.*, the nodes appearing in the observed entries (the infinite inactive nodes do not have any observed data), we can marginalize (7) into finite measures. Suppose we have D_k active nodes in each mode k. For convenience, we index them (the observed nodes) by $1, \ldots, D_k$, and the inactive nodes by $D_{k+1}, \ldots, \infty$[2]. In the first level, for each G_k, we consider $\boldsymbol{\beta}^k = [\beta_1^k, \ldots, \beta_{D_k}^k, \beta_{>D_k}^k]$ where $\beta_{>D_k}^k = \sum_{j=D_k+1}^{\infty} \beta_j^k$, *i.e.*, the DP weights for every active node and the aggregated weight of all the inactive nodes. In the second level, we partition the sample space of each H_r^k into $D_k + 1$ subsets, $\{\{\boldsymbol{\theta}_1^k\}, \ldots, \{\boldsymbol{\theta}_{D_k}^k\}, \{\text{Others}\}\}$. Since $H_r^k \sim \text{DP}(\gamma_r^k, G_k)$, its measures of the $K+1$ subsets will jointly follow a Dirichlet distribution. The values of these measures are $\boldsymbol{\omega}_r^k = \{\omega_{r1}^k, \ldots, \omega_{rD_k}^k, \omega_{r>D_k}^k\}$, where $\omega_{r>D_k}^k = \sum_{j=D_k+1}^{\infty} \omega_{rj}^k$, namely, the sociability of each active node and the aggregated sociability of all the inactive nodes. The Dirichlet distribution is parameterized by $[\gamma_r^k G_k(\{\boldsymbol{\theta}_1^k\}), \ldots, \gamma_r^k G_k(\{\boldsymbol{\theta}_{D_k}^k\}), \gamma_r^k G_k(\{\text{Others}\})] = \gamma_r^k \boldsymbol{\beta}^k$. Therefore, we have

$$p(\boldsymbol{\omega}_r^k | \gamma_r^k, \boldsymbol{\beta}^k) = \text{Dir}(\boldsymbol{\omega}_r^k | \gamma_r^k \boldsymbol{\beta}^k) = \frac{\Gamma(\gamma_r^k)}{\prod_j \Gamma(\gamma_r^k \beta_j^k)} \prod_j \left(\omega_{rj}^k\right)^{\gamma_r^k \beta_j^k - 1}, \tag{11}$$

where $\Gamma(\cdot)$ is a Gamma function. Given the sociabilities of the active nodes, we have already been able to sample the observed entry indices from (9). The remaining problem is how to obtain the prior of $\boldsymbol{\beta}^k$ for each G_k. From the stick-breaking construction (Ishwaran & James, 2001), we have

$$\xi_j^k \sim \text{Beta}(1, \widetilde{\alpha}), \quad \beta_j^k = \xi_j^k \prod_{t=1}^{j-1} (1 - \xi_t^k) \ (1 \le j \le \infty),$$

from which we can obtain a reverse mapping,

$$(\xi_1^k, \ldots, \xi_{D_k}^k) = \left(\frac{\beta_1^k}{\Lambda_1^k}, \ldots, \frac{\beta_{D_k}^k}{\Lambda_{D_k}^k}\right), \tag{12}$$

where $\Lambda_j^k = 1 - \sum_{t=1}^{j-1} \beta_t^k = \prod_{t=1}^{j-1}(1 - \xi_t^k)$. Therefore, we can use probability density transformation to obtain the prior distribution of $\boldsymbol{\beta}^k$. Since each Λ_j^k is only determined by $\{\beta_1^k, \ldots, \beta_{j-1}^k\}$, the Jacobian $\mathbf{J}$ is a lower-triangular matrix and $|\mathbf{J}| = \prod_{j=1}^{D_k} \frac{1}{\Lambda_j^k}$. Accordingly, we can derive

$$p(\boldsymbol{\beta}^k) = \prod_{j=1}^{D_k} \text{Beta}\left(\frac{\beta_j^k}{\Lambda_j^k} | 1, \widetilde{\alpha}\right) \frac{1}{\Lambda_j^k}. \tag{13}$$

Note that since $\beta_{>D_k}^k = 1 - \sum_{j=1}^{D_k} \beta_j^k$, it does not need to be explicitly computed in the prior. To conveniently estimate each $\boldsymbol{\beta}^k$ and $\boldsymbol{\omega}_r^k$ and to avoid incorporating additional constraints (nonnegative, summation is one), we parameterize $\boldsymbol{\beta}^k = \text{softmax}(\widetilde{\boldsymbol{\beta}}^k)$ and $\boldsymbol{\omega}_r^k = \text{softmax}(\widetilde{\boldsymbol{\omega}}_r^k)$ and estimate the free parameters $\{\widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\omega}}_r^k\}$ instead.

Next, to scale to a large number of observed entries, we use random Fourier features to construct a sparse GP approximation (Rahimi et al., 2007; Lázaro-Gredilla et al., 2010). Specifically, according to Bochner's theorem (Rudin, 1962), any stationary kernel $\kappa(\mathbf{b}_1, \mathbf{b}_2) = \kappa(\mathbf{b}_1 - \mathbf{b}_2)$ can be viewed as an expectation, $\kappa(\mathbf{b}_1 - \mathbf{b}_2) = \kappa(\mathbf{0}) \mathbb{E}_{p(\mathbf{z})}[e^{i\mathbf{z}^\top \mathbf{a}_1} \left(e^{i\mathbf{z}^\top \mathbf{a}_2}\right)^\dagger]$, where $\dagger$ is the complex conjugate, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \tau\mathbf{I})$ is computed from the Fourier transform of $\kappa(\cdot)$. Hence, we can draw M frequencies $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$ from $p(\mathbf{z})$ to construct an Monte-Carlo approximation, $\kappa(\mathbf{b}_1, \mathbf{b}_2) \approx \frac{\kappa(\mathbf{0})}{M} \sum_{m=1}^{M} e^{i\mathbf{s}_m^\top \mathbf{b}_1} \left(e^{i s_m^\top \mathbf{b}_2}\right)^\dagger = \frac{\kappa(\mathbf{0})}{M} \boldsymbol{\phi}(\mathbf{b}_1)^\top \boldsymbol{\phi}(\mathbf{b}_2)$, where

$$\boldsymbol{\phi}(\mathbf{b}) = [\cos(\mathbf{z}_1^\top \mathbf{b}), \sin(\mathbf{z}_1^\top \mathbf{b}), \ldots, \cos(\mathbf{z}_M^\top \mathbf{b}), \sin(\mathbf{z}_M^\top \mathbf{b})].$$

Here, $\boldsymbol{\phi}(\cdot)$ is a $2M$ dimensional nonlinear feature mapping, which is called random Fourier features. We can then use a Bayesian linear regression model over the random Fourier features to approximate the GP. Specifically, we use the RBF kernel, $\kappa(\mathbf{b}_1, \mathbf{b}_2) = \exp(-\frac{1}{2}\tau \|\mathbf{b}_1 - \mathbf{b}_2\|^2)$. The corresponding frequency distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \tau\mathbf{I})$. We sample M frequencies, $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$, from $p(\mathbf{Z}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{z}_m|\mathbf{0}, \tau\mathbf{I})$, and a weight vector $\mathbf{g}$ from $p(\mathbf{g}) = \mathcal{N}(\mathbf{g}|\mathbf{0}, \frac{1}{M}\mathbf{I})$. We then represent the function $f(\cdot)$ in (10) by

$$f(\mathbf{x}_{\mathbf{i}_n}) = \boldsymbol{\phi}(\mathbf{x}_{\mathbf{i}_n})^\top \mathbf{g}, \tag{14}$$

[2] In practice, it is natural to index the observed nodes by consecutive integers, and there seems no rationality or necessity to leave gaps between these indices (e.g., 1, 3, 7, 8. . .) to pre-allocate indices for nodes that never appear in the data (*i.e.*, inactive nodes). However, it is straightforward for our algorithm to handle the case where the indices of active and inactive nodes are interleaving. All we need is to compute the measure over each node indexed from one to the index of the last active node, *i.e.*, including all the active nodes and the inactive nodes whose indices are less than the largest index of the active nodes. See the details in the following presentation.

where the input $\mathbf{x}_{\mathbf{i}_n}$ includes the factors of all the nodes associated with $\mathbf{i}_n$. Since the sociabilities are often very small and close to zero, we use their softmax parameters $\{\widetilde{\omega}^k_{i_{nk}}\}$ in $\mathbf{x}_{\mathbf{i}_n}$. When we marginalize out the weight vector $\mathbf{g}$, the joint Gaussian distribution is recovered, and the kernel takes the form of the Monte-Carlo approximation. We will keep $\mathbf{g}$ to prevent computing the full covariance matrix, and to scale to a large number of observed entries.

Combining (13)(11)(9)(14)(10), we obtain the joint probability of our model,

$$p(\text{Joint}) = \prod_{k=1}^{K} p(\boldsymbol{\beta}^k) \left(\prod_{j=1}^{D_k} \text{Uniform}(\boldsymbol{\theta}^k_j | [0, \alpha]^{R_1}) \right) \cdot \left(\prod_{r=1}^{R_2} \text{Gamma}(\gamma^k_r | 1, \widetilde{\alpha}) \text{Dir}(\boldsymbol{\omega}^k_r | \gamma^k_r \boldsymbol{\beta}^k) \right) \mathcal{N}(\mathbf{g} | \mathbf{0}, \frac{1}{M}\mathbf{I}) \cdot \prod_{m=1}^{M} \mathcal{N}(\mathbf{z}_m | \mathbf{0}, \tau\mathbf{I}) \prod_{n=1}^{N} w_{\mathbf{i}_n} \cdot \mathcal{N}(y_{\mathbf{i}_n} | \boldsymbol{\phi}(\mathbf{x}_{\mathbf{i}_n})^\top \mathbf{g}, \sigma^2).$$

We use variational inference (Wainwright & Jordan, 2008) for model estimation. We introduce a variational posterior $q(\mathbf{g}) = \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}, \mathbf{L}\mathbf{L}^\top)$, and construct a variational evidence lower bound (ELBO), $\mathcal{L} = \mathbb{E}_q [\log p(\text{Joint}) - \log q(\mathbf{g})]$. We maximize the ELBO to estimate $q(\mathbf{g})$ and all the other parameters, including $\{\widetilde{\boldsymbol{\beta}}^k\}$, $\{\boldsymbol{\theta}^k_j\}$, $\{\gamma^k_r, \widetilde{\boldsymbol{\omega}}^k_r\}$, $\mathbf{Z}$, *etc*. Due to the additive structure over the observed entries in $\mathcal{L}$, it is straightforward to use mini-batch stochastic optimization for efficient and scalable estimation.

Complexity. The time complexity of our inference is $O(NM^2 + R\sum_{k=1}^{K} D_k)$ where $R = R_1 + R_2$, and N is the number of observed entries. Since M is fixed and $M \ll N$, the time complexity is linear in N. The space complexity is $O(M^2 + M + \sum_k D_k R)$, which is to store the latent factors, the frequencies, and the variational posterior.

5 Related Work

Numerous tensor factorization methods have been developed, *e.g.*, (Chu & Ghahramani, 2009; Yang & Dunson, 2013; Choi & Vishwanathan, 2014; Du et al., 2018; Liu et al., 2018; Fang et al., 2021a;b). While most methods adopt the multilinear forms of the classical Tucker (Tucker, 1966) and CP decompositions (Harshman, 1970), a few GP based factorization models (Xu et al., 2012; Zhe et al., 2015; 2016a;b; Tillinghast et al., 2020; Pan et al., 2020b) were recently proposed to capture the nonlinear relationships between the tensor nodes.

Most methods model dense tensors. From the Bayesian viewpoint, they all belong to random function prior models (Lloyd et al., 2012) on exchangeable arrays. Recently, Tillinghast & Zhe (2021) used Gamma processes (ΓPs), a commonly used completely random measure (CRM) (Kingman, 1967; 1992; Lijoi et al., 2010), to construct a Poisson random measure that represents asymptotically sparse tensors. Their work can be viewed as an extension of the pioneer works Caron & Fox (2014); Williamson (2016); Caron & Fox (2017) in sparse random graph modeling. However, their single-level ΓPs and accordingly single-level Dirichlet processes (DPs) for finite projection impose a severe restriction on the type of the learned factors. For each tensor node, either all the factors are exclusively sociability factors or only one factor can be the sociability factor. The former misses the location factors that represent the intrinsic properties of the node, and the latter is often inadequate to capture the extrovert interaction activity across different communities. To overcome this limitation, our model uses hierarchical Gamma processes (HΓPs) to construct the sparse tensor model. Due to the sharing of the locations of ΓPs in the second level, our model does not have the node alignment issue and is able to estimate an arbitrary set of sociability and location factors to fully capture the extrovert activities and inward attributes respectively, hence giving more interpretability and an additional flexibility to select the number of two types of factors. We couple the normalized HΓPs and GPs to sample both the sparse entry indices (hyperlinks) and entry values, so as to absorb both the sparse structural knowledge and complex relationships into the factor estimates. We point out that the normalized HΓP is HDP (Teh et al., 2006), an important extension of DP (Ferguson, 1973). HDPs are popular in nonparametric mixture modeling in text mining (Hong & Davison, 2010; Wang et al., 2011; McFarland et al., 2013), which is often referred to as topic models. For inference, Tillinghast & Zhe (2021) directly used the stick-breaking construction of DPs, which, however, are not available for our model, because it is intractable for HDP inference. Instead, we used DP measures over a finite partition of the sample space to integrate the measures of non-active nodes, and density transformation to obtain the density of finite point masses. The strategy was also used in HDP mixture models, *e.g.*, (Liang et al., 2007; Bryant & Sudderth, 2012). Schein et al. (2016a) used Gamma and Poisson distributions to develop a Bayesian Poisson Tucker factorization model. They discussed that if one uses a Gamma process instead of the Gamma distribution to sample the latent core tensor, the core tensor will be asymptotically sparse. However, it is unclear if the tensor itself will also be sparse. Recently, (Ayed & Caron, 2021) developed a Poisson factorization model with completely random measures to infer unbounded, overlapping communities, and used the model for network analysis.

Another recent line of research (Zhe & Du, 2018; Pan et al., 2020a; Wang et al., 2011) uses GPs to construct different point processes for temporal events modeling and embedding the participants of these events. While these problems are formulated as decomposing a special type of tensors, *i.e.*, event-tensors, where each entry is a sequence of temporal events (rather than numerical values), these models focus

on building expressive non-Poisson processes to capture complex temporal dependencies between the events, *e.g.*, the Hawkes processes with local or global triggering effects (Zhe & Du, 2018; Pan et al., 2020b), the non-Poisson, non-Hawkes process (Wang et al., 2020) that captures the long-term, short-term, triggering and inhibition effects.

6 Experiment

6.1 Simulation

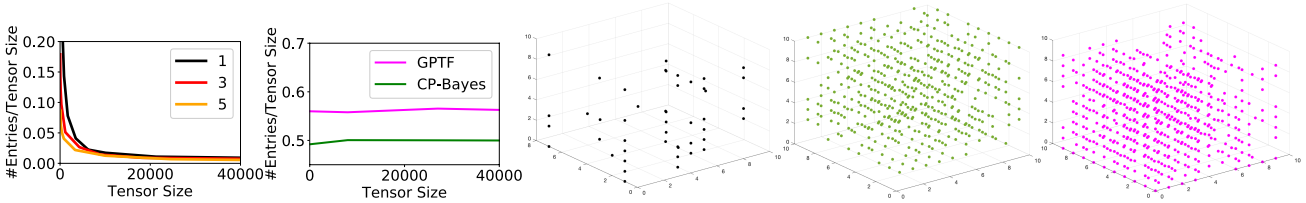
First, we examined the sparsity of the tensor generated by our model. Following (Tillinghast & Zhe, 2021), we generated a series of tensors with increasingly more present entries, and examined how the sparse ratio — the proportion of the present entries — varied accordingly (see details in Appendix). We fixed $R_1 = 1$ and varied the number of sociability factors (R_2) from $\{1, 3, 5\}$, and α from $[1, 15]$. For each particular α, we ran the simulation for 100 times and calculated the average ratio. As shown in Fig. 1 a, the sparse ratio drops rapidly with the growth of the active tensor and is tending to zero (the gap between the curves and the horizontal axis is decreasing). This is consistent with the (asymptotic) sparse guarantee in Lemma 3.2. In contrast to Fig. 1 b, the popular factorization models, CP-Bayes (Zhao et al., 2015; Du et al., 2018) and GPTF (Zhe et al., 2016b), generated dense tensors, where the sparse ratio is almost constant, reflecting that the present entry number grows linearly with the tensor size. In Fig. 1c-e, we showcase exemplar tensors generated by each model. We can see that the tensors entries sampled from CP-Bayes and GPTF (Fig. 1d and e) are much denser than ours (Fig. 1c) and more uniform. This is consistent with the fact that their priors are exchangeable and symmetric, and each entry is sampled with the same probability (see Sec. 2).

6.2 Predictive Performance in Practical Applications

Next, we evaluated the predictive performance in three real-world benchmark datasets: (1) *Alog* (Zhe et al., 2016b), extracted from a file access log, depicting the access frequency among *users*, *actions*, and *resources*, of size $200 \times 100 \times 200$. The value of each present entry is the logarithm of the access frequency. The proportion of present entries is 0.3%. (2) *MovieLens* (`https://grouplens.org/datasets/movielens/100k/`), a three-mode *(user, movie, time)* tensor, of size $1000 \times 1700 \times 31$. There are 100K present entries, taking 0.19% of the whole tensor. The entry values are movie ratings (1-5). We divided every value by 5 to normalized them to $[0, 1]$. (3) *SG* (Li et al., 2015; Liu et al., 2019), extracted from Foursquare Singapore data, representing *(user, location, point-of-interest)* check-ins, of size $2321 \times 5596 \times 1600$. The value of each present entry is the normalized check-in frequency in $[0, 1]$. The number of present entries is $105,764$, taking 0.0005% of whole tensor.

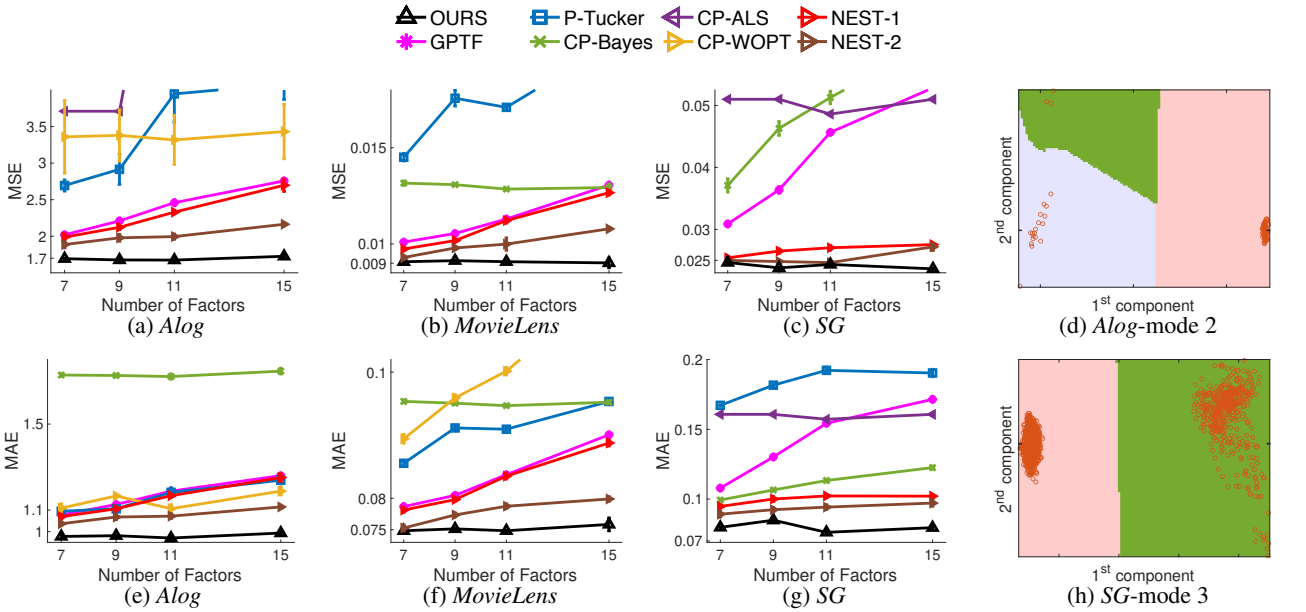
Methods and Settings. We compared with the following state-of-the-art tensor factorization methods. (1) CP-Bayes (Zhao et al., 2015; Du et al., 2018), a Bayesian CP factorization model with Gaussian likelihood. (2) CP-ALS (Bader et al., 2015), CP factorization using alternating least squares to update the latent factors. (3) CP-WOPT (Acar et al., 2011), a fast CP factorization method using conjugate gradient descent to optimize the latent factors. (4) P-Tucker (Oh et al., 2018), a highly efficient Tucker factorization algorithm that updates the factor matrices row-wisely in parallel. (5) GPTF (Zhe et al., 2016b; Pan et al., 2020b), nonparametric tensor factorization based on GPs. It is the same as our modeling the entry value as a latent function of the latent factors (see (10)), except that it samples the latent factors from a standard Gaussian prior. The same sparse GP approximation, *i.e.*, random Fourier features, was employed for scalable model estimation. (6) NEST-1 and (7) NEST-2, which are the two models proposed by Tillinghast & Zhe (2021). NEST-1 uses a single DP in each mode to sample the sparse tensors, and can only estimate one sociability factor for each node; the remaining factors are location factors (from DP locations). NEST-2 uses multiple GEM distributions to draw DP weights for sparse tensor modeling, hence all the factors are sociability factors and there are no location factors. We implemented our method with PyTorch (Paszke et al., 2019). CP-Bayes, GPTF, NEST-1 and NEST-2 were implemented with TensorFlow (Abadi et al., 2016). Since all these methods used stochastic mini-batch optimization, we chose the learning rate from $\{10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}\}$, and set the mini-batch size to 200 for *Alog* and *MovieLens*, and 512 for *SG*. We ran 700 epochs on all the three datasets, which is enough for convergence. We used the original MATLAB implementation of CP-ALS and CP-WOPT (`http://www.tensortoolbox.org/`), and C++ implementation of P-Tucker (`https://github.com/sejoonoh/P-Tucker`), and their default settings. Note that CP-ALS needs to fill nonexistent entries with zero-values to manipulate the whole tensor.

Missing Entry Value Completion. We first tested the accuracy of predicting the missing entry values. To this end, we randomly split the existent entries in each dataset into 80% for training and 20% for test. We then ran each method, and evaluated the Mean Square Error (MSE) and Mean Absolute Error (MAE). We varied the number of factors R from $\{7, 9, 11, 15\}$, and for each setting, we ran the experiment for five times. Since our approach can flexibly estimate the two types of factors, *i.e.*, sociability and location factors, to identify an appropriate trade-off, we ran an extra validation in the training set to select R_1 and R_2 ($R_1 + R_2 = R$) (In Fig. 5 of Appendix, we show how the performance of our method varies along with all possible combinations of



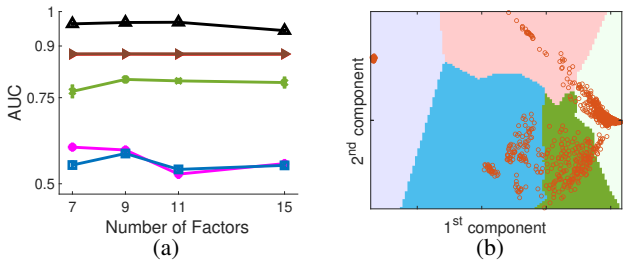
(a) Sparsity: Ours (b) Sparsity: Dense models (c) Our model (d) CP-Bayes (e) GPTF

Figure 1: The ratio between the sampled entries and tensor size (a, b), and exemplar tensors generated by each model (c, d, e), of size $10 \times 10 \times 10$. The legend in (a) indicates the number of sociability factors.



(a) *Alog* (b) *MovieLens* (c) *SG* (d) *Alog*-mode 2
(e) *Alog* (f) *MovieLens* (g) *SG* (h) *SG*-mode 3

Figure 2: Prediction accuracy of entry values (a-c, e-g) and the structures of the estimated factors by our method (d, h). The results of some approaches were much worse than the others and were not included, *e.g.*, the broken curves of CP-WOPT and CP-ALS in (a).



(a) (b)

Figure 3: Entry index prediction accuracy and structure of the estimated factors (mode 2) on *MovieLens*. Note that NEST-1 and NEST-2 overlap in (a).

R_1 and R_2). While NEST also estimates the two types of factors, their number choice is strictly fixed: for NEST-1, $R_1 = 1, R_2 = R - 1$, and NEST-2, $R_1 = R, R_2 = 0$. We report the average MSE, average MAE and their standard deviation in Fig. 2 a-c and e-g. Our method always obtains the best performance. In general, our method and NEST outperform the dense models by a large margin, which confirm the power of the sparse tensor modeling. Our method further improves upon NEST in nearly all the cases ($p < 0.05$), which demonstrates the advantage of our method that can estimate an arbitrary number of sociability and location factors, and allow the selection of their trade-off.

Missing Entry (Link) Prediction. Next, we examined our model in predicting missing entry indices. To this end, from each dataset, we randomly sampled 80% of existent entries, and used their indices for training. Then we used the remaining 20% existent entries plus ten-times nonexistent entries (randomly generated) for test. We did not incorporate all the nonexistent entries for test in order to prevent them from

Alog	$R = 7$	$R = 9$	$R = 11$	$R = 15$
Ours	$\mathbf{0.9939 \pm 0.0004}$	$\mathbf{0.9938 \pm 0.0003}$	$\mathbf{0.9938 \pm 0.0003}$	$\mathbf{0.9913 \pm 0.0004}$
GPTF	0.5862 ± 0.01296	0.6391 ± 0.0056	0.7544 ± 0.0223	0.7594 ± 0.0433
P-Tucker	0.5879 ± 0.0138	0.6606 ± 0.0336	0.6577 ± 0.0215	0.6577 ± 0.0481
CP-Bayes	0.9618 ± 0.0010	0.9634 ± 0.0017	0.9613 ± 0.0004	0.9926 ± 0.0005
NEST-1	0.9908 ± 0.0005	0.9848 ± 0.0028	0.9534 ± 0.0193	0.9648 ± 0.0157
NEST-2	0.9915 ± 0.0003	0.9915 ± 0.0003	0.9904 ± 0.0003	0.9905 ± 0.0006
SG				
Ours	$\mathbf{0.9903 \pm 0.0004}$	$\mathbf{0.9923 \pm 0.0002}$	$\mathbf{0.9786 \pm 0.0004}$	$\mathbf{0.9910 \pm 0.0007}$
GPTF	0.4779 ± 0.0176	0.4755 ± 0.0171	0.4656 ± 0.0131	0.5178 ± 0.0288
P-Tucker	0.5062 ± 0.0149	0.5128 ± 0.0173	0.5271 ± 0.0215	0.5743 ± 0.0396
CP-Bayes	0.5443 ± 0.0021	0.5708 ± 0.0023	0.5715 ± 0.0111	0.5834 ± 0.01286
NEST-1	0.9763 ± 0.0006	0.9763 ± 0.0006	0.9767 ± 0.0004	0.9767 ± 0.0010
NEST-2	0.9765 ± 0.0004	0.9767 ± 0.0004	0.9767 ± 0.0003	0.9767 ± 0.0009

Table 1: Prediction accuracy (AUC) on entry indices (links). The results were averaged over five runs.

dominating the result — they are too many. We compared with CP-Bayes, GPTF, P-Tucker, NEST-1 and NEST-2. For CP-Bayes and GPTF, we used a Bernoulli distribution to sample the presence (or existence) of an entry, for P-Tucker we used a zero-thresholding function. We computed the area under ROC curve (AUC) of predictions on the test entries. We repeated the experiment for five times and computed the average AUC and its standard deviation. We show the result on *MovieLens* in Fig. 3 a, and the other in Table 1. We can see that our method shows better prediction accuracy than all the competing models. The improvement over the dense models is particularly large. While NEST is much closer, our method still significantly outperforms NEST in all the cases ($p < 0.05$), especially a large margin is shown on *MovieLens*.

6.3 Pattern Discovery

Finally, we investigated if our method can discover hidden patterns within the tensor nodes, as compared with the popular dense factorization models. To this end, we set $R = 11$ where $R_1 = 5$ and $R_2 = 6$, and ran our method on all the three datasets. We then used Principal Component Analysis (PCA) to project the learned factors in each mode onto a plain. The positions of the points represent the first and second principal components. To find the patterns, we ran the k-means algorithm and filled the cluster regions with different colors. We used the elbow method (Ketchen & Shook, 1996) to select the cluster number. In Fig. 2d, h and 3 b, we show the results of the second mode on *Alog* and *MovieLens* and the third mode on *SG*. Each point corresponds to a tensor node. These results reflect clear, interesting clustering structures. By contrast, the factors estimated by CP-Bayes and GPTF, as shown in Fig. 4 d-i of the Appendix, do not exhibit patterns and their principal components are distributed like a symmetric Gaussian. While NEST-1 and NEST-2 can find some patterns on *MovieLens* and *SG* as well (see Fig. 4k, l, n and o in Appendix), they are less structured than ours. Furthermore, on *Alog*, both NEST-1 and NEST-2 did not discover any patterns (see Fig. 4 j and m). Together this has demonstrated that our model can potentially discover more interesting and informative patterns and further enhance the interpretability.

7 Conclusion

We have presented a novel nonparametric tensor factorization method based on hierarchical Gamma processes. Our model not only can sample sparse tensors to match the nature of data sparsity in practice, but also can estimate an arbitrary number of sociability and location factors so as to discover interesting and important patterns.

Acknowledgments

This work has been supported by NSF IIS-1910983 and NSF CAREER Award IIS-2046295.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283, 2016.

Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. Scalable tensor factorizations for incomplete data. Chemometrics and Intelligent Laboratory Systems, 106 (1):41–56, 2011.

Aldous, D. J. Representations for partially exchangeable arrays of random variables. Journal of Multivariate Analysis, 11(4):581–598, 1981.

Ayed, F. and Caron, F. Nonnegative bayesian nonpara-

metric factor models with completely random measures. Statistics and Computing, 31(5):1–24, 2021.

Bader, B. W., Kolda, T. G., et al. Matlab tensor toolbox version 2.6. Available online, February 2015. URL http://www.sandia.gov/~tgkolda/TensorToolbox/.

Bryant, M. and Sudderth, E. Truly nonparametric online variational inference for hierarchical dirichlet processes. Advances in Neural Information Processing Systems, 25: 2699–2707, 2012.

Caron, F. and Fox, E. B. Sparse graphs using exchangeable random measures. arXiv preprint arXiv:1401.1137, 2014.

Caron, F. and Fox, E. B. Sparse graphs using exchangeable random measures. Journal of the Royal Statistical Society. Series B, Statistical Methodology, 79(5):1295, 2017.

Choi, J. H. and Vishwanathan, S. Dfacto: Distributed factorization of tensors. In Advances in Neural Information Processing Systems, pp. 1296–1304, 2014.

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

Du, Y., Zheng, Y., Lee, K.-c., and Zhe, S. Probabilistic streaming tensor decomposition. In 2018 IEEE International Conference on Data Mining (ICDM), pp. 99–108. IEEE, 2018.

Engen, S. A note on the geometric series as a species frequency model. Biometrika, 62(3):697–699, 1975.

Fang, S., Kirby, R. M., and Zhe, S. Bayesian streaming sparse Tucker decomposition. In Uncertainty in Artificial Intelligence, pp. 558–567. PMLR, 2021a.

Fang, S., Wang, Z., Pan, Z., Liu, J., and Zhe, S. Streaming Bayesian deep tensor factorization. In International Conference on Machine Learning, pp. 3133–3142. PMLR, 2021b.

Ferguson, T. S. A bayesian analysis of some nonparametric problems. The annals of statistics, pp. 209–230, 1973.

Griffiths, R. C. Lines of descent in the diffusion approximation of neutral wright-fisher models. Theoretical population biology, 17(1):37–50, 1980.

Harshman, R. A. Foundations of the PARAFAC procedure: Model and conditions for an"explanatory"multi-mode factor analysis. UCLA Working Papers in Phonetics, 16: 1–84, 1970.

Hong, L. and Davison, B. D. Empirical study of topic modeling in twitter. In Proceedings of the first workshop on social media analytics, pp. 80–88, 2010.

Hoover, D. N. Relations on probability spaces and arrays of random variables. Preprint, Institute for Advanced Study, Princeton, NJ, 2, 1979.

Hougaard, P. Survival models for heterogeneous populations derived from stable distributions. Biometrika, 73(2):387–396, 1986.

Hu, C., Rai, P., and Carin, L. Zero-truncated poisson tensor factorization for massive binary tensors. In UAI, 2015.

Ishwaran, H. and James, L. F. Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association, 96(453):161–173, 2001.

Kang, U., Papalexakis, E., Harpale, A., and Faloutsos, C. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 316–324. ACM, 2012.

Ketchen, D. J. and Shook, C. L. The application of cluster analysis in strategic management research: an analysis and critique. Strategic management journal, 17(6):441–458, 1996.

Kingman, J. Completely random measures. Pacific Journal of Mathematics, 21(1):59–78, 1967.

Kingman, J. Poisson Processes, volume 3. Clarendon Press, 1992.

Kolda, T. G. Multilinear operators for higher-order decompositions, volume 2. United States. Department of Energy, 2006.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. SIAM Review, 51(3):455–500, September 2009.

Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. Sparse spectrum gaussian process regression. The Journal of Machine Learning Research, 11:1865–1881, 2010.

Li, X., Cong, G., Li, X.-L., Pham, T.-A. N., and Krishnaswamy, S. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp. 433–442, 2015.

Liang, P., Petrov, S., Jordan, M. I., and Klein, D. The infinite pcfg using hierarchical dirichlet processes. In Proceedings of the 2007 joint

conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 688–697, 2007.

Lijoi, A., Prünster, I., et al. Models beyond the Dirichlet process. Bayesian nonparametrics, 28(80):342, 2010.

Liu, B., He, L., Li, Y., Zhe, S., and Xu, Z. Neuralcp: Bayesian multiway data analysis with neural tensor decomposition. Cognitive Computation, 10(6):1051–1061, 2018.

Liu, H., Li, Y., Tsang, M., and Liu, Y. CoSTCo: A Neural Tensor Completion Model for Sparse Tensors, pp. 324–334. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450362016. URL https://doi.org/10.1145/3292500.3330881.

Lloyd, J. R., Orbanz, P., Ghahramani, Z., and Roy, D. M. Random function priors for exchangeable arrays with applications to graphs and relational data. In Advances in Neural Information Processing Systems 24, pp. 1007–1015, 2012.

McCloskey, J. W. A model for the distribution of individuals by species in an environment. Michigan State University. Department of Statistics, 1965.

McFarland, D. A., Ramage, D., Chuang, J., Heer, J., Manning, C. D., and Jurafsky, D. Differentiating language usage through topic models. Poetics, 41(6):607–625, 2013.

Oh, S., Park, N., Lee, S., and Kang, U. Scalable Tucker factorization for sparse tensors-algorithms and discoveries. In 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1120–1131. IEEE, 2018.

Pan, Z., Wang, Z., and Zhe, S. Scalable nonparametric factorization for high-order interaction events. In International Conference on Artificial Intelligence and Statistics, pp. 4325–4335. PMLR, 2020a.

Pan, Z., Wang, Z., and Zhe, S. Streaming nonlinear Bayesian tensor decomposition. In Conference on Uncertainty in Artificial Intelligence, pp. 490–499. PMLR, 2020b.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703, 2019.

Rahimi, A., Recht, B., et al. Random features for large-scale kernel machines. In NIPS, volume 3, pp. 5. Citeseer, 2007.

Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., and Carin, L. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In Proceedings of the 31th International Conference on Machine Learning (ICML), 2014.

Rai, P., Hu, C., Harding, M., and Carin, L. Scalable probabilistic tensor factorization for binary and count data. In IJCAI, 2015.

Rasmussen, C. E. and Williams, C. K. I. Gaussian Processes for Machine Learning. MIT Press, 2006.

Rudin, W. Fourier analysis on groups, volume 121967. Wiley Online Library, 1962.

Schein, A., Paisley, J., Blei, D. M., and Wallach, H. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1045–1054. ACM, 2015.

Schein, A., Zhou, M., Blei, D., and Wallach, H. Bayesian poisson tucker decomposition for learning the structure of international relations. In International Conference on Machine Learning, pp. 2810–2819. PMLR, 2016a.

Schein, A., Zhou, M., Blei, D. M., and Wallach, H. Bayesian poisson tucker decomposition for learning the structure of international relations. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pp. 2810–2819. JMLR.org, 2016b. URL http://dl.acm.org/citation.cfm?id=3045390.3045686.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical dirichlet processes. Journal of the american statistical association, 101(476):1566–1581, 2006.

Tillinghast, C. and Zhe, S. Nonparametric decomposition of sparse tensors. In International Conference on Machine Learning, pp. 10301–10311. PMLR, 2021.

Tillinghast, C., Fang, S., Zhang, K., and Zhe, S. Probabilistic neural-kernel tensor decomposition. In 2020 IEEE International Conference on Data Mining (ICDM), pp. 531–540. IEEE, 2020.

Tucker, L. Some mathematical notes on three-mode factor analysis. Psychometrika, 31:279–311, 1966.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. Now Publishers Inc, 2008.

Wang, C., Paisley, J., and Blei, D. Online variational inference for the hierarchical dirichlet process. In Proceedings

of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 752–760. JMLR Workshop and Conference Proceedings, 2011.

Wang, Z., Chu, X., and Zhe, S. Self-modulating nonparametric event-tensor factorization. In International Conference on Machine Learning, pp. 9857–9867. PMLR, 2020.

Williamson, S. A. Nonparametric network models for link prediction. The Journal of Machine Learning Research, 17(1):7102–7121, 2016.

Xu, Z., Yan, F., and Qi, Y. A. Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. In ICML, 2012.

Yang, Y. and Dunson, D. Bayesian conditional tensor factorizations for high-dimensional classification. Journal of the Royal Statistical Society B, revision submitted, 2013.

Zhao, Q., Zhang, L., and Cichocki, A. Bayesian cp factorization of incomplete tensors with automatic rank determination. IEEE transactions on pattern analysis and machine intelligence, 37(9):1751–1763, 2015.

Zhe, S. and Du, Y. Stochastic nonparametric event-tensor decomposition. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 6857–6867, 2018.

Zhe, S., Xu, Z., Chu, X., Qi, Y., and Park, Y. Scalable nonparametric multiway data analysis. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pp. 1125–1134, 2015.

Zhe, S., Qi, Y., Park, Y., Xu, Z., Molloy, I., and Chari, S. Dintucker: Scaling up gaussian process models on large multidimensional arrays. In Thirtieth AAAI conference on artificial intelligence, 2016a.

Zhe, S., Zhang, K., Wang, P., Lee, K.-c., Xu, Z., Qi, Y., and Ghahramani, Z. Distributed flexible nonlinear tensor factorization. In Advances in Neural Information Processing Systems, pp. 928–936, 2016b.

Appendix

A Tensor Sampling Details

In general, to simulate a Poisson random measure (PRM) with the rate measure $\mu(\cdot)$ on $(\Omega, \mathcal{B})$ where Ω is the universal set and $\mathcal{B}$ is a σ-algebra that includes Ω, we conduct two steps.

- Sample the number of points $N \sim \text{Poisson}\big(\mu(\Omega)\big)$.
- Sample N points (locations) i.i.d with the normalized rate measure $\frac{\mu}{\mu(\Omega)}$ (which is a probability measure).

We use this standard procedure to sample sparse tensors with our model. Given a particular α, we first sample the total mass

$$A([0,\alpha]^{R_1}) = \frac{1}{R_2}\sum_{r=1}^{R_2} W^{\alpha}_{1,r}([0,\alpha]^{R_1}) \times \ldots \times W^{\alpha}_{K,r}([0,\alpha]^{R_1}).$$

To sample the total mass, we need to sample each $W^{\alpha}_{k,r}([0,\alpha]^{R_1})$ where $1 \leq k \leq K$ and $1 \leq r \leq R_2$, which according to the definition of ΓPs, follows $\text{Gamma}(1, L^{\alpha}_{k}([0,\alpha]^{R_1}))$. Recursively, $L^{\alpha}_{k}([0,\alpha]^{R_1})$ is sampled from $\text{Gamma}(1, \lambda_{\alpha}([0,\alpha]^{R_1}))$. We then sample the number of present entries (*i.e.*, points) N from a Poisson distribution with the total mass as the mean. Next, we use the normalized HΓPs, namely, HDPs defined in Sec 3.2 of the main paper to sample each entry. To obtain the probability measure over all possible entries, namely, the normalized rate measure in (8) of the main paper, we first sample the weights β^k_j in the first level of DPs (see (6) of the main paper). We use the stick-breaking construction. Usually, when $j > 2000$, the weights are below the machine precision and automatically truncated to zero, and we do not need to store an infinite number of weights. Following (Teh et al., 2006), the stick-breaking construction of the weights in a second-level DP H^k_r is

$$\nu^k_{rj} \sim \text{Beta}\left(\gamma^k_r \beta^k_j, \gamma^k_r \left(1 - \sum_{l=1}^{j} \beta^k_j\right)\right),\ \omega^k_{rj} = \nu^k_{rj} \prod_{t=1}^{j-1}(1-\nu^k_{rt})\ (1 \leq j \leq \infty).$$

Accordingly, ν^k_{rj} will be truncated to zero at which β^k_j is zero, and so is the weight ω^k_{rj}. Hence, we obtain a finite set of nonzero weights, with which we calculate the probability measure to sample the present entries.

B Combination of R_1 and R_2

We investigated the performance of our method with different combinations of R_1 and R_2 (the number of location and sociability factors) on *Alog* and *SG* datasets. To this end, we fixed the total number of factors to 11 and varied R_2, *i.e.*, the number of sociability factors. We used the same data splits as in Sec. 6.2 of the main paper. The average MSE and MAE for different combinations of R_1 and R_2 are reported in Fig. 5. We can see that the best setting on *Alog* is $R_2 = 6, R_1 = 5$, and on *SC* is $R_2 = 2, R_1 = 9$ (MSE) and $R_2 = 1, R_1 = 10$ (MAE). While the results of all the settings are better than all the competing dense tensor models, different combinations of R_1 and R_2 result in quite different prediction accuracies. The best trade-off is application specific. The results also demonstrate the advantage of our method over NEST models (Tillinghast & Zhe, 2021), which restrict the types of factors to be estimated and do not have the flexibility to choose different R_1 and R_2.

C Running Time

The speed of our method is faster than NEST, comparable to GPTF, and slower than the other competing methods. For example, for $R = 5$, on *MovieLens* and *SG* datasets, the average running time (in seconds) of each method is {Ours: 825.1, GPTF: 706.5, NEST-1: 1218.9, NEST-2: 1193.4, CP-Bayes: 323.2, P-Tucker: 7.9, CP-WOPT: 6.1, CP-ALS: 0.8} and {Ours: 1213.5, GPTF: 1096.1, NEST-1: 2112.2, NEST-2: 1910.5, CP-Bayes: 566.4, P-Tucker: 10.8, CP-WOPT: 10.6, CP-ALS: 0.7}. This is reasonable, because the GP based methods, although using sparse approximations to avoid computing the full covariance (kernel) matrix, still need to estimate the frequencies and perform extra nonlinear feature transformations (*i.e.*, random Fourier features). Hence, they are much more complex. In addition, our method introduces nonparametric priors over sparse tensor entries, and the estimation involves discrete, hierarchical probability measures. Therefore, it requires more computation.

D Proof of Lemma 3.1 and 3.2

D.1 Proof of Lemma 3.1

To ease the notation, we consider the case where $R_1 = R_2 = 1$. It is straightforward to extend the proof to arbitrary R_1 and R_2. Given a particular $\alpha < \infty$, denote the total number of points in T (see (5) in the main paper) by D^α. We observe that by the properties of the Poisson random measure, $D^\alpha \sim \text{Poisson}(W_1^\alpha([0,\alpha]) \times \cdots \times W_K^\alpha([0,\alpha]))$. Since these points might overlap, the number of the distinct points (*i.e.*, entries) $N^\alpha \leq D^\alpha$. According to the definition of Gamma process, $W_k^\alpha([0,\alpha])|L_k^\alpha$ is a Gamma distributed random variable with mean given by $L_k^\alpha([0,\alpha])$, and $L_k^\alpha([0,\alpha])$ is another Gamma random variable with mean α. Therefore, $L_k^\alpha([0,\alpha]) < \infty$ almost surely, $W_k^\alpha([0,\alpha])|L_k^\alpha < \infty$ almost surely, and then $W_k^\alpha([0,\alpha]) < \infty$ almost surely (because $p(W_k^\alpha([0,\alpha]) < \infty) = \int p\left(W_k^\alpha([0,\alpha]) < \infty | L_k^\alpha\right) \mathrm{d}p(L_k^\alpha) = 1$). Hence, their product $b^\alpha = W_1^\alpha([0,\alpha]) \times \cdots \times W_K^\alpha([0,\alpha])$ is finite almost surely. Since $D^\alpha \sim \text{Poisson}(b^\alpha)$, we have $D^\alpha < \infty$ almost surely and $N^\alpha \leq D^\alpha$ implies $N^\alpha < \infty$ almost surely.

Now we consider $W_k^\alpha((j-1,j])|L_k^\alpha$ where $j \in \mathbb{N}^+$. This is a Gamma distributed random variable with mean given by $L_k^\alpha((j-1,j])$. However $L_k^\alpha((j-1,j])$ is a mean-one Gamma distributed random variable. Thus if we integrate out L_k^α for $1 \leq k \leq K$, $\{W_k^\alpha((j-1,j])\}_{k,j}$ are independent identically distributed random variables because L_k^α as a Gamma process itself is independent on disjoint sets. Note that given j, $\{W_k^\alpha((j-1,j])\}$ across k have already been independent. Furthermore, we have $\mathbb{E}[W_k^\alpha((j-1,j])] = 1$, and $\mathbb{E}[W_1^\alpha((j-1,j]) \times \ldots \times W_K^\alpha((j-1,j])] = 1$.

We consider $S_{\lfloor\alpha\rfloor} = \sum_{j=1}^{\lfloor\alpha\rfloor} \mathbb{1}(X_j > 0)$ where $\mathbb{1}(\cdot)$ is the indicator function, and

$$X_j \sim \text{Poisson}\left(W_1^\alpha((j-1,j]) \times \ldots \times W_K^\alpha((j-1,j])\right).$$

Obviously, $S_{\lfloor\alpha\rfloor} \leq N^\alpha$. For convenience, let us define $\zeta_j = W_1^\alpha((j-1,j]) \times \ldots \times W_K^\alpha((j-1,j])$. We have $\mathbb{E}[X_j|\zeta_j] = \zeta_j$ and $\mathbb{E}[X_j] = \mathbb{E}[\zeta_j] = 1$. Since all $\{X_j\}_j$ are i.i.d, the indicators $\{\mathbb{1}(X_j > 0)\}_j$ are i.i.d as well. According to the strong law of large numbers,

$$\lim_{\lfloor\alpha\rfloor \to \infty} \frac{S_{\lfloor\alpha\rfloor}}{\lfloor\alpha\rfloor} = \mathbb{E}[\mathbb{1}(X_j > 0)] = 1 - e^{-1} > 0 \ a.s.$$

Therefore, when $\alpha \to \infty$, $\lfloor\alpha\rfloor \to \infty$ and $S_{\lfloor\alpha\rfloor} \to \infty$ a.s. Since $N^\alpha \geq S_{\lfloor\alpha\rfloor}$, we have $\lim_{\alpha\to\infty} N^\alpha = \infty$ a.s.

D.2 Proof of Lemma 3.2

Our tensor-variate stochastic process when $R_2 = 1$ is summarized in the following.

$$\begin{aligned} L_k^\alpha &\sim \Gamma\text{P}(\lambda_\alpha), \\ W_k^\alpha \mid L_k^\alpha &\sim \Gamma\text{P}(L_k^\alpha), \\ T \mid \{W_k^\alpha\}_{k=1}^\infty &\sim \text{PRM}(W_1^\alpha \times \cdots \times W_K^\alpha). \end{aligned}$$

We will follow the similar strategy in (Tillinghast & Zhe, 2021) to prove the lemma. That is, given α, we define M_k^α the number of active nodes in mode k and N^α the number of sampled entries. Then we have $M_k^\alpha = \#\{\theta_i^k \in [0,\alpha] | T(A_{k,\theta_i^k}^\alpha) > 0\}$, where $A_{k,\theta_i^k}^\alpha = [0,\alpha] \times \cdots \times \{\theta_i^k\} \times \cdots \times [0,\alpha]$. The goal of the first step is to show $\lim_{\alpha\to\infty} \frac{\alpha}{M_k^\alpha} = 0 \ \ a.s.$ for all $k \in \{1,\ldots,K\}$, and the second step is $\limsup_{\alpha\to\infty} N^\alpha/\alpha^K < \infty \ \ a.s.$ Based on these, it is trivial to show

$$\lim_{\alpha\to\infty} \frac{N^\alpha}{\prod_{k=1}^K M_k^\alpha} = 0 \ \ a.s.$$

Step 1. Following the same steps as in (Tillinghast & Zhe, 2021), we can show that

$$\begin{aligned} &\mathbb{E}[M_k^\alpha | \{W_j^\infty\}_{j\neq k}, L_k^\alpha] \\ &= \mathbb{E}\left[\sum_{\theta_i^k \in [0,\alpha]} 1 - \exp\left(-W_k^\infty(\{\theta_i^k\} \times \prod_{j\neq k} W_j^\infty([0,\alpha])\right) \middle| \{W_j^\infty\}_{j\neq k}, L_k^\alpha\right]. \end{aligned}$$

Note that different from (Tillinghast & Zhe, 2021), the expectation here is conditioned on L_k^α, an additional level of ΓPs. We then apply Campbell's Theorem (Kingman, 1992) and obtain

$$\begin{aligned}&\mathbb{E}[M_k^\alpha|\{W_j^\infty([0,\alpha])\}_{j\neq k}, L_k^\alpha]\\&= L_k^\alpha([0,\alpha])\int_0^\infty\left(1-\exp\left(-w\times\prod_{i\neq k}W_i^\infty([0,\alpha])\right)\right)w^{-1}e^{-w}dw.\end{aligned}$$

Since $L_k^\alpha([0,\alpha])$ is a Gamma random variable with mean α, we further take the expectation with respect to the measure induced by L_k^α. Then, we obtain

$$\mathbb{E}[M_k^\alpha|\{W_j^\infty([0,\alpha])\}_{j\neq k}] = \alpha\cdot\int_0^\infty\left(1-\exp\left(-w\times\prod_{i\neq k}W_i^\infty([0,\alpha])\right)\right)w^{-1}e^{-w}dw.$$

Now, we arrive at the same intermediate result as in (Tillinghast & Zhe, 2021). It follows that

$$\lim_{\alpha\to\infty}\frac{\alpha}{M_k^\alpha} = 0 \;\; a.s.$$

The result points out that in each mode, the number of active nodes M_k^α grows faster than α.

Step 2. We consider the distribution of $W_k([j,j+1]) \mid L_k^\alpha$. This will be Gamma distributed with mean $L_k^\alpha([j,j+1])$. But $L_k^\alpha(j,j+1)$ is Gamma distributed with mean 1 for all $j<\alpha-1$. Thus by marginalizing out $L_k^\alpha(j,j+1)$ we see that $W_k([j,j+1])$ is identically distributed for all $j<\alpha-1$. By the independence of the completely random measure (Kingman, 1967) on disjoint sets, it follows immediately by the strong law of large numbers

$$\lim_{j\to\infty}\frac{W_k^\infty([0,j])}{j} = \frac{\sum_{i=1}^j W_k^\infty((i-1,i])}{j} = E[W_k^\infty([0,1])] = 1\, a.s.$$

as $W_k^\infty((i-1,i])$ are i.i.d random variables. This implies

$$\lim_{j\to\infty}\frac{\prod_{k=1}^K W_k^\infty([0,j])}{j^K} = 1\, a.s. \tag{15}$$

Let us denote D^α by the actual number of points sampled in $T([0,\alpha]^K)$. Note $N^\alpha \leq D^\alpha$. Applying Lemma 17 in (Caron & Fox, 2014) implies

$$Pr\left(\lim_{j\to\infty}\frac{D^j}{\prod_{k=1}^K W_k^\infty([0,j])} = 1 \middle| \{W_i^\infty\}_{i=1}^K\right) = 1.$$

Taking the expectation of both sides of the above expression and combining with equation (15) implies

$$\lim_{j\to\infty}\frac{D^j}{j^K} = 1\, a.s.$$

Now, we can use the same bounding technique as in (Tillinghast & Zhe, 2021) to show that

$$\lim_{\alpha\to\infty}\frac{N^\alpha}{\alpha^K} \leq \lim_{\alpha\to\infty}\frac{D^\alpha}{\alpha^K} = 1 < \infty.$$

The result in the second step points out the number of sampled entries N^α grows slower than or as fast as α^K.

Finally, we extend the case when $R_2 > 1$ by simply applying Poisson superposition theorem.

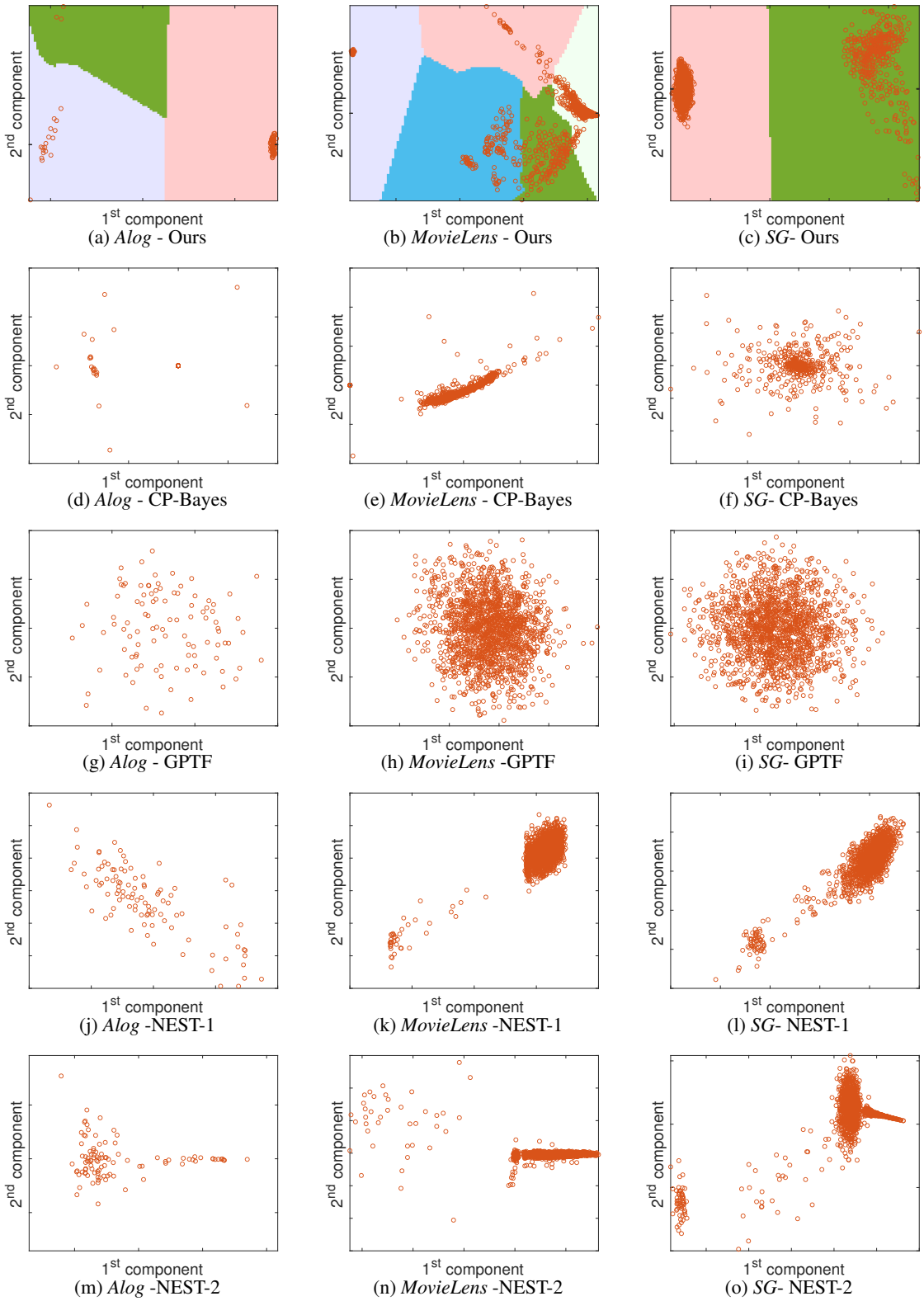
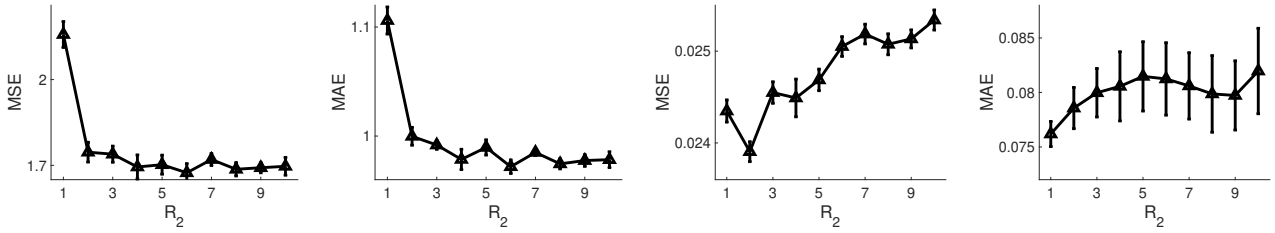


Figure 4: Structures of the estimated latent factors for the second mode of *Alog* and *MovieLens* dataset and third mode of *SG* dataset, by our method, CP-Bayes, GPTF, NEST-1 and NEST-2.



(a) *Alog* (b) *Alog* (c) *SG* (d) *SG*

Figure 5: The performance of our approach with different combinations of R_1 and R_2 (the number of location and sociability factors respectively). The total number of factors R is fixed to 11 ($R = R_1 + R_2$).