The Convolution-Closed Hurdle Motif With an Application to Tensor Decomposition

John Hood [1] **Aaron Schein** [1]

Abstract

This paper introduces a novel inference scheme for a class of hurdle priors that exploits sparsity to scale inference in potentially high-dimensional models with convolution-closed non-negative likelihoods, such as the Poisson. We apply an instance of the class of hurdle priors, the hurdle gamma prior, to a probabilistic non-negative Tucker decomposition and derive an inference scheme that scales with only the nonzero latent parameters in the core tensor. This scheme avoids the typical exponential blowup in computational cost present in Tucker decomposition, efficiently mapping the data to a high-dimensional latent space. We derive and implement a closed-form Gibbs sampler for full posterior inference and fit our model to longitudinal microbiome data. Using our inference motif to quickly fit our model, we reveal interpretable qualitative structure and encouraging classification results.

1. Introduction

Sparse and high-dimensional data are ubiquitous in scientific applications. Practitioners often seek to build complex models for such data, whose parameters and latent variables are themselves high-dimensional. To tailor complex models to sparse data, modelers routinely employ regularization or sparsity-inducing (e.g., "spike-and-slab") priors, which encourage many parameters to be (near) zero, and thus promote parsimony even in high-dimensional models.

Techniques that promote model sparsity are typically motivated either to avoid overfitting (i.e., as regularizers) or to improve model interpretability. Although often effective for both, such techniques often introduce increased computational cost to inference. In probabilistic or Bayesian settings,"spike-and-slab" priors promote sparsity by introducing many additional binary parameters, which must themselves be fit, and which may take combinatorically many values (Bai et al., 2021). As a result, these priors are typically not employed directly, and replaced instead with continuous relaxations, or other approximations.

In some cases, model sparsity can improve rather than exacerbate the computational cost of inference. One such setting is that of trained neural networks, whose weight matrices are high-dimensional but often found to be sparse (Molchanov et al., 2017). Louizos et al. (2017) exploit this sparsity explicitly to improve the computational cost of both training and inference. Inspired by this and other examples, we aim to develop complex probabilistic models that similarly benefit computationally from model sparsity.

This paper builds on classical statistical motifs for modeling sparse data, specifically conditionally conjugate models and hurdle models (Cragg, 1971). We first review the hurdle model as a classical way to model sparsity in probabilistic modeling and refer to a subset of these hurdle models as *hurdle conjugate priors*. We apply hurdle conjugate priors to impose sparsity in the latent parameter space.

We couple our class of priors with a class of closed-convolution likelihoods and present a data augmentation technique that avoids the computational costs that often come with imposing sparsity in high-dimensional, probabilistic settings. Our scheme yields efficient complete conditional updates critical for Bayesian inference such as Gibbs sampling and variational methods.

To demonstrate this scheme, we build a model for sparse count tensors with a closed-convolution likelihood (Poisson) and a hurdle conjugate prior. Building on work that leverages the Poisson likelihood's scalability, we show how the closed-convolution hurdle motif reduces computation by orders of magnitude relative to previous approaches. More specifically, we show how the hurdle conjugate prior allows us to fit a non-negative Tucker decomposition model without suffering the "exponential blowup" in computational cost that Tucker decomposition typically suffers. We implement a fast Gibbs sampler to demonstrating the computational advantage of our method for efficient posterior inference.

[1]Department of Statistics, University of Chicago, Chicago, IL, USA. Correspondence to: John Hood <johnhood@uchicago.edu>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

2. A Family of Hurdle Conjugate Priors

The hurdle model is defined by its sampling scheme. For $k \in [K]$,

$$b_k \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\rho), \tag{1}$$

$$\lambda_k \mid b_k \overset{\text{ind.}}{\sim} \begin{cases} \delta_0 & \text{if } b_k = 0 \\ g_\theta(\lambda_k) & \text{otherwise} \end{cases} \tag{2}$$

where $0 \notin \text{supp}(g_\theta)$ and $\rho \in (0, 1)$ is the hurdle parameter. We consider the setting in which $y_k \mid \lambda_k \sim F_{c_k \lambda_k}$ for some likelihood distribution F_λ and constant c_k, parameterized by λ (i.e. $F_\lambda(y) = \text{Poisson}(y; \lambda)$). $(y_k)_{k=1}^K$ may be observed or latent. We use Y throughout to denote the observed data.

Hurdle models, like spike-and-slab priors (which allow for $0 \in \text{supp}(g_\theta)$), are useful for probabilistically modeling sparsity in high-dimensional spaces, as they provide hard sparsity (exact zeros) in a clear and interpretable manner. However, inference in hurdle and spike-and-slab models involves learning b_k, a prohibitively expensive procedure for large enough K (Bai et al., 2021). We show that for a class of likelihoods, we may streamline this procedure.

2.1. Convolution-Closed Likelihoods

Suppose we have a conditionally-conjugate pair, as in (3-5), where (3) specifies the conditionally conjugate prior, (4) specifies the likelihood, and (5) specifies the conditional posterior (where $(c_k)_{k=1}^K$ are scaling constants). For $k \in [K]$,

$$\lambda_k \overset{\text{iid}}{\sim} g_\theta(\lambda_k), \tag{3}$$

$$y_k \mid \lambda_k \overset{\text{ind.}}{\sim} F_{c_k \cdot \lambda_k}(\cdot), \tag{4}$$

$$(\lambda_k \mid y_k, c_k) \overset{\text{ind.}}{\sim} g_{\theta'}(\lambda_k), \tag{5}$$

for some θ' that depends on c_k and y_k.

We consider the setting where F_λ is convolution-closed, as defined in Definition 2.1.

Definition 2.1. A distribution F_λ is *convolution-closed* in λ if for independently sampled $X_1 \sim F_{\lambda_1}, X_2 \sim F_{\lambda_2}$, the sum $X_1 + X_2 \sim F_{\lambda_1 + \lambda_2}$.

Examples of convolution-closed distributions include the Gaussian, Poisson, binomial, negative binomial, gamma, multivariate Gaussian, inverse Gaussian, generalized Poisson, Tweedie, and multinomial, many of which have conjugate priors. This paper focuses on non-negative data: Table 1 lists commonly used non-negative convolution-closed likelihoods and their accompanying closed-form conjugate priors.

2.2. Data Augmentation

We would like to approximate the exact posterior $\text{P}((\lambda_k, b_k)_{k=1}^K \mid \text{Y})$ using methods that leverage conditional conjugacy, such as Gibbs sampling and mean field variational inference (Casella & George, 1992; Wainwright et al., 2008; Blei et al., 2017). These algorithms iteratively update $(b_k \mid -)$, $(\lambda_k \mid -)$, and $(y_k \mid -)$, conditional on all other parameters fixed. To perform the updates to each $(b_k \mid -)$ efficiently, we propose a data augmentation scheme derived from Proposition 2.2.

Closed-Convolution Likelihood	Conjugate Prior
$y \sim \text{Poisson}(\lambda)$	$\lambda \sim \text{Gamma}(\cdot, \cdot)$
$y \sim \text{Binomial}(n, p)$	$p \sim \text{Beta}(\cdot, \cdot)$
$y \sim \text{NegativeBinomial}(r, p)$	$p \sim \text{Beta}(\cdot, \cdot)$
$\boldsymbol{y} \sim \text{Multinomial}(n, \boldsymbol{p})$	$\boldsymbol{p} \sim \text{Dirichlet}(\cdot)$
$y \sim \text{Exponential}(\lambda)$	$\lambda \sim \text{Gamma}(\cdot, \cdot)$
$y \sim \text{Gamma}(\alpha, \beta)$	$\beta \sim \text{Gamma}(\cdot, \cdot)$
$y \sim \text{Weibull}(\theta, \beta)$	$\theta \sim \text{InvGamma}(\cdot, \cdot)$

Table 1. Some convolution-closed likelihoods and their corresponding conjugate priors for modeling non-negative data.

Proposition 2.2. *Let* $y_k \overset{ind.}{\sim} F_{c_k \cdot \lambda_k}(\cdot)$, $\bar{c} = \max_k c_k$, *and* F_λ *be convolution-closed. Then for* $\tilde{y}_k \overset{ind.}{\sim} F_{(\bar{c} - c_k)\lambda_k}$, $\bar{y}_k \equiv y_k + \tilde{y}_k \sim F_{\bar{c}\lambda_k}$ *independent of* c_k.

Proposition 2.2 follows directly from fact that F_λ is convolution-closed. For each $y_k \sim F_{c_k \lambda_k}$, we sample an auxiliary $\tilde{y}_k$, such that $\bar{y}_k \equiv y_k + \tilde{y}_k \sim F_{\bar{c}\lambda_k}(\cdot)$ which does not depend on c_k. If F_λ is convolution-closed, then for

$$y_k \sim F_{c_k \lambda_k}, \tilde{y}_k \sim F_{(\bar{c} - c_k)\lambda_k} \text{ independently}, \tag{6}$$

$$\bar{y}_k \equiv y_k + \tilde{y}_k \sim F_{\bar{c}\lambda_k}. \tag{7}$$

Under the convolution-closed augmentation scheme, inference can be simplified to performing inference under the generating process given by (8-10).

$$b_k \sim \text{Bernoulli}(p), \tag{8}$$

$$\lambda_k \mid b_k \sim \begin{cases} \delta_0 & \text{if } b_k = 0 \\ g(\lambda_k) & \text{otherwise} \end{cases} \tag{9}$$

$$\bar{y}_k \mid \lambda_k \sim F_{\bar{c}\lambda_k}(\bar{y}_k) \tag{10}$$

In this framework, posterior inference requires iteratively updating $(b_k \mid \lambda_k, \bar{y}_k, \bar{c})$, $(\lambda_k \mid b_k, \bar{y}_k, \bar{c})$, $(y_k \mid b_k, \lambda_k, c_k)$, and the auxiliary $(\tilde{y}_k \mid b_k, c_k, \bar{c}, \lambda_k)$.

Note that F_0 is not defined for some distributions, such as the Poisson. For these distributions, we define F_0 to be a delta spike at 0, $F_0(\cdot) = \delta_0(\cdot)$. That is,

$$y_k \sim F_0 \implies y_k = 0. \tag{11}$$

Updating $\boldsymbol{\lambda_k \mid b_k, \bar{c}, \bar{y}_k}$. If $b_k = 0$, then $\lambda_k = 0$. Conditional on $b_k = 1, \bar{y}_k, \bar{c}$, $(\lambda_k \mid b_k = 1, \bar{y}_k, \bar{c})$ has a closed-form conjugate update, as given by (5).

Auxiliary Updates. Auxiliary sampling scales with $||\boldsymbol{b}||_0$, the number of nonzero b_k. When $b_k = 0$, then $\lambda_k = 0$ and so $\bar{c}\lambda_k = c_k\lambda_k = 0$. We avoid sampling auxiliary $\tilde{y}_k$, as y_k is already distributed as $F_{\bar{c}\lambda_k} = F_0$. Otherwise, sample $\tilde{y}_k \sim \mathrm{F}_{\bar{c}\lambda_k}$ and set $\bar{y}_k = \tilde{y}_k + y_k$.

Updating $b_k \mid \boldsymbol{\lambda_k}, \bar{c}, \bar{y}_k$. It is evident that

$$(b_k \mid \lambda_k \neq 0, \bar{c}, \bar{y}_k) = 1, \tag{12}$$
$$(b_k \mid \lambda_k = 0, \bar{c}, \bar{y}_k) = 0, \tag{13}$$

since $\lambda_k = 0 \implies b_k = 0$ and $\lambda_k \neq 0 \implies b_k = 1$. Updates to b_k are immediate, but a Markov chain based on these updates violates detailed balance (i.e. the Markov chain will get stuck in this state and not explore the full posterior). As such, we collapse out λ_k to derive an update

$$\mathrm{P}(b_k = 1 \mid \bar{c}, \bar{y}_k) \propto \mathrm{P}(b_k = 1)\mathrm{P}(\bar{y}_k \mid b_k = 1, \bar{c}) \tag{14}$$
$$\propto p \cdot \underbrace{\int \underbrace{\mathrm{P}(\bar{y}_k \mid \lambda, \bar{c})}_{F_{\bar{c}\lambda}(\bar{y}_k)} \underbrace{\mathrm{P}(\lambda \mid b_k = 1)}_{g(\lambda)} d\lambda}_{f(\bar{c}, \bar{y}_k)} \tag{15}$$

which is a function of $\bar{y}_k$ and $\bar{c}$ independent of c_k. Then for $k \neq k'$, $\bar{y}_k = \bar{y}_{k'}$, b_k and $b_{k'}$ are i.i.d. in their complete conditional:

$$\mathrm{P}(b_k = 1 \mid \bar{y}_k, \bar{c}) = \mathrm{P}(b_{k'} = 1 \mid \bar{y}_{k'}, \bar{c}). \tag{16}$$

In particular, this implies that for $\bar{y}_k = \bar{y}_{k'} = 0$,

$$\mathrm{P}(b_k = 1 \mid \bar{y}_k = 0, \bar{c}) = \frac{p \cdot f(\bar{c}, 0)}{p \cdot f(\bar{c}, 0) + (1 - p)} \tag{17}$$
$$= \mathrm{P}(b_{k'} = 1 \mid \bar{y}_{k'} = 0, \bar{c}) \tag{18}$$

for $k \neq k'$. That is, for all k such that $\bar{y}_k = 0$, these b_k are i.i.d in their complete conditional. We can sample these Bernoulli random variables by sampling a *single* binomial random variable and thinning it. Letting

$$\tilde{p} = \mathrm{P}(b_k = 1 \mid \bar{y}_k = 0, \bar{c}), \tag{19}$$

we update the $\{b_k : \bar{y}_k = 0\}$ jointly by sampling

$$n \sim \mathrm{Binomial}(\sum_k 1\{\bar{y}_k = 0\}, \tilde{p}) \tag{20}$$

and then sample n of the classes $\{k : \bar{y}_k = 0\}$ uniformly without replacement.

Computational benefit in sparse regimes. The greater the difference between n and K, the greater the reduction in computational cost. We see the most reduction when $\boldsymbol{\lambda}$ is high-dimensional but sparse. Our method only requires sampling the auxiliary $\tilde{y}_k$ for those k such that $\lambda_k > 0$, a step that scales as $O(||\boldsymbol{\lambda}||_0)$ instead of $O(K)$. Similarly, we must only re-sample the λ_k for which $b_k = 1$, a significant reduction in cost when $||\boldsymbol{b}||_0 << K$.

Computing the evidence. Note that collapsing out λ_k relies on the ability to compute the evidence term

$$\mathrm{P}(\bar{y}_k \mid b_k = 1) = \int F_{\bar{c}\lambda_k}(\bar{y}_k) g(\lambda_k) d\lambda_k. \tag{21}$$

Generally, computing the evidence term is intractable. Since $\bar{y}_k > 0$ implies $b_k > 0$, we only need to resample b_k (and thus compute the evidence term) when $\bar{y}_k = 0$. In many cases, the likelihood $F_{\bar{c}\lambda}(0)$ simplifies when $y = 0$, and so $\mathrm{P}(0)$ is tractable and cheap to compute. This yields cheap updates to $(b_k \mid \bar{c}, \bar{y}_k = 0)$.

Extending to continuous data. The closed-convolution hurdle motif relies on $\bar{y}_k = \bar{y}_{k'}$ for $k \neq k'$. When the support of F is finite or discrete, our scheme takes exploits repetition for computational advantage. For discrete $F_\lambda(y)$ with support at 0, for example, our motif presents realizable benefits. However, for continuous data, true zeros are not observed in practice (and more generally, $\bar{y}_k \neq \bar{y}_{k'}$). We provide an approximate method to the continuous likelihood case in Appendix B. Our proposal introduces a tradeoff between computational cost and approximation error and we argue that in many cases the approximation error will be quite small.

3. Sparse Non-Negative Tucker Decomposition

We couple a hurdle gamma prior with the Poisson likelihood to model sparse count tensor data using the hurdle motif. The gamma hurdle (Hilbe, 2014) is an established tool for statistical modeling and sparsity in Tucker is an established idea. Sparse alternatives have been applied to the Tucker decomposition, including spike-and-slab priors on the individual core elements (Fang et al., 2021; Park et al., 2021; Zhang & Ng, 2022). However, these methods do not exploit sparsity for computational benefit, but instead for interpretability and generalization. As such, these methods scale poorly with the size of the core tensor. We attempt to alleviate the typical computational costs, exploiting sparsity for computation and interpretability.

3.1. Tucker Decomposition

The Tucker decomposition of a tensor $\mathrm{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ decomposes Y into a core tensor $\Lambda \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ and M factor matrices $\theta^{(m)} \in \mathbb{R}^{I_m \times J_m}$. Tucker reconstructs Y as a sum of $|\Lambda| = \prod_{m=1}^M J_m$ weighted outer products:

$$\widehat{\mathrm{Y}} \equiv \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} \lambda_{j_1,\ldots,j_M} \theta^{(1)}_{j_1} \circ \cdots \circ \theta^{(M)}_{j_M}, \tag{22}$$

where each element of the core tensor $\lambda_{j_1,\ldots,j_M}$ corresponds to a *weight* assigned to each outer product. We use

$\boldsymbol{i} = (i_1, \ldots, i_M)$ to index the observed tensor Y and $\boldsymbol{j} = (j_1, \ldots, j_M)$ to index the core tensor.

Recent work advocates for modeling complex network data using the non-negative Tucker decomposition (Schein et al., 2016; De Bacco et al., 2017; Aguiar et al., 2022). Tucker yields expressive, rich latent structure. In the case of complex network data, Tucker embeds individuals into clusters, with distinct modalities to capture latent structure (i.e. temporal and spatial) and how these entities interact. However, its usefulness is limited: computation generically scales linearly with the size of the core tensor, which makes fitting a large-core Tucker decomposition difficult in practice.

For high-dimensional, sparse count data, it is natural to adopt a conditionally Poisson likelihood, so that $\mathrm{Y} \sim \text{Poisson}(\widehat{\mathrm{Y}})$ and constrain the parameters of $\theta^{(m)}$ and Λ to be non-negative. We refer to this adaptation of Tucker as the Poisson Tucker decomposition (Schein et al., 2016).

Previous work argues for using the Poisson likelihood to model sparse data for its interpretable appeal near zero and computational tractability (Chi & Kolda, 2012). Evaluating the log-likelihood of a matrix or tensor Y scales computationally with the number of non-zeros in the data structure. Moreover, the closed-convolution characteristic of the Poisson permits a latent *parts-based* representation (Lee & Seung, 1999) of the observed data.

Inference involves allocating observed counts $y_{\boldsymbol{i}}$ across K latent classes through multinomial thinning. Expectation-maximization, Gibbs sampling, and variational inference methods built around this scheme scale as $\mathrm{O}(K \cdot ||\mathrm{Y}||_0)$ (Gopalan et al., 2013; 2014; Schein et al., 2015). In Tucker decomposition, $K = |\Lambda| = \prod_{m=1}^{M} J_m$, the size of the core.

Bayesian Poisson Tucker decomposition. We assume priors on the parameters $\lambda_{\boldsymbol{j}}$, $\theta^{(m)}_{i_m j_m}$ and infer them through posterior estimation. For instance, $\lambda_{\boldsymbol{j}} = \mathrm{P}(\lambda_{\boldsymbol{j}} \mid \phi)$, where ϕ parameterizes the prior distribution over $\lambda_{\boldsymbol{j}}$. Under this formulation, computation generally scales with the size of the core tensor, which experiences an exponential blowup in parameters (exponential, since size of the core tensor is exponential in M). Workarounds such as the ALℓ_0CORE tensor decomposition (Hood & Schein, 2024) have been proposed, which places an ℓ_0 constraint on Λ and infers the nonzero locations and values in the core tensor. The authors derive an inference scheme that scales computationally as $\mathrm{O}(||\Lambda||_0 \cdot ||\mathrm{Y}||_0)$ and enforce a strict upper bound on $||\Lambda||_0$. Inspired by the explicit l_0-constraint on the core tensor, this work places a hurdle prior on each element of the core tensor to *implicitly* provide ℓ_0 regularization. We apply the hurdle prior to the factor matrices in addition to the core tensor, incorporating sparsity across all of Tucker's latent components.

The hurdle gamma prior. The conjugate prior to the Poisson distribution is the gamma distribution, defined as

$$\text{Gamma}(\lambda; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \tag{23}$$

for $\lambda, \alpha, \beta > 0$. In Bayesian Poisson matrix and tensor factorization models (including Poisson Tucker) the gamma prior yields easy-to-compute complete conditionals, $\mathrm{P}(\lambda \mid -)$, conducive to efficient posterior estimation via Markov chain Monte Carlo and variational inference methods. Since the gamma distribution places zero density at $\lambda = 0$, gamma priors constrain parameters to be positive and dense. We model the elements of the factor matrices and core tensor with hurdle gamma priors and couple them with the (convolution-closed) Poisson likelihood. We apply the convolution-closed hurdle motif to speed up inference.

We refer the reader to Appendix C for the complete generative model and inference scheme. At a high level, inference involves alternating between allocating counts to latent sub-counts, as described above, and updating parameters conditional on these latent sub-counts. Upon allocating, inference simplifies to computing the complete conditionals under the following generative model:

$$b \sim \text{Bernoulli}(\rho), \tag{24}$$

$$\lambda \mid b \sim \begin{cases} \delta_0 & \text{if } b = 0 \\ \text{Gamma}(\alpha, \beta) & \text{otherwise} \end{cases} \tag{25}$$

$$y \mid \lambda \sim \text{Poisson}(c\lambda), \quad c \in \mathbb{R}_{\geq 0} \tag{26}$$

which yields closed-form complete conditional updates for each of the parameters b and λ.

Updating b. As in (12) and (13), conditioning on λ determines b so we marginalize out λ, which yields the following update rule.

Proposition 3.1. *When $y > 0$, then $b = 1$. Otherwise, $b \mid c, y = 0 \sim$ Bernoulli$(\tilde{p})$, where*

$$\tilde{p} = \frac{\rho\beta^\alpha}{(1-\rho)(\beta + c)^\alpha + \rho\beta^\alpha}. \tag{27}$$

Updating $\boldsymbol{\lambda}$. By the hurdle and conditional conjugacy,

$$(\lambda \mid b, y) \sim \begin{cases} 0 & \text{if } b = 0 \\ \text{Gamma}(\alpha + y, \beta + c) & \text{otherwise.} \end{cases} \tag{28}$$

Convolution-closed Poisson augmentation. We introduce auxiliary Poisson random variables to sample $b_{\boldsymbol{j}}$ jointly. Updates to $\lambda_{\boldsymbol{j}}$ condition on the latent sub-counts $y_{\boldsymbol{j}}$. The update to $b_{\boldsymbol{j}} \mid y_{\boldsymbol{j}}, c_{\boldsymbol{j}}$ yields different Bernoulli success parameters $\tilde{p}_j$ for each $b_{\boldsymbol{j}}$. Naively, each Gibbs update iterates over all $b_{\boldsymbol{j}}$, scaling computationally with the size of the core. We apply the convolution-closed hurdle motif to work

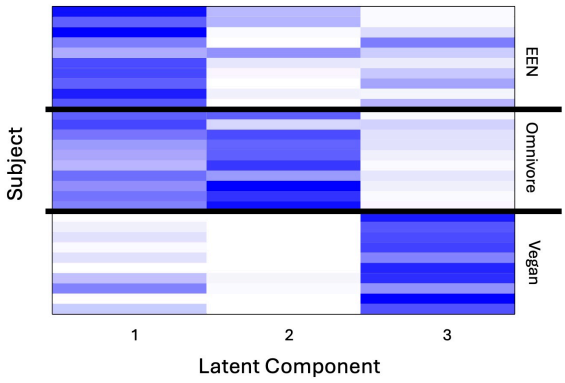

Figure 1. Our method's learned subject modality factor matrix, where subjects (one subject per row) are grouped by phenotype. After training, each latent component so that it's values lie in the unit interval, and the factor elements at zero are plotted white, while those at 1 are dark blue.

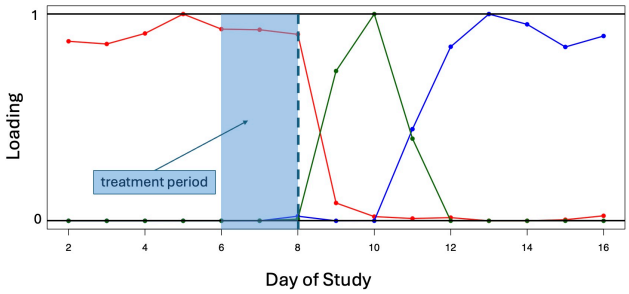

Figure 2. Time series for each latent time component, normalized to lie in the unit interval. In days 6-8, highlighted in blue, subjects receive antibiotic treatment. Factor 1, in blue, is active from day 11 through the end of the study. Factor 2, in red, is active in days 2-9. Factor 3, in green, is most active in days 9-11.

around this problem, as the Poisson is a convolution-closed likelihood and the gamma is its conjugate prior.

Letting $\bar{c} = \max_{\boldsymbol{j}}(c_{\boldsymbol{j}})$, we sample auxiliary counts $\tilde{y}_{\boldsymbol{j}} \sim \text{Poisson}((\bar{c} - c_{\boldsymbol{j}})\lambda_{\boldsymbol{j}})$, such that

$$\bar{y}_{\boldsymbol{j}} \equiv y_{\boldsymbol{j}} + \tilde{y}_{\boldsymbol{j}} \sim \text{Poisson}(\bar{c}\lambda_{\boldsymbol{j}}), \tag{29}$$

since the Poisson is convolution-closed.

Computation. Let Y^{Λ} denote the tensor of size $|\Lambda|$ containing the latent sub-counts $(y_{\boldsymbol{j}})$. Conditional on $\bar{y}_{\boldsymbol{j}}$, $\tilde{p} = \tilde{p}_{\boldsymbol{j}} = \tilde{p}_{\boldsymbol{j}'}$ for all $\boldsymbol{j} \neq \boldsymbol{j}'$, $\bar{y}_{\boldsymbol{j}} = \bar{y}_{\boldsymbol{j}'} = 0$. As such, we sample the $b_{\boldsymbol{j}}$ jointly, sampling $n \sim \text{Binomial}(|\Lambda| - ||\mathrm{Y}^{\Lambda}||_0, \tilde{p})$ and then sampling n new multi-indices in the core at random without replacement. The greater the difference between n and $|\Lambda|$, the greater the reduction in computational cost.

Our method only requires sampling $\tilde{y}_{\boldsymbol{j}}$ for those $\lambda_{\boldsymbol{j}} > 0$, which scales as $\mathrm{O}(||\Lambda||_0)$, the l_0 norm of Λ, instead of $\mathrm{O}(|\Lambda|)$, the size of the core tensor. As in (28), $b = 0$ implies that $\lambda = 0$. Otherwise,

$$(\lambda \mid b = 1, \bar{y}, \bar{c}) \sim \text{Gamma}(\alpha + \bar{y}, \beta + \bar{c}) \tag{30}$$

by conditional conjugacy. Iterating between these updates (on the factor matrices and core tensor) and allocating observed counts to latent sub-counts, which scales $\mathrm{O}(||\Lambda||_0 \cdot ||\mathrm{Y}||_0)$, forms a Gibbs sampler, which generates samples from a Markov chain with stationary distribution equal to the exact posterior distribution, the target of interest.

4. Experimental Results

Data. We demonstrate the closed-convolution hurdle motif's effectiveness by fitting Tucker to data from a microbiome longitudinal study, where Y_{ijt} denotes the gene count of gene i of subject j at time t. We qualitative and quantitatively evaluate our method on the FARMM cohort (Tanes et al., 2021), where the observed tensor $\mathrm{Y} \in \mathbb{N}_0^{343\times30\times16}$. A description of the FARMM dataset may be found at (Tanes et al., 2021). We consider 343 genes, 30 subjects, each with phenotype of {vegan, omnivore, or EEN}, and 16 days. Subjects receive antibiotic treatment on days 6-8 of the study. We omit day 1 observations to be consistent with previous methods (none of the vegan subjects record observations on the first day). The observed tensor is 76% sparse and contains approximately 11% missing values. Our approach treats each missing data point naturally as its own latent variable that is imputed during inference.

Hyperparameter selection. For simplicity, we place independent hurdle gamma priors with hurdle parameter $\rho = 0.9$ and $\text{Gamma}(1, 1)$ priors on each element of the core tensor $\lambda_{\boldsymbol{j}}$. For each of the factor matrix elements, we specify hurdle $\text{Gamma}(1, 10)$ priors. To let sparsity levels differ across factors, we use Beta priors (conjugate to the binomial) $\rho_{j_m}^{(m)} \sim \text{Beta}(1, 1)$, where

$$\text{Beta}(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{31}$$

The sparse factor matrices distinguish subjects by phenotype in the posterior even though our model does not have access to labels while training, as shown in Figure 1.

Qualitative evaluation. Our method identifies latent temporal structure in the time-specific factor matrix and latent structure corresponding to known phenotypes in the subject-specific factor matrix. The core tensor allows for all possible multi-linear latent gene-subject-time interactions. We examine the inferred values in the core tensor to evaluate inferred latent interactions, plot each subject's loading onto each latent factor, and plot the time series for each temporal latent factor. Figures 1-3 show interpretable inferred latent structure. The qualitative and quantitative results are taken

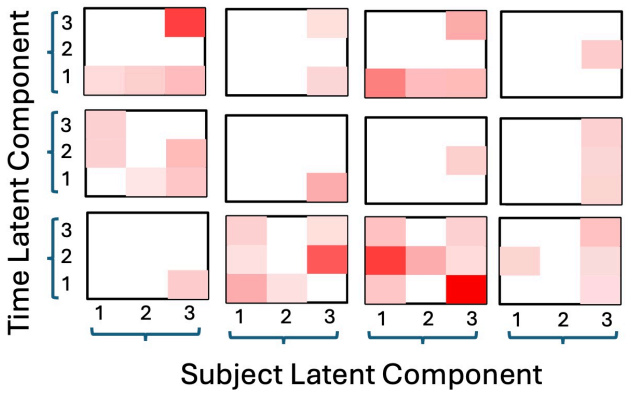

Figure 3. The core tensor. White element indicate core elements that are exactly zero. Each subfigure corresponds to a slice of a different latent component specific to the gene mode of the core tensor, where each slice shows different interactions between time and subject latent factors, organized by gene factor. Of the 15 latent factors, 12 are active (non-zero). We depict heterogenous structure and repetitive structure to demonstrate the complex structure in the core.

from observing one sample from the posterior at random (we simply use the last saved sample). We discard the first 500 samples and keep the remaining 1,000.

Quantitative evaluation. Our model outputs a set of posterior samples, each which contains a sample core tensor and factor matrices. We train a logistic regression classifier on the subject factor matrix to classify subjects by phenotype (EEN, or not EEN) and predict each subject's phenotype using a leave-one-out procedure. We report the area under the precision-recall curve error (1 - AUCPR) for a variety of different core tensor sizes, ranging from $(3, 3, 3)$ to $(25, 3, 3)$, as depicted in Figure 4. We repeat this procedure for 20, 30, 40, and 50% missing data points, holding out samples from different time points at random.

We also compare our method's ability to classify subjects by phenotype to baselines (Martino et al., 2021; Ma & Li, 2023). Each method outputs a 30×3 subject by latent component factor matrix. We normalize each component to have zero mean and unit variance using the `scale` function in `R` and perform spectral clustering (Ng et al., 2001) using the `specc` package from the `kernlab` library in `R`. We compare each method's classification accuracy using the metric of mutual information between the ground-truth empirical distribution over phenotypes and the empirical distribution derived from each method's clusters.

Our method quantifies uncertainty around parameter estimates via Gibbs sampling, which samples parameters from the exact posterior distribution. One drawback of Gibbs sampling is its runtime, as sampling $1,000$ samples

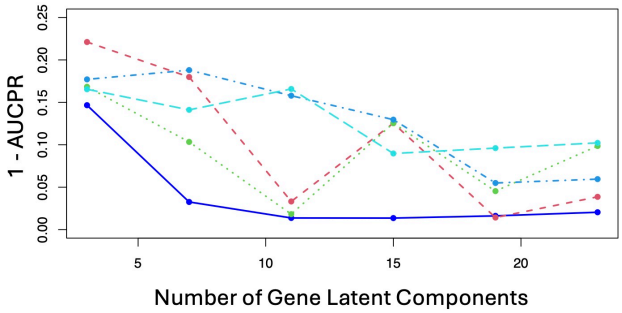

Figure 4. 1 - AUCPR (median over 10 masks) as a function of the gene-specific latent dimension, keeping the subject-specific latent dimension fixed at $K = 3$, for missing data proportions ranging from 11-50%. Missing proportions are 11% (dark blue), 20% (red), 30% (green), 40% (teal), and 50% (light blue).

takes longer than a typical optimization procedure such as EM or VI. However, we note that on the FARMM dataset, for a $(15, 3, 3)$ core tensor with approximately 100 nonzero elements, one Gibbs iteration takes about 0.1 seconds on a laptop. We suspect that as the size of the observed tensor and size of the core increases, the gap between our method and a naive implementation grows, since $|\Lambda| >> ||\Lambda||_0$.

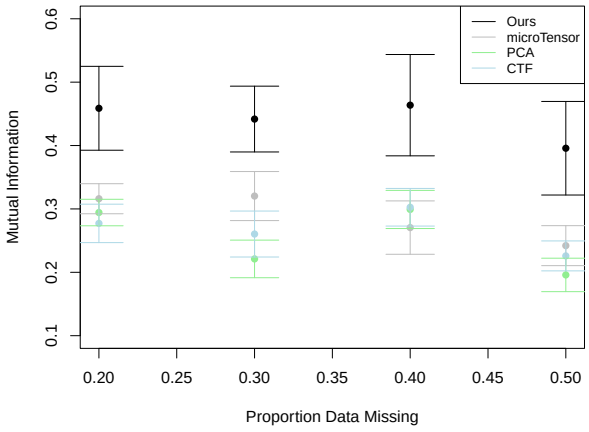

Figure 5. Mutual information between the ground-truth labels and each method's predicted labels after spectral clustering. Each point is the mean mutual information across 10 masks, with error bars representing standard errors.

4.1. Qualitative Results

We fit a model with core tensor $\Lambda \in \mathbb{R}^{15\times3\times3}$. Our method yields a 71.1% sparse core tensor, with 96 out of 135 core elements exactly zero. The estimated 343×15 gene fac-

tor matrix is 45.4% sparse, while the 30×3 subject and 15×3 time factor matrices are 12.2% and 31.1% sparse, respectively.

Factor matrices. Figure 1 shows each subject's loading onto the normalized latent components. Our method demonstrates that the subject-specific latent factors delineate between phenotypes. The EEN phenotype corresponds mostly to factor 1, while the omnivore phenotype corresponds mostly to factor 2 and vegan to factor 3. We consider the identification of 3 distinct and informative latent factors that match the ground-truth phenotypes an advantage of our model.

Figure 2 shows distinct normalized latent factors corresponding to different temporal patterns associated around the study's treatment period, highlighted in blue. Factor 2 (red) captures temporal structure before antibiotic treatment, while factors 1 (blue) and 3 (green) capture relatively chronic and acute responses to treatment, respectively.

Core slices. The Tucker decomposition allows for all possible multi-linear interactions between latent gene, subject, and time factors. We identify heterogeneous responses to antibiotic treatment by latent subject component that corresponds to known phenotype groupings. Darker colors represent higher values in the core, while white indicates exact zeros.

4.2. Quantitative Results

We find an interesting relationship between classification accuracy and core size that motivates the use of a large core tensor in practice. As the number of gene specific latent components grows, our classifier achieves lower error, even though the subject factor matrix is fixed in size. Increasing the core tensor size along one dimension yields more precise latent structure in other modalities, despite fixing the latent dimension specific to that modality.

We leverage the large core size to classify subjects in an unsupervised learning task. After fitting a $(25, 3, 3)$ instance of our model, we spectral cluster to classify subjects, as outlined above, and compare our results to existing methods. Our method's mutual information score (median across 10 random masks) is higher than that of existing methods, as shown in Figure 5, across all proportions of missing data, and we see this as a promising sign.

5. Conclusion

We present the closed-convolution hurdle motif, an inference scheme for a class of models that exploits sparsity for computational savings. We demonstrate the interpretable and computational benefits of our motif by imposing a sparse, high-dimensional latent space on a probabilistic non-negative Tucker decomposition. More generally, we view the idea of exploiting sparsity for computational benefit in high-dimensional probabilistic models as a promising future direction.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2140001.

References

Aguiar, I., Taylor, D., and Ugander, J. A tensor factorization model of multilayer network interdependence. *arXiv preprint arXiv:2206.01804*, 2022.

Bai, R., Ročková, V., and George, E. I. Spike-and-slab meets lasso: A review of the spike-and-slab lasso. *Handbook of Bayesian variable selection*, pp. 81–108, 2021.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Casella, G. and George, E. I. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Chi, E. C. and Kolda, T. G. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

Cragg, J. G. Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica: journal of the Econometric Society*, pp. 829–844, 1971.

De Bacco, C., Power, E. A., Larremore, D. B., and Moore, C. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4):042317, 2017.

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

Gopalan, P., Hofman, J. M., and Blei, D. M. Scalable recommendation with Poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.

Gopalan, P. K., Charlin, L., and Blei, D. Content-based recommendations with Poisson factorization. *Advances in neural information processing systems*, 27, 2014.

Hilbe, J. M. *Modeling Count Data*. Cambridge University Press, 2014.

Hood, J. and Schein, A. J. The ALℓ_0CORE tensor decomposition for sparse count data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4654–4662. PMLR, 2024.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755): 788–791, 1999.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Ma, S. and Li, H. A tensor decomposition model for longitudinal microbiome studies. *The Annals of Applied Statistics*, 17(2):1105–1126, 2023.

Martino, C., Shenhav, L., Marotz, C. A., Armstrong, G., McDonald, D., Vázquez-Baeza, Y., Morton, J. T., Jiang, L., Dominguez-Bello, M. G., Swafford, A. D., et al. Context-aware dimensionality reduction deconvolutes gut microbial community dynamics. *Nature biotechnology*, 39(2): 165–168, 2021.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. In *International conference on machine learning*, pp. 2498–2507. PMLR, 2017.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

Park, M., Jang, J.-G., and Sael, L. VeST: Very sparse Tucker factorization of large-scale tensors. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 172–179. IEEE, 2021.

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, 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, 2016.

Tanes, C., Bittinger, K., Gao, Y., Friedman, E. S., Nessel, L., Paladhi, U. R., Chau, L., Panfen, E., Fischbach, M. A., Braun, J., et al. Role of dietary fiber in the recovery of the human gut microbiome and its metabolome. *Cell Host & Microbe*, 29(3):394–407, 2021.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Zhang, X. and Ng, M. K. Sparse nonnegative Tucker decomposition and completion under noisy observations. *arXiv preprint arXiv:2208.08287*, 2022.

A. Implementation

A.1. Code

The source code for this paper may be found at https://github.com/jhood3/HCPCCL. Our Gibbs sampler is implemented in Julia.

A.2. Implementation Details

All experiments were performed on a 2020 M1 Macbook Air. We ran each Gibbs chain for 500 samples of burn-in, which we discard, and an additional 1000 samples from the target (exact posterior) distribution.

Initialization. We initialize dense instances of the core and factor matrices, such that $b^{(m)}_{i_m j_m} = b_{\boldsymbol{j}} = 1$ for all combinations of i_m, j_m. During the burn-in phase, we threshold the core elements $\lambda_{\boldsymbol{j}}$ such that if $\lambda_{\boldsymbol{j}} < \tau$ for some threshold τ, we set $\lambda_{\boldsymbol{j}} = 0$. We find that this burn-in procedure yields quicker convergence of the chains than initializing values from the prior. For the FARMM study, we set $\tau = 0.003$. We only use this thresholding procedure during the burn-in phase.

B. Continuous Extension

Consider the following generative model:

$$b_k \sim \text{Bern}(p), \tag{32}$$

$$\lambda_k \mid b_k \sim \begin{cases} \delta_0 & \text{if } b_k = 0 \\ g(\cdot) & \text{otherwise} \end{cases} \tag{33}$$

$$y_k \mid b_k, \lambda_k, c_k \sim \begin{cases} \text{Unif}(a_0, b_0) & \text{if } b_k = 0 \\ F_{c_k \lambda_k}(\cdot) & \text{otherwise} \end{cases} \tag{34}$$

$$\tilde{y}_k \mid b_k, \lambda_k, c_k, \bar{c} = \max_k c_k \sim \begin{cases} 0 & \text{if } b_k = 0 \\ F_{(\bar{c}-c_k)\lambda_k}(\tilde{y}_k) & \text{otherwise} \end{cases} \tag{35}$$

$$\bar{y}_k = y_k + \tilde{y}_k \tag{36}$$

where (35) arises from the convolution-closed augmentation scheme. Letting $A = (a_0, b_0)$, we relax the delta spike assumption in the main paper and can derive complete conditional updates. We turn our attention to updates for b_k and propose a method to sample them jointly.

Approximate updates for b_k. If $\bar{y}_k \notin A$, then $b_k = 1$. If $\bar{y}_k \in A$, then

$$P(b_k = 1 \mid \bar{y}_k, \bar{c}) = \frac{P(b_k = 1)P(\bar{y}_k \mid \bar{c}, b_k = 1)}{P(b_k = 1)P(\bar{y}_k \mid \bar{c}, b_k = 1) + P(b_k = 0)\text{Unif}(\bar{y}_k; a_0, b_0)} \tag{37}$$

$$= \frac{p \int F_{\bar{c}\lambda_k}(\bar{y}_k) g(\lambda_k) d\lambda_k}{p \int F_{\bar{c}\lambda_k}(\bar{y}_k) g(\lambda_k) d\lambda_k + (1-p)\frac{1}{b-a}} \tag{38}$$

$$= \frac{p f(\bar{c}, \bar{y}_k)}{p f(\bar{c}, \bar{y}_k) + (1-p)\frac{1}{b-a}} \tag{39}$$

$$= \frac{f(\bar{c}, \bar{y}_k)}{f(\bar{c}, \bar{y}_k) + \frac{1-p}{p(b-a)}} \tag{40}$$

where $f(\bar{c}, \bar{y}_k) = \int F_{\bar{c}\lambda_k}(\bar{y}_k) g(\lambda_k) d\lambda_k$ is the marginal density of $\bar{y}_k$. We note that (40) is $\frac{p(b-a)}{1-p}$- Lipschitz in f on $[0, 1]$, suggesting that if $f(\bar{c}, \bar{y}_k) \approx f(\bar{c}, \bar{y}_{k'})$ for $\bar{y}_k, \bar{y}_{k'} \in A$, then $P(b_k = 1 \mid \bar{y}_k, \bar{c}) \approx P(b_{k'} = 1 \mid \bar{y}_{k'}, \bar{c})$, particularly when $b - a$ is small. Assuming equality, we can sample from the complete conditionals jointly by sampling $n \sim \text{Binomial}(\sum_{k=1}^{K} 1\{\bar{y}_k \in 0\}, \tilde{p})$, and then sampling the n classes uniformly at random without replacement from the set $\{k : \bar{y}_k \in A\}$.

C. Mathematical Details

Area under the precision-recall curve. Consider predictions made by a logistic regression model that lie in the interval $[0, 1]$. For a specific binary threshold $t \in (0, 1)$, let TP_t, TN_t, FP_t, FN_t denote the number of true positives, true negatives, false positives, and false negatives. Then define

$$\text{precision}_t = \frac{\text{TP}_t}{\text{TP}_t + \text{FP}_t}, \qquad \text{recall}_t = \frac{\text{TP}_t}{\text{TP}_t + \text{FN}_t}. \tag{41}$$

Each classification threshold t yields a point on the 2-dimensional precision-recall plane. The area under the precision-recall curve (AUCPR) obtained by integrating over classification thresholds $t \in [0, 1]$ is a value between 0 and 1, and we define the *AUCPR error* as 1 - AUCPR.

Mutual Information. The mutual information $I(X;Y)$ for two discrete random variables X and Y is defined as the KL divergence between the joint distribution of X and Y and the product of their marginals, i.e.

$$\begin{aligned} I(X;Y) &= D_{KL}(\text{P}(X,Y)||\text{P}(X)\text{P}(Y)) && (42)\\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \text{P}(x,y)\log\left(\frac{\text{P}(x,y)}{\text{P}(x)\text{P}(y)}\right) && (43)\\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \text{P}(x,y)\log\left(\frac{\text{P}(x)\text{P}(y \mid x)}{\text{P}(x)\text{P}(y)}\right) && (44)\\ &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \text{P}(x)\text{P}(y \mid x)\log\left(\frac{\text{P}(y \mid x)}{\text{P(y)}}\right) && (45)\\ &= E_X[D_{KL}\left(p(Y \mid X)||p(Y)\right)]. && (46) \end{aligned}$$

$I(X;Y)$ is a measure of dependence between X and Y, where $I(X;Y) = 0$ implies independence between X and Y. For a classification task with n subjects and K classes, we compute the mutual information between the ground-truth labels and predicted labels using the empirical joint and marginal distributions over labels. The higher the mutual information between predicted labels and true labels, the better the quality of the predictions.

C.1. The Poisson Likelihood and Sparsity

Evaluating the log-likelihood of a matrix or tensor Y requires evaluating the log-likelihood at the nonzero values of Y only

$$\begin{aligned} \log(\text{P}(\text{Y} \mid \widehat{\text{Y}}(\Theta))) &= \sum_{\boldsymbol{i}} \log(\text{Poisson}(y_{\boldsymbol{i}}; \hat{y}_{\boldsymbol{i}})) && (47)\\ &\propto_{\widehat{\text{Y}}} \sum_{\boldsymbol{i}} y_{\boldsymbol{i}}\log(\hat{y}_{\boldsymbol{i}}) - \hat{y}_{\boldsymbol{i}}, && (48) \end{aligned}$$

a significant reduction in computational complexity when Y is sparse (i.e. $||\text{Y}||_0 << |\text{Y}|$). The *parts-based* representation of the observed data enables computationally efficient inference. Each observed count $y_{\boldsymbol{i}}$ is the sum of K latent Poisson random variables $\{y_{\boldsymbol{i}k}\}_{k=1}^K$, such that

$$y_{\boldsymbol{i}k} \sim \text{Poisson}(\mu_{\boldsymbol{i}k}), \qquad y_{\boldsymbol{i}} = \sum_k y_{\boldsymbol{i}k}. \tag{49}$$

EM, Gibbs sampling, and variational inference methods built around this scheme scale as $\text{O}(K \cdot ||\text{Y}||_0)$. Inference involves conditioning on the observed counts $y_{\boldsymbol{i}}$ and allocating them across K latent classes through multinomial thinning,

$$(y_{\boldsymbol{i}k})_{k=1}^K \mid y_{\boldsymbol{i}} \sim \text{Multinomial}\left(y_{\boldsymbol{i}}, \left(\frac{\lambda_{\boldsymbol{i}k}}{\sum_{k=1}^K \lambda_{\boldsymbol{i}k}}\right)_{k=1}^K\right) \tag{50}$$

and updating parameters conditional on these latent sub-counts.

C.2. Generative Model

Our specified generative model for the probabilistic Tucker follows from the sequence of (24), (25) and (26).

Factor matrices. For each factor matrix element $\theta^{(m)}_{i_m j_m}$, $m \in [M]$, $i_m \in [I_m]$, $j_m \in [J_m]$,

$$b^{(m)}_{i_m j_m} \overset{\text{ind.}}{\sim} \text{Bernoulli}(\rho^{(m)}_{j_m}), \tag{51}$$

$$\theta^{(m)}_{i_m j_m} \mid b^{(m)}_{i_m j_m} \overset{\text{ind.}}{\sim} \begin{cases} \delta_0 & \text{if } b^{(m)}_{i_m j_m} = 0 \\ \text{Gamma}(1,1) & \text{otherwise.} \end{cases} \tag{52}$$

We also place a Beta prior on each $\rho^{(m)}_{j_m}$, such that for each $j_m \in [J_m], m \in [M]$, $\rho^{(m)}_{j_m} \overset{\text{i.i.d.}}{\sim} \text{Beta}(1,1)$.

Core elements. Similar to the factor elements, we specify the prior for each core element $\lambda_{j_1,\ldots,j_M} = \lambda_{\boldsymbol{j}}$ as the following hurdle gamma prior.

$$b_{\boldsymbol{j}} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(0.9), \tag{53}$$

$$\lambda_{\boldsymbol{j}} \mid b_{\boldsymbol{j}} \overset{\text{ind.}}{\sim} \begin{cases} \delta_0 & \text{if } b_{\boldsymbol{j}} = 0 \\ \text{Gamma}(\alpha, \beta) & \text{otherwise} \end{cases} \tag{54}$$

where $\alpha = 1$ and $\beta = 10$.

Likelihood. We assume a conditionally independent Poisson likelihood for each observed count $y_{\boldsymbol{i}}$, where

$$y_{\boldsymbol{i}} \overset{\text{ind.}}{\sim} \text{Poisson}(\hat{y}_{\boldsymbol{i}}), \tag{55}$$

$$\hat{y}_{\boldsymbol{i}} \equiv \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} \lambda_{\boldsymbol{j}} \prod_{m=1}^{M} \theta^{(m)}_{i_m j_m}, \tag{56}$$

which describes the Poisson Tucker decomposition. Since the sum in (56) decomposes, we use the Poisson's parts-based representation to define unobserved latent subcounts $y_{\boldsymbol{ij}} \overset{\text{ind.}}{\sim} \text{Poisson}(\lambda_{\boldsymbol{j}} \prod_{m=1}^{M} \theta^{(m)}_{i_m j_m})$ for each $y_{\boldsymbol{i}}$, such that $y_{\boldsymbol{i}} = \sum_{\boldsymbol{j}} y_{\boldsymbol{ij}}$. We infer the latent subcounts as latent variables during inference, along with the remaining latent variables.

C.3. Complete Conditionals.

The complete conditionals for the core tensor elements and factor matrix elements follow from the relations given by (24) through (26). Therefore, we begin by justifying them. While most justifications are relatively elementary, the results are crucial for inference, so we include derivations here.

Proof of Proposition 3.1. By Bayes' theorem,

$$P(b = 1 \mid y > 0, c) = \frac{P(y > 0 \mid b = 1, c) P(b = 1 \mid c)}{P(y > 0 \mid b = 1, c) P(b = 1 \mid c) + P(b = 0 \mid c) P(y > 0 \mid b = 0, c)} \tag{57}$$

$$= \frac{P(y > 0 \mid b = 1, c) P(b = 1 \mid c)}{P(y > 0 \mid b = 1, c) P(b = 1 \mid c) + 0} = 1. \tag{58}$$

Othrerwise, when $y = 0$, for constant c,

$$P(b = 1 \mid c, y = 0) = \frac{P(b = 1 \mid c) P(y = 0 \mid b = 1, c)}{P(b = 1 \mid c) P(y = 0 \mid b = 1, c) + P(b = 0 \mid c) P(y = 0 \mid b = 0, c)} \tag{59}$$

$$= \frac{P(b = 1) P(y = 0 \mid b = 1, c)}{P(b = 1) P(y = 0 \mid b = 1, c) + P(b = 0) P(y = 0 \mid b = 0, c)} \tag{60}$$

$$= \frac{\rho P(y = 0 \mid b = 1, c)}{\rho P(y = 0 \mid b = 1, c) + (1 - \rho) \cdot 1} = \frac{\rho P(y = 0 \mid b = 1, c)}{\rho P(y = 0 \mid b = 1, c) + (1 - \rho)}, \tag{61}$$

which depends on $P(y = 0 \mid b = 1, c)$. We can solve for $P(y = 0 \mid b = 1, c)$ analytically, as

$$P(y = 0 \mid b = 1, c) = \int_0^\infty P(y = 0, \lambda \mid b = 1, c) d\lambda = \int_0^\infty P(y = 0 \mid \lambda, b = 1, c) P(\lambda \mid b = 1, c) d\lambda \tag{62}$$

$$= \int \text{Poisson}(0; c\lambda)\text{Gamma}(\lambda; \alpha, \beta) d\lambda \tag{63}$$

$$= \int_0^\infty e^{-c\lambda} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} d\lambda \tag{64}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^\infty e^{-(\beta+c)\lambda} \lambda^{\alpha-1} d\lambda \tag{65}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha)}{(\beta + c)^\alpha} = \frac{\beta^\alpha}{(\beta + c)^\alpha} \tag{66}$$

by u-substitution. Plugging in $\frac{\beta^\alpha}{(\beta+c)^\alpha} = P(y = 0 \mid b = 1, c)$, we have that

$$P(b = 1 \mid y > 0, c) = \frac{\rho \frac{\beta^\alpha}{(\beta+c)^\alpha}}{(1 - \rho) + \rho \frac{\beta^\alpha}{(\beta+c)^\alpha}} = \frac{\rho \frac{\beta^\alpha}{(\beta+c)^\alpha}}{(1 - \rho) \frac{(\beta+c)^\alpha}{(\beta+c)^\alpha} + \rho \frac{\beta^\alpha}{(\beta+c)^\alpha}} \tag{67}$$

$$= \frac{\rho\beta^\alpha}{(1 - \rho)(\beta + c)^\alpha + \rho\beta^\alpha}. \tag{68}$$

Justification of Equation (28). When $b = 0$, then by the definition of the hurdle model, $\lambda = 0$ almost surely.

When $b = 1$,

$$P(\lambda \mid b = 1, y, c) \propto_\lambda P(\lambda \mid b = 1, c) P(y \mid \lambda, b = 1, c) \tag{69}$$

$$\propto_\lambda \text{Gamma}(\lambda; \alpha, \beta)\text{Poisson}(y; c\lambda) \tag{70}$$

$$\propto_\lambda e^{-\beta\lambda} \lambda^{\alpha-1} e^{-c\lambda} \lambda^y \tag{71}$$

$$\propto_\lambda e^{-(\beta+c)\lambda} \lambda^{(\alpha+y)-1}. \tag{72}$$

That is, the full conditional is proportional to the kernel of the Gamma distribution with shape parameter $\alpha + y$ and rate parameter $\beta + c$. As such, $P(\lambda \mid b = 1, y, c) = \text{Gamma}(\lambda; \alpha + y, \beta + c)$.

Equipped with the two justifications above, we derive the the complete conditionals for the factor matrix and core tensor elements that are critical for Gibbs sampling.

Factor matrix elements $\theta^{(m)}_{i_m j_m}$. If $b^{(m)}_{i_m j_m} = 0$, then $\theta^{(m)}_{i_m j_m} = 0$. Otherwise, we apply (28), with

$$P(\theta^{(m)}_{i_m j_m} \mid -) = \text{Gamma}(\theta^{(m)}_{i_m j_m}; 1 + y_{i_m j_m}, 1 + c_{i_m j_m}), \tag{73}$$

where the $y_{i_m j_m}$ and $c_{i_m j_m}$ are defined as:

$$y_{i_m j_m} = \sum_{\boldsymbol{i}', \boldsymbol{j}'} 1\{i'_m = i_m, j'_m = j_m\} \cdot y_{\boldsymbol{i}'\boldsymbol{j}'}, \quad c_{i_m j_m} = \sum_{\boldsymbol{i}', \boldsymbol{j}'} 1\{i'_m = i_m, j'_m = j_m\} \cdot \lambda_{\boldsymbol{j}'} \prod_{m' \neq m} \theta^{(m')}_{i_{m'} j_{m'}}. \tag{74}$$

Auxilliary latent subcounts $\tilde{y}_{\boldsymbol{j}}$. Define $y_{\boldsymbol{j}} = \sum_{\boldsymbol{i}} y_{\boldsymbol{i}\boldsymbol{j}}$. Then marginally, by Poisson additivity, $y_{\boldsymbol{j}} \sim \text{Poisson}(\lambda_{\boldsymbol{j}} \sum_{\boldsymbol{i}} \prod_{m=1}^M \theta_{i_m j_m})$. Define $c_{\boldsymbol{j}} = \sum_{\boldsymbol{i}} \prod_{m=1}^M \theta_{i_m j_m}$, such that $y_{\boldsymbol{j}} \sim \text{Poisson}(\lambda_{\boldsymbol{j}} c_{\boldsymbol{j}})$ and $\bar{c} = \max_{\boldsymbol{j}} c_{\boldsymbol{j}}$. Then

$$P(\tilde{y}_{\boldsymbol{j}} \mid -) = \text{Poisson}(\lambda_{\boldsymbol{j}}(\bar{c} - c_{\boldsymbol{j}})). \tag{75}$$

Core elements $\lambda_{\boldsymbol{j}}$. If $b_{\boldsymbol{j}} = 0$, then $\lambda_{\boldsymbol{j}} = 0$. Otherwise, by conditional conjugacy, we apply (28) again, such that

$$P(\lambda_{\boldsymbol{j}} \mid -) = \text{Gamma}(1 + \bar{y}_{\boldsymbol{j}}, 10 + \bar{c}) \tag{76}$$

where $\bar{c} = \max_{\boldsymbol{j}} \sum_{\boldsymbol{i}} \prod_{m=1}^M \theta^{(m)}_{i_m j_m}$ and $\bar{y}_{\boldsymbol{j}} = y_{\boldsymbol{j}} + \tilde{y}_{\boldsymbol{j}}$.

Latent subcounts $y_{\boldsymbol{ij}}$. Marginally, $y_{\boldsymbol{i}} \sim \text{Poisson}(\sum_{\boldsymbol{j}} \lambda_j \prod_{m=1}^{M} \theta^{(m)}_{i_m j_m})$. As such, by multinomial thinning,

$$P\left((y_{\boldsymbol{ij}})_{\boldsymbol{j}} \mid -\right) = \text{Multinomial}\left(y_{\boldsymbol{i}}, \left(\frac{\lambda_{\boldsymbol{j}} \prod_{m=1}^{M} \theta^{(m)}_{i_m j_m}}{\sum_{\boldsymbol{j}} \lambda_{\boldsymbol{j}} \prod_{m=1}^{M} \theta^{(m)}_{i_m j_m}}\right)_{\boldsymbol{j}}\right). \tag{77}$$

Hurdle parameters $\rho^{(m)}_{j_m}$. Let $n^{(m)}_{j_m} = \sum_{i_m=1}^{I_m} b^{(m)}_{i_m j_m}$. By Beta-binomial conjugacy,

$$P(\rho^{(m)}_{j_m} \mid -) = \text{Beta}(\rho^{(m)}_{j_m}; 1 + n^{(m)}_{j_m}, 1 + I_m - n^{(m)}_{j_m}). \tag{78}$$