

Conditional Hierarchical Bayesian Tucker Decomposition for Genetic Data Analysis

Adam Sandler

*Engineering Sciences and Applied Mathematics
Northwestern University
Evanston, IL*

ADAMSANDLER2021@U.NORTHWESTERN.EDU

Diego Klabjan

*Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL*

D-KLABJAN@NORTHWESTERN.EDU

Yuan Luo

*Preventive Medicine (Health and Biomedical Informatics)
Northwestern University
Chicago, IL*

YUAN.LUO@NORTHWESTERN.EDU

Editor: Editor’s name

Abstract

We develop methods for reducing the dimensionality of large data sets, common in biomedical applications. Learning about patients using genetic data often includes more features than observations, which makes direct supervised learning difficult. One method of reducing the feature space is to use latent Dirichlet allocation to group genetic variants in an unsupervised manner. Latent Dirichlet allocation describes a patient as a mixture of topics corresponding to genetic variants. This can be generalized as a Bayesian tensor decomposition to account for multiple feature variables. Our most significant contributions are with hierarchical topic modeling. We design distinct methods of incorporating hierarchical topic modeling, based on nested Chinese restaurant processes and Pachinko Allocation Machine, into Bayesian tensor decomposition. We apply these models to examine patients with one of four common types of cancer (breast, lung, prostate, and colorectal) and siblings with and without autism spectrum disorder. We linked the genes with their biological pathways and combine this information into a tensor of patients, counts of their genetic variants, and the genes’ membership in pathways. We find that our trained models outperform baseline models, with respect to coherence, by up to 40%.

1. Introduction

A tensor is a multidimensional array of data. Tensors can be decomposed into parts for easier analysis. This decomposition is typically done by grouping data along each mode into topics. Particularly, we examine two variations on the tensor decomposition formula: Bayesian tensor decomposition, i.e., the use of Bayesian inference to compute the components, and hierarchical decomposition, which applies a structure to the topics.

Bayesian non-hierarchical and non-Bayesian hierarchical tensor decompositions have already been studied. For example, [Yang and Dunson \(2016\)](#) developed a conditional Bayesian

Tucker decomposition, the probability of a categorical response depending on p categorical predictors. Grasedyck (2010) developed a hierarchical Tucker decomposition format. Also, Luo et al. (2015) constructed a Tucker decomposition with a hierarchy of modes.

We first define a non-hierarchical conditional Bayesian Tucker decomposition as multiple feature variables depending on a sample variable. Then, we develop hierarchical topic models for multiple modes. We use these topic models in the conditional Bayesian Tucker decomposition to group patients based on their counts of genetic variants and pathways. Then, we examine and compare the groups generated by different models.

Our contributions are as follows.

- We define a new formulation for conditional Bayesian Tucker decomposition.
- We develop methods for incorporating hierarchical topic models into the Bayesian Tucker decomposition.
- We study three known properties of Chinese restaurant processes (CRP) in the context of multimodal topic models.
- We derive a collapsed Gibbs sampler for the conditional Bayesian Tucker decomposition with an arbitrary number of modes.
- We present a novel method of feature reduction for sparse count data.

In Section 2, we review existing literature related to our research. In Section 3, we define our decomposition and topic models. In Section 4, we discuss which properties apply to our models. In Section 5, we discuss algorithms used to compute our decompositions. In Section 6, we evaluate our models trained on the example data sets.

2. Related Work

Two tensor decomposition methods are considered generalizations of singular value decomposition: Tucker decomposition, which decomposes a tensor as a core tensor multiplied by matrices along each mode, and CANDECOMP/PARAFAC (CP) decomposition, which decomposes a tensor as the sum of rank-one tensors (Kolda and Bader, 2009). Luo et al. (2017b,a) suggested using tensor decompositions to identify latent groups in biomedical fields, including genotyping and phenotyping.

Models exist for decomposing probability tensors using a Dirichlet prior. Dunson and Xing (2009); Zhou et al. (2015) proposed using a Bayesian model to decompose a joint probability tensor using CP decomposition. Also, Yang and Dunson (2016) proposed a Bayesian model to decompose a conditional probability tensor according to the Tucker decomposition. They used a finite sized core tensor, which can be adjusted as part of the posterior algorithm. Dunson and Xing (2009); Yang and Dunson (2016); Zhou et al. (2015) imposed Dirichlet priors (or a Dirichlet stick-breaking prior) on the components of the decomposition and proposed a Gibbs sampler for the posterior computation. Dunson and Xing (2009); Zhou et al. (2015) used a CP decomposition, while we use a Tucker decomposition. Yang and Dunson (2016) used a Tucker decomposition with 1 response variable conditional on p predictor variables, while we use p feature variables conditional on

1 sample variable. In addition to this different formulation, we add a hierarchical structure which has not been studied before in the tensor context.

If a conditional Bayesian tensor decomposition consists of only two modes, one sample variable and one feature variable, then both [Yang and Dunson’s \(2016\)](#) and our model are equivalent. In this case, both decompositions simplify to a matrix factorization method, latent Dirichlet allocation (LDA), which has been studied extensively. [Hoffman et al. \(2010\)](#) and [Buntine \(2002\)](#) stated that LDA can be viewed as a probabilistic matrix factorization of the counting matrix of words in each document into a matrix of topic weights and a dictionary of topics. Also, [Schein et al. \(2016\)](#) noted that there is a connection between Poisson matrix factorization and LDA.

Some researchers studied Bayesian Tucker decomposition with other priors. [Schein et al. \(2016\)](#) modeled country-to-country interactions using a four-mode tensor to represent an action performed between two countries in a certain month. They applied a Bayesian Poisson Tucker decomposition to group countries, actions, and time steps. [Xu et al. \(2012\)](#) proposed a model for computing the Tucker decomposition of a tensor using a normal prior and a variational expectation maximization posterior algorithm.

[Chi and Kolda \(2012\)](#) developed a Poisson CP decomposition, called CP Alternating Poisson Regression, and fit their model using a log-likelihood score.

Hierarchical decompositions were studied in a non-Bayesian context. [Hackbusch and Kühn \(2009\)](#) defined the hierarchical Tucker format, which defines the hierarchy according to vector spaces and subspaces. [Grasedyck \(2010\)](#) developed algorithms for computing decompositions in the hierarchical Tucker format based on hierarchical singular value decomposition. [Song et al. \(2013\)](#) defined a recursive decomposition algorithm for estimating a latent tree graphical model of a hierarchical tensor decomposition. Their model depicts the joint probability of a set of observed variables as nodes, dependent on their hidden parents. [Schifanella et al. \(2014\)](#) proposed a method for hierarchical decomposition of tensors by adjusting the resolution or size of the core tensor to provide different resolution decompositions of the same data. Unlike our model, none of these models used Bayesian inference nor expressed the hierarchy of latent topics in each mode. Instead, these models depicted a hierarchy of the vector spaces, a hierarchy of hidden variables, or different resolutions. Also, [Teh et al. \(2006\)](#) developed hierarchical Dirichlet process to cluster grouped data.

In summary, our hierarchical Bayesian Tucker decomposition model is different from previous Bayesian tensor decomposition models because we use a unique formulation and impose a hierarchical structure to the latent topics.

3. Models

In what follows, we use bold font to denote vectors, matrices, and tensors and non-bold to denote scalars. If \mathbf{u} is a vector, we denote u_j as its j^{th} component. [Table 1](#) summarizes the variables used and their corresponding definitions for the autism spectrum disorder (ASD) and cancer examples.

In [Section 3.1](#), we define our conditional non-hierarchical Bayesian Tucker decomposition. In [Section 3.2](#), we discuss hierarchical topic models for a single mode. In [Section 3.3](#), we define the conditional hierarchical Bayesian Tucker (HBT) decomposition. In [Appendix C.1](#), we discuss generalizations of the hierarchical topic models to $p \geq 3$ feature modes.

	Definition	Example
x	sample variable id	patient index
\mathbf{y}	feature variables	genes and pathways
z	hidden topic	gene and pathway group
$\boldsymbol{\pi}$	conditional probability tensor $P(\mathbf{y} x)$	patients' gene and pathway prevalence
$\boldsymbol{\phi}$	topic prevalence for each sample	each patient's topic distribution
$\boldsymbol{\psi}$	prevalence of each feature in each topic	prevalence of genes and pathways in their respective topics
λ_x	total count for sample x	total genetic variants for patient x
c_j	specific value of $\begin{cases} x, & \text{if } j = 0 \\ y_j, & \text{if } j > 0 \end{cases}$	a specific patient, gene, or pathway
d_j	size of tensor in mode j	number of patients, genes, or pathways
p	number of feature modes	2 modes (genes and pathways)
K_j	number of topics for mode j	number of gene or pathway topics
\mathcal{K}	set of possible topics	set of gene and pathway topics
\mathbf{h}	specific topic set	specific gene and pathway topics
\mathbf{T}_x	sample x 's path through hierarchical model	path of patient x through PAM
L	depth of hierarchical model	3-level PAM
ℓ	level in hierarchical model	level of PAM
$\tau^{(\ell,j)}$	number of topics in level ℓ for mode j	number of topics in specific level of PAM
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	priors for $\boldsymbol{\phi}, \boldsymbol{\psi}$	uniform prior with value 1
γ	prior or parameter for \mathbf{T}	uniform prior over topics in next level

Table 1: Notation

3.1. Conditional Bayesian Tucker Decomposition

First, we define our tensor decomposition for a counting tensor without a hierarchical structure. This model details how the Bayesian Tucker decomposition is constructed and performed, i.e., which quantities the core tensor and factor matrices represent. In the context of applications to ASD and cancer, this model decomposes a tensor of the count of each patient's genetic variants and pathways as the mixture of the patient's genetic variant and pathway groups, the mixture of genetic variants in each genetic variant group, and the mixture of pathways in pathway groups. The decomposition provides proportions of genetic variant and pathway groups to which each patient belongs, as well as the proportion of each genetic variant within the genetic variant groups and pathways within the pathway groups.

Given a counting tensor $\mathcal{B} = \{b_{c_0 \dots c_p}\}$,¹ we first normalize it by dividing by $\lambda_{c_0} = \sum_{c_1=1}^{d_1} \dots \sum_{c_p=1}^{d_p} b_{c_0 \dots c_p}$ to obtain the $d_0 \times \dots \times d_p$ conditional probability tensor $\pi_{c_0 \dots c_p} = \frac{b_{c_0 \dots c_p}}{\lambda_{c_0}} = P(y_1 = c_1, \dots, y_p = c_p \mid x = c_0)$, where \mathbf{y} and x are the feature and sample variables. In the genetic applications, $\boldsymbol{\pi} = P(\text{genetic variants, pathways} \mid \text{patient})$, \mathbf{y} represents the genes and pathways, and x is a patient. We define the conditional Bayesian Tucker decomposition as the Tucker decomposition of the conditional probability tensor, $\pi_{c_0 \dots c_p} = \sum_{\mathbf{h} \in \mathcal{K}} \phi_{c_0 \mathbf{h}} \prod_{j=1}^p \psi_{h_j c_j}^{(j)}$, with $\mathcal{K} = \{(h_1, \dots, h_p) \mid h_j \in \{1, \dots, K_j\} \forall j \in \{1, \dots, p\}\}$, latent classes \mathbf{z} , $\phi_{c_0 h_1 \dots h_p} = P(z_1 = h_1, \dots, z_p = h_p \mid x = c_0)$, and $\psi_{h_j c_j}^{(j)} = P(y_j =$

1. Convention dictates that if \mathcal{B} is a tensor, $b_{c_0 \dots c_p}$ are its elements, for $c_i \in \{1, \dots, d_i\}$ and $i \in \{1, \dots, p\}$.

Algorithm 1: Generative Process

<pre> for $x = 1, \dots, d_0$ do Draw core tensor $\tilde{\phi}_x \sim \text{Dir}(\boldsymbol{\alpha}) \in \mathcal{S}_K$ for $\mathbf{k} \in \mathcal{K}$ do $\phi_{x\mathbf{k}} = \tilde{\phi}_{x\text{vec}(\mathbf{k})}$ for $j = 1, \dots, p$ do for $k = 1, \dots, K_j$ do Draw auxiliary matrices $\boldsymbol{\psi}_k^{(j)} \sim \text{Dir}(\boldsymbol{\beta}^{(j)}) \in \mathcal{S}_{d_j}$ </pre>	<pre> for $x = 1, \dots, d_0$ do for $i = 1, \dots, \lambda_x$ do Draw latent topics $\boldsymbol{\varepsilon} \sim \text{Mult}(\{1, \dots, K\}, \tilde{\phi}_x)$ $\mathbf{z}_i^{(x)} = \text{vec}^{-1}(\boldsymbol{\varepsilon})$ for $j = 1, \dots, p$ do Draw features $\mathbf{y}_{ij}^{(x)} \sim \text{Mult}(\{1, \dots, d_j\}, \boldsymbol{\psi}_{\mathbf{z}_i^{(x)}}^{(j)})$ </pre>
---	--

$c_j \mid z_j = h_j$ for all j . In our example, $\phi = P(\text{gene groups, pathway groups} \mid \text{patient})$, $\boldsymbol{\psi}^{(1)} = P(\text{genes} \mid \text{gene groups})$, and $\boldsymbol{\psi}^{(2)} = P(\text{pathways} \mid \text{pathway groups})$. In this model, for each x , a joint topic distribution over topic vectors $\mathbf{h} \in \mathcal{K}$ is first selected, governed by core tensor ϕ . Next, for all modes $j > 0$, the selected topic h_j is a mixture of choices $\{1, \dots, d_j\}$, governed by auxiliary matrix $\boldsymbol{\psi}^{(j)}$. We note that $\sum_{\mathbf{h} \in \mathcal{K}} \phi_{c_0 \mathbf{h}} = 1$ for

all $c_0 \in \{1, \dots, d_0\}$ and $\sum_{c_j=1}^{d_j} \psi_{h_j c_j}^{(j)} = 1$ for all j and h_j .

For ease of notation, we define $K = \prod_{i=1}^p K_i$ and map $\text{vec} : \mathcal{K} \mapsto \{1, \dots, K\}$ as a one-to-one mapping from a tuple of topics to a single topic index. Our model does not depend on the choice of such map.² The generative process is presented in Algorithm 1, where $\mathcal{S}_d = \left\{ \mathbf{v} \in \mathbb{R}_+^d \mid \sum_{i=1}^d v_i = 1 \right\}$ is the d -dimensional probability simplex.

The model probability is given by the product of components: factor matrices given their priors, core tensor given its prior, and individual count probabilities. The probability of the factor matrices $\boldsymbol{\psi}$ given its priors $\boldsymbol{\beta}$ is a nested product over modes j and topics h_j in that mode given the prior for that mode:

$$P(\boldsymbol{\psi} \mid \boldsymbol{\beta}) = \prod_{j=1}^p \prod_{h_j=1}^{K_j} P\left(\boldsymbol{\psi}_{h_j}^{(j)} \mid \boldsymbol{\beta}^{(j)}\right). \quad (1)$$

The probability of the core tensor ϕ given its prior $\boldsymbol{\alpha}$ is a product over all samples x given the prior:

$$P(\phi \mid \boldsymbol{\alpha}) = \prod_{x=1}^{d_0} P(\phi_x \mid \boldsymbol{\alpha}). \quad (2)$$

The individual count probabilities are a nested product over samples x and counts within each sample λ_x of the hidden topics $\mathbf{z}_i^{(x)}$'s probability given the core tensor $\boldsymbol{\psi}_x$ and the

2. An example mapping would be $\text{vec}(\mathbf{k}) = k_1 + (k_2 - 1)K_1 + \dots + (k_p - 1) \prod_{i=1}^{p-1} K_i$. This is a generalization of the column-major order map.

feature variable $\mathbf{y}_i^{(x)}$'s probability given the feature matrices $\boldsymbol{\psi}_{\mathbf{z}_i^{(x)}}$:

$$P(\mathbf{Y}, \mathbf{Z} | \boldsymbol{\phi}, \boldsymbol{\psi}) = \prod_{x=1}^{d_0} \prod_{i=1}^{\lambda_x} P\left(\mathbf{y}_i^{(x)} | \boldsymbol{\psi}_{\mathbf{z}_i^{(x)}}\right) P\left(\mathbf{z}_i^{(x)} | \boldsymbol{\phi}_x\right). \quad (3)$$

The overall model probability is a product of the components in (1)-(3):

$$P(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\boldsymbol{\psi} | \boldsymbol{\beta}) P(\boldsymbol{\phi} | \boldsymbol{\alpha}) P(\mathbf{Y}, \mathbf{Z} | \boldsymbol{\phi}, \boldsymbol{\psi}). \quad (4)$$

3.2. Overview of CRP & PAM

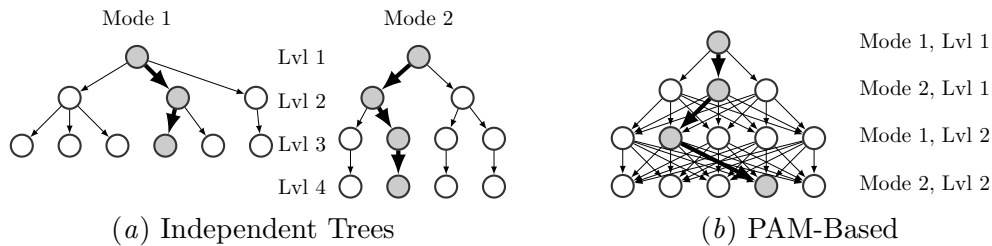
As a prelude to our discussion on hierarchical topic models in multiple modes, we first discuss relevant hierarchical topic models for a single feature mode. We start with the nested Chinese Restaurant Process (nCRP) and hierarchical LDA (hLDA). Then, we define the Pachinko Allocation Machine (PAM) and hierarchical PAM (hPAM).

Blei et al. (2003) defined nCRP as follows. Imagine a city with an infinite number of infinite-table Chinese restaurants. One restaurant is designated as the root. On each table in each restaurant, there is a card referring to another restaurant. The root restaurant never appears on a card and other restaurants appear on one card within the entire city. Customers visit this city for a set number of nights, starting at the root restaurant and each subsequent night visiting the restaurant on the previous night's card. Tables within a restaurant are selected based on the CRP. Thus, the restaurants are organized into an infinitely-branched hierarchical tree of set depth L (the length of stay), with each restaurant having an associated level. Furthermore, each customer x has a corresponding set of restaurants \mathbf{T}_x that they visited (their path through the tree), which determines their set of topics. In natural language processing, customers correspond to documents and restaurants correspond to topics. The transition from LDA to hLDA is made by drawing a path through the nCRP for each document, then using the set of restaurants visited by a customer as the set of topics in a document (only these topics have non-zero prevalence).

Li and McCallum (2006) defined PAM as a model that connects topics with a directed acyclic graph (DAG). PAM samples a topic path through the DAG, starting at the root, and sampling each child according to the multinomial distribution of the current topic. Although a PAM can use an arbitrary DAG, we use a DAG with a leveled structure, i.e., for a set number of levels L , each node on level $\ell \in \{1, \dots, L-1\}$ is a parent of every node on the next level $\ell+1$. In this model, the number of topics $\tau^{(\ell)}$ are predetermined for each level ℓ . Mimno et al. (2007) defined hPAM as a PAM model where all nodes are associated with distributions over the vocabulary, rather than only those on the lowest level. In this model, a path is drawn through the PAM for each document, where the set of nodes visited determines the set of topics in a document.

3.3. Conditional Hierarchical Bayesian Tucker Decomposition

Next, we extend our conditional Bayesian Tucker decomposition to topic hierarchies. For simplicity, we assume $p = 2$ feature modes, but we discuss generalizations to $p \geq 3$ in Appendix C.1.


 Figure 1: Example Hierarchical Topic Models ($p = 2$)

As outlined in Section 3.2, the transition from LDA to hLDA uses a nCRP. The challenge with implementing this in our context is that we have multiple modes of topics. It is not clear how to generalize nCRP with each table representing a pair of topics rather than a single topic. In the case of hLDA, when a customer sits at a new table, it represents a new topic. But if each table represents a pair of topics rather than a single topic, what does a new table represent? A new topic in one or both modes or a new combination of existing topics? There is no clear way of determining what a new table represents without imposing an order on pairs of groups; but we prove in Section 4.2 that there is no natural order where the order of customers does not matter.

We describe two solutions to this problem: the independent topic model and the hierarchical topic model. The hierarchical model requires an order of modes since the choice of topic for each mode depends on the topic in its parent mode. Given $x \in \{1, \dots, d_0\}$, let \mathbf{T}_x be its path through a conceptual topic model. The topic distribution ϕ_x along any such \mathbf{T}_x is drawn from $\text{Dir}(\boldsymbol{\alpha}) \in \mathcal{S}^L$, with L being the length of \mathbf{T}_x . In other words, \mathbf{T}_x dictates which topics have positive probabilities.

The probability for a HBT decomposition (illustrated in its plate diagram, Figure 4 in Appendix A) is constructed similarly to that of the non-hierarchical model (4). While the factor matrices (1) and individual count probabilities (3) are the same, the core tensor probability (2) must be modified to incorporate the hierarchical model. Here, we incorporate the probability of each sample's x path \mathbf{T}_x given the parameter γ . The core tensor ϕ_x 's probability for each sample depends on both prior $\boldsymbol{\alpha}$ and path \mathbf{T}_x since the path dictates which topics have non-zero probabilities:

$$P(\boldsymbol{\phi}, \mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\gamma}) = \prod_{x=1}^{d_0} P(\boldsymbol{\phi}_x | \boldsymbol{\alpha}, \mathbf{T}_x) P(\mathbf{T}_x | \boldsymbol{\gamma}). \quad (5)$$

Combining (1), (3), and (5) yields the overall model probability:

$$P(\mathbf{Y}, \mathbf{Z}, \boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = P(\boldsymbol{\psi} | \boldsymbol{\beta}) P(\boldsymbol{\phi}, \mathbf{T} | \boldsymbol{\alpha}, \boldsymbol{\gamma}) P(\mathbf{Y}, \mathbf{Z} | \boldsymbol{\phi}, \boldsymbol{\psi}). \quad (6)$$

We define an independent topic model ($p = 2$) as containing two separate topic models for each mode (example shown in Figure 1(a)). Each customer x draws two paths, one for each mode, $\mathbf{T}_x^{(1)}$ and $\mathbf{T}_x^{(2)}$, represented as a set of topics in that mode. The overall topic list \mathbf{T}_x consists of all possible pairs in $\mathbf{T}_x^{(1)}$ and $\mathbf{T}_x^{(2)}$. A common choice draws these paths $\mathbf{T}_x^{(1)}$ and $\mathbf{T}_x^{(2)}$ by means of two independent nCRPs as follows: Let $\mathbf{P}^{(x, \ell, m)}$ be the

probabilities associated with two independent CRPs, with hyper-parameters $\gamma_m^{(\ell)}$, where $m \in \{1, 2\}$ is the mode and $\ell \in \{1, \dots, L_m\}$ is the level of the tree. In this model, the number of tables (topics) in each mode $\tau^{(\ell, m)}$ varies. Specifically,

$$P_i^{(x, \ell, m)} = \begin{cases} \frac{n_i}{\gamma_m^{(\ell)} + n - 1}, & \text{table } i \text{ occupied} \\ \frac{\gamma_m^{(\ell)}}{\gamma_m^{(\ell)} + n - 1}, & \text{next unoccupied table } i \end{cases},$$

where customer x is the n^{th} customer at the restaurant and n_i customers are already seated at table i . While others have imposed independent hierarchical structures on the topics of different modes (Song et al., 2013; Schifanella et al., 2014), none have incorporated nCRP into their topic models.

We define a hierarchical topic model ($p = 2$) as a single topic model with a dominant mode (example depicted in Figure 1(b)). Without loss of generality, we assume the first mode is dominant. At the first level, each customer x starts at the root topic 1 in mode 1; then, each customer chooses a topic j in mode 2, according to probability $\mathbf{P}_1^{(1,1)}$. At each subsequent level $\ell \in \{2, \dots, L\}$, each customer chooses a topic i in mode 1, according to $\mathbf{P}_j^{(\ell-1,2)}$, then a topic j in mode 2, according to $\mathbf{P}_i^{(\ell,1)}$. Here, $\mathbf{P}_i^{(\ell, m)}$ is the probability distribution over its children topics. There are two obvious choices for the overall topic list \mathbf{T}_x , the pairs of topics visited at each level of the DAG (which we call the level method) or all possible pairs of elements in the topic lists for each mode (we call this the Cartesian method). If PAM ideas are used, $\mathbf{P}_i^{(\ell, m)}$ (for all $i \in \{1, \dots, \tau^{(\ell, 1)}\}$ and $m \in \{1, 2\}\}$) are multinomials drawn from Dirichlet distributions, i.e., $\mathbf{P}_i^{(\ell, m)} \sim \text{Dir}(\gamma_i^{(\ell, m)}) \in \mathcal{S}_{\tau^{(\ell, m)}}$. While PAM has been used to model interactions between variables for LDA (Li and McCallum, 2006), it has not been used to model topic interactions between multiple modes.

4. Properties

First, in Section 4.1, we define the exchangeability, partition, and rich-get-richer properties, based on those of the CRP, which govern topic models. Blei (2007) defined the exchangeability property as the probability of a seating being invariant under permutations of customers. The partition property implies that two seating arrangements, consisting of the same number of tables with equal occupants, have the same probability. In the single mode case, this property follows from exchangeability, but we find that this is not necessarily true for multiple modes. Teh (2010) described the rich-get-richer property as large clusters growing faster than smaller clusters. For simplicity, we assume $p = 2$ feature modes and study a single level. In Section 4.2, we prove that all of these properties cannot strictly apply in the $p = 2$ case.

4.1. Definitions

In order to define these properties, we first need to define a partition, an arrangement of customers into topics. Here, we assume the number of customers n is fixed.

Definition 1 *The count of customers assigned to each topic is a **partition**. For partition $\rho = \{\rho_{ij}\}$, ρ_{ij} is the count of customers assigned topic i in mode 1 and topic j in mode 2.*

In other words, ρ_{ij} is the number of customers sitting at the same table. Here, $\boldsymbol{\rho}$ is a matrix, with rows and columns representing the topics in each mode. Note that $n = \sum_{i,j} \rho_{ij}$.

We first define the exchangeability property. This property is necessary because if the order of customers affects the performance of a model it would be impossible to determine how well it would perform inference, i.e., predict new customers.

Definition 2 We say that a topic model has the **exchangeability** property if the probability of a partition does not depend on the order of the customers, i.e., $P(\boldsymbol{\rho}|\text{order of customers}) = P(\boldsymbol{\rho}|\Pi(\text{order of customers}))$, where Π is a permutation of the order.

Next, we define two variants of the partition property: strict and loose. This property implies that the label and order of topics does not matter.

Definition 3 We say that a topic model has the **strict partition** property if for all partitions $\boldsymbol{\rho}$ and $\Pi(\boldsymbol{\rho})$ permutations of the elements of $\boldsymbol{\rho}$, $P(\boldsymbol{\rho}) = P(\Pi(\boldsymbol{\rho}))$.

Definition 4 We say that a topic model has the **loose partition** property if for all partitions $\boldsymbol{\rho}$ and $\Pi(\boldsymbol{\rho})$ permutations of the rows and columns of $\boldsymbol{\rho}$, $P(\boldsymbol{\rho}) = P(\Pi(\boldsymbol{\rho}))$.

Note that if strict partition holds, so does loose partition. However, the converse does not necessarily hold.

Lastly, we define the rich-get-richer property under the assumption of the exchangeability and partition properties. The general idea is that new customers are more likely to join topics with more customers. Let $\rho_{(\cdot)i} = \sum_{j=1}^{K_1} \rho_{ji}$ and $\rho_{i(\cdot)} = \sum_{j=1}^{K_2} \rho_{ij}$. In addition, let ξ_i be the probability of a new customer being assigned to topic i in mode 1 and θ_{ij} be the probability of a new customer being assigned to topic j in mode 2 given that the customer was assigned to topic i in mode 1. Due to the exchangeability property, ξ_i and θ_{ij} do not depend on the order of the customers. Also, for the independent topic model, we can drop the dependency in θ on the topic in the first mode and denote θ_i as the probability of a new customer being assigned to topic i in mode 2.

Definition 5 We say that an independent topic model has the **rich-get-richer** property if

1. assuming $\rho_{i(\cdot)} \neq 0$ and $\rho_{j(\cdot)} \neq 0$, then $\xi_i > \xi_j$ if and only if $\rho_{i(\cdot)} > \rho_{j(\cdot)}$ (Mode 1), and
2. assuming $\rho_{(\cdot)i} \neq 0$ and $\rho_{(\cdot)j} \neq 0$, then $\theta_i > \theta_j$ if and only if $\rho_{(\cdot)i} > \rho_{(\cdot)j}$ (Mode 2).

Definition 6 We say that a hierarchical topic model has the **rich-get-richer** property if

1. assuming $\rho_{i(\cdot)} \neq 0$ and $\rho_{j(\cdot)} \neq 0$, then $\xi_i > \xi_j$ if and only if $\rho_{i(\cdot)} > \rho_{j(\cdot)}$ (Mode 1), and
2. assuming $\rho_{ki} \neq 0$ and $\rho_{kj} \neq 0$, then $\theta_{ki} > \theta_{kj}$ if and only if $\rho_{ki} > \rho_{kj}$, $\forall k$ (Mode 2).

The difference between definitions 5 and 6 is that the independent version only compares the fibers (rows and columns) of the partition, while the hierarchical version compares individual elements within each fiber.

4.2. Conditions

Now, we determine under which conditions the loose and strong partition properties hold in multi-modal hierarchical models. While we sketch our proofs here, detailed proofs are in Appendix E. We begin with the PAM-based model ($p = 2$), giving the specific case in which the loose partition property holds.

Theorem 7 *The loose partition property holds in the PAM model ($p = 2$) if and only if the parameters of the Dirichlet distributions are symmetric.*

In the proof, we consider a single Dirichlet distribution and show that a uniform prior is sufficient for the loose partition property to hold. Then, we use a non-uniform example to show that uniformity is necessary for this property to hold. Lastly, we argue that the theorem holds for the entire PAM model if and only if it holds for every node in the model.

Next, we prove that a hierarchical topic model ($p = 2$) for which all three properties apply does not exist. First, we show that the probability of a new customer being assigned a topic in each mode has a functional form.

Lemma 8 *If there exists a hierarchical topic model ($p = 2$) where the rich-get-richer strong partition, and exchangeability properties hold, then ξ and θ are of the form: $\xi_i \propto \begin{cases} f(\rho_{i(\cdot)}), & \rho_{i(\cdot)} > 0 \\ \gamma_0(K_1), & \rho_{i(\cdot)} = 0 \end{cases}$ and $\theta_{ij} \propto \begin{cases} g_i(\rho_{ij}), & \rho_{ij} > 0 \\ \gamma_i(K_2), & \rho_{ij} = 0 \end{cases}$, for some functions f and g_i ($\forall i$).*

We show that this formulation is sufficient to satisfy the strong partition property and necessary to satisfy the rich-get-richer and exchangeability properties.

Next, we show that these functions must be linear.

Lemma 9 *Given the assumptions and results of Lemma 8, f and g_i (for all i) are linear.*

We use the formula from Lemma 8 to compute the probabilities for two different orderings. Given the exchangeability property, these formulations must be equal. From this, we conclude f must be linear.

Finally, we use the linear form to show that such a model does not exist.

Theorem 10 *There does not exist a hierarchical topic model ($p = 2$) where the rich-get-richer, strong partition, and exchangeability properties all hold.*

We exhibit in Appendix E an example to show that the strict partition property is violated by the formulation in Lemma 9.

However, the independent trees model allows all three properties. This follows from the properties of each independent CRP.

Theorem 11 *The exchangeability, loose partition, and rich-get-richer properties hold in the independent trees model.*

5. Algorithms

We use a Gibbs sampler to compute the posterior model based on our prior and samples. This algorithm alternates between drawing new paths through the hierarchical topic model and solving the Bayesian Tucker decomposition problem. This scheme is similar to that developed for hLDA by Blei et al. (2003). We present a general overview of our Gibbs sampler in Algorithm 2.

Algorithm 2: Algorithm Overview

Initialize hierarchical topic model and Bayesian Tucker decomposition

for $i = 1, \dots, I$ **do**

- └ Draw Bayesian Tucker decomposition ϕ, ψ and latent topics \mathbf{z} (Algorithm 3 or 6)
 - └ Draw hierarchical topic model \mathbf{T} (Algorithm 4, 5, or 7)
-

In Section 5.1 we present two algorithms for sampling from the Bayesian Tucker decomposition, and in Section 5.2 we present Gibbs samplers for both the independent trees and PAM-based topic models. In Appendix F, we present variations and modifications that can be made to these models to improve performance.

5.1. Sampling from Bayesian Tucker Decomposition

Here, we present a collapsed Gibbs sampler for our Bayesian Tucker decomposition, Algorithm 3, which we derive in Appendix C. We give a non-collapsed version, Algorithm 6, in Appendix B. When applying a topic model, we constrain these algorithms so that $\phi_{x\mathbf{k}}$ is positive for $\mathbf{k} \in \mathbf{T}_x$ and zero for $\mathbf{k} \notin \mathbf{T}_x$, for all x .

While these algorithms are based on the Yang and Dunson (2016) algorithm, they did not derive a collapsed sampler and their model uses a different decomposition form. Our collapsed algorithm makes sampling easier by integrating out the Dirichlet distributions. Here, n_x^k is the count of topic k given sample x , $m_{hy}^{(j)}$ is the count of feature y in mode j given topic h , and a superscript $-xi$ indicates omitting count i . Recall that the bold font implies $\mathbf{m}_h^{(j)}$ is a vector over y 's.

Algorithm 3: Collapsed Bayesian Tucker Decomposition Gibbs Sampler

for $x = 1, \dots, d_0$ **do**

- └ **for** $i = 1, \dots, \lambda_x$ **do**
 - └ Compute latent topic probabilities

$$P(\mathbf{z}_i^{(x)} = \mathbf{k} \mid \mathbf{n}^{-xi}, \mathbf{m}^{-xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto (n_x^{\mathbf{k}, -xi} + \alpha_{\mathbf{k}}) \prod_{j=1}^p \frac{m_{k_j y_j}^{(j), -xi} + \beta_{y_j}^{(j)}}{\sum_{y=1}^{d_j} m_{k_j y}^{(j), -xi} + \beta_y^{(j)}}$$
 - └ Draw latent topic $\mathbf{z}_i^{(x)}$ from $P(\mathbf{z}_i^{(x)} = \mathbf{k} \mid \mathbf{n}^{-xi}, \mathbf{m}^{-xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
-

5.2. Hierarchical Topic Models

Next, we present Gibbs samplers for the independent trees and PAM-based hierarchical topic models. For the PAM-based model, we give both non-collapsed and collapsed versions.

Algorithm 4 describes how paths are drawn through an independent trees model. This algorithm is a generalization of the hLDA algorithm by Blei et al. (2003). Here, $m_{c_{x,\ell},y}^{(j),-x}$ is the count of features y assigned to topic $c_{x,\ell}$ in mode j , $m_{c_{x,\ell},(\cdot)}^{(j),-x} := \sum_{y=1}^{d_j} m_{c_{x,\ell},y}^{(j),-x}$, and subscript $-x$ indicates omitting sample x .

Algorithm 4: Independent Trees Algorithm

```

for  $j = 1, \dots, p$  do
    for  $x = 1, \dots, d_0$  do
        for  $\ell = 1, \dots, L_j$  do
            
$$P\left(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}^{(j)}, \mathbf{Z}^{(j)}\right) = \frac{\Gamma\left(m_{c_{x,\ell},(\cdot)}^{(j),-x} + d_j \beta^{(j)}\right) \prod_y \Gamma\left(m_{c_{x,\ell},y}^{(j)} + \beta^{(j)}\right)}{\prod_y \Gamma\left(m_{c_{x,\ell},y}^{(j),-x} + \beta^{(j)}\right) \Gamma\left(m_{c_{x,\ell},(\cdot)}^{(j)} + d_j \beta^{(j)}\right)}$$

            
$$P\left(c_{x,\ell}^{(j)} | \mathbf{c}_{-x\ell}^{(j)}\right) = \text{prior imposed by CRP}$$

            
$$P\left(c_{x,\ell}^{(j)} | \mathbf{Y}^{(j)}, \mathbf{c}_{-x,\ell}^{(j)}, \mathbf{Z}^{(j)}\right) \propto P\left(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}_\ell^{(j)}, \mathbf{Z}^{(j)}\right) P\left(c_{x,\ell}^{(j)} | \mathbf{c}_{-x\ell}^{(j)}\right)$$


```

We provide collapsed (Algorithm 5) and non-collapsed (Algorithm 7 in Appendix B) algorithms to draw paths through the PAM-based model. The collapsed algorithm is derived by integrating out the Dirichlet distribution and dropping a constant. Li and McCallum (2006) and Mimno et al. (2007) gave collapsed Gibbs algorithms for four-layered PAM and hPAM models, but not for arbitrary DAGs. This algorithm uses an arbitrary PAM-based hierarchical structure, as described in Appendix C.1. In Algorithm 5, $n_{ik}^{(\ell,j)}$ is the count of customers assigned to topic k in the mode j at level ℓ and parent topics i .

Algorithm 5: Collapsed PAM-Based Hierarchical Topic Model Algorithm

```

for  $\ell = 1, \dots, L$  do
    for  $j = 1, \dots, p$  do
        if  $\ell \neq 1$  or  $j$  is not a root mode then
            for  $x = 1, \dots, d_0$  do
                
$$P\left(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}^{(j)}, \mathbf{Z}^{(j)}\right) = \frac{\Gamma\left(m_{k,(\cdot)}^{(j),-x} + d_j \beta^{(j)}\right) \prod_y \Gamma\left(m_{k,y}^{(j)} + \beta^{(j)}\right)}{\prod_y \Gamma\left(m_{k,y}^{(j),-x} + \beta^{(j)}\right) \Gamma\left(m_{k,(\cdot)}^{(j)} + d_j \beta^{(j)}\right)}$$

                
$$P\left(c_{x,\ell}^{(j)} = k | \mathbf{Y}^{(j)}, \mathbf{c}_{-x,\ell}^{(j)}, \mathbf{Z}^{(j)}\right) \propto P\left(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}_\ell^{(j)}, \mathbf{Z}^{(j)}\right) \left(\gamma_{ik}^{(\ell,j)} + n_{ik}^{(\ell,j)}\right)$$


```

6. Model Evaluation

First, we describe the cancer and ASD data that we used to train and analyze our models in Section 6.1. We compare the coherence of various models trained on all three data sets in Section 6.2. Then, we use a held-out non-parametric likelihood estimate to compare our cancer models to each other, described in Appendix G.7. In Appendix G.5, we discuss classification results for cancer.

6.1. Data Sets

The first data set we examined is from The Cancer Genome Atlas and contains patients with one of four common types of cancer: breast, lung, prostate, or colorectal. It is believed that genetic factors affect the patients’ risk of developing cancer. This data has a clear hierarchical structure: variants are on genes, which are part of biological pathways,³ which are within patients. Furthermore, pathways and genetic variants can be hierarchically grouped based on functions and interactions. A HBT decomposition would be useful in the classification of cancer, where we have two modes of feature variables (genetic variants and pathways) and a clear hierarchical structure of groupings of genetic variants and pathways. We used the Reactome pathway data set (Fabregat et al., 2018; Croft et al., 2014) to determine which genes are in each pathway.

Another data set we analyzed is from the National Database for Autism Research and contains paired siblings – one with ASD and another without – and counts of their genetic variants. We used the same process for including pathway information as with the cancer data. ASD refers to a group of neurodevelopment disorders defined by a range of behavioral patterns and difficulty with social interaction. The Centers for Disease Control and Prevention estimates 1 in 68 children have ASD, but it is more prevalent in boys than girls. Currently, ASD is typically diagnosed by parent and doctor observation of a child’s behavior and development. Experts believe ASD is caused by a mixture of genetic and environmental factors (National Institute of Neurological Disorders and Stroke, 2017).

More details about the data are in Appendix G.1.

6.2. Coherence

We compared the coherence of models trained on the cancer and ASD data. Xing et al. (2019) stated that co-occurrence and posterior based methods are common evaluation techniques for LDA models. The main posterior method, topic stability, can be undermined by high frequency genes and pathways (in our case) causing poor topics to have high stability. Newman et al. (2010) showed that pointwise mutual information (PMI/UCI) consistently outperformed other methods, in terms of correlation with human subjects. However, we also provide Mimno et al.’s (2011) measure (UMass), a coherence measure intended for use without an external reference corpus. Some details on our use of these measures are in Appendix G.2.

While most of the models we analyzed were Bayesian and hierarchical (using the independent trees model), the CP TensorLy model (Kossaifi et al., 2019) was deterministic and non-hierarchical. In our tables, we bolded the best performing measures for emphasis. We further explain our methodology in Appendix G.3.

Figures 2 and 3 show that the best HBT models’ topics were more coherent than the baseline models’ (hLDA, CP tree, and CP TensorLy) on the cancer and ASD data. On cancer data, our largest improvement was on UMass pathways coherence, where the unweighted HBT model using UMass coherence outperformed the best baseline, the CP tree model using UMass coherence, by 34.52%. However, on the other three measures, the best HBT models outperformed the best baseline models by only 0.76%-2.44%. On ASD data,

3. We define a pathway as a set of genes working together for a specific biological function. We do not consider the nature of interactions between genes, only their membership in the pathway.

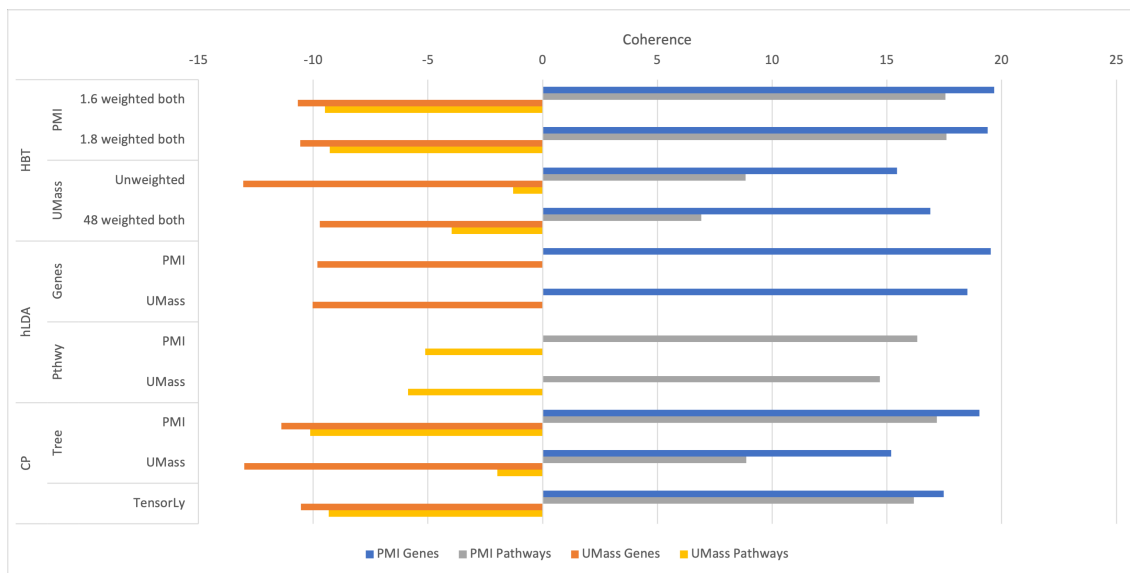


Figure 2: Coherence of Cancer Models

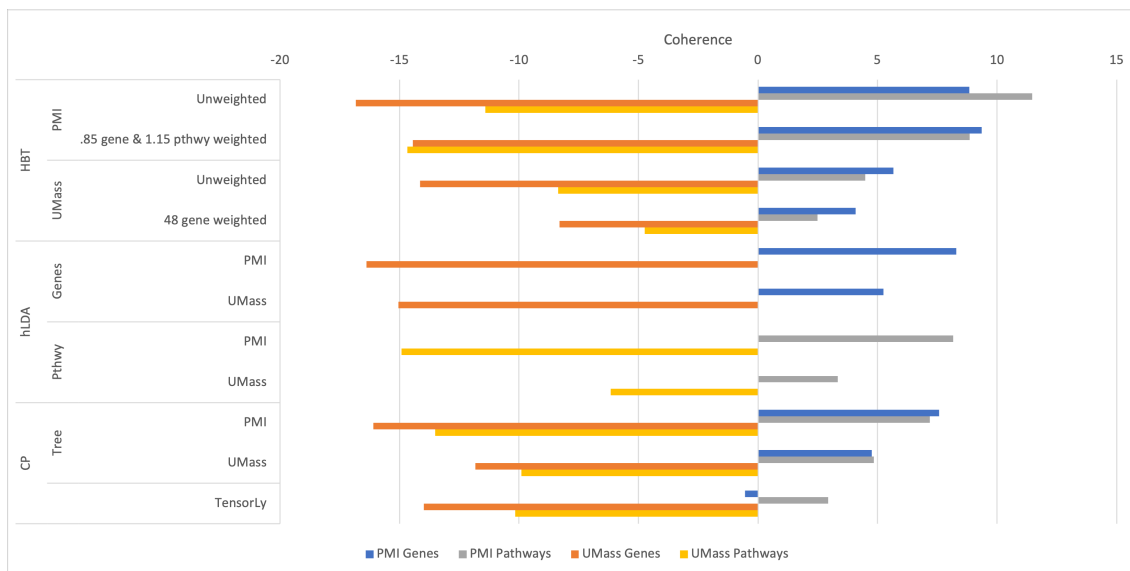


Figure 3: Coherence of ASD Models

we saw more significant improvement on all four measures. Our smallest relative improvement was on PMI gene coherence, where the HBT model using PMI coherence and .85 gene and 1.15 pathway weighting outperformed hLDA using PMI coherence by 12.65%. Our largest improvement margin was on PMI pathway coherence, where the unweighted HBT model using PMI coherence outperformed hLDA using PMI coherence by 40.56%. While no model performed best on all measures, the HBT models using PMI coherence outperformed

the baseline models on all PMI coherence measures and the weighted HBT UMass models outperformed the baselines on all UMass coherence measures.

In Appendix G.4, we applied a HBT structure to a different context, natural language processing, and achieved promising results. Here, our tensor consisted of articles, phrases, and words. While HBT performed worse than CP TensorLy on UMass phrase coherence, the HBT model using UMass coherence outperformed the best baseline models by up to 58% on the other three coherence measures.

In general, we found that HBT generates more coherent, and thus more interpretable, topics than other models. The choice of specific model depends on the coherence measure, relative priority and structure of modes, number of topics, memory requirements, etc. Other models have their own drawbacks. hLDA can only evaluate one mode at a time. CP Trees imposes a more rigid structure, with a single topic hierarchy for both modes. CP Tensorly does not provide a hierarchical topic structure and is worse suited for sparse counting tensors than the Bayesian models.

7. Conclusion

We developed methods to perform a Bayesian Tucker decomposition and designed strategies for incorporating dependent and independent hierarchical topic models in the decomposition. The independent trees model uses independent CRPs to generate hierarchical structures for each mode, while the PAM-based model creates a single hierarchical structure across modes. Furthermore, we presented a blueprint for generalizing these models to more than two feature modes. We generalized the properties of CRP to multiple modes and prove that the strict versions of these properties cannot apply to a hierarchical model with multiple modes. Additionally, we derived a collapsed Gibbs sampler for the Bayesian Tucker decomposition, with an arbitrary number of feature modes. Lastly, we trained our models on real-world examples and found that our models' topics were more coherent than existing methods.

Acknowledgments

Research reported in this publication was supported, in part, by the National Library of Medicine, Grant Number T32LM012203. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- Brett W Bader and Tamara G Kolda. Tensor toolbox for MATLAB, version 3.2.1, Apr 2021. URL <http://www.tensortoolbox.org/>.
- David Blei. COS 597c: Bayesian nonparametrics, September 2007. URL <https://www.cs.princeton.edu/courses/archive/fall07/cos597C/scribe/20070921.pdf>.
- David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Proceedings of the*

- 16th International Conference on Neural Information Processing Systems, NIPS'03*, page 17–24. MIT Press, Cambridge, MA, USA, 2003.
- Wray Buntine. *Variational Extensions to EM and Multinomial PCA*, pages 23–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.
- David Croft, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R Kamdar, Bijay Jassal, Steven Jupe, Lisa Matthews, Bruce May, Stanislav Palatnik, Karen Rothfels, Veronica Shamovsky, Heeyeon Song, Mark Williams, Ewan Birney, Henning Hermjakob, Lincoln Stein, and Peter D'Eustachio. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 42(Database issue):D472–477, Jan 2014.
- Peter J. Diggle and Richard J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society, Series B: Methodological*, 46: 193–227, 1984.
- Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936, 2020.
- Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- David B. Dunson and Chuanhua Xing. Nonparametric Bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.
- Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 46(D1):D649–D655, Jan 2018.
- Lars Grasedyck. Hierarchical singular value decomposition of tensors. *Society for Industrial and Applied Mathematics Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.
- W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, Oct 2009.
- Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent Dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc., 2010.

- Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.
- Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. TensorLy: Tensor learning in python. *Journal of Machine Learning Research*, 20(26):1–6, 2019. URL <http://jmlr.org/papers/v20/18-277.html>.
- Wei Li and Andrew McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584, New York, NY, USA, 2006. ACM.
- Yuan Luo, Yu Xin, Ephraim Hochberg, Rohit Joshi, Ozlem Uzuner, and Peter Szolovits. Subgraph augmented non-negative tensor factorization (SANTF) for modeling clinical narrative text. *Journal of the American Medical Informatics Association : JAMIA*, 22, 04 2015.
- Yuan Luo, Faraz S Ahmad, and Sanjiv J Shah. Tensor factorization for precision medicine in heart failure with preserved ejection fraction. *Journal of cardiovascular translational research*, 10(3):305–312, 2017a.
- Yuan Luo, Fei Wang, and Peter Szolovits. Tensor factorization toward precision medicine. *Briefings in bioinformatics*, 18(3):511–514, 2017b.
- David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, New York, NY, USA, 2007. ACM.
- David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 262–272. Association for Computational Linguistics, 2011.
- National Institute of Neurological Disorders and Stroke. Autism spectrum disorder fact sheet, Dec 2017.
- David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Los Angeles, California, June 2010. Association for Computational Linguistics. URL <https://aclanthology.org/N10-1012>.
- Aaron Schein, Mingyuan Zhou, David M. Blei, and Hanna Wallach. Bayesian Poisson tucker decomposition for learning the structure of international relations. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 2810–2819. JMLR.org, 2016.

- Claudio Schifanella, K. Selçuk Candan, and Maria Luisa Sapino. Multiresolution tensor decompositions with mode hierarchies. *Association for Computing Machinery Transactions on Knowledge Discovery from Data*, 8(2):10:1–10:38, June 2014.
- Le Song, Mariya Ishteva, Ankur Parikh, Eric Xing, and Haesun Park. Hierarchical tensor decomposition of latent tree graphical models. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 334–342, Atlanta, Georgia, USA, June 2013. PMLR.
- Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010.
- Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes. *Journal of the american statistical association*, 101(476):1566–1581, 2006.
- Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C. Denny, Abel Kho, You Chen, Bradley A. Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. *KDD : proceedings. International Conference on Knowledge Discovery & Data Mining*, 2015:1265–1274, Aug 2015. ISSN 2154-817X. doi: 10.1145/2783258.2783395. URL <https://pubmed.ncbi.nlm.nih.gov/31452969>.
- Linzi Xing, Michael J. Paul, and Giuseppe Carenini. Evaluating topic quality with posterior variability. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, November 2019. Association for Computational Linguistics.
- Zenglin Xu, Feng Yan, and Alan Qi. Infinite tucker decomposition: Nonparametric Bayesian models for multiway data analysis. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning, ICML '07*, pages 1023–1030, New York, NY, USA, 2012. ACM.
- Yun Yang and David B. Dunson. Bayesian conditional tensor factorizations for high-dimensional classification. *Journal of the American Statistical Association*, 111(514): 656–669, Feb 2016.
- Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *CoRR*, abs/1401.6497, 2014. URL <http://arxiv.org/abs/1401.6497>.
- Jing Zhou, Anirban Bhattacharya, Amy H. Herring, and David B. Dunson. Bayesian factorizations of big sparse tensors. *Journal of the American Statistical Association*, 110(512):1562–1576, Feb 2015.

Appendices

Appendix A. Plate Diagram

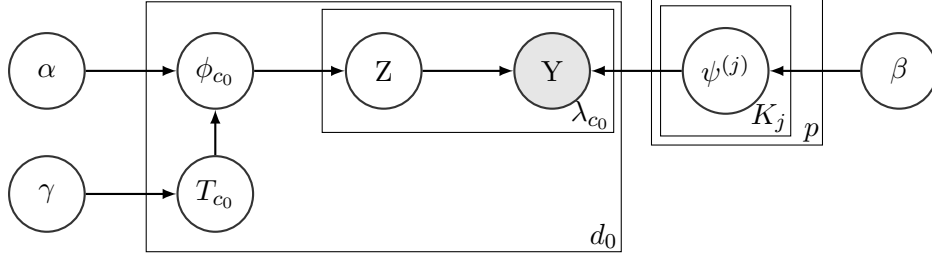


Figure 4: Plate Diagram for a Conditional Hierarchical Bayesian Tucker Decomposition

Appendix B. Non-Collapsed Algorithms

Algorithm 6: Non-Collapsed Bayesian Tucker Decomposition Gibbs Sampler

```

for  $x = 1, \dots, d_0$  do
    Draw core tensor  $\tilde{\phi}_x \sim \text{Dir}(\boldsymbol{\alpha} + \mathbf{n}_x) \in \mathcal{S}_K$ 
    for  $\mathbf{k} \in \mathcal{K}$  do
         $\phi_{x\mathbf{k}} = \tilde{\phi}_{x\text{vec}(\mathbf{k})}$ 
for  $j = 1, \dots, p$  do
    for  $k = 1, \dots, K_j$  do
        Draw auxiliary matrices  $\boldsymbol{\psi}_k^{(j)} \sim \text{Dir}(\boldsymbol{\beta}^{(j)} + \mathbf{m}_h^{(j)}) \in \mathcal{S}_{d_j}$ 
for  $x = 1, \dots, d_0$  do
    for  $i = 1, \dots, \lambda_x$  do
        Compute topic probabilities  $P(\mathbf{z}_i^{(x)} = \mathbf{k} | -) \propto \phi_{x\mathbf{k}} \boldsymbol{\psi}_{k y_i}^{(j)}$ 
        Draw  $\mathbf{z}_i^{(x)}$  from  $P(\mathbf{z}_i^{(x)} = \mathbf{k} | -)$ 
    
```

Algorithm 7: Non-Collapsed PAM-Based Hierarchical Topic Model Algorithm

```

for  $\ell = 1, \dots, L$  do
    for  $j = 1, \dots, p$  do
        if  $\ell \neq 1$  or  $j$  is not a root mode then
            for  $i \in \{\text{possible parent topics}\}$  do
                 $\mathbf{P}_i^{(\ell, j)} \sim \text{Dir}(\boldsymbol{\gamma}_i^{(\ell, j)} + \mathbf{n}_i^{(\ell, j)})$ 
            for  $x = 1, \dots, d_0$  do
                
$$P(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}^{(j)}, \mathbf{Z}^{(j)}) = \frac{\Gamma(m_{k,(\cdot)}^{(j), -x} + d_j \beta^{(j)})}{\prod_y \Gamma(m_{k,y}^{(j), -x} + \beta^{(j)})} \frac{\prod_y \Gamma(m_{k,y}^{(j)} + \beta^{(j)})}{\Gamma(m_{k,(\cdot)}^{(j)} + d_j \beta^{(j)})}$$

                
$$P(\mathbf{c}_{x,\ell}^{(j)} = k | \mathbf{Y}^{(j)}, \mathbf{c}_{-x,\ell}^{(j)}, \mathbf{Z}^{(j)}) \propto P(\mathbf{Y}_x^{(j)} | \mathbf{Y}_{-x}^{(j)}, \mathbf{c}_\ell^{(j)}, \mathbf{Z}^{(j)}) P_{ik}^{(\ell, j)}$$

    
```

Appendix C. Collapsed Gibbs Derivation

Here, we derive equations for collapsed Gibbs sampling of a conditional Bayesian Tucker decomposition. This derivation is similar to that of LDA. We begin with the total probability of our model and integrate out ϕ and ψ :

$$\begin{aligned} P(\mathbf{Y}, \mathbf{Z} | \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \int_{\phi} \int_{\psi} P(\mathbf{Y}, \mathbf{Z}, \phi, \psi | \boldsymbol{\alpha}, \boldsymbol{\beta}) d\psi d\phi \\ &= \int_{\psi} \prod_{j=1}^p \prod_{h=1}^{K_j} P(\psi_h^{(j)} | \boldsymbol{\beta}) \prod_{i=1}^{\lambda_x} P(\mathbf{y}_{xi} | \psi_{z_{xi}}) d\psi \int_{\phi} \prod_{x=1}^{d_0} P(\phi_x | \boldsymbol{\alpha}) \prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) d\phi. \end{aligned}$$

All ψ 's and ϕ 's are independent from each other and thus can be treated separately. We first examine the ϕ 's:

$$\int_{\phi} \prod_{x=1}^{d_0} P(\phi_x | \boldsymbol{\alpha}) \prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) d\phi = \prod_{x=1}^{d_0} \int_{\phi_x} P(\phi_x | \boldsymbol{\alpha}) \prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) d\phi_x.$$

Now, we look at a single ϕ :

$$\int_{\phi_x} P(\phi_x | \boldsymbol{\alpha}) \prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) d\phi_x = \int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{\alpha_k - 1} \prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) d\phi_x.$$

Letting n_x^k denote the count of topic(s) k given independent variable x , we can express

$$\prod_{i=1}^{\lambda_x} P(\mathbf{z}_{xi} | \phi_x) = \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k}.$$

Thus, the ϕ_x integral can be rewritten as

$$\int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{\alpha_k - 1} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k} d\phi_x = \int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k + \alpha_k - 1} d\phi_x.$$

According to the functional expression of the Dirichlet distribution,

$$\int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k + \alpha_k - 1} d\phi_x = 1.$$

We apply this equation to get rid of the integral, resulting in a fraction made up of products of Gamma functions,

$$\begin{aligned}
 \int_{\phi_x} P(\phi_x | \alpha) \prod_{i=1}^{\lambda_x} P(z_{xi} | \phi_x) d\phi_x &= \int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k + \alpha_k - 1} d\phi_x \\
 &= \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)} \int_{\phi_x} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)} \prod_{k=1}^{K_1 \cdots K_j} \phi_{xk}^{n_x^k + \alpha_k - 1} d\phi_x \\
 &= \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)}.
 \end{aligned}$$

Similarly, we derive the ψ part, letting $m_{hy}^{(j)}$ denote the count of dependent variable y in the j^{th} mode given topic h :

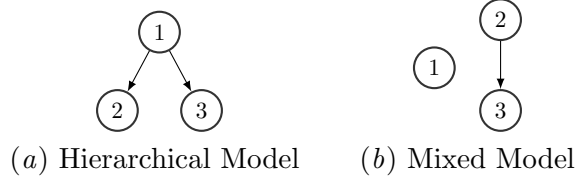
$$\begin{aligned}
 \int_{\psi} \prod_{j=1}^p \prod_{h=1}^{K_j} P(\psi_h^{(j)} | \beta) \prod_{i=1}^{\lambda_x} P(y_{xi} | \psi_{z_{xi}}) d\psi &= \prod_{j=1}^p \prod_{h=1}^{K_j} \int_{\psi_h^{(j)}} P(\psi_h^{(j)} | \beta) \prod_{i=1}^{\lambda_x} P(y_{xi}^{(j)} | \psi_{z_{xi}}^{(j)}) d\psi_h^{(j)} \\
 &= \prod_{j=1}^p \prod_{h=1}^{K_j} \int_{\psi_h^{(j)}} \frac{\Gamma\left(\sum_{y=1}^{d_j} \beta_y^{(j)}\right)}{\prod_{y=1}^{d_j} \Gamma(\beta_y^{(j)})} \prod_{y=1}^{d_j} \left(\psi_{z_{xi}^{(j)} y}^{(j)}\right)^{\beta_y^{(j)} - 1} \prod_{y=1}^{d_j} \left(\psi_{z_{xi}^{(j)} y}^{(j)}\right)^{m_{hy}^{(j)}} d\psi_h^{(j)} \\
 &= \prod_{j=1}^p \prod_{h=1}^{K_j} \int_{\psi_h^{(j)}} \frac{\Gamma\left(\sum_{y=1}^{d_j} \beta_y^{(j)}\right)}{\prod_{y=1}^{d_j} \Gamma(\beta_y^{(j)})} \prod_{y=1}^{d_j} \left(\psi_{z_{xi}^{(j)} y}^{(j)}\right)^{m_{hy}^{(j)} + \beta_y^{(j)} - 1} d\psi_h^{(j)} \\
 &= \prod_{j=1}^p \prod_{h=1}^{K_j} \frac{\Gamma\left(\sum_{y=1}^{d_j} \beta_y^{(j)}\right) \prod_{y=1}^{d_j} \Gamma\left(m_{hy}^{(j)} + \beta_y^{(j)}\right)}{\prod_{y=1}^{d_j} \Gamma(\beta_y^{(j)}) \Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j)} + \beta_y^{(j)}\right)}.
 \end{aligned}$$

By combining the expressions from the ϕ and ψ parts, we obtain,

$$P(\mathbf{Y}, \mathbf{Z} | \alpha, \beta) = \prod_{x=1}^{d_0} \frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)} \times \prod_{j=1}^p \prod_{h=1}^{K_j} \frac{\Gamma\left(\sum_{y=1}^{d_j} \beta_y^{(j)}\right)}{\prod_{y=1}^{d_j} \Gamma(\beta_y)} \frac{\prod_{y=1}^{d_j} \Gamma\left(m_{hy}^{(j)} + \beta_y\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j)} + \beta_y^{(j)}\right)}.$$

Next, we need to derive an expression for the probability distribution of $\mathbf{z}_i^{(c)}$, which denotes the hidden variable(s) for the i^{th} count in $x = c$, where $\mathbf{y} = \mathbf{v}$. Let a superscript $-ci$ denote the count, excluding the i^{th} count in $x = c$. By Bayes' Theorem,

$$P(\mathbf{z}_i^{(c)} | \mathbf{Z}^{-ci}, \mathbf{Y}, \alpha, \beta) = \frac{P(\mathbf{z}_i^{(x)}, \mathbf{Z}^{-ci}, \mathbf{Y} | \alpha, \beta)}{P(\mathbf{Z}^{-ci}, \mathbf{Y} | \alpha, \beta)}.$$


 Figure 5: General p Topic Model Examples ($p = 3$)

By applying this and dropping the denominator, we can express the probability distribution of $\mathbf{z}_i^{(c)}$ as proportional to the expression we derived above,

$$\begin{aligned}
 P(\mathbf{z}_i^{(c)} = \mathbf{k} | \mathbf{Z}^{-ci}, \mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto P(\mathbf{z}_i^{(c)} = \mathbf{k}, \mathbf{Z}^{-ci}, \mathbf{Y} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
 &\propto \left(\frac{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} \alpha_k\right)}{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(\alpha_k)} \right)^{d_0} \prod_{x \neq c} \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_x^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_x^k + \alpha_k\right)} \times \prod_{j=1}^p \left(\frac{\Gamma\left(\sum_{y=1}^{d_j} \beta_y^{(j)}\right)}{\prod_{y=1}^{d_j} \Gamma(\beta_y^{(j)})} \right)^{K_j} \prod_{h=1}^{K_j} \prod_{y \neq v_j} \Gamma\left(m_{hy}^{(j)} + \beta_y^{(j)}\right) \\
 &\quad \times \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_c^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_c^k + \alpha_k\right)} \prod_{j=1}^p \prod_{h=1}^{K_j} \frac{\Gamma\left(m_{hv_j}^{(j)} + \beta_{v_j}^{(j)}\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j)} + \beta_y^{(j)}\right)}.
 \end{aligned}$$

We simplify this expression by dropping multiplicative constants,

$$\begin{aligned}
 &\propto \frac{\prod_{k=1}^{K_1 \cdots K_j} \Gamma(n_c^k + \alpha_k)}{\Gamma\left(\sum_{k=1}^{K_1 \cdots K_j} n_c^k + \alpha_k\right)} \prod_{j=1}^p \prod_{h=1}^{K_j} \frac{\Gamma\left(m_{hv_j}^{(j)} + \beta_{v_j}^{(j)}\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j)} + \beta_y^{(j)}\right)} \\
 &\propto \prod_{h=1}^{K_1 \cdots K_j} \Gamma\left(n_c^h + \alpha_h\right) \prod_{j=1}^p \prod_{h=1}^{K_j} \frac{\Gamma\left(m_{hv_j}^{(j)} + \beta_{v_j}^{(j)}\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j)} + \beta_y^{(j)}\right)}.
 \end{aligned}$$

We now split this expression to obtain an h -independent summation, which can be dropped,

$$\begin{aligned}
 &\propto \prod_{h \neq \mathbf{k}} \Gamma\left(n_c^h, -ci + \alpha_h\right) \prod_{j=1}^p \prod_{h \neq k_j} \frac{\Gamma\left(m_{hv_j}^{(j), -ci} + \beta_{v_j}^{(j)}\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{hy}^{(j), -ci} + \beta_y^{(j)}\right)} \times \Gamma\left(n_c^{\mathbf{k}}, -ci + \alpha_{\mathbf{k}} + 1\right) \prod_{j=1}^p \frac{\Gamma\left(m_{k_j v_j}^{(j), -ci} + \beta_{v_j}^{(j)} + 1\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{k_j y}^{(j), -ci} + \beta_y^{(j)} + 1\right)} \\
 &\propto \Gamma\left(n_c^{\mathbf{k}}, -ci + \alpha_{\mathbf{k}}\right) \prod_{j=1}^p \frac{\Gamma\left(m_{k_j v_j}^{(j), -ci} + \beta_{v_j}^{(j)}\right)}{\Gamma\left(\sum_{y=1}^{d_j} m_{k_j y}^{(j), -ci} + \beta_y^{(j)}\right)} \times \left(n_c^{\mathbf{k}}, -ci + \alpha_{\mathbf{k}}\right) \prod_{j=1}^p \frac{m_{k_j v_j}^{(j), -ci} + \beta_{v_j}^{(j)}}{\sum_{y=1}^{d_j} m_{k_j y}^{(j), -ci} + \beta_y^{(j)}} \\
 &\propto \left(n_c^{\mathbf{k}}, -ci + \alpha_{\mathbf{k}}\right) \prod_{j=1}^p \frac{m_{k_j v_j}^{(j), -ci} + \beta_{v_j}^{(j)}}{\sum_{y=1}^{d_j} m_{k_j y}^{(j), -ci} + \beta_y^{(j)}}.
 \end{aligned}$$

C.1. Generalizations to $p \geq 3$

For the independent topic model, we generalize to p independent hierarchical models (such as trees) and our topic tuples \mathbf{T}_x are all possible combinations of $\mathbf{T}_x^{(1)}$, \dots , and $\mathbf{T}_x^{(p)}$. For the hierarchical model, with general p , it is possible to have more complex dependence relations between modes. For example, Figure 5(a) shows a model where modes 2 and 3 depend on mode 1 but do not directly depend on each other. To this end, we assume we are given a DAG representing the dependency structure of the p modes. At each level, each customer moves through the DAG, selects a topic in each mode (ordered topologically)

based on probability $P_i^{(\ell,m)}$, where ℓ is the level, m is the mode, and i represents the topics of all parent modes. Similarly to the $p = 2$ case, the tuples of topics visited at each level of the DAG or all possible tuples of elements in the topic lists for each mode correspond to the topic list for each x (those topics with non-zero prevalence). Furthermore, it is possible to have a mixture of independent and hierarchical topic models; a certain mode or set of modes could be independent from the other modes. For example, Figure 5(b) presents a model where mode 1 is independent of modes 2 and 3, while mode 3 depends on mode 2.

Appendix D. Derivatives of Log-Likelihood

We define Log-Likelihood LL as the log of our model probability, found in Section 3.3, (6).

$$LL = \log P(\mathbf{Y}, \mathbf{Z}, \phi, \psi, \mathbf{T} | \alpha, \beta, \gamma) = \log P(\psi | \beta) + \log P(\phi, \mathbf{T} | \alpha, \gamma) + \log P(\mathbf{Y}, \mathbf{Z} | \phi, \psi)$$

First, we derive the derivative with respect to α_i :

$$\begin{aligned} \frac{\partial LL}{\partial \alpha_i} &= \frac{\partial}{\partial \alpha_i} \log P(\phi, \mathbf{T} | \alpha, \gamma) = \frac{\partial}{\partial \alpha_i} \log \prod_{x=1}^{d_0} P(\phi_x | \alpha, \mathbf{T}_x) P(\mathbf{T}_x | \gamma) \\ &= \frac{\partial}{\partial \alpha_i} \sum_{x=1}^{d_0} \log P(\phi_x | \alpha, \mathbf{T}_x) = \sum_{x=1}^{d_0} \frac{1}{P(\phi_x | \alpha, \mathbf{T}_x)} \frac{\partial P(\phi_x | \alpha, \mathbf{T}_x)}{\partial \alpha_i} \\ &= \sum_{x=1}^{d_0} \frac{B(\alpha)}{\prod_{j=1}^{\mathbf{K}} \phi_{xj}^{\alpha_j - 1}} \frac{\partial}{\partial \alpha_i} \left[\frac{1}{B(\alpha)} \prod_{j=1}^{\mathbf{K}} \phi_{xj}^{\alpha_j - 1} \right], \text{ where } B(\cdot) \text{ is the Beta function} \\ &= \sum_{x=1}^{d_0} \left[B(\alpha) \frac{\partial}{\partial \alpha_i} \frac{1}{B(\alpha)} + \frac{\alpha_i - 1}{\phi_{xi}} \right] \\ &= \sum_{x=1}^{d_0} \left[\Psi^{(0)} \left(\sum_{j=1}^{\mathbf{K}} \alpha_j \right) - \Psi^{(0)}(\alpha_i) + \frac{\alpha_i - 1}{\phi_{xi}} \right], \text{ where } \Psi^{(0)}(\cdot) \text{ is the Polygamma function.} \end{aligned}$$

Next, we similarly derive the derivative with respect to $\beta_i^{(j)}$:

$$\begin{aligned} \frac{\partial LL}{\partial \beta_i^{(j)}} &= \frac{\partial}{\partial \beta_i^{(j)}} \log P(\psi | \beta) = \sum_{h_j=1}^{K_j} \log P(\psi_{h_j}^{(j)} | \beta^{(j)}) \\ &= \sum_{h_j=1}^{K_j} \left[\Psi^{(0)} \left(\sum_{y=1}^{d_j} \beta_y^{(j)} \right) - \Psi^{(0)}(\beta_i^{(j)}) + \frac{\beta_i^{(j)} - 1}{\psi_{h_j i}^{(j)}} \right] \end{aligned}$$

Appendix E. Conditions Proofs

First, we prove Theorem 7, showing that the parameters of the Dirichlet distributions in our PAM model are symmetric.

Proof First, we look at a single Dirichlet distribution, i.e. a single-node PAM model and show that the partition property applies if and only if the parameters are symmetric. Without loss of generality, assume the parameters of the Dirichlet distribution $\gamma = (\gamma_1, \dots, \gamma_K)$

are ordered such that $\gamma_1 \leq \dots \leq \gamma_K$, where K is the number of topics. Also, by PAM $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K) \sim \text{Dir}(\boldsymbol{\gamma})$.

First, we show sufficiency. To this end, assume $\gamma_p = \gamma_q := \gamma$ for any p, q . Here, both the probability distribution and expectation of all θ 's are equal (this is clear from examining the probability distribution function of the Dirichlet distribution). We denote the probability of m_p people to topic p and m_q people to topic q , along with our assignments of people to each of the other topics, as $P(m_p, m_q, -)$. Thus this probability,

$$\begin{aligned} P(m_p, m_q, -) &\propto \int_0^1 \int_0^1 P(m_p, m_q | \theta_p, \theta_q) P(\theta_p, \theta_q) d\theta_p d\theta_q \\ &\propto \int_0^1 \int_0^1 \theta_p^{m_p + \gamma - 1} \theta_q^{m_q + \gamma - 1} d\theta_p d\theta_q \\ &= \int_0^1 \theta^{m_p + \gamma - 1} d\theta \int_0^1 \theta^{m_q + \gamma - 1} d\theta, \end{aligned}$$

is equal to the probability of assigning m_q people to topic p and m_p people to topic q ,

$$P(m_q, m_p, -) \propto \int_0^1 \theta^{m_p + \gamma - 1} d\theta \int_0^1 \theta^{m_q + \gamma - 1} d\theta.$$

Similarly, since the probability distributions over all θ 's are the same, the probability of assigning m people to topics 1 through K is equal to the probability of assigning any permutation of m people to topics 1 through k . Thus the partition property holds.

Next, we show necessity. To this end, assume $\gamma_p < \gamma_q$ (for some p and q). The probability of assigning m_p people to topic p and m_q people to topic q ,

$$P(m_p, m_q, -) \propto \int_0^1 \int_0^1 \theta_p^{m_p + \gamma_p - 1} \theta_q^{m_q + \gamma_q - 1} d\theta_p d\theta_q = \frac{1}{(m_p + \gamma_p)(m_q + \gamma_q)},$$

is not equal to the probability of assigning m_q people to topic p and m_p people to topic q ,

$$P(m_q, m_p, -) \propto \frac{1}{(m_p + \gamma_q)(m_q + \gamma_p)},$$

for all m_p and m_q . If $m_p < m_q$, then elementary algebra shows $(m_p + \gamma_p)(m_q + \gamma_q) < (m_p + \gamma_q)(m_q + \gamma_p)$ and if $m_p > m_q$, then $(m_p + \gamma_p)(m_q + \gamma_q) > (m_p + \gamma_q)(m_q + \gamma_p)$. Thus, the partition property does not hold.

If and only if the partition property holds for single nodes of the PAM model, it is possible to re-arrange topics in both modes (i.e., the loose partition property applies). ■

Next, we prove Lemma 8, showing that for the given properties to hold, then $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ must be of the given form.

Proof For such a model, $\xi_i = \xi_j$ if and only if $\rho_{i(\cdot)} = \rho_{j(\cdot)}$ and $\theta_{ki} = \theta_{kj}$ if and only if $\rho_{ki} = \rho_{kj}$. This is sufficient because of the strong partition property and necessary because of the rich-get-richer property. By the strong partition property and the chain rule,

$\xi_i \theta_{ij} = \xi_k \theta_{kl}$ if $\rho_{ij} = \rho_{kl}$. Also, because of the rich-get-richer and exchangeability properties, we can express $\xi_i \propto \begin{cases} f(\rho_{i(\cdot)}), & \rho_{i(\cdot)} > 0 \\ \gamma_0(K_1), & \rho_{i(\cdot)} = 0 \end{cases}$ and $\theta_{ij} \propto \begin{cases} g_i(\rho_{ij}), & \rho_{ij} > 0 \\ \gamma_i(K_2), & \rho_{ij} = 0 \end{cases}$. ■

Then, we prove Lemma 9, showing that the functions must be linear.

Proof Suppose we want to assign x people to topic 1 and one person to topic 2. One way (case one) to do this would be to assign all x people at topic 1, then one person to topic 2. Another way (case two) to do this would be to assign $x - 1$ people to topic 1, then one person to topic 2, then one more customer to topic 1. The probabilities of these cases can be expressed as:

$$P(\text{case one}) = \frac{f(1)}{\gamma_0(1) + f(1)} \cdots \frac{f(x-2)}{\gamma_0(1) + f(x-2)} \frac{f(x-1)}{\gamma_0(1) + f(x-1)} \frac{\gamma_0(1)}{\gamma_0(1) + f(x)}$$

$$P(\text{case two}) = \frac{f(1)}{\gamma_0(1) + f(1)} \cdots \frac{f(x-2)}{\gamma_0(1) + f(x-2)} \frac{\gamma_0(1)}{\gamma_0(1) + f(x-1)} \frac{f(x-1)}{\gamma_0(2) + f(x-1) + f(1)}$$

Thus, the differences in the probabilities are that the first case has $\gamma_0(1) + f(x)$ in the last denominator while the second case has $\gamma_0(2) + f(x-1) + f(1)$ in the last denominator. If the exchangeability property applies, we have $P(\text{case one}) = P(\text{case two})$ and in turn $\gamma_0(1) + f(x) = \gamma_0(2) + f(x-1) + f(1)$. Since this must apply for all x , by induction we have $f(x) = xf(1) + \gamma_0(2) - \gamma_0(1)$. Thus f is linear.

Most generally, $\xi_i \propto \begin{cases} \rho_{i(\cdot)} - \gamma_{02}, & \rho_{i(\cdot)} > 0 \\ \gamma_{01} + \gamma_{02}K_1, & \rho_{i(\cdot)} = 0 \end{cases}$ and $\theta_{ij} \propto \begin{cases} \rho_{ij} - \gamma_{i2}, & \rho_{ij} > 0 \\ \gamma_{i1} + \gamma_{i2}K_2, & \rho_{ij} = 0 \end{cases}$. Note that this is a generalized nCRP. ■

Finally, we prove Theorem 10, showing that such a model does not exist.

Proof Given the forms of ξ and θ from Lemma 9, suppose we take two elements $(i, j) \neq (m, n)$, $i \neq m$ and swap them within ρ . Then the ratio ν of the original probability with the swapped probability is:

$$\omega(\rho_{ij}, \rho_{mn}) := \frac{\Gamma(\rho_{m(\cdot)} - \gamma_{02})\Gamma(\rho_{i(\cdot)} - \gamma_{02})}{\Gamma(\rho_{m(\cdot)} - \rho_{mn} + \rho_{ij} - \gamma_{02})\Gamma(\rho_{i(\cdot)} - \rho_{ij} + \rho_{mn} - \gamma_{02})}$$

$$\nu = \frac{\omega(\rho_{ij}, \rho_{mn}) \Gamma(\rho_{ij} - \gamma_{i2})\Gamma(\rho_{mn} - \gamma_{m2})}{\Gamma(\rho_{ij} - \gamma_{m2})\Gamma(\rho_{mn} - \gamma_{i2})}$$

If the strict partition property is satisfied, then $\nu = 1$ for all ρ_{ij} , ρ_{mn} , $\rho_{i(\cdot)}$, and $\rho_{m(\cdot)}$. Note that $\frac{\Gamma(\rho_{ij} - \gamma_{i2})\Gamma(\rho_{mn} - \gamma_{m2})}{\Gamma(\rho_{ij} - \gamma_{m2})\Gamma(\rho_{mn} - \gamma_{i2})} = 1$ for all ρ_{ij} and ρ_{mn} if and only if $\gamma_{i2} = \gamma_{m2}$, however $\omega(\rho_{ij}, \rho_{mn}) \neq 1$ for all ρ_{ij} , ρ_{mn} , $\rho_{i(\cdot)}$, and $\rho_{m(\cdot)}$. For example, if $\rho_{ij} = 1$, $\rho_{mn} = 2$, $\rho_{i(\cdot)} = 3$, $\rho_{m(\cdot)} = 3$, and $\gamma_{02} = 0$, then $\omega(\rho_{ij}, \rho_{mn}) = \frac{2!2!}{1!3!} = \frac{2}{3} \neq 1$. This shows that $\nu = 1$ is not always possible. ■

Appendix F. Model Variations

In our efforts to boost the performance of our models, we implemented and developed a number of variations and modifications to our original methodology, including:

- **Different seeds:** Train the model using ten different random number generator seeds, picking the best model based on either log-likelihood or coherence.
- **Keep best:** Check the log-likelihood or coherence during training every ten iterates, saving the best model rather than the final one. Another variation on this method is resetting to the current best model after each check.
- **MAP estimate:** When using the collapsed sampler, use the maximum a posteriori (MAP) estimate of the auxiliary matrices and core tensors, rather than performing one non-collapsed iterate.
- **Adjust the number of counts:** For example, double the number of counts or set all non-zero counts to one.
- **Sparse cutoff:** Set a cutoff value, below which all proportional probabilities are set to zero.
- **Initialization:** Initialize the auxiliary matrices with those trained using a different method.
- **Relative number of topic model iterates to Bayesian Tucker:** Do two Bayesian Tucker sample iterations for every draw from the hierarchical topic model, or vice-versa.
- **Set a topics goal:** Decide on an ideal number of topics. Adjust γ during training to reach that goal. The formula we used was:

$$\gamma_{\text{new}} = \gamma_{\text{old}} * \max \left\{ \min \left\{ \left(\frac{\text{topics goal}}{\# \text{ of topics}} \right)^{1/\prod_{i=1}^p (L_i-1)}, 2 \right\}, 0.5 \right\}.$$

- **Exponential weighting:** Apply an exponential weight to the relative probabilities in the Collapsed Gibbs Sampler (Algorithm 3) as such:

$$P \left(z_i^{(x)} = \mathbf{k} | - \right) \propto \left(n_x^{\mathbf{k}, -xi} + \alpha_{\mathbf{k}} \right) \prod_{j=1}^p \left[\frac{m_{k_j y_j}^{(j), -xi} + \beta_{y_j}^{(j)}}{\sum_{y=1}^{d_j} m_{k_j y}^{(j), -xi} + \beta_y^{(j)}} \right]^{w_p},$$

where w_p is the weight for mode p . This allows us to adjust the relative variance, i.e., the uniformity of each mode.

For our coherence measure experiments (Section 6.2 and Appendix G.4), we used different seeds, keep best, MAP estimate, topics goals, and exponential weighting. For our classification models (Appendix G.5), we used keep best. While we tried some of the other above methods, they did not perform as well. For the likelihood models (Appendix G.7), we did not use any of the above modifications.

Appendix G. Experiments

Code available at github.com/ars2240/asdHB Tucker.

Cancer type	TCGA designation
Breast	Breast Invasive Carcinoma (BRCA)
Lung	Lung Squamous Cell Carcinoma (LUSC) or Lung Adenocarcinoma (LUAD)
Prostate	Prostate Adenocarcinoma (PRAD)
Colorectal	Colon Adenocarcinoma (COAD) or Rectum Adenocarcinoma (READ)

Table 2: Cancer types and TCGA designation

G.1. Data Sets

The cancer data contains 3,037 patients. The Cancer Genome Atlas (TCGA) designations we considered for each type of cancer are given in Table 2. This data set contained 1,044 patients with breast cancer, 1,066 patients with lung cancer, 494 patients with prostate cancer, and 433 patients with colorectal cancer. After linking with the Reactome pathways, we were left with 7,846 genes and 1,678 pathways.

Once we similarly linked the ASD genetic variants to the pathways, our data set contains 7,211 genetic variants and 1,413 pathways.

The Reuters data contains 5,501 articles, 8,820 phrases, and 6,837 unique words.

In our experiments, we split each data set into a 30% held-out test set and performed 10-fold cross-validation (CV) on the remaining training/validation data.

G.2. Coherence Measure

In order to utilize this coherence measure in our context, we made some modifications: 1) our metric is intrinsic and does not utilize an external corpus to determine the gene or pathway probabilities and co-occurrence probabilities; and 2) we determined co-occurrence as having a variant on a pair of genes or pathways, not accounting for sequences as in many natural language processing examples.

G.3. Coherence Models

For pre-processing our data, we removed genes that appeared in fewer than 200 or more than 2,000 patients and words that appeared in fewer than 200 or more than 2,000 documents, eliminating both rare and very common genes or words. We also removed phrases numbers that were present in fewer than 10 articles. Words without an assigned phrase were then assigned to a single word phrase corresponding to the given word. We modified [Ding et al.’s \(2020\)](#) implementation to process the R8 data into the article, phrase, word tensor structure. The CP TensorLy model was trained on the entire training data set, then split into folds; while the other models were split into folds before fitting the decompositions. Our coherence measures looked at the top 5 genes, pathways, sentences, or words in each topic.

The Bayesian cancer and ASD models had a topics goal of 500 and the R8 models had a topics goal of 50; while the CP TensorLy model had 200 topics for all data sets. The hierarchical models used three levels.

For the R8 models with bad phrases removed, we removed words and phrases.

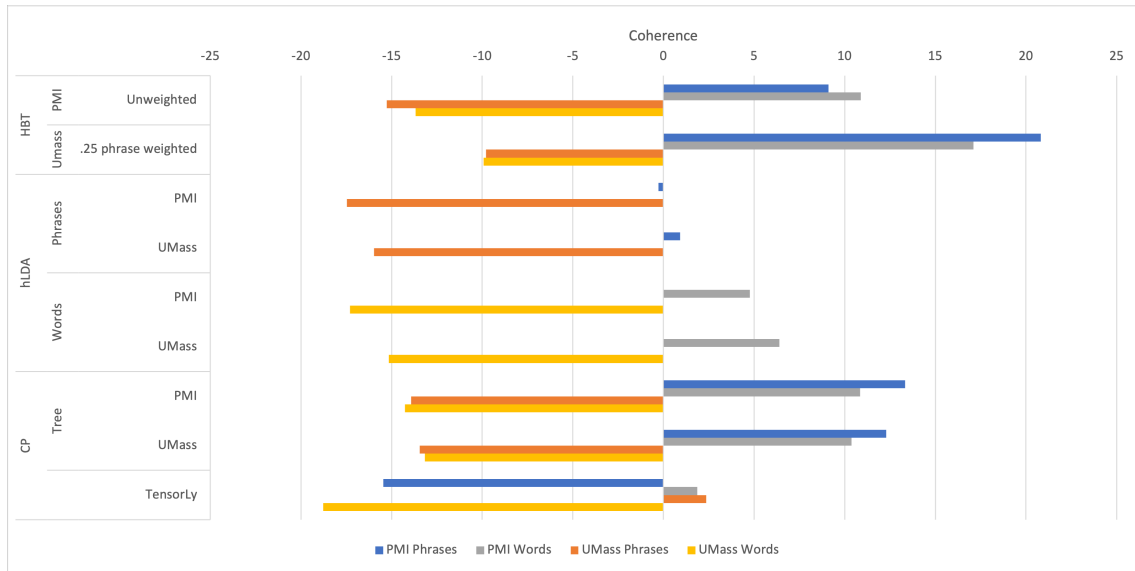


Figure 6: Coherence of R8 Models

Model	Phrases		Words	
	PMI	UMass	PMI	UMass
HBT (PMI)	11.80	-12.91	12.38	-12.37
HBT (UMass)	12.61	-11.19	13.39	-10.79
CP (TensorLy)	-15.47	-6.64	-4.00	-20.04

Table 3: Coherence of R8 Models with Bad Phrases Removed

G.4. Reuters Experiments

We also applied a similar structure from our genetic models to natural language processing. Incorporating phrases as another mode may improve grouping of articles. The hierarchical structure here would be: words make up phrases, which make up articles. We looked at the eight largest classes (earnings, acquisitions, money - foreign exchange, grain, crude, trade, interest, and shipping) in the Reuters-21578 data set (denoted R8) (Dua and Graff, 2017). We used SpaCy (Honnibal and Montani, 2017) to group words into phrases (noun chunks). This gives us a count of words in each phrase in each article.

While our best HBT model outperformed the baseline models on three-of-four coherence measures on the R8 data, it under-performed on UMass phrase coherence. Figure 6 shows that the HBT model using UMass coherence and .25 phrase weighting outperformed the CP model using PMI coherence, the best baseline model, on PMI phrase and word coherence by 56.07% and 57.84% respectively. This HBT model also outperformed the CP model using UMass coherence by 24.54% on UMass words coherence. However, while the CP TensorLy model performed worst on these three coherence measures, it outperformed our best model on UMass phrase coherence (2.35 to -9.79). After examining this data set further, we removed words and phrases that frequently showed up in topics and had few co-occurrences with other words in the topic. While we tried various word sets and were

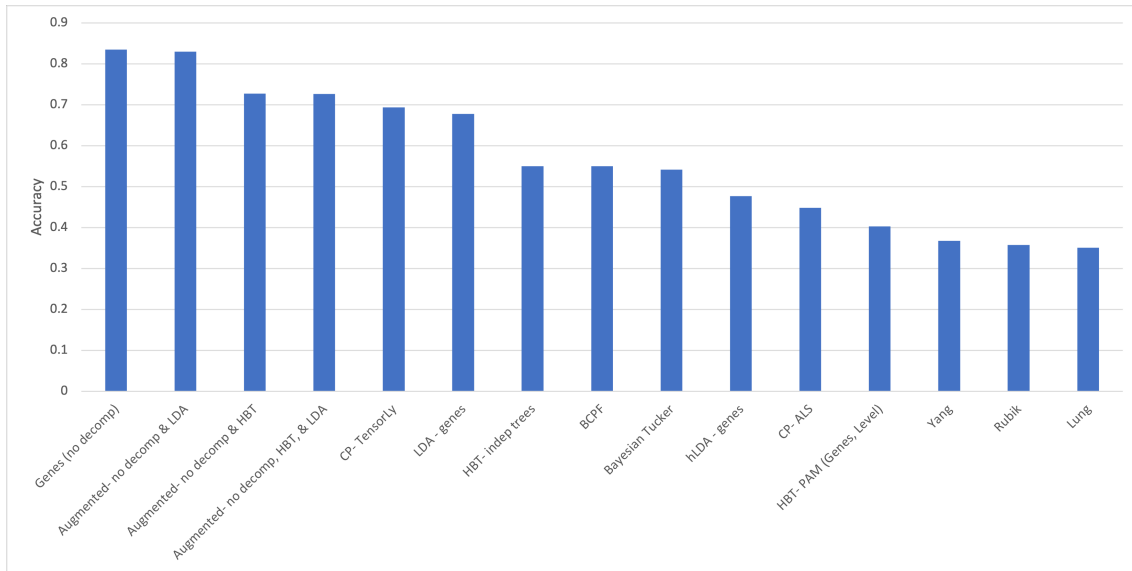


Figure 7: Accuracy of Cancer Models

able to make improvements, due to the Bayesian nature of the algorithms, words with fewer co-occurrences are more likely to be placed in the same topic than in a deterministic model. Table 3 shows that while these changes narrowed the gap in performance on UMass phrase coherence, the CP TensorLy model still outperformed our best HBT model on this measure.

G.5. Classification

We trained logistic regression models using patient groups from the decomposition models in order to predict each patients’ cancer diagnosis. Figure 7 shows the accuracy of these models and Appendix G.6 details our implementation. Unfortunately, none of the models we trained, including the baseline decomposition models, outperformed classifying based on the genetic variant counts (i.e., not using a decomposition model), which had an accuracy of 83.49%. Augmenting the genetic variant data with our decomposition models did not improve the accuracy of our predictions. Of the decomposition models, CP TensorLy performed best (69.33% accurate). The other CP models, Zhao et al.’s (2014) Bayesian CP Factorization (BCPF) (54.94% accurate) and Bader and Kolda’s (2021) alternating least squares (ALS) CP (44.78% accurate) performed significantly worse. The second-best decomposition model was LDA on the genetic variants (67.74% accurate), while our genes-based hLDA model had a 47.68% accuracy. Our independent trees HBT model and our non-hierarchical Bayesian Tucker model performed about the same, with accuracies of 54.98% and 54.14%, respectively. Our PAM-based HBT model, using genes as the dominant mode and the level method, performed worse (40.31% accurate). Yang and Dunson’s (2016) Bayesian Conditional Tensor factorization (36.74% accurate) and Wang et al.’s (2015) Rubik model (35.70% accurate) only slightly outperformed diagnosing all patients with Lung cancer, the most prevalent class, at 35.10% prevalence.

G.6. Classification Models

For the CP TensorLy model, we removed genes that appeared in fewer than 200 or more than 2,000 patients and used 200 topics. For the LDA model, we modified Hoffman et al.’s (2010) variational Bayes implementation, using 40 topics. For our independent trees HBT model, we removed genes that appeared in fewer than 400 or more than 1,000 patients and used three levels, $\gamma = 0.1$, and our “keep best” methodology. For our Bayesian Tucker model, we removed genes that appeared in fewer than 200 or more than 2,000 patients and used 10 topics on each mode and our “keep best” methodology. For our hLDA model, we used two levels and $\gamma = 0.1$. For our CP ALS model, we used 25 topics. For our PAM-based HBT, we removed genes that appeared in fewer than 400 or more than 1,000 patients and used three levels and 10 topics-per-level. For the Rubik model, we used 5 topics.

G.7. Likelihood

One issue with comparing likelihoods across models is that the probabilities from Section 3 are not comparable due to differing hierarchical model structures. To compute the held-out likelihood, we would need to sum over or integrate out our hierarchical model variables, which does not have a closed-form solution.

To solve this problem, we use a non-parametric likelihood estimate, similar to Li and McCallum (2006) and based on empirical likelihood (Diggle and Gratton, 1984). First, we randomly generate one thousand patients, using the trained generative process. Then, we compute the probabilities of a held-out test or validation patients as a mixture of the generated patients. Unlike other likelihood measures, this method is stable, easy to compute, and yields values that can reasonably be compared across models.

We trained HBT decomposition models using various hierarchical models and computed the mean validation log-likelihood (over the ten CV folds, using the above methodology). For each hierarchical model, we trained with varying levels $L \in \{2, 3, 4, 5\}$. Additionally, we trained the independent trees model with three CRP hyperparameters $\gamma \in \{0.5, 1, 2\}$. For the PAM-based model, we compared each choice of dominant mode (genes or pathways) and topic set composition (Cartesian or level set method, see Section 3.3 for definitions). We also used varying topics per level $\tau \in \{10, 25, 50\}$.

We plotted the results in Figure 8 and give the values in Table 4. This figure depicts the mean validation log-likelihood for hierarchical models trained on the cancer data set, comparing it to the total number of topics (product of the number of gene and pathway topics) for the model. In the case of the independent trees model, the number of topics is an average over the CV folds. Though the CRP hyperparameter (γ) and number of topics per level (τ) are not displayed on the plot, they dictate the number of total topics.

Figure 8 shows that the log-likelihood for models with fewer topics is often larger than those with more topics (14 of 20 lines peak at the fewest topics), indicating that some models are overfitted. We observed that some PAM-based models outperformed the independent trees models, which we hoped would be the case given the inherent hierarchical structure between genes and pathways, but there was not a clear reason as to which models were better than others (other than the total number of topics). The PAM-based models with level set topic composition are the most sensitive to changes in the other parameters (the standard deviation of the log-likelihoods of all such models with genes as the dominant mode

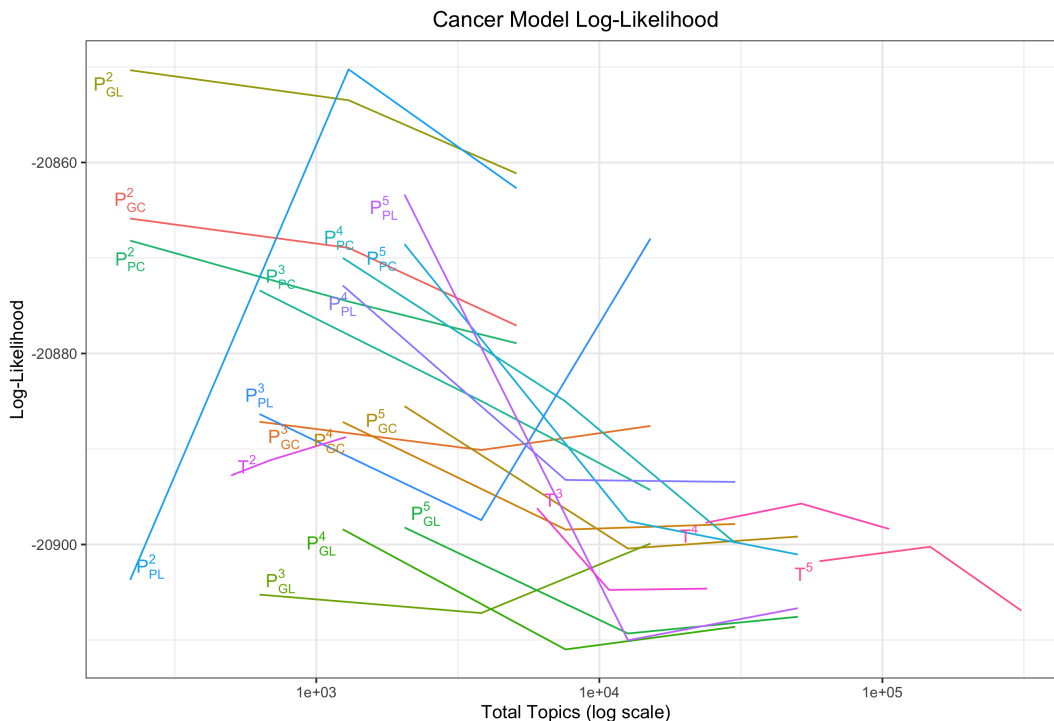


Figure 8: Each line has a coded label: The large letter indicates if the point is from the independent trees model (“T”) or the PAM-based model (“P”). The superscript indicates the number of levels in the model. The subscript (for the PAM-based models) indicates the dominant mode, genes (“G”) or pathways (“P”), followed by the topic set composition method, Cartesian (“C”) or level set (“L”). Every label (combination of model type, number of levels, dominant mode, and topic set composition, if applicable) has the same color.

was 23.15 and 19.86 for pathway-dominant models). The independent tree models were the least sensitive (with standard deviation of 5.68). The PAM-based models with Cartesian topic composition were in between (with standard deviation of 11.52 for gene-dominant models and 12.56 for pathway-dominant models). However, the differences between these models is well within the margin of error, as the standard deviation in log-likelihood over the CV folds for each model is about 2,090 (or 10% of the log-likelihood).

Table 4 gives the hierarchical topic model (independent trees or PAM-based), dominant mode (for PAM-based model, genes or pathways), whether the topic sets are created using the Cartesian or level method (See Section 3.3 for definitions), the value of γ used (hyperparameter in CRP for the independent trees model, a uniform Dirichlet prior was used in PAM), the topics per level τ (for the PAM-based model), the number of hierarchical levels, the mean validation log-likelihood (over the 10-fold CV, computed using the method described in Section G.7), the standard deviation of the log-likelihood (over the 10-fold CV), the number of gene and pathway topics (or mean number of topics across the CV folds in

the case of the independent trees model), and the total number of topics (the product of the number of gene and pathway topics). This data is presented in Section [G.7](#).

Table 4: Cancer Log-Likelihood

Topic Model	Dominant Mode	Topic Set	γ	τ	Levels	Mean	StDev	Gene Topics	Pathway Topics	Total Topics
Trees		Cartesian	0.5		2	-20,892.77	2,090.08	22.4	22.2	500.1
Trees		Cartesian	1		2	-20,891.20	2,091.51	26.6	25.6	683.7
Trees		Cartesian	2		2	-20,888.75	2,093.63	34.6	36.9	1,273.3
Trees		Cartesian	0.5		3	-20,896.20	2,088.46	77.7	77.5	6,021.0
Trees		Cartesian	1		3	-20,904.76	2,093.13	104.5	103.4	10,820.0
Trees		Cartesian	2		3	-20,904.62	2,093.29	159.2	151.1	24,055.1
Trees		Cartesian	0.5		4	-20,897.73	2,089.66	157.4	151.0	23,729.0
Trees		Cartesian	1		4	-20,895.72	2,094.83	225.8	229.0	51,709.1
Trees		Cartesian	2		4	-20,898.37	2,088.85	322.6	327.3	105,598.5
Trees		Cartesian	0.5		5	-20,901.75	2,093.03	256.7	233.5	59,938.1
Trees		Cartesian	1		5	-20,900.24	2,092.58	376.9	391.0	147,520.1
Trees		Cartesian	2		5	-20,906.93	2,091.26	551.1	563.1	309,852.2
PAM	Genes	Cartesian		10	2	-20,865.87	2,085.96	11	20	220
PAM	Genes	Cartesian		25	2	-20,868.94	2,093.25	26	50	1300
PAM	Genes	Cartesian		50	2	-20,877.08	2,089.14	51	100	5100
PAM	Genes	Cartesian		10	3	-20,887.15	2,094.10	21	30	630
PAM	Genes	Cartesian		25	3	-20,890.11	2,091.12	51	75	3825
PAM	Genes	Cartesian		50	3	-20,887.59	2,090.89	101	150	15150
PAM	Genes	Cartesian		10	4	-20,887.17	2,092.55	31	40	1240
PAM	Genes	Cartesian		25	4	-20,898.44	2,094.26	76	100	7600
PAM	Genes	Cartesian		50	4	-20,897.85	2,092.47	151	200	30200
PAM	Genes	Cartesian		10	5	-20,885.52	2,084.77	41	50	2050
PAM	Genes	Cartesian		25	5	-20,900.43	2,090.06	101	125	12625
PAM	Genes	Cartesian		50	5	-20,899.17	2,086.49	201	250	50250
PAM	Genes	Level		10	2	-20,850.34	2,090.04	11	20	220
PAM	Genes	Level		25	2	-20,853.48	2,100.40	26	50	1300
PAM	Genes	Level		50	2	-20,861.15	2,087.41	51	100	5100
PAM	Genes	Level		10	3	-20,905.25	2,091.74	21	30	630
PAM	Genes	Level		25	3	-20,907.18	2,091.60	51	75	3825
PAM	Genes	Level		50	3	-20,899.90	2,092.68	101	150	15150
PAM	Genes	Level		10	4	-20,898.41	2,097.51	31	40	1240
PAM	Genes	Level		25	4	-20,910.99	2,093.91	76	100	7600
PAM	Genes	Level		50	4	-20,908.62	2,086.51	151	200	30200
PAM	Genes	Level		10	5	-20,898.22	2,083.74	41	50	2050
PAM	Genes	Level		25	5	-20,909.32	2,100.66	101	125	12625
PAM	Genes	Level		50	5	-20,907.57	2,093.01	201	250	50250
PAM	Pathways	Cartesian		10	2	-20,868.18	2,089.82	20	11	220
PAM	Pathways	Cartesian		25	2	-20,874.56	2,091.57	50	26	1300
PAM	Pathways	Cartesian		50	2	-20,878.93	2,088.94	100	51	5100
PAM	Pathways	Cartesian		10	3	-20,873.40	2,093.43	30	21	630
PAM	Pathways	Cartesian		25	3	-20,884.98	2,091.50	75	51	3825
PAM	Pathways	Cartesian		50	3	-20,894.30	2,093.27	150	101	15150
PAM	Pathways	Cartesian		10	4	-20,869.99	2,091.49	40	31	1240
PAM	Pathways	Cartesian		25	4	-20,885.04	2,092.28	100	76	7600
PAM	Pathways	Cartesian		50	4	-20,899.85	2,095.19	200	151	30200
PAM	Pathways	Cartesian		10	5	-20,868.56	2,087.76	50	41	2050
PAM	Pathways	Cartesian		25	5	-20,897.56	2,089.86	125	101	12625
PAM	Pathways	Cartesian		50	5	-20,901.05	2,091.75	250	201	50250
PAM	Pathways	Level		10	2	-20,903.71	2,092.12	20	11	220
PAM	Pathways	Level		25	2	-20,850.25	2,088.73	50	26	1300
PAM	Pathways	Level		50	2	-20,862.70	2,092.73	100	51	5100
PAM	Pathways	Level		10	3	-20,886.34	2,090.20	30	21	630
PAM	Pathways	Level		25	3	-20,897.45	2,093.63	75	51	3825
PAM	Pathways	Level		50	3	-20,867.97	2,094.37	150	101	15150
PAM	Pathways	Level		10	4	-20,872.88	2,097.50	40	31	1240
PAM	Pathways	Level		25	4	-20,893.24	2,090.46	100	76	7600
PAM	Pathways	Level		50	4	-20,893.45	2,089.18	200	151	30200
PAM	Pathways	Level		10	5	-20,863.37	2,095.64	50	41	2050
PAM	Pathways	Level		25	5	-20,910.03	2,091.86	125	101	12625
PAM	Pathways	Level		50	5	-20,906.67	2,089.15	250	201	50250