Topics in Applied Statistics

by

Patrick M. LeBlanc

Department of Statistical Science
Duke University

Date:_____

Approved:

_____
David Banks, Advisor

_____
Li Ma, Advisor

_____
Amy Herring

_____
Anru Zhang

Dissertation submitted in partial fullfilment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2023

ABSTRACT

Topics in Applied Statistics

by

Patrick M. LeBlanc

Department of Statistical Science
Duke University

Date:_____

Approved:

_____
David Banks, Advisor

_____
Li Ma, Advisor

_____
Amy Herring

_____
Anru Zhang

An abstract of a dissertation submitted in partial fullfilment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2023

Abstract

One of the fundamental goals of statistics is to develop methods which provide improved inference in applied problems. This dissertation will introduce novel methodology and review state-of-the-art existing methods in three different areas of applied statistics. Chapter 2 focuses on modelling subcommunity dynamics in gut microbiome data. Existing methods ignore cross-sample heterogeneity in subcommunity composition; we propose a novel mixed-membership model which models cross-sample heterogeneity using the phylogenetic tree and as a result is robust to mispecifying the number of subcommunities. Chapter 3 reviews state-of-the-art methods in recommender systems, including collaborative filtering, content-based filtering, hybrid recommenders, and active recommender systems. Existing literature has focused primarily on bespoke applications; statisticians have an opportunity to build recommender system theory. Chapter 4 proposes a novel method of accounting for time-based design inconsistencies in Bayesian network meta-analysis models and discovers nonlinear time trends in the effectiveness of vancomycin as a MRSA treatment. Chapter 5 provides some concluding remarks.

Acknowledgements

I would like to thank my co-advisors David Banks and Li Ma for their guidance during my Ph.D. They generously shared their time and enthusiasm with me, offered invaluable advice, and encouraged my research interests. I would also like to thank the members of my co-advisors research groups, for sharing common interests, ideas, feedback, and collaborations. The members of my prelim and dissertation committees — Surya Tokdar, Jichun Xie, Amy Herring, and Anru Zhang — provided critical feedback, support, and guidance. I thank them for this. In addition, I would like to thank the Department of Statistical Science staff, in particular Lori Rauch and Nikki Scott who had to remind me of deadlines on more than one occasion. Their patience is appreciated.

Most of all, I would like to thank my parents and my fiancee. They have been loving and have offered support even in the most challenging times. I couldn't have done this without them.

Contents

List of Tables

List of Figures

1 Introduction

This dissertation is based off of three manuscripts written during the author's Ph.D. program. They all involve the development and use of statistical methods in different areas of application. Each manuscript has a corresponding chapter; the text appearing in the chapter will be largely identical to the text in the manuscript.

In Chapter 2, we propose a novel mixed-membership (MM) for gut microbiome data. MM models such as Latent Dirichlet Allocation (LDA) have been applied to microbiome compositional data to identify latent subcommunities of microbial species. These subcommunities are informative for understanding the biological interplay of microbes and for predicting health outcomes. However, microbiome compositions typically display substantial cross-sample heterogeneities in subcommunity compositions – that is, the variability in the proportions of microbes in shared subcommunities across samples – which is not accounted for in prior analyses. As a result, LDA can produce inference which is highly sensitive to the specification of the number of subcommunities and often divides a single subcommunity into multiple artificial ones. To address this limitation, we incorporate the logistic-tree normal (LTN) model into LDA to form a new MM model. This model allows cross-sample variation in the composition of each subcommunity around some "centroid" composition that defines the subcommunity. Incorporation of auxiliary Pólya-Gamma variables enables a computationally efficient collapsed blocked Gibbs sampler to carry out Bayesian inference under this model. By accounting for such heterogeneity, our new model restores the robustness of the inference in the specification of the number of subcommunities and

allows meaningful subcommunities to be identified. Chapter 2 is based off of LeBlanc and Ma [2022], and was done jointly with one of the author's co-advisors, Li Ma.

In Chapter 3, we review statistical methods for recommender systems. Recommender systems are the engine of online advertising. Not only do they suggest movies, music, or romantic partners, but they also are used to select which advertisements to show to users. We review the fundamentals of recommender system methodology: collaborative filtering leverages rating data to recommend items to users, while content-based filtering uses content descriptions of items to make recommendations [Park et al., 2012]. In practice, many recommender systems do not restrict themselves to one strategy and instead combine multiple strategies in order to improve performance—these types of approaches are collectively known as hybrid filtering [Adomavicius and Tuzhilin, 2005a]. We also review the emerging field of active recommender systems. Active recommender systems interact with the user and can mimic how humans operate by, e.g., asking the user questions. If someone asks a person for a book recommendation, that person will typically respond by asking "What kind of books do you like?" Despite its statistical nature, most research on recommender systems has been performed by computer scientists and researchers in industry and there is a corresponding lack of theory. Statisticians may be able to target this gap in the literature. The manuscript Chapter 3 was based off of was done jointly with one of the author's co-advisors, David Banks, as well as Linhui Fu, Mingyan Li, Zhengyu Tang, and Qiuyi Wu.

In Chapter 4, we propose a novel method to account for time-based design inconsistencies in Bayesian network meta-analysis (BNMA) models motivated by the

prevalence of methicillin-resistant *Staphylococus Aureus* (MRSA). The presence of MRSA in complicated skin and soft structure infections (cSSSI) is associated with greater health risks and economic costs to patients. There is concern that MRSA is becoming resistant to other "gold standard" treatments such as vancomycin, and there is disagreement about the relative efficacy of vancocymin compared to linezolid. There are several review papers employing BNMAs to investigate which treatments are best for MRSA related cSSSIs, but none address time-based design inconsistencies. This paper proposes a time-varying BNMA (tBNMA), which models time-varying treatment effects across studies using a Gaussian Process kernel. A dataset is compiled from nine existing MRSA cSSSI NMA review papers containing 58 studies comparing 19 treatments over 19 years. tBNMA finds evidence of a non-linear trend in the treatment effect of vancomycin—it became less effective than linezolid between 2002 and 2007, but has since recovered statistical equivalence. Chapter 4 is based off of LeBlanc and Banks [2023], and was done jointly with one of the author's co-advisors, David Banks.

2 Microbiome subcommunity learning with logistic-tree normal latent Dirichlet allocation

2.1 Introduction

The human gut microbiome is the genetic content of all bacteria, archaea, viruses, and eukaryotic microbes residing in the human gut and is commonly used to profile the composition of the gut microbiota. Advances in next-generation sequencing techniques have substantially reduced the cost of this approach and made it widely accessible. One cost-effective microbiome profiling strategy is based on targeting a single marker gene, the 16S ribosomal RNA (rRNA) gene, through amplicon-based sequencing [Li, 2015]. A more expensive, but more precise, approach is whole-genome shotgun metagenomic sequencing [Weber and Myers, 1997]. Traditionally, sequencing reads have been clustered into Operational Taxonomic Units (OTUs), which serve as the basic unit of microbial taxa. Recently, amplicon sequencing variants (ASVs) have come into wider use as they can achieve more precise characterization of microbial species and resolve the sample-specificity issue of the OTU [Callahan et al., 2017]. Our work is applicable to either method of characterizing microbial taxa; in the following we shall generically refer to the basic unit as ASVs.

Gut microbiome studies often involve highly heterogeneous samples due to the multitude of factors that can influence an individual's gut microbiota. A useful data analytical strategy for microbiome compositions is to sort microbiome samples into clusters characterized by particular compositional signatures. In the context of gut microbiome, these clusters are called "enterotypes" [Siezen and Kleerebezem, 2011]

and are associated with health outcomes [Del Chierico et al., 2014]. One of the most popular microbiome clustering methods is the Dirichlet-multinomial mixture (DMM) model [Holmes et al., 2012, Nigam et al., 2000], which uses a hierarchical structure to allow within-cluster cross-sample variability in subcommunity compositions. However, the DMM is too restrictive to realistically characterize the within-cluster cross-sample variance in microbiome data [Tang et al., 2018, Wang and Zhao, 2017] as it uses a single scalar parameter to characterize the entire covariance structure across all microbial taxa. More general methods have recently been introduced to alleviate, though not eliminate, this limitation through the use of Dirichlet-tree models [Dennis III, 1991, Wang and Zhao, 2017].

Such clustering analysis, however, makes the implicit assumption that each microbiome sample must belong to a *single* signature "community" characterized by the cluster centroid. This assumption is often unrealistic and overly restrictive for complex environments such as the gut microbiome [Holmes et al., 2012, Mao et al., 2020]. Recent developments embrace the more relaxed biological hypothesis that the ASVs characterizing a microbiota sample hail from a combination of multiple microbial "clusters", or more precisely "subcommunities".

Mixed-membership (MM) models are generalizations of clustering models that provide a generative modeling framework for data involving subcommunity structure as they allow each sample to be composed of multiple subcommunities. Sankaran and Holmes [2019] applied the most well-known MM model, latent Dirichlet allocation (LDA), to microbiome profiling. Earlier, Shafiei et al. [2015] and Deek and Li [2019] proposed variations of LDA accounting for environmental factors and inflated zero-

counts, respectively, in the microbiome context.

The key motivation for our paper is the observation that existing MM models such as LDA and its variations—originally developed for other contexts such as topic modeling [Blei et al., 2003] and population genetics [Pritchard et al., 2000] — do not incorporate key features of microbiome compositions. Most notably, they assume that a microbial subcommunity's composition must remain *exactly the same* across all samples. This is unrealistic in the vast majority of microbiome studies collected from diverse environments such as the gut where samples often possess large heterogeneities [Jeganathan and Holmes, 2021, Tang et al., 2018]. It is interesting to note that such heterogeneity has been well-recognized in clustering models for microbiome data [Holmes et al., 2012, Mao et al., 2020], but has been largely ignored in existing MM models. Additionally, choosing the number of subcommunities for LDA is not trivial in the presence of cross-sample heterogeneity, and LDA-based approaches often lead to overestimates in the number of subcommunities in microbiome applications [Fukuyama et al., 2022].

We introduce a generalization of LDA that aims to appropriately incorporate cross-sample heterogeneity, or "random effects", in microbiomal subcommunity compositions due to unmeasured sources, thereby leading to more accurate identification of subcommunities in MM models. Our approach takes advantage of the availability of a natural tree structure relating the microbial taxa—the phylogenetic tree—which allows us to decompose the compositional vector into a collection of binomial observations on the tree nodes. This transform serves two purposes. First, it allows us to model the heterogeneity by modeling the vector of log-odds transforms of the binomial

probabilities at each node as Gaussian. By modeling the subcommunity compositions as realizations from this logistic-tree normal (LTN) [Wang et al., 2021b] distribution, we are able to impose constraints on the underlying covariance structure to ensure the identifiability of the subcommunities. A second purpose of the tree-based transform is computational. By utilizing the Pólya-Gamma (PG) data augmentation technique [Polson et al., 2013], Bayesian inference under the resulting MM model can be readily accomplished through fully conjugate collapsed blocked Gibbs sampling. We term our new model logistic-tree normal latent Dirichlet allocation (LTN-LDA).

Several other relevant prior works are worth mentioning. Graph-Sparse LDA [Doshi-Velez et al., 2015] also incorporates random effects from subcommunity-to-subcommunity using a tree structure. However, in the context of microbiome compositions, it would assume that every node of the tree is an ASV which can occur in a sample and is thus incompatible with the phylogenetic tree. Other tree-based MM methods include Tam and Schultz [2007], which uses trees to model the abundance of subcommunities in samples, and Andrzejewski et al. [2009], which uses mixtures of trees to model subcommunity composition by explicitly modelling which ASVs must co-occur and which cannot.

In the following, we will briefly review the LDA and LTN models before introducing the LTN-LDA model. We will augment the LTN-LDA model using a class of auxiliary Pólya-Gamma variables [Polson et al., 2013] and present a collapsed blocked Gibbs sampler for carrying out fully Bayesian inference. We will demonstrate in simulations that, in the presence of cross-sample heterogeneity, inference by LTN-LDA is robust with respect to overspecifying the number of subcommunities while infer-

ence by LDA can be highly sensitive to the choice of the number of subcommunities. We apply LTN-LDA to the dataset of Dethlefsen and Relman [2011], which has been used for demonstrating MM models in the microbiome settings [Sankaran and Holmes, 2019], and compare our results to LDA.

2.2 Methods

2.2.1 Latent Dirichlet allocation

Let there be D samples consisting of counts of V unique ASVs indexed by $1, 2, \ldots, V$. For sample d, let $\mathbf{x}_d = (x_{d,1}, \ldots, x_{d,V})$ be the vector of ASV counts such that $x_{d,v}$ is the total count for ASV v in sample d. Let $N_d = \sum_{v=1}^{V} x_{d,v}$ be the sum of counts in sample d, which is determined by the sequencing depth. Subcommunities are defined to be collections of ASVs that co-occur in samples at given relative proportions. An ASV can occur in multiple subcommunities at various abundances and the key assumption underlying an MM model, in contrast to a clustering model, is that different instances (i.e., different sequencing reads) of the same ASV in a sample can arise from the participation of that ASV in multiple microbial subcommunities. Key parameters of interest in MM models are subcommunity abundance, i.e., the proportions of the various subcommunities in each sample, and subcommunity composition, i.e., the proportions of the ASVs in each subcommunity.

To describe LDA, it is convenient to introduce categorical indicators for each read and its associated subcommunity identity. For $d = 1, 2, \ldots, D$, let $\mathbf{w}_d$ be a vector $\mathbf{w}_d = (w_{d,1}, \ldots, w_{d,N_d})$ where $w_{d,n} \in \{1, 2, \ldots, V\}$ is the categorical indicator of the ASV associated with the nth read in the sample. We refer to the elements $w_{d,n}$ in

this vector as "tokens" to draw analogy with topic modelling. There, each token is a word in a document; here, each token corresponds to a read in a sample. We also note that $x_{d,v} = \sum_{n=1}^{N_d} \mathbf{1}_{\{w_{d,n}=v\}}$.

Let $\boldsymbol{\phi}_d = (\phi_d^1, \phi_d^2, \ldots, \phi_d^K)' \in \Delta^{K-1}$, where Δ^S is the S-dimensional simplex, be the subcommunity abundance vector. That is, ϕ_d^k represents the relative abundance of subcommunity k in sample d, and so $\boldsymbol{\phi}_d$ specifies the categorical distribution of each token over the K underlying subcommunities in sample d. Let $z_{d,n}$ represent the subcommunity from which the n^{th} token in sample d arises from and let $\mathbf{z}_d$ be the vector all such assignments for sample d. Also, let $\boldsymbol{\beta}_k = (\beta_k^1, \beta_k^2, \ldots, \beta_k^V)' \in \Delta^{V-1}$ be the subcommunity composition for subcommunity k. That is, $\boldsymbol{\beta}_k$ gives the relative proportions of the V unique ASVs in subcommunity k. For $d = 1, \ldots, D$ and $n = 1, \ldots, N_d$ and while $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are hyperparameters, the LDA model (Figure 1a) [Blei et al., 2003] is then

$$w_{d,n} \,|\, z_{d,n}, \boldsymbol{\beta}_{z_{d,n}} \overset{\text{ind}}{\sim} \text{Cat}(\boldsymbol{\beta}_{z_{d,n}}) \quad z_{d,n} \,|\, \boldsymbol{\phi}_d \overset{\text{ind}}{\sim} \text{Cat}(\boldsymbol{\phi}_d)$$

$$\boldsymbol{\phi}_d \,|\, \boldsymbol{\alpha} \overset{\text{iid}}{\sim} \text{Dir}(\boldsymbol{\alpha}) \qquad \boldsymbol{\beta}_k \,|\, \boldsymbol{\gamma} \overset{\text{iid}}{\sim} \text{Dir}(\boldsymbol{\gamma}).$$

Though LDA can be applied in the microbiome context [Sankaran and Holmes, 2019], it does not account for cross-sample heterogeneity in subcommunity composition. In particular, it assumes that the $\boldsymbol{\beta}_k$ are the *exact same* across all samples. This is inconsistent with the empirical behavior of the microbiome where large cross-sample heterogeneities exist [Holmes et al., 2012]. LDA thus tends to interpret cross-sample heterogeneity as the presence of additional subcommunities.

2.2.2 Incorporating cross-sample heterogeneity

We shall enrich the LDA framework to allow the subcommunity compositions to vary across samples. There are several hierarchical models for microbiome compositions such as the Dirichlet-Multinomial (DM) model [Holmes et al., 2012, Nigam et al., 2000] and Aitchinson's log-ratio based normal (LN) models [Aitchison, 1982], which could be embedded into LDA for this purpose. However, the DM is highly restrictive in its ability to characterize the underlying cross-sample variability as the Dirichlet distribution has only one scalar variance parameter, while the LN models are computationally challenging due to lack of conjugacy to the multinomial sampling model. To resolve these difficulties, we adopt the recently introduced logistic-tree normal (LTN) model [Wang et al., 2021b]. In particular, we will show that the LTN model can be embedded into the LDA model to accommodate cross-sample heterogeneity and that posterior inference can be accomplished through simple collapsed blocked Gibbs sampling using a data-augmentation technique called Pólya-Gamma augmentation. Moreover, since the adoption of the LTN model requires specifying a dyadic partition tree on the ASVs, the phylogenetic tree relating the taxa is a natural choice.

2.2.3 The phylogenetic tree

Let $\mathcal{T}$ denote a phylogenetic tree capturing genetic similarities between the observed ASVs. The leaf nodes in the tree correspond to the observed ASVs in the data set. Each interior node is the inferred common ancestral taxon for the ASVs lying in the corresponding descendant subtree at the node. Each node (or taxon) A in the phylogenetic tree $\mathcal{T}$ can be represented by the collection of its descendant

ASVs. In particular, each leaf node A contains a single ASV, whereas each internal node A contains multiple ASVs. In the following, we let $\mathcal{I}$ be the set of internal nodes. Throughout this work, we shall assume that the phylogenetic tree is rooted and binary in the sense that each $A \in \mathcal{I}$ has exactly two child nodes (i.e., direct descendants): let A_l and A_r be the left and right children of A, respectively.

2.2.4 The logistic-tree normal model

We shall adopt the logistic-tree normal (LTN) model [Wang et al., 2021b] as the sampling model for the ASV count distribution within each subcommunity. LTN is a distribution on a tree-based log-odds transform of the categorical probabilities $\boldsymbol{\beta} = (\beta^1, \beta^2, \ldots, \beta^V)' \in \Delta^{V-1}$. Specifically, given the phylogenetic tree $\mathcal{T}$, for each interior node we define $\theta(A) = \frac{\sum_{v \in A_l} \beta^v}{\sum_{v \in A} \beta^v}$: the probability that a token belongs to an ASV in A_l given that it belongs to an ASV in A. The collection of $\theta(A)$ on all $A \in \mathcal{I}$ gives an equivalent reparametrization of $\boldsymbol{\beta}$. In Figure 2 we plot an example phylogenetic tree over 6 ASVs with labelled nodes (Figure 2a) and with labelled β^v and $\theta(A)$ (Figure 2b) to demonstrate the link between the β^v and the $\theta(A)$.

After taking the logit transform of these binomial probabilities on the tree nodes, $\psi(A) = \log \frac{\theta(A)}{1-\theta(A)}$, let $\boldsymbol{\psi}$ be the vector of $\psi(A)$ with respect to an ordering on the p internal nodes of $\mathcal{T}$. LTN is simply a Gaussian model on these tree-based log-odds: $\boldsymbol{\psi} \,|\, \boldsymbol{\mu}, \Sigma \overset{\text{iid}}{\sim} \text{MVN}(\boldsymbol{\mu}, \Sigma)$ for some mean $\boldsymbol{\mu}$ and covariance Σ parameters that specify the overall average profile of the count distribution and the cross-sample variability.

Posterior computation under LTN, which we will describe later, relies on an equivalent representation of the categorical sampling on the leaves of the tree as a collection of sequential binomial experiments on the internal nodes of the tree. Specifically,

generating a categorical draw from the probability vector $\boldsymbol{\beta}$ can be achieved by sequentially "dropping" the token from top-to-bottom along the phylogenetic tree: at each node determine whether the token belongs to the left or right child node with probabilities $\theta(A)$ and $1-\theta(A)$, respectively. More formally, for each node $A \in \mathcal{T}$, we use $y(A)$ to denote the total counts associated with the ASVs descended from node A. That is, $y(A) = \sum_{n=1}^{N} 1_{w_n \in A}$ where w_n represents the nth count. Generating a multinomial count vector with probabability $\boldsymbol{\beta}$ can be achieved by sequentially drawing $y(A_l)$ given $y(A)$ from $\text{Bin}(y(A_l) \,|\, y(A), \theta(A))$. Putting the pieces together, and letting $\text{expit}(\psi) = 1/(1+e^{-\psi})$, LTN is the following generative model: for all internal nodes $A \in \mathcal{T}$,

$$y(A_l) \,|\, y(A), \psi(A) \overset{\text{ind}}{\sim} \text{Bin}(y(A), \theta(A) = \text{expit}(\psi(A))) \quad \text{and} \quad \boldsymbol{\psi} \,|\, \boldsymbol{\mu}, \Sigma \overset{\text{ind}}{\sim} \text{MVN}(\boldsymbol{\mu}, \Sigma).$$

2.2.5 LTN-LDA

We incorporate the LTN model into LDA to allow cross-sample heterogeneity in subcommunity compositions. The resulting model is termed logistic-tree normal latent Dirichlet allocation (LTN-LDA). Specifically, for $d = 1, \ldots, D$, $k = 1, \ldots, K$, $n = 1, \ldots, N_d$, and $A \in \mathcal{I}$, where the subscripts d, k, and n indicate the corresponding quantities associated with the dth sample, kth subcommunity, and nth read, the model is as follows

$$y_{d,k}(A_l) \,|\, y_{d,k}(A), \psi_{d,k}(A) \overset{\text{ind}}{\sim} \text{Bin}(y_{d,k}(A), \text{expit}(\psi_{d,k}(A)))$$

$$y_{d,k}(A) = \sum_{n=1}^{N_d} 1_{z_{d,n}=k} 1_{w_{d,n} \in A} \qquad z_{d,n} \,|\, \boldsymbol{\phi}_d \overset{\text{ind}}{\sim} \text{Cat}(\boldsymbol{\phi}_d)$$

$$\boldsymbol{\phi}_d \,|\, \boldsymbol{\alpha} \overset{\text{iid}}{\sim} \text{Dir}(\boldsymbol{\alpha}) \qquad \boldsymbol{\psi}_{d,k} \,|\, \boldsymbol{\mu}_k, \Sigma_k \overset{\text{ind}}{\sim} \text{MVN}(\boldsymbol{\mu}_k, \Sigma_k),$$

$$\boldsymbol{\mu}_k \,|\, \boldsymbol{\mu}_0, \Lambda_0 \overset{\text{iid}}{\sim} \text{MVN}(\boldsymbol{\mu}_0, \Lambda_0) \qquad \Sigma_k \,|\, \text{G} \overset{\text{iid}}{\sim} \text{G}$$

Note that we also endowed the subcommunity mean $\boldsymbol{\mu}_k$ and covariance Σ_k, with corresponding priors $\text{MVN}(\boldsymbol{\mu}_0, \Lambda_0)$ and G, which will be specified later. Figure 1*b* provides the graphical model representation for this full hierarchical model. The key distinction between LTN-LDA and LDA is that LTN-LDA uses a hierarchical kernel, namely LTN, to model cross-sample heterogeneity. In particular, the composition in sample d of subcommunity k is determined by $\boldsymbol{\psi}_{d,k}$ and is explicitly allowed to vary across samples.

Without additional constraints on the high-dimensional covariance matrices for each subcommunity, Σ_k, the model is too flexible [Haffari and Teh, 2009], and can become unidentifiable. Additional structural constraints serving the purpose of regularization on the covariance structure are thus necessary and so we assume that Σ_k is a diagonal covariance matrix. An LTN distribution with diagonal covariance is similar in distributional properties to a Dirichlet-tree multinomial (DTM) distribution [Dennis III, 1991, Wang and Zhao, 2017] but is computationally more efficient because there are no known conjugate priors for the mean and variance parameters under the DTM model. While this limitation is manageable when the DTM is used as a standalone model or the top layer in a hierarchical model, when embedded as a

kernel within an MM model such as LDA the incurred numerical computational cost becomes prohibitive. For more details, see Supporting Information A.1.

While the covariance constraint may appear strong, we note that the dependence among the tree-based log-odds ratios is generally much weaker than the complex dependence structure among the ASV counts themselves. In a sense, the tree-based log-odd transform of the abundance vectors "decorrelates" the data. For the interested reader, this decorrelation phenomenon is analogous to the so-called "whitening" effects in wavelet analysis [Nason, 2008], as the dyadic tree transform we incorporate here is the counterpart of Haar-wavelet transform on functions. In Supporting Information Section A.2 we investigate the effects of relaxing the diagonal covariance to a blocked diagonal covariance, and the results show that the additional sophistication does not lead to noticeable improvement in the inference.

Aside from the diagonal covariance, we also assume that the amount of variability for each node depends on that node's distance to the bottom (i.e., leaf) level of the tree. In particular, we assume that taxa close to the bottom of the phylogenetic tree have larger cross-sample variability in the corresponding log-odds ratio than those which are distant. This is motivated by the biological intuition that taxa close to each other on deep levels of the phylogenetic tree tend to have comparable functionality; the relative proportions of such taxa thus often display elevated levels of variance [Jeganathan and Holmes, 2021].

Specifically, let $|A|$ measure the distance of A from the leaf level by denoting the number of leaves descended from node A. For $i = 1, \ldots, p$, $k = 1, \ldots, K$, $C \in \mathbb{N}$ (a tuning parameter), and $\boldsymbol{\tau}_k = (\tau_k^1, \ldots, \tau_k^p)$, the prior we adopt has the form $\Sigma_k \,|\, \boldsymbol{\tau}_k =$

$\text{diag}(\boldsymbol{\tau_k})$ where

$$\tau_k^i \,|\, a_1, a_2, b \overset{\text{iid}}{\sim} \begin{cases} \text{IG}(a_1, b) & |A_i| \geq C \\ \text{IG}(a_2, b) & |A_i| < C \end{cases}$$

We default to $(a_1, a_2, b) = (10^4, 10, 10)$ and note that while we still refer to the $\psi_{d,k}$ as being drawn from a multivariate normal distribution, we have $\psi_{d,k}^i \,|\, \mu_k^i, \tau_k^i \overset{\text{iid}}{\sim} \text{N}(\mu_k^i, \tau_k^i)$.

This choice of priors ensures conjugate updating and avoids identifiability issues. Further, it partitions the internal nodes of the tree in two: we shall refer to these sets as the upper tree $\mathcal{U} = \{A \in \mathcal{I} : |A| \geq C\}$ and the lower tree $\mathcal{L} = \{A \in \mathcal{I} : |A| < C\}$. In $\mathcal{U}$, the hyperparameters a_1 and b are such that the τ_k^i will be small and the $\psi_{d,k}^i$ will vary little around μ_k^i; in $\mathcal{L}$, the hyperparameters a_2 and b are such that the τ_k^i are allowed to be large and the $\psi_{d,k}^i$ can vary significantly across samples. This implies that if A_c is the child of A, and $A_c \in \mathcal{L}$ but $A \in \mathcal{U}$, then all ASVs descended from A_c can substitute for each other across samples in a given subcommunity. We call sets of ASVs which are allowed to substitute for each other substitution sets. All ASVs are either part of a substitution set or singletons. The tree structure is critical to how LTN-LDA models cross-sample heterogeneity, and we include an analysis on the robustness to misspecified trees in Supporting Information A.3.

2.2.6 Bayesian inference by collapsed blocked Gibbs sampling

While the LTN-LDA model is not conditionally conjugate by itself, one can restore conjugacy by introducing a class of Pólya-Gamma latent variables [Polson et al., 2013] $v_{d,k}(A)$ — one for each interior node A — which are independent of $y_{d,k}(A_l)$ conditioned on $y_{d,k}(A)$ and $\psi_{d,k}(A)$: $v_{d,k}(A) \,|\, y_{d,k}(A), \psi_{d,k}(A) \sim \text{PG}(y_{d,k}(A), \psi_{d,k}(A))$. The

full conditional for $\psi_{d,k}(A)$ is then proportional to $\exp\left((y_{d,k}(A_l) - \frac{y_{d,k}(A)}{2})\psi_{d,k}(A) - \frac{v_{d,k}(A)\psi_{d,k}(A)^2}{2}\right)$, which takes a quadratic form in the exponent and thus is conjugate to the Gaussian model on $\psi_{d,k}(A)$. The graphical model for LTN-LDA with the Pólya-Gamma variables is presented in Figure 1*c*. To speed up the sampling of Pólya-Gamma variables we adopt an approximate sampler proposed by Glynn et al. [2019] for $y_{d,k}(A) \geq 30$. Further, we integrate $\boldsymbol{\phi}_d$ out of the sampling model to improve convergence as in Griffiths and Steyvers [2004]. The algorithm scales linearly with D, K, V, and N_d; for details, see Supporting Information A.4.

2.3 Numerical experiments

2.3.1 Robustness in choosing the number of subcommunities

The true number of subcommunities K in a given dataset is typically unknown and it is common to treat K as a tuning parameter. However, for data with large cross-sample heterogeneity such as microbiome data, intuition suggests that a model assuming zero heterogeneity will confuse sample-specific variation around a subcommunity mean with the presence of additional subcommunities. This results in difficulty estimating K and inference sensitive to K; indeed, LDA encounters both of these difficulties [Fukuyama et al., 2022].

To verify this intuition, we generated data from a known LTN-LDA model which induces cross-sample heterogeneity. In particular, we simulated $D = 50$ samples, and $N_d = 10,000$ reads per sample; we set $\alpha = 1$, $\mu = 0$, $\Lambda = I$, $a_1 = 10^4$, $a_2 = b = 10$, and $(K, C) = (4, 5)$. The underlying phylogenetic tree is presented in Supporting Information A.5: there are $V = 49$ ASVs. We then contrasted LDA and LTN-LDA

by running Gibbs samplers on the data generated above with $K \in \{4, 5, 7, 10\}$ and $C = 5$. In the left part of Figure 3, we plot the posterior means of the subcommunity abundances $\boldsymbol{\phi}_d$ for both LDA and LTN-LDA. We corrected for label switching and estimated the $\boldsymbol{\phi}_d$ as in Griffiths and Steyvers [2004].

With K set to truth, LDA performs comparably to LTN-LDA in estimating the true values of $\boldsymbol{\phi}_d$; however, as we increase K, the inference provided by LDA worsens. While it still recovers the abundances for subcommunities 1 and 2, it does a worse job at recovering subcomunities 3 and 4. Moreover, LDA detects the presence of additional subcommunities which do not exist in the true generative model. LTN-LDA, in contrast, is remarkably stable when K is overspecified. No matter the modelled value of K, it detects the four true subcommunities with approximately the same abundances while estimating that additional subcommunities have little abundance. For $K = 10$, we plotted the subcommunity compositions on the right part of Figure 3. (This figure appears in color in the electronic version of this article, and any mention of color refers to that version.) For LTN-LDA, distributions for the $\boldsymbol{\beta}_{d,k}$ are in blue and the $\boldsymbol{\beta}_k$ are in red; the LDA $\boldsymbol{\beta}_k$ distributions are in black. LTN-LDA finds moderate levels of cross-sample heterogeneity in subcommunity 2, and a high levels in samples 3 and 4.

These figures imply that LDA is able to recover the subcommunity abundances only for those subcommunities with low cross-sample heterogeneity. LDA fails to recover the subcommunity abundances for those subcommunities with high cross-sample heterogeneity, mistaking heterogeneity for additional subcommunities. In effect, LDA splits true heterogeneous subcommunities into many smaller subcommunities with no

heterogeneity and ASVs which ought to belong in the same subcommunity are separated. LTN-LDA, on the other hand, provides stable and accurate inference as the modelled K increases. This thus confirms our intuition about the behavior of LDA in the presence of cross-sample heterogeneity.

2.3.2 Predictive scoring as a device for choosing tuning parameters

While incorporating cross-sample heterogeneity enhances the robustness of LTN-LDA to overspecifying the number of subcommunities, it is still useful to have a generally applicable strategy for setting the tuning parameters for LTN-LDA: K and C. One option is to use out-of-sample predictive performance to identify suitable choices of the tuning parameters. A popular performance measure for MM models is perplexity [Wallach et al., 2009]: a transform of out-of-sample predictive likelihood such that lower perplexity is preferred.

We thus implement the simple strategy of computing the average out-of-sample perplexity score for different choices of (K, C) and examine whether that can lead to a practical way of choosing these parameters. We will also examine whether this strategy could be adopted for models without cross-sample heterogeneity, namely LDA, to alleviate their limitations. We follow the procedure in Section 5.1 of Wallach et al. [2009] for computing the perplexity for LDA, and generalize that strategy to LTN-LDA. For details, see Supporting Information A.6. We generated 200 simulated datasets. In each, there are $D = 50$ samples and $N_d = 10,000$ counts per sample; we set $\alpha = 1$, $\mu = 0$, $\Lambda = I$, $a_1 = 10^4$, $a_2 = b = 10$, and $(K, C) = (4, 5)$. For each dataset, we also generate a test set of the same size where the sample specific parameters are generated using $\alpha = 1$ and the training set's $\boldsymbol{\mu}_k$ and Σ_k.

Fixing C to truth, we varied K and computed average perplexity for LDA and LTN-LDA in Figure 4(a). There are three main observations: (i) LTN-LDA significantly outperforms LDA for K near truth, (ii) the perplexity curve for LTN-LDA decreases until it stabilizes at the true value of K, (iii) the perplexity curve for LDA continues to decrease as the modelled K is increased past its true value. The main reason for the difference is that LDA interprets the presence of cross-sample heterogeneity as extra subcommunities and so finds as many subcommunities as are modelled. While this improves out-of-sample predictive performance, it does not improve inference on the underlying truth. Thus, using perplexity to select the modelled number of subcommunities for LDA is a poor method if there is significant cross-sample heterogeneity. LTN-LDA is more robust and parsimonious in its representation of the data because it incorporates cross-sample heterogeneity in subcommunity compositions.

Fixing K to truth, we computed average perplexity for LTN-LDA as we varied C in Figure 4(b). The perplexity curve decreases until it stabilizes at the true value of C. In addition to perplexity, we also computed the L_2 distances between the posterior mean estimates and the true values for the $\boldsymbol{\phi}_d$, $\boldsymbol{\beta}_{d,k}$, and $\boldsymbol{\beta}_k$ distributions (Figure 4(c)). Unlike the perplexity curves, the L_2 distances are lowest around $C = 5$ and increase as C increases. Thus, if the modelled value of C is increased too far above truth, inference becomes unreliable.

The above results suggest a simple two-stage strategy for choosing (K, C) using perplexity. First, let (K, C) vary jointly on a grid and use cross-validation to compute the average perplexity, giving K perplexity curves over C. Set C to be the inflection point in these curves. Second, vary K and set the value of K to be the inflection

point of the resulting perplexity curve. Note that this strategy may fail for LDA: as our numerical examples show below, due to the lack of cross-sample heterogeneity in LDA, the perplexity score generally continues to improve as one increases the number of subcommunities beyond truth. This in turn leads to misleading inference on subcommunity abundance and composition.

2.4 Evaluation on a microbiome study

We apply LTN-LDA to identify subcommunity dynamics in the dataset of Dethlefsen and Relman [2011], which has been previously investigated by Sankaran and Holmes [2019] using LDA. The data includes gut microbiome samples of three patients who were administered two five-day courses of ciprofloxacin over a ten-month span. We focus on the 54 samples from patient F, each consisting of approximately $10,000$ reads. Ciproflaxin was administered during samples 12-23 and 41-51. There are $2,852$ unique ASVs in the dataset; we merged ASVs into taxa at the finest known level and pruned all taxa which did not total at least 100 sequencing reads. This left 44 taxa comprising 99.86 percent of the original counts. The resulting phylogenetic tree is included in Supporting Information A.7.

We implemented the strategy outlined above to choose tuning parameters. In particular, we implement a 4-fold cross-validation letting K vary in $\{2, 3, \ldots, 8\}$ and C in $\{1, 2, \ldots, 21\}$. The resulting K perplexity curves over C are presented in Figure 4(d). The inflection point in the curve appears at $C = 8$. Setting $C = 8$ and varying K gives the results in Figure 4(e); for comparison, we also applied LDA to the data over varying K. LTN-LDA has strictly lower perplexity than LDA, indicating that

there are significant levels of cross-sample heterogeneity in the dataset. Moreover, LTN-LDA experiences a noticeable inflection point (near $K = 5$) in contrast to LDA whose perplexity decays slowly.

We now present more detailed analysis for LTN-LDA and LDA with $C = 8$. For $K \in \{3, 4, 7\}$ we plotted the subcommunity abundance on the left side of Figure 5, after manually correcting for label switching. The grey regions indicate periods of ciproflaxin treament. The subcommunities found by LTN-LDA are remarkably stable as K changes. Subcommunities 1, 2, and 3 have almost the exact same abundance, and additional subcommunities have minimal abundance. LDA, however, finds as many subcommunities as are modelled: it will split a heterogenous subcommunitiy into multiple subcommunities with no heterogeneity. For $K = 7$, we plotted the ASV-subcommunity distributions on the right side of Figure 5. Distributions for the $\boldsymbol{\beta}_{d,k}$ are in blue, the $\boldsymbol{\beta}_k$ in red, and the LDA distributions in black. The 3 most prevalent ASVs in each subcommunity are presented in Figure 6 for LDA and LTN-LDA. These demonstrate that LTN-LDA finds significant levels of cross-sample heterogeneity and subcommunities with meaningfully different compositions than LDA.

LTN-LDA thus provides two major advantages. First, LTN-LDA is more robust with respect to modelling differing numbers of subcommunities than LDA. This is similar to our simulations and indicates that LTN-LDA better accounts for the cross-sample heterogeneity in the data than does LDA. Moreover, the three subcommunities found by LTN-LDA are biologically interpretable. The first subcommunity is composed mostly of Lachnospiraceae and Ruminococcaceae and displays significant levels of cross-sample heterogeneity, indicating that LTN-LDA has found these two ASVs

can substitute for each other. Haak et al. [2018] found this phenomena in humans undergoing ciproflaxin treatment. LTN-LDA can thus learn when two ASVs substitute for each other across samples from the data, with no prior knowledge. The second subcommunity, composed mainly of Bacteroides, increases in abundance during the antibiotic treatments. Studies in mice [Zhu et al., 2020] and humans [Stewardson et al., 2015] indicate that the abundance of Bacteroides increases during ciproflaxcin treatment. The third subcommunity has a small spike in abundance only on the first day of the second antibiotic course, and is composed mostly of Dialister and Veillonella. Ciproflaxin has been shown to be effective against Dialister [Morio et al., 2007] which may explain the decrease in this subcommunity after treatment began.

2.5 Discussion

We have proposed a novel mixed-membership model which seeks to appropriately incorporate cross-sample heterogeneity in subcommunity compositions: a characteristic of the data prevalent in most microbiome studies. By incorporating the logistic-tree normal model for the sample-specific compositions of each subcommunity, we explicitly allow the composition of subcommunities to vary across samples. We have shown that incorporating cross-sample heterogeneity into MM models can lead to substantially improved inference over models which assume zero cross-sample heterogeneity. LTN-LDA is substantially more robust than LDA with respect to overspecifying K and significantly outperforms LDA in terms of predictive performance. Moreover, perplexity can be a useful device to set the tuning parameters for LTN-LDA but not for LDA. Posterior computation on LTN-LDA can proceed through collapsed blocked

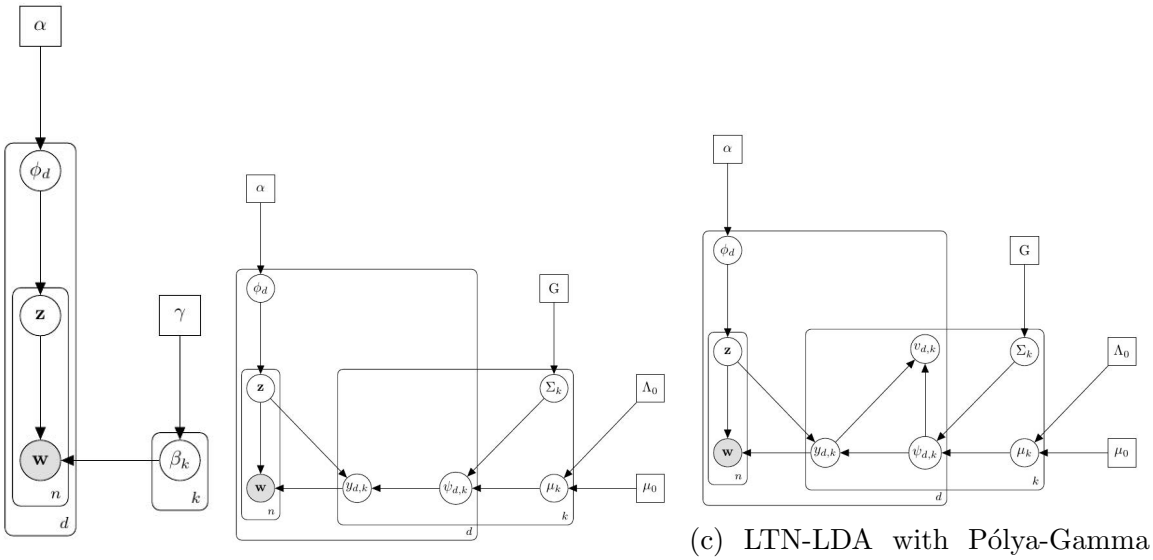
Gibbs-sampling with the assistance of Pólya-Gamma augmentation, and as such implementation for LTN-LDA is convenient. Moreover, LTN-LDA is a fully Bayesian model and the Gibbs sampler allows for posterior uncertainty quantification.

In comparison to LDA, LTN-LDA incorporates two new features: the tree structure and the random effects allowing cross-sample heterogeneity. The tree structures provides guidance on how to parsimoniously model the random effects without causing non-identifiability. We carried out an additional numerical experiment that shows that using the tree structure as a way to parametrize the model without adding random effects does not lead to improved inference. For a more detailed discussion see Supporting Information A.8. While LTN-LDA relies on the tree structure to incorporate random effects, we note there are several alternative approaches to incorporating random effects in microbiome compositions [Grantham et al., 2017, Ren et al., 2020a, Zhang and Lin, 2019]. In principle it is possible to incorporate random effects without a tree structure in the MM model.

Like other unsupervised learning methods, LTN-LDA is unable to differentiate between different scenarios giving rise to the same sampling distributions. That is, LTN-LDA, or any other models for that matter, cannot distinguish between multiple subcommunities and a single over-dispersed one if the two give rise to the same sampling distributions. Domain knowledge is necessary to identify such possibilities; traditionally, there are two strategies to incorporate such domain knowledge. The first is through modeling assumptions, such as modelling how large the single-subcommunity dispersion is through the hyperpriors on the τ_k^i. The other strategy is using a decision theoretic formulation that introduces certain loss functions to carry

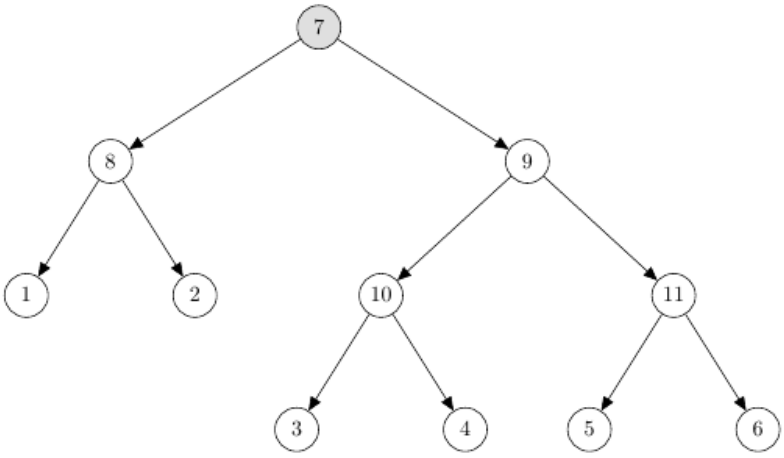
out post-hoc merging of the identified topics.

Moreover, we believe that the idea of incorporating cross-sample heterogeneity in MM models could be valuable beyond the context of microbiome compositions. In topic models, for example, one might expect different authors to write on the same topic using different vocabulary. LTN-LDA has the potential to be applicable to these other contexts as well, though the immediate challenge is finding an appropriate tree structure.

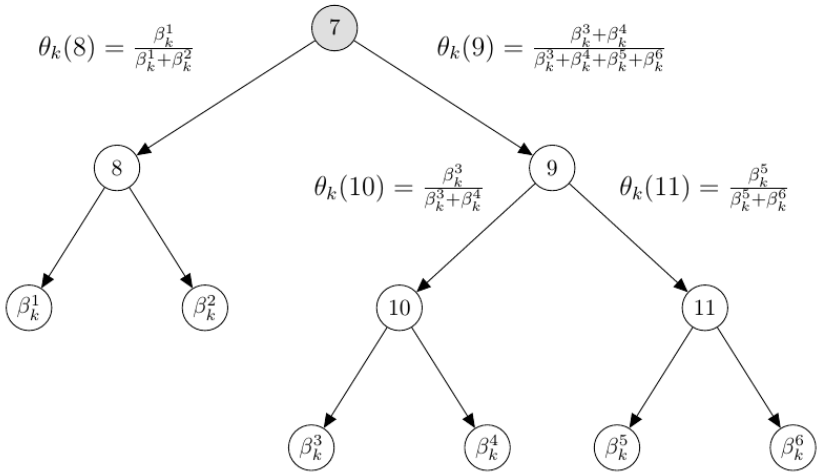


(a) LDA (b) LTN-LDA (c) LTN-LDA with Pólya-Gamma Variables

Figure 1: Graphical model representations for LDA and LTN-LDA.



(a) Notation for nodes



(b) Present θ_k and β_k

Figure 2: An example phylogenetic tree for 6 ASVs and the graphical relationship between μ_k and $\psi_{d,k}$.

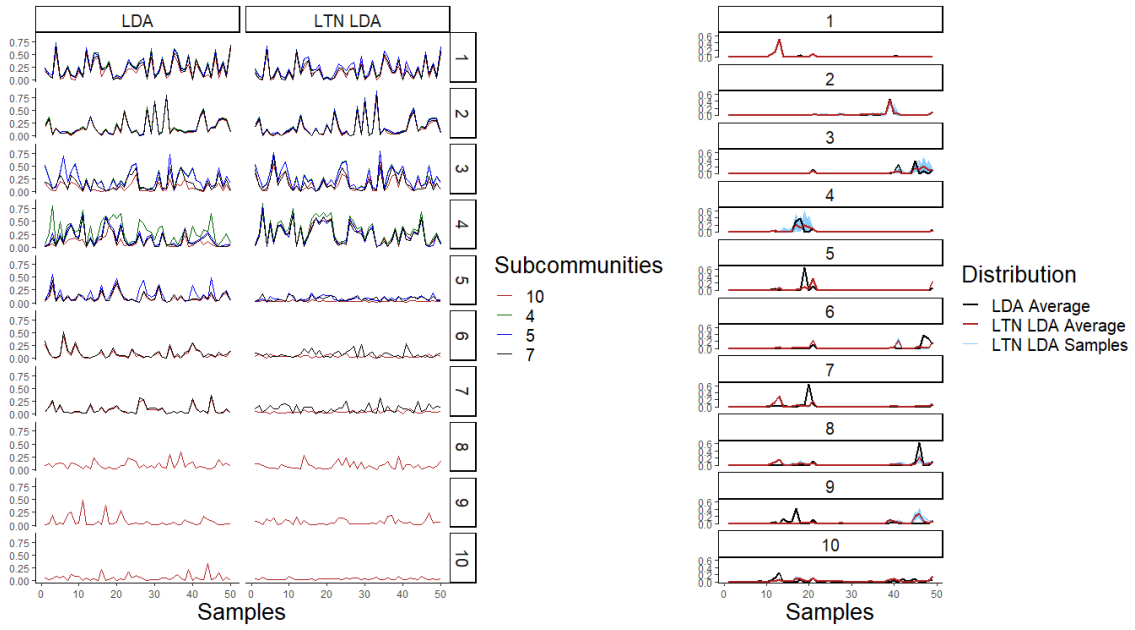
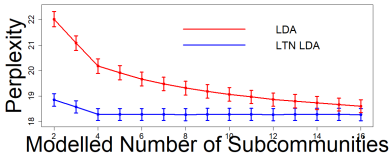
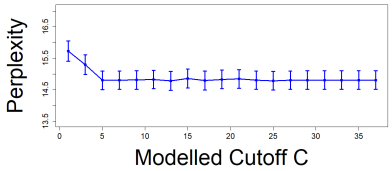


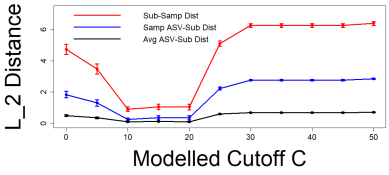
Figure 3: (Left) Subcommunity abundance for $\boldsymbol{\phi}_d$ for all samples over four different numbers of subcommunities $K \in \{4, 5, 7, 10\}$ for LDA (left) and LTN-LDA (right). The estimated abundances are noticeably more stable over different values of K for the LTN-LDA. (Right) Estimated subcommunity compositions for all samples. Blue indicates the sample-specific composition under LTN-LDA ($\boldsymbol{\beta}_{d,k}$, red indicates the average subcommunity composition under LTN-LDA ($\boldsymbol{\beta}_k$) and black indicates the average subcommunity composition ($\boldsymbol{\beta}_k$) under LDA. The 49 ASVs are on the x-axis.



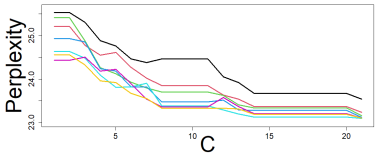
(a)



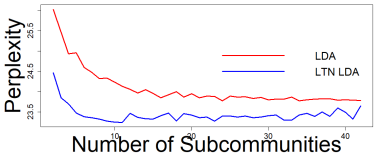
(b)



(c)



(d)



(e)

Figure 4: **(a)** Perplexity for LDA (red) and LTN-LDA (blue) as the modelled number of subcommunities K varies. **(b)** Perplexity for LTN-LDA as the modelled threshold C varies. **(c)** L_2 distances for $\boldsymbol{\phi}_d$ (red), $\boldsymbol{\beta}_{k,d}$ (blue), and $\boldsymbol{\beta}_k$ (black) for LTN-LDA as the modelled threshold C varies. **(d)** Perplexity for varying levels on K on the Dethlefsen and Relman data as we vary C. **(e)** Perplexity for LTN-LDA and LDA as K varies while $C = 8$.

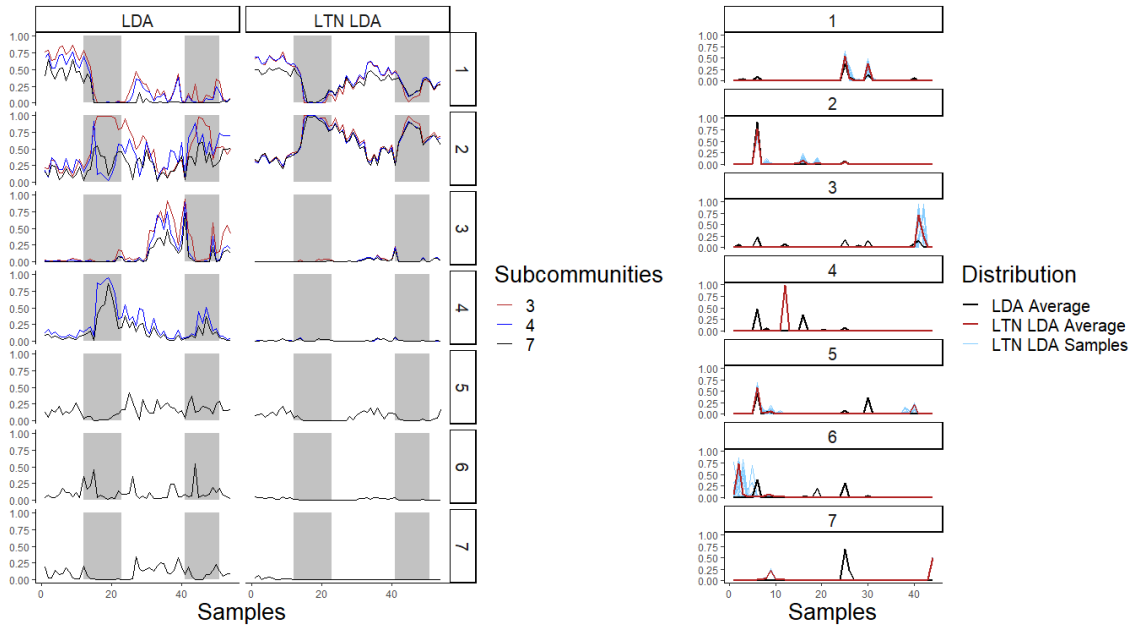


Figure 5: (Left) Subcommunity abundance for $\boldsymbol{\phi}_d$ for all samples over three different numbers of subcommunities $K \in \{3, 4, 7\}$ for LDA (left) and LTN-LDA (right). The estimated abundances are noticeably more stable over different values of K for the LTN-LDA. (Right) Estimated subcommunity compositions for all samples. Blue indicates the sample-specific composition under LTN-LDA ($\boldsymbol{\beta}_{k,d}$, red indicates the average subcommunity composition under LTN-LDA ($\boldsymbol{\beta}_k$) and black indicates the average subcommunity composition ($\boldsymbol{\beta}_k$) under LDA. The 44 ASVs are on the x-axis.

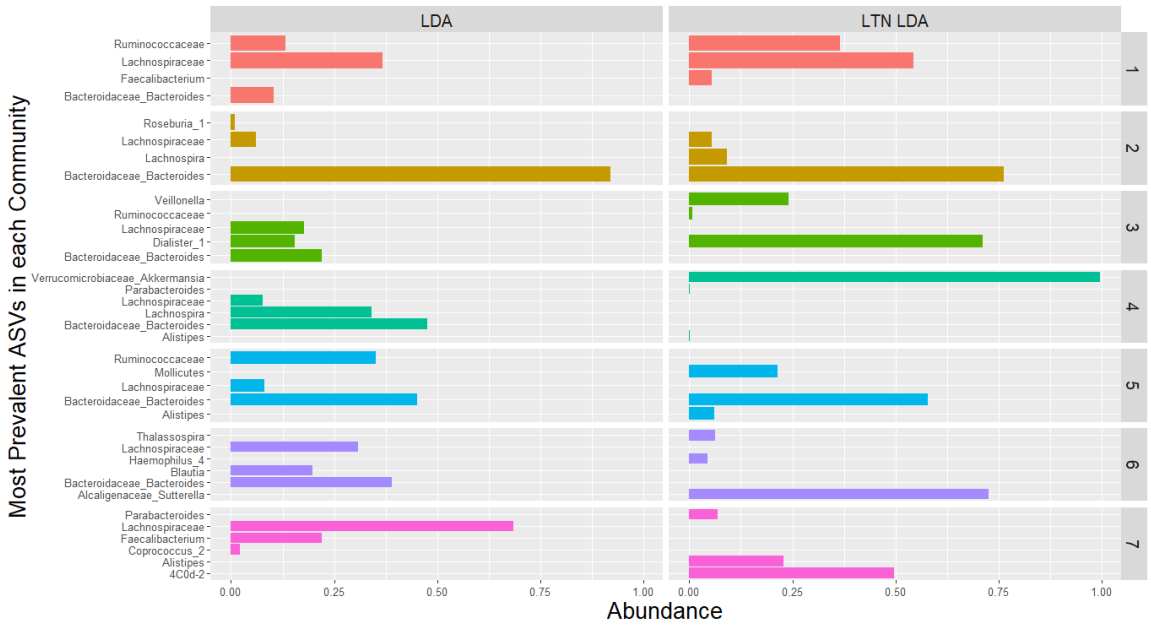


Figure 6: The 5 most prevalent ASVs in each subcommunity for LDA and LTN-LDA, $K = 7$, $C = 8$.

3 The Statistics of Recommender Systems

3.1 Introduction

Recommender systems save users time and help them find products that better meet their needs. They are continually improving, because algorithms are improving and because existing algorithms are learning more about users and products. Research is extending the reach of recommender systems to new kinds of applications. Statistics is integral to their success.

There are many different strategies employed by recommender systems. Two of the most common are collaborative filtering and content-based filtering. Collaborative filtering leverages rating data to recommend items to users, while content-based filtering uses content descriptions of items to make recommendations [Park et al., 2012]. In practice, many recommender systems do not restrict themselves to one strategy and instead combine multiple strategies in order to improve performance—such approaches are called hybrid filtering [Adomavicius and Tuzhilin, 2005a]. Many approaches have been developed, and, for commercial applications, they are generally tailored to a specific recommendation task since suggesting books is different from suggesting music or health insurance plans. Even within a specific task, there can be differences; e.g., suggesting a murder mystery probably employs different criteria than recommending a romance novel.

We describe collaborative filtering, content-based filtering, and hybrid methods before discussing the emerging field of active recommender systems. Active recommender systems interact with the user and can mimic how humans operate by, e.g.,

asking the user questions. If someone asks a person for a book recommendation, that person may by asking "What kind of books do you like?" One can imagine that one day when a user logs onto Amazon's kindle website, Amazon will ask a set of individually tailored questions to determine which book the user will be most likely to purchase.

Throughout this paper, we shall use the MovieLens 25M data set [Harper and Konstan, 2015] to illustrate the ideas and to benchmark performance. But there are ideas that do not generalize beyond the movie context. A recommender system that uses movie-specific features, such as actors, does not extend to such applications as suggesting books.

Section 2 reviews general ideas in the recommender system field, including common challenges and performance metrics. Section 3 describes collaborative filtering, Section 4 reviews content-based filtering, and Section 5 discusses hybrid procedures. Section 6 lays out some of the statistical issues in active recommender systems. Section 7 concludes.

3.2 Background

We formalize a mathematical framework for recommender systems following Adomavicius and Tuzhilin [2005a]. Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ be a set of N users, $\mathcal{M} = \{i_1, \ldots, i_M\}$ be a set of M items, and $f : \mathcal{U} \times \mathcal{M} \to \mathcal{R}$ be a utility function which maps a user-item pair (u_n, i_m) to a utility r_{nm} in a totally ordered set $\mathcal{R}$. In practice, the utility must be estimated from either explicit or implicit user feedback [Zhao et al., 2018]. Feedback is explicit if the user directly gives information stating their

opinion of items, e.g. rating a movie between 1 and 5 stars. Feedback is implicit if the system passively observes user behavior instead. There are three main types of implicit feedback: "examination", which measures how a user examines an item; "retention", which measures to what degree a user stores information on an item for later use; and "reference", which measures how users connect different items [Oard and Kim, 1998]. There is a trade-off between explicit and implicit feedback—explicit feedback may be more informative but implicit feedback is easier to collect [Nichols, 1998]. Despite this, there is relatively little research comparing the use of explicit or implicit feedback in recommender systems[Zhao et al., 2018] and most recommender systems exclusively use just one kind of feedback [Jawaheer et al., 2010]. Zhao et al. [2018] finds that using both kinds of feedback can improve recommender systems by, e.g., increasing user engagement.

Most of this paper uses the MovieLens 25M data set as a motivating example and so will assume explicit feedback in the form of ratings data stored in a ratings matrix $\boldsymbol{R}$: row n corresponds to user u_n, column m corresponds to movie i_m, and the value in the n^{th} row and m^{th} column, R_{nm}, is the rating given to movie i_m by user u_n. If user u_n did not give a rating to i_m, then R_{nm} is set to 0. Of course, this zero indicates missing data rather than an actual rating of zero.

Implicit feedback can be turned into explicit feedback of this matrix form by following Lee et al. [2008], which implements a collaborative filtering-based recommender system using a pseudo-ratings matrix constructed from implicit feedback. An example of a pure implicit feedback recommender can be found in Morita and Shinoda [1994], which predicts what rating a user will give an item based on the time the

user spends on a website. Koren [2008] proposes a hybrid method with both explicit and implicit feedback.

For each user u_n, a recommender system attempts to recommend the item i_{u_n} that maximizes the user's utility:

$$i_{u_n} = \text{argmax}_{i_m \in \mathcal{M}} f(u_n, i_m).$$

This, however, is only one of several possible criteria a recommender system could maximize. Gunawardana and Shani [2009] classifies recommender systems into three groups according to their specific goal: (1) to recommend a subset of a set of good but interchangeable items, (2) to optimize utility, e.g. to maximize value to a company by increasing revenue, and (3) to predict ratings over a possibly large set of user-item pairs. Each of these goals requires knowledge of the utility function f and so inference on f is critical.

A fundamental challenge is extreme sparsity. Consider the MovieLens 25M dataset [Harper and Konstan, 2015]. Released in December 2019, it contains data collected by the MovieLens movie recommendation service from January 9, 1995, to November 21, 2019. There are about 25 million ratings from $162,000$ users on $62,000$ movies. Thus only 0.25% of user-item pairs are rated: 99.75% of the data are missing. Furthermore, the data are not Missing Completely at Random (MCAR) or Missing at Random (MAR) [cf. Little, 1988], since the probability that a user watches a movie depends in part on how much they expect to enjoy it, and the probability they rate a movie is surely related to their enjoyment.

A subset of these movies is characterized by the tag genome [Vig et al., 2012]. The tag genome is a set $\mathcal{G}$ of 1,128 tags corresponding to some feature of a movie, such as "action", "sci-fi", or "spielberg". For each movie i_m with tag genome data, there is a vector $\boldsymbol{g}_m \in [0,1]^{1128}$ such that the jth entry in this vector, g_m^j, is a relevance score describing the pertinence of the jth tag to movie i_m. We restrict our analysis to the subset of tagged movies.

Many recommender system strategies make inferences on the utility function. Collaborative filtering and content-based filtering are the two oldest and most common [Park et al., 2012]. Collaborative filtering uses only the information contained in the ratings matrix to make recommendations; content-based approaches also use descriptions of the items. There are other approaches and contexts which are less common but nonetheless interesting: demographic filtering, which uses the demographic information of users [Prasad, 2012]; reciprocal recommenders, in which users are recommended to other users; context-aware recommenders, which take variables such as location and time into account; as well as a suite of deep-learning based approaches [Batmaz et al., 2019]. Related, but distinct, is the field of computational advertising which seeks to find the "best match" between users and advertisements subject to a number of constraints [Yang et al., 2017]. In practice, most recommender systems are hybrids that combine ideas from multiple methods and which are tuned to a specific application.

There are several metrics to evaluate recommender system performance [Herlocker et al., 2004], and the appropriate metric depends upon the domain and the goal [Gunawardana and Shani, 2009]. Predictive accuracy metrics, such as mean absolute

error (MAE) and root mean squared error (RMSE), have been used for as long as recommender systems have been studied [Herlocker et al., 2004] to assess how well recommender systems can predict ratings for any user-item pair. These metrics are convenient to compare the predictive performance of recommender systems and have been used in, e.g., the Netflix Prize competition. However, the rating prediction task does not by itself constitute a recommender system, as it must be combined with a decision rule governing which items to recommend [Gunawardana and Shani, 2009]. One such decision rule may be to recommend the "best" predicted items for a user.

Rank accuracy metrics measure how well a recommender system produces an ordered top k recommendation list that matches a user's top k list [Herlocker et al., 2004]. Utility maximization metrics measure how well a recommender system maximizes a company's utility and depend heavily on what that utility is. If a company maximizes its utility by providing an infinitely long ranked recommendation list to its users, then the half-life utility score is a useful performance metric [Breese et al., 1998, Gunawardana and Shani, 2009].

Classification accuracy metrics measure the observed frequency with which recommender systems correctly label items as good [Herlocker et al., 2004]. They include measures such as precision, which is the ratio of good items recommended to the number of items recommended, and recall, which is the ratio of good items recommended to the number of good items [Cleverdon and Kean, 1968]. The F-measure is a function of precision and recall which combines them into one statistic [Cremonesi et al., 2008]. Additionally, one may construct a receiver operating characteristic (ROC) curve on the entire dataset by plotting the true positive rate against the false positive rate or

take the area under the curve (AUC) as a summary statistic [Schein et al., 2005]. Variants of the ROC curve for the recommendation context, such as the Customer ROC (CROC), which imposes the constraint that each user is recommended the same number of items, have also been studied [Schein et al., 2005].

In our MovieLens example, we compare results in terms of RMSE as it is traditional and reflects overall accuracy. Learning user taste can be equally informed by what the user is indifferent to and dislikes as it is by what the user likes. Obviously, depending on the goal, other measures would be more appropriate.

To evaluate the performance of recommender systems, one may use online or offline experiments. Online experiments offer recommendations to real users and measure the user's response in terms of implicit and/or explicit feedback. Offline experiments use existing datasets, such as the MovieLens 25M dataset, and split the data into training and test sets to assess performance. Online experiments are better tailored to provide inputs and give feedback to a specific recommender system and better simulate how it would perform in the wild; however, they are significantly more expensive than offline experiments. The majority of academic research in recommender systems uses offline experiments. Beel et al. [2013] compares how recommender systems in research papers perform in offline and online experiments and find that a recommender system's performance in offline experiments can be a poor predictor of performance in online experiments.

Beyond concerns introduced by experiment type, there is a reproducibility crisis. Ekstrand et al. [2011] noted that research often did not follow best practices. Many papers presented their algorithms only as mathematical formulations without pro-

viding publicly available code implementations, leading other researchers to try to imperfectly duplicate their work. Moreover, there was no consistent basis for evaluating recommender systems, and many proposed methods were not compared to the best existing methods. These issues, to some extent, persist. Dacrema et al. [2019] analyzed 18 recently proposed neural recommendation approaches: only 7 could be reproduced and of those 6 were outperformed by basic nearest-neighbor methods. Dacrema et al. [2021] analyzed 12 neural recommendation approaches proposed at "prestigious conferences" and found that 11 of them were surpassed by simple methods such as nearest neighbors and linear models.

Much of the recommender system research is being done outside of academia and remains unpublished. Companies such as Amazon and Netflix pour resources into designing and tuning their recommender systems, but they are incentivized to keep progress secret. Academics lack their resources but can focus on developing theory.

3.3 Collaborative Filtering

Collaborative filtering uses only rating data to make recommendations—it uses no information about the users or items. There are two main approaches: memory-based methods and model-based methods.

Most memory-based methods follow the same procedure [Bobadilla et al., 2013]. First, use a similarity score to measure how alike the active user, u_n, is to each of the other users. The set $\{R_{nm}\}_{m=1}^{M}$ contains all of the ratings assigned by u_n; one calculates how similar u_n is to, say, u_a by calculating a score between the two sets $\{R_{nm}\}_{m=1}^{M}$ and $\{R_{am}\}_{m=1}^{M}$. Cosine similarity is widely used [Adomavicius and

Tuzhilin, 2005a]:

$$\text{sim}(u_n, u_a) = \frac{\sum_{m=1}^{M} R_{nm} R_{am}}{\sqrt{\sum_{m=1}^{M} R_{nm}^2} \sqrt{\sum_{m=1}^{M} R_{am}^2}}.$$

Other common measures are Pearson's correlation coefficient and Euclidean distance [Jeong et al., 2010].

Similarity scores determine k-nearest groups for each user u_n: simply find the k other users with the highest similarity score to u_n. To predict the rating u_n would give to an item i_m, use an aggregation strategy, e.g. an average or a weighted sum, to combine the ratings given to i_m by other users in the group who have rated i_m [Bobadilla et al., 2013]. Perhaps the most successful memory-based collaborative filtering implementation is Amazon's recommender system ("Customers who bought this item also bought ...") [Hardesty, 2019]. Breese et al. [2013] compares aggregation methods that use Bayesian networks and cluster analysis.

While these can be effective and are simple to implement, memory-based methods face challenges from data sparsity [Su and Khoshgoftaar, 2009]. They rely on different users having rated a common set of items, and the sparser the data, the smaller this set. Also, memory-based methods are computationally expensive: they compute similarity measures between all pairs of users and, as they use all ratings generated before a specific recommendation is made [Bobadilla et al., 2013], they must be rerun whenever a new rating is added.

In contrast, model-based collaborative filtering methods use the ratings matrix to learn an underlying model which is then used to predict ratings and make recommendations [Adomavicius and Tuzhilin, 2005a]. One such class of models is neighborhood

formation models, which cluster users (through k-means clustering, mixture modeling, Manhattan or normalized Euclidean distances) in order to predict ratings based upon that user's cluster [Candillier et al., 2005, 2007, Su and Khoshgoftaar, 2009]

A second class of models is Bayesian belief nets [Su and Khoshgoftaar, 2009], which use directed acyclic graphs (DAGs) to model conditional dependencies among variables [Cheng and Greiner, 2001]. Miyahara and Pazzani [2000] applied a collaborative filtering algorithm using Naive Bayes to binary rating data and found that it outperformed memory-based methods. More advanced models, such as Naive Bayes optimized by extended logistic regression [Greiner and Zhou, 2002, Shen et al., 2003] can be applied to recommender systems and can outperform memory-based methods [Su and Khoshgoftaar, 2006]. Wang and Tan [2011] relaxes the conditional independence assumption in Naive Bayes and finds improved performance.

A third class is latent semantic models [Su and Khoshgoftaar, 2009]. These aim to discover user communities and prototypical interest profiles and are more accurate than memory-based methods [Hofmann, 2004]. The aspect model [Hofmann and Puzicha, 1999] is an example of a latent semantic model; it models individual preferences as a convex combination of preference factors. Each user-item pair is associated with a latent class variable. Conditional on this variable, users and items are independent.

Latent factor models are another class of model-based collaborative filtering approaches which mitigate the high-dimensionality and sparsity of the problem by seeking to predict the ratings characterizing users and items in some lower dimensional latent factor space [Koren et al., 2009]. For movies, latent factors may correspond to

genres, quality of acting, or be uninterpretable [Koren, 2008]. Matrix factorization models are a subclass of latent factor models [Mehta and Rana, 2017]. There are four main matrix factorization techniques: Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF) [Goldberg et al., 2001, Luo et al., 2014], Singular Value Decomposition (SVD) [Sarwar et al., 2000], and Probabilistic Matrix Factorization (PMF) [Salakhutdinov and Mnih, 2007]. (Latent Semantic Indexing (LSI) [Deerwester et al., 1990, Littman et al., 1998] was first proposed in an information retrieval context and uses SVD; it has been used to develop recommender systems [Su and Khoshgoftaar, 2009] but is distinct from the latent semantic models cited above.) We now cover the SVD and PMF techniques in more detail.

SVD factors the $N \times M$ ratings matrix $\boldsymbol{R}$ by using a D dimensional low-rank approximation:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top,$$

where $\boldsymbol{S}$ is a D-dimensional diagonal matrix, $\boldsymbol{U} \in \mathbb{R}^{N \times D}$ is a latent user feature matrix, and $\boldsymbol{V} \in \mathbb{R}^{M \times D}$ is a latent item feature matrix. The $\boldsymbol{S}$ can be factored into $\boldsymbol{U}$ and $\boldsymbol{V}$, resulting in $\boldsymbol{R}$ becoming the product of two matrices. The SVD captures latent relationships between users and items and computes a low-dimensional representation of the original user-item space, which is used for neighborhood formation [Sarwar et al., 2000].

When applied to problems such as the Netflix prize competition, SVD encounters the problem of sparsity since most entries of $\boldsymbol{R}$ are empty and set equal to zero. Standard SVD struggles with pairs having zero entries when they are more accurately

seen as missing. Regularized SVD (RSVD), proposed in a seminal blog post [Funk, 2006], is constrained to consider only the observed user-item ratings.

Probabilistic Matrix Factorization (PMF) is similar to RSVD. PMF scales linearly with the number of observations and performs well in sparse and unbalanced datasets [Salakhutdinov and Mnih, 2007]. Let $\boldsymbol{U} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{V} \in \mathbb{R}^{M \times D}$ be latent user and item feature matrices, respectively. In particular, $\boldsymbol{U}_n$ is the feature vector for the n^{th} user, and $\boldsymbol{V}_m$ is the feature vector for the m^{th} item. PMF assumes that the likelihood of the observed ratings is

$$p(\boldsymbol{R}|\boldsymbol{U}, \boldsymbol{V}, \sigma^2) = \prod_{n=1}^{N} \prod_{m=1}^{M} [N(R_{nm}|\boldsymbol{U}_n^\top \boldsymbol{V}_m, \sigma^2)]^{I_{nm}},$$

where σ^2 is the variance parameter and I_{nm} is the indicator function for the event that user u_n rated item i_m. The priors on the user feature matrix $\boldsymbol{U}$ and the item feature matrix $\boldsymbol{V}$ are mean zero spherical Gaussian distributions,

$$p(\boldsymbol{U}|\alpha_U) = \prod_{n=1}^{N} N(\boldsymbol{U}_n|\boldsymbol{0}, \alpha_U^{-1}\boldsymbol{I}), \qquad p(\boldsymbol{V}|\alpha_V) = \prod_{m=1}^{M} N(\boldsymbol{V}_m|\boldsymbol{0}, \alpha_V^{-1}\boldsymbol{I}).$$

One learns the model by maximizing the log-posterior over the user and item features given the fixed hyperparameters $\boldsymbol{\alpha} = (\sigma^2, \alpha_U, \alpha_V)$:

$$\ln p(\boldsymbol{U}, \boldsymbol{V}|\boldsymbol{R}, \boldsymbol{\alpha}) = \ln p(\boldsymbol{R}|\boldsymbol{U}, \boldsymbol{V}, \sigma^2) + \ln p(\boldsymbol{U}|\alpha_U) + \ln p(\boldsymbol{V}|\alpha_V) + C,$$

where C is a constant that does not depend on $\boldsymbol{U}$ and $\boldsymbol{V}$. Maximizing the log-posterior is equivalent to minimizing the sum of squared errors in the objective function with

quadratic regularization, i.e.,

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} I_{nm}(R_{nm} - \boldsymbol{U}_n^\top \boldsymbol{V}_m)^2 + \frac{\lambda_U}{2}\sum_{n=1}^{N}\|\boldsymbol{U}_n\|_F^2 + \frac{\lambda_V}{2}\sum_{m=1}^{M}\|\boldsymbol{V}_m\|_F^2,$$

where $\lambda_U = \alpha_U/\sigma^2$, $\lambda_V = \alpha_v/\sigma^2$, and $\|\cdot\|_F$ is the Frobenius norm. One can use gradient descent in $\boldsymbol{U}$ and $\boldsymbol{V}$ to find a local minimum.

Jannach et al. [2013] shows that using a single latent factor does not personalize the system—it just recommends the movie that has the greatest overall popularity. But more factors provide person-specific rating estimates, and this generally improves as one adds latent factors until one begins to overfit. PMF and RSVD run quickly and efficiently, outperform previous methods in terms of predictive accuracy, and are the basis for many recommender systems [Mehta and Rana, 2017].

There is also a fully Bayesian version of the PMF model, Bayesian PMF (BPMF) [Salakhutdinov and Mnih, 2008]. The likelihood is

$$p(\boldsymbol{R}|\boldsymbol{U}, \boldsymbol{V}, \sigma^2) = \prod_{n=1}^{N}\prod_{m=1}^{M}[N(R_{nm}|\boldsymbol{U}_n^\top \boldsymbol{V}_m, \alpha^{-1})]^{I_{nm}},$$

which is the same as the standard PMF likelihood, except that we parameterize in terms of precision. The user and item latent feature vectors are given Gaussian priors:

$$p(\boldsymbol{U}|\mu_U, \Lambda_U) = \prod_{n=1}^{N} N(\boldsymbol{U}_n|\mu_U, \Lambda_U^{-1}), \qquad p(\boldsymbol{V}|\mu_V, \Lambda_V) = \prod_{m=1}^{M} N(\boldsymbol{V}_m|\mu_V, \Lambda_V^{-1}).$$

Gaussian-Wishart hyperpriors are placed on the user and item hyperparameters $\Theta_U =$

$\{\mu_U, \Lambda_U\}$ and $\Theta_V = \{\mu_V, \Lambda_V\}$:

$$p(\Theta_U|\Theta_0) = p(\mu_U|\Lambda_U)p(\Lambda_U) = N(\mu_U|\mu_0, (\beta_0\Lambda_U)^{-1})W(\Lambda_U|W_0, \nu_0)$$

$$p(\Theta_V|\Theta_0) = p(\mu_V|\Lambda_V)p(\Lambda_V) = N(\mu_V|\mu_0, (\beta_0\Lambda_V)^{-1})W(\Lambda_V|W_0, \nu_0),$$

where $\Theta_0 = \{\mu_0, \beta_0, \nu_0, W_0\}$. Inference is done via variational methods or Gibbs sampler. BPMF tends to outperform PMF and RSVD in terms of predictive accuracy for a fixed D, and its performance improves as D increases instead of overfitting [Salakhutdinov and Mnih, 2008].

Another popular subset of model-based collaborative filtering methods that are ones that adopt deep learning techniques [Zhang et al., 2019a]. The Neural Collaborative Filtering model replaces the inner product in matrix factorization approaches with neural architecture that can learn arbitrary functions from the data [He et al., 2017]; Collaborative Metric Learning replaces the inner product with a Euclidean distance metric [Hsieh et al., 2017]. There are several approaches that use autoencoders to make recommendations, especially AutoRec [Sedhain et al., 2015] and its extensions, such as CFN [Strub et al., 2016], which generate the Collaborative Denoising Auto-Encoder [Wu et al., 2016], and Multi-VAE and Multi-DAE [Liang et al., 2018]. Tang and Wang [2018] proposed sequential recommendations using convolutional neural networks. Such models are not necessarily distinct from those outlined above—PMF, for instance, can be regarded as a machine learning model—but they represent a growing body of research [Batmaz et al., 2019].

Collaborative filtering faces a number of challenges: sparsity; high-dimensionality;

scalability; synonymy, which occurs when equivalent items appear with different names; gray sheep, which are groups of users with idiosyncratic views who do not benefit from collaborative filtering; shilling attacks, which occur when an adversary gives large amounts of either positive or negative reviews to influence recommendations; and others [Su and Khoshgoftaar, 2009]. Pure collaborative filtering faces another chronic issue that we will highlight: the cold start problem, which occurs when either a new user or a new item is introduced [Schein et al., 2002].

The new-user problem is critical to recommender systems that wish to expand their user base. The challenge is that there are no data for on users. Collaborative filtering leverages rating data to make recommendations, so they are ill-suited for dataless users. A proposed solution is to ask a new user to rate a sequence of items until the system has enough information [Rashid et al., 2002]. Finding the optimal sequence is a problem unto itself; active learning, the process by which a recommender system decides which unrated item would provide the most information about user preferences if it were rated and prompts the user to rate said item, is one approach [Elahi et al., 2016, Rubens et al., 2016].

The new-item problem is important to recommender systems that are already in operation and which wish to introduce new items. It is, however, more critical for some applications than others. Many people will watch new movies without them having been recommended, so the problem is less severe for movies, but blog recommender systems are not so fortunate [Bobadilla et al., 2012]. One approach is to have a subcommunity of users who volunteer to rate new items [Bobadilla et al., 2012].

To illustrate, we apply PMF to a subset of the MovieLens 25M dataset. The code and data are available as supplementary files. The data were determined by randomly selecting users and including all ratings those users gave to movies with tag genome data. This resulted in $3,092$ users with $493,792$ ratings on $12,887$ movies. We partitioned this data into a training and a test set based on the timestamp; every rating given on or before January 16, 2016, was assigned to the training set. About $\frac{3}{4}$ of the ratings are used for training. This procedure mimics systems that observe ratings up to a time point and need to estimate ratings in the future.

The RMSE for PMF on the test data was 1.026, but we emphasize that in commercial use, much more work would be done to tune the system. We also found the top five recommendations (among movies not previously rated) for $\text{user}_{160,747}$, chosen because he or she rated the most movies. The top recommendation was *Pieces of April* (2003), followed by *The Signal* (2007), *Hoodwinked!* (2005), *Jakob the Liar* (1999), and *Captain Corelli's Mandolin* (2001). This implementation of PMF finds a local minimum, which need not be unique, so rerunning the algorithm with a different initialization produces different recommendations.

3.4 Content-Based Filtering

Content-based filtering uses item descriptions to make recommendations. Whether a user likes an item depends on its content; users will prefer items similar to items they have liked in the past [Balabanovic and Shoham, 1997]. Unlike collaborative filtering, users are independent of each other [Adomavicius and Tuzhilin, 2005a, Lops et al., 2011]. There are three main steps in content-based filtering: extracting item

features, learning user preferences, and recommending items that fit the user's preferences [Bobadilla et al., 2012].

Item features can be unstructured and high-dimensional. A movie might contain hours of audiovisual data, while a book could have over a hundred-thousand words. Data on this scale is problematic, so a feature extraction algorithm is necessary to represent item content in some structured low-dimensional fashion, commonly a vector in $\mathbb{R}^q$ for some practical q [Lops et al., 2011]. This item representation is taken as the input to the preference learning and filtering algorithms.

There are many feature extraction algorithms, and these are usually tailored to the application. One of the earliest commonly used methods, developed for text-based data, is a Vector Space Model (VSM) with term-frequency inverse document frequency (TF-IDF) weighting [Adomavicius and Tuzhilin, 2005a, B. Thorat et al., 2015, Lops et al., 2011, van Meteren, 2000]. A VSM represents a text document as a vector of weights in $\mathbb{R}^q$ where the ℓth entry characterizes the relevance of the ℓth keyword to the document. While many weighting schemes could be used, the most common is the TF-IDF scheme which accounts for both how frequently ℓ occurs in the document and how specific it is to the document.

Machine learning methods are also used as feature extraction algorithms. Wang and Blei [2011] uses Latent Dirichlet Allocation to identify the topics which characterize research papers, and van den Oord et al. [2013] applies deep convolutional neural networks directly to audio data to generate item profiles for songs. Vig et al. [2012] uses machine learning methods to generate the tag genome used in this paper, which estimates how relevant different tags, or keywords, are to movies.

From the item features and the ratings, a user is assigned to a subset of items. One then learns user preferences and constructs a user profile [Aggarwal, 2016, Lops et al., 2011]. The user profile is combined with an item description, allowing one to predict the rating that the user would assign to that item. But there is no single method of learning a user profile.

If the item description is a q-dimensional VSM using TF-IDF weights, the user profile may be a q-dimensional vector of weights where each entry characterizes the user's value for that keyword [Adomavicius and Tuzhilin, 2005a]. An averaging approach such as the Rocchio algorithm [Rocchio, 1971] can be used to compute the user profile Lang [1995]. Ratings are predicted for a user-item pair by evaluating a similarity score, such as cosine similarity, on the relevant user and item vectors.

There are other ways to learn user preferences. For instance, classification algorithms, such as Naive Bayes [Mooney and Roy, 2000], can be used to learn preferences and can perform well [Domingos and Pazzani, 2004]. Likewise, a decision tree may be used [Pazzani and Billsus, 2004]. Nearest-neighbor algorithms can cluster ratings based on their item description; the rating of an unrated item is predicted using the rating of the other items in its neighborhood [Billsus et al., 2000].

Once the features have been extracted and the user profile learned, one uses a filtering algorithm to recommend items [Bobadilla et al., 2012]. The best algorithm depends on the purpose of the recommender system. It may, for instance, entail recommending the top-k items the user has not yet rated. Or it may favor items that generate more profit.

Content-based filtering provides several advantages over other approaches. They

model each user independently, which reduces computational burden. They are easily explained to a user, and so provide a measure of transparency. And content-based filtering methods do not suffer from the new item cold-start problem once feature extraction has been performed [B. Thorat et al., 2015].

Despite these advantages, pure content-based filtering approaches are rare because they face challenges not seen by other approaches. First, it can be difficult to extract features, especially in domains with complicated and unstructured data such as blogs or music [Bobadilla et al., 2012]; moreover, if features are extracted incorrectly, the recommender system will be unreliable. A second problem is overspecialization [Lops et al., 2011]. Content-based approaches tend to recommend items similar to those already rated by users, leading to a lack of diversity in recommendations. Third, it is harder to acquire user feedback in this setting than in other settings, making it difficult to determine whether the recommendations are correct [Bobadilla et al., 2012]. For these reasons, content-based approaches are often combined with other methods.

We now build a content-based recommender system on the MovieLens data using the tagged genome information. This system is intended for illustration and would be outperformed by state-of-the-art methods.

First, extract user features. The tag genome provides a characterization of movie content; we generate content profiles for each movie by transforming the relevance scores. Let $\boldsymbol{w}_m \in \mathbb{R}^{1128}$ be the vector of weights for movie i_m. Then define the jth

entry as

$$w_{i_m}^j = \frac{g_m^\ell}{\sum_j g_m^\ell} \ln \frac{M}{\sum_m g_m^\ell},$$

where $g_{i_m}^\ell$ indicates the ℓth relevance score of movie i_m and M is the number of movies. This is a modified version of TF-IDF designed to work with relevance scores.

Second, we generate user profiles as weighted averages of the item profiles $\boldsymbol{w}_m$ of the movies rated by a user. Let $\boldsymbol{p}_n$ be the user profile for user u_n. Then,

$$p_n^\ell = \frac{\sum_m R_{nm} w_{i_m}^\ell}{\sum_m R_{nm}}.$$

We predict the rating R_{nm} by fitting a linear regression with $\boldsymbol{p}_n$ and $\boldsymbol{w}_m$ as covariates.

As before, we use standard code to train and test the recommender system. Its RMSE is 0.96, and the top five recommendations for $\text{user}_{160,747}$ among movies not already rated are, in descending order, *Planet Earth II* (2016), *Blue Planet II* (2001), *Band of Brothers* (2001), *Planet Earth* (2006), and *Hud* (1963). These results are more cohesive than those found by collaborative filtering: there are three films about our planet and two acclaimed character-driven historical dramas.

3.5 Hybrid Filtering

Hybrid filtering combines two or more types of recommender systems in order to improve performance and compensate for disadvantages of other types of recommender system [Burke, 2002]. In principle, any kind and number of recommender systems could be combined, but in practice, most hybrid recommender systems combine collaborative filtering with a different technique in order to address one of five

problems: cold-start, data sparsity, accuracy, scalability, and recommendation diversity [Cano and Morisio, 2017]. Moreover, and despite their importance, there has been relatively little work that explicitly surveys hybrid systems [Cano and Morisio, 2017].

Burke [2007] analyzes seven hybridization strategies that combine some sets of collaborative filtering, content-based filtering, demographic filtering, and knowledge-based filtering. The seven different strategies are called weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level [Burke, 2007]. Of these strategies, weighted and feature combinations are most prominent in recent literature [Cano and Morisio, 2017].

Weighted strategies combine the predictions of the component recommenders using numerical weights [Burke, 2007]. There are many ways to choose weights, such as averaging over all recommenders, using user feedback [Claypool et al., 1999], or using linear regression [Bell et al., 2008]. Cano and Morisio [2017] finds that twenty-nine percent of recently proposed hybrid systems use weighting strategies, making it the most common approach. One reason for their popularity is strength of performance: the winning entry to the Netflix competition was a weighting method that blended the results of over one hundred recommender systems. Another is the ease of implementation—it is simpler to implement and average the results from a suite of relatively basic recommender systems than to design and implement a complex bespoke model.

Feature combination strategies use features from one type of recommender system, such as collaborative filtering, as input to a different type of recommender system,

such as content-based filtering [Burke, 2007]. Bedi et al. [2013] uses collaborative filtering to generate book recommendations for each user and includes these recommendations as features in a content-based filtering algorithm; the hybrid method outperformed the individual methods. Despite being the second most popular strategy, recent papers have employed feature combination at only half the rate of weighting [Cano and Morisio, 2017].

Meta-level, feature augmentation, switching, and cascade strategies are each employed in about ten percent of recent papers [Cano and Morisio, 2017]. In meta-level strategies, one recommender system creates a model which is used as input for another recommender system [Burke, 2007]. It is common to use content-based recommenders to build item representation profiles and then use memory-based collaborative filtering methods to compare item and user profiles [Cano and Morisio, 2017]. Feature augmentation is similar to feature combination in that one recommender technique is used to compute a set of features which is part of the input of the next technique; however, instead of using features drawn from the contributing recommender's domain, feature augmentation generates new features for each item [Burke, 2007]. In switching strategies, the recommender system chooses one of the constituent recommender systems to make the recommendation [Burke, 2007]. For instance, one can use a collaborative filtering method as a default recommender and switch to other methods when there is a lack of data, as in a cold-start scenario. Cascade methods impose a hierarchy on their constituent systems; a weaker recommender is used to break a tie between two recommendations from the stronger recommender [Burke, 2007].

A mixed strategy runs multiple recommender systems in isolation and combines their recommendations into a single list [Burke, 2007]. This technique is rare, being used in only four percent of studies [Cano and Morisio, 2017].

Popularity in the literature is likely to be a good predictor of performance-by-type-of-strategy, but it is not a perfect one. Popularity is biased towards methods that are easy to implement, such as weighting strategies, and against methods that are difficult, such as meta-level strategies. Further, while the taxonomy proposed by Burke [2007] is extensive, covering eighty-seven percent of hybrid systems studied by Cano and Morisio [2017], it is not exhaustive and does not classify the remaining thirteen percent.

In addition to hybrid approaches which explicitly combine multiple recommender systems, one can create a hybrid recommender that has only one recommender system but which is not easily categorized: e.g., it is not clearly a collaborative or content-based method [Bobadilla et al., 2012]. These are common but can be quite varied. We highlight one recent hybrid recommender that uses both collaborative and content-based techniques. Bi et al. [2016] proposes a group-specific singular value decomposition method that generalizes SVD by incorporating between-subject dependency and using missingness in the ratings matrix.

Let x_{nm} be a covariate vector for user u_n and item i_m. If R_{nm} is the rating data, then take the demeaned $\tilde{R}_{nm} = R_{nm} - x_{nm}^{\top}\hat{\boldsymbol{\beta}}$ as the new ratings data, where $\hat{\boldsymbol{\beta}}$ is a vector of regression coefficients. If there is no covariate information, then an ANOVA model with global mean, user effects, and item effects is used to demean the data.

Let $\theta_{nm} = \mathbb{E}[\tilde{R}_{nm}]$ be the average rating and fit the model

$$\theta_{nm} = (\boldsymbol{p}_n + \boldsymbol{s}_{v_n})^\top (\boldsymbol{q}_m + \boldsymbol{t}_{j_m})$$

where $\boldsymbol{p}_n$ and $\boldsymbol{q}_m$ are K-dimensional user and item latent vectors as in standard SVD and $\boldsymbol{s}_{v_n}$ and $\boldsymbol{t}_{j_m}$ are K-dimensional group effects. There are V user clusters $V_v = \{u_n | v_n = v\}$ and J item clusters $J_j = \{i_m | j_m = j\}$. Individuals within the same cluster share group effects while individuals from different clusters are independent.

While multiple methods of clustering are possible, such as clustering by covariate information, Bi et al. [2016] clusters users and items using the nonignorable missingness of the data: the number of ratings per user or item. Missingness-related information is usually available for new users and items, so $\boldsymbol{s}_{v_n}$ and $\boldsymbol{t}_{j_m}$ can be used to solve the cold-start problem.

To carry out inference and address the scalability problem, Bi et al. [2016] proposes an method that embeds a backfitting algorithm into alternating least squares. Moreover, they avoid operating upon and storing large matrices, enabling scalable computation.

To illustrate, we applied the code used in Bi et al. [2016] to the MovieLens 25M data with the tagged genome data. The RMSE was 2.04, which is surprisingly large compared to the other implementations. The top five recommendations for $\text{user}_{160,747}$ were *Rivers and Tides* (2001), *The Comedians of Comedy* (2005), *Facing Windows (La Finestra di Fronte)* (2003), *Ghost Rider: Spirit of Vengeance* (2012), and *Fay Grim* (2006). Of course, we do not know how $\text{user}_{160,747}$ would have rated these

movies, but it appears that content-based filtering produced results that were more tightly themed, and may be more robust.

3.6 Active Recommender Systems

Most traditional recommender systems are "static recommenders" [Lei et al., 2020a]: they use a fixed dataset containing a user's rating or use history and do not interact with the user aside from providing recommendations. Static recommendation systems face several important limitations. If user preferences are not accurately represented in the rating data, possibly because the user has not rated a distinctive subset of items, then static recommenders will struggle [Jannach et al., 2021]. This can happen if user preferences change over time [Rafailidis and Nanopoulos, 2016] or if a new user requires the recommender to confront the cold start problem. Another source of worry is "natural noise", which negatively affects recommendations and occurs when, e.g., users incorrectly rate items [Amatriain et al., 2009].

Even if the data are correct, static recommenders cannot infer why a user was interested in an item [Gao et al., 2021]. Different users are interested in different items for different reasons at different times, and knowledge of a user's purpose is important for the recommendation. For example, does the user want to watch a comedy or a drama tonight? More broadly, static recommenders ignore confounding variables that change user preferences. Baltrunas and Amatriain [2009] demonstrates that the songs users want to hear depend on the time of day. Sometimes users are not even aware of their preferences and might construct them in a context-dependent manner [Tversky and Simonson, 1993]. In this case, an interactive decision aid tool

is helpful in exploring item space [Wang and Benbasat, 2013].

A solution to all of these limitations is a class of recommender systems we call active recommenders. These systems request specific user feedback. (Recommender systems that question the user have also been referred to as knowledge-based [Burke, 2000] and session-based recommender systems [Wang et al., 2021a].) Three archetypes that have recently emerged are interactive [Gao et al., 2021], critique-based [Jannach et al., 2021], and conversational recommender systems [Lei et al., 2020a].

Adomavicius and Tuzhilin [2005b] details the iterative personalization process at the core of interactive recommenders: (1) gather data on users, (2) make personalized recommendations, and (3) collect feedback on these recommendations. Allowing users to give feedback increases the effectiveness of a recommender system [He et al., 2016]. Hariri et al. [2014] uses Thompson sampling to learn a multi-arm bandit which takes context changes into account—it is an example of an interactive system that uses user feedback to improve recommendations. Interactive recommenders have been used to solve the new-user cold start problem in collaborative filtering. Sarwar et al. [2000] asks new users to rate items one at a time and iteratively chooses items based off of previous ratings. Loepp et al. [2014] employs a similar procedure, but asks users if they prefer one of two sets of items at each step. Active learning can guide which items are presented at each step to maximize the information learned [Elahi et al., 2016, Rubens et al., 2016]. While interactive recommenders can improve over static recommenders, Gao et al. [2021] notes that they can suffer low-efficiency issues because the recommenders cannot efficiently explore the item space.

Chen and Pu [2012] outlines the general algorithm of a critique-based recom-

mender system. First, the user initializes the system by either picking a starting product or a desired set of properties. The iterative part of the process begins here: the recommender suggests an item based on the initialization, then user chooses to accept or offer a critique. How the critiques work, and what form they take, varies by system. Burke et al. [1997] proposes a critique-based recommender with a method termed "tweaking", where a content-based feature of an item is modified to be different; e.g., the system might present options for movies like *Termiator II* but less violent. Reilly et al. [2004] extends this paradigm from unit critiques, a critique on one content feature, to compound critiques, which respond to multiple features at once. Ricci and Nguyen [2007] allows users to distinguish the strength of their critiques by specifying which are "musts" and which are "wishes", as well as integrating simple dialogue into the recommender. Some recommenders, such as Viappiani et al. [2006], give their users more flexibility in specifying critiques by allowing them to specify which feature they wish to critique instead of selecting from a prespecified list of options.

Underlying the concept of critique-based recommendation is the notion of content—to critique a product, one must know what characterizes that product. Such recommenders need a measure of how near any two items are in content space [McGinty and Smyth, 2003]; this is similar to a content-based recommender. Hong et al. [2010] implements a critique-based recommender using an embedded conversational agent (ECA) for e-novel recommendation; in particular, it uses an algorithm that alternates between supervised and unsupervised machine learning techniques to cluster items along multiple dimensions. Vig et al. [2011] implements a systems-suggested

critique-based recommender using the tag genome as a natural content-based characterization of movies in the MovieLens dataset.

Most critique-based recommenders rely on forms; i.e., they rely on a prespecified dialogue or methods of interaction [Jannach et al., 2021]. Moreover, a critique-based recommender will suggest an item after receiving feedback, even when it is not certain enough of user preferences to make a good recommendation Gao et al. [2021].

There is no universal definition of conversational recommender systems (CRS); however, a key component of a CRS is a multi-turn dialogue [Gao et al., 2021, Jannach et al., 2021, Lei et al., 2020a]. Multi-turn conversation allows the CRS to learn the user's current preferences and motivation. Most CRSs follow a general algorithm. First, the CRS is initialized, possibly on offline data [Christakopoulou et al., 2016] or by asking the user to provide a starting point [Sun and Zhang, 2018]. Next, the CRS chats with the user to learn the user's preferences. This can take different forms depending on the CRS involved—e.g. asking questions [Zou et al., 2020] or providing lists of recommendations [Sun and Zhang, 2018]—but it is differentiated from critique-based recommenders by its multi-turn structure. The CRS can ask multiple questions in a row without providing recommendations, improving the recommendation process. Finally, the CRS makes recommendation. If the user does not like the recommendation, then the CRS returns to querying user preferences.

There is no standard anatomy of a CRS, but every CRS must include three components: (1) a user interface [Gao et al., 2021], (2) a recommendation algorithm [Lei et al., 2020a], and (3) a preference elicitation algorithm [Christakopoulou et al., 2016]. The components are models unto themselves and may include further sub-

components. CRSs are thus in some sense more ambitious than other recommender systems because they must integrate additional modeling components.

The details of the user interface—and any subcomponents, such as a dialogue management system [Jannach et al., 2021]—are outside the scope of this review, which focuses on the statistics of recommender systems. They bear some similarities to conversational search, dialog systems, traditional web search, or faceted search, but differ in that their function is focused on recommendation [Zhang et al., 2018]. User interfaces can support a wide variety of interaction modalities. Natural language is the most attractive method of interaction, but it currently faces technical challenges which make it less effective [Jannach et al., 2021]. End-to-end CRSs, for instance, lead to broken conversations in about one-third of all interactions [Jannach and Manzoor, 2020]. Recent papers have found evidence that mixed modality strategies outperform purely natural language methods in terms of user experience [Ma et al., 2021, Narducci et al., 2020]. This is still an open question, however; Ren et al. [2020b] uses adversarial learning to improve end-to-end learning and generate more human-like conversations.

The user interface is closely linked to the recommender system and the preference elicitation algorithm. As the part of the CRS which communicates with the user, it must pass recommendation and preference information between the user and the other parts of the CRS. Any change in the user interface may induce change in the recommender system or preference elicitation and *vice versa*. Zhou et al. [2020a], for instance, uses knowledge-graph-based semantic fusion to model language data, and this necessitates modification of the recommender engine.

Collaborative, content-based, and hybrid approaches have all been employed as

the recommender systems in a CRS [Jannach et al., 2021]. In theory, any sort of recommender system could be used here. Christakopoulou et al. [2016] tries a version of PMF which has been modified to work with their preference elicitation algorithm; Zou et al. [2020] employs a novel matrix factorization method referred to as QMF; Sun and Zhang [2018] employs a factorization machine trained on dialogue state, user information, and item information; Zhang et al. [2018] uses personalized multi-memory networks; Zhou et al. [2020a] develops a knowledge-graph-enhanced recommender module using machine learning methods. To be most useful, a recommender in a CRS should be able to provide guidance to the preference elicitation algorithm.

The preference elicitation algorithm determines how a CRS discovers user preferences. It is similar to the onboarding process described in Sarwar et al. [2000], as well as the active learning process [Elahi et al., 2016, Rubens et al., 2016]. It is common for different CRSs to employ different techniques to elicit preferences. Because the user interface, the recommender system, and the preference elicitation algorithm need to work together, it is unlikely that any two CRSs employ exactly the same techniques. Christakopoulou et al. [2016] characterizes items in terms of features and uses multi-arm bandits to decide which features to query users about or whether to make recommendations. Zou et al. [2020] asks questions about item features extracted from textual information describing the items using General Binary Search to select a sequence of questions. Sun and Zhang [2018] employs a deep policy network to decide, at each step, whether to inquire about user preferences or make a recommendation. Zhang et al. [2018] employs a pre-trained multi-modal network to decide whether to elicit preferences or to make recommendations.

A CRS can engage in multi-turn dialogue; it can ask the user multiple questions in a row without making a recommendation. At each point in the conversation, the CRS decides whether to elicit a preference or make a recommendation. Eliciting preferences helps improve recommendations, but if too many questions are asked in a then the user may grow bored and discard the CRS [Lei et al., 2020a]. Lei et al. [2020b] proposes a method for framing multi-turn conversational strategy as an interactive path reasoning problem on a graph. Habib et al. [2020] follows a more rules-based approach, requesting, for example, additional preference information if the number of items meeting the current preferences exceeds a predefined threshold. Multi-turn conversations can be thought of as a version of an explore-exploit problem. This problem is more intense for new users in a cold-start scenario, where the CRS has no past information [Gao et al., 2021]. Christakopoulou et al. [2016] explores this problem by evaluating eight multi-arm bandit strategies.

Evaluating an active recommender system in general, and a CRS in particular, is more difficult than evaluating other strategies such as content-based recommenders. The dynamic interactions between users and the system on which active recommender systems are based are difficult to capture with traditional offline datasets such as the MovieLens dataset [Gao et al., 2021]. Moreover, for systems with multiple components, each component can be evaluated independently and according to different metrics, which complicates performance assessment [Jannach et al., 2021]. For this reason, synthetic datasets and *in vivo* experiments have become more common [Iovine et al., 2020, Narducci et al., 2020]. Zhang and Balog [2020] proposes a method to simulate users for use in synthetic environments. Synthetic environments, however,

will often fail to generalize to real applications, and the best way to evaluate models is through online experiments, as in, e.g., Zhou et al. [2020b].

3.7 Conclusion

This paper highlights several key findings. The first is that recommender system methodology is quite complex—there are many approaches, and these can be combined in many ways. To compound things, recommender systems can be used for multiple goals (e.g., recommend the best, recommend an item that will be liked, and maximize profit). There are correspondingly many metrics for recommender system performance. Table 1 shows the MovieLens 25 analysis results based on the three main recommender methods we have discussed.

One consequence of this combinatorial explosion of methodologies is that there is no general theory. Each application requires a bespoke analysis, tuned to the specifics of the problem. But in many industries, such as computational advertising or Amazon sales, an improvement of a fraction of a percentage point in predicting a user's preference can translate into millions of dollars of revenue. So it is worthwhile to invest in building good systems and continually improving them as more data and better algorithms become available.

The review also emphasized statistical aspects of recommender systems, both in their development and their assessment. As was shown, experimental design, latent space methods, and machine learning techniques are core parts of the design and refinement of these tools.

Finally, the main conclusion of this paper is that recommender systems are a

hugely important area, but too few statisticians are engaged in such research. Much of this territory has been ceded to computer scientists, but statisticians have the potential to make consequential contributions.

Table 1: MovieLens 25 Analysis

Recommender	RMSE	Top5 Recommended Movies
Collaborative Filtering	1.026	Pieces of April (2003) The Signal (2007) Hoodwinked! (2005) Jakob the Liar (1999) Captain Corelli's Mandolin (2001)
Content-based Filtering	0.96	Planet Earth II (2016) Blue Planet II (2001) Band of Brothers (2001) Planet Earth (2006) and Hud (1963)
Hybrid Filtering	2.04	Rivers and Tides (2001) The Comedians of Comedy (2005) Facing Windows (La Finestra di Fronte) (2003) Ghost Rider: Spirit of Vengeance (2012) Fay Grim (2006)

4 Time-varying Bayesian Network Meta-Analysis

4.1 Introduction

Methicilin-resistant *Staphylococcus aureus* (MRSA) infections are a threat to public health. MRSA increases mortality, hospital stays, and costs [Crum et al., 2006, McCollum et al., 2007, Shorr, 2012]. The incidence of MRSA rose globally in the late 1900's and early 2000's [Hersh et al., 2008]. The SENTRY antimicrobial surveillance program, for instance, observed increasing prevalence of MRSA in complicated skin and soft structure infections (cSSSI) [Moet et al., 2007]. More recent findings suggest that MRSA prevalence peaked in 2008 and has been declining since in the European Union and the United States [Diekema et al., 2019, Klein et al., 2017]; see Figure 7 for a plot of MRSA prevalence over time. This may be because medical professionals began to implement clincial interventions to reduce the spread of MRSA [Liebowitz, 2009]. Yet MRSA remains the second most common cause of antibiotic-resistant bacterial infections in the European Union [Gasser et al., 2019] and is stable in the Asia-Pacific region [Lim et al., 2018].

Growing antibiotic resistance in MRSA is a potential problem [Nathwani, 2009, Wilcox, 2009]. *S. aureus* is possibly developing resistance to other treatments, such as fusidic acid and mupirocin [Brown et al., 2021]. The Infectious Disease Society of America (IDSA) has long recommended vancomycin as a treatment for MRSA [Gould et al., 2012], and vancomycin is regarded as the "gold standard" of MRSA treatments [Shorr, 2012]. Daum [2007] and Cosgrove et al. [2004] state that the increase in MRSA prevalence resulted in increasing use of vancomycin and the emergence of

vancomycin resistant *S. aureus*. Diekema et al. [2019] finds that there was no increase in vancomycin-resistant MRSA from 2013-2016. There remains an "evidence gap" with respect to vancomycin-resistant *S. aureus* [Brown et al., 2021].

Many randomized controlled trials (RCT) have been conducted to assess the effectiveness of treatments for MRSA-related cSSSIs. These studies provide a mix of direct and indirect evidence for the treatments, so Bayesian network meta-analyses (BNMAs) have been used to estimate treatment effects; in particular, there is disagreement about whether linezolid is more effective than vancomycin [Brown et al., 2021, Feng et al., 2021, Guest et al., 2017, Lan et al., 2019, Li and Xu, 2018, Liu et al., 2016, Mccool et al., 2017, Thom et al., 2015, Zhang et al., 2019b]. If MRSA is developing antibiotic resistance, however, treatment effects would vary across time. The selection of treatments for a given RCT, which must take place over a short time period, will be confounded with the estimated effects of those treatments. This type of design inconsistency [Higgins et al., 2012] must be accounted for by modelling time-varying treatment effects across RCTs.

Literature that incorporates time effects into models has focused on capturing time effects within individual RCTs rather than addressing time-based design inconsistencies. This paper highlights some methods; Tallarita et al. [2019] provides a more exhaustive review. Jansen [2011] uses fractional polynomials to model the hazard ratio in a network of RCTs which each report the hazard ratio in some longitudinal format. Jansen et al. [2015] generalizes this approach to other types of longitudinal data. Mawdsley et al. [2016] proposes a framework, model-based network meta-analysis (MBNMA), which adapts methods from model-based meta analysis (MBMA) to the

network setting in order to capture dose-response relationships. Pedder et al. [2019] extends this approach to time-course models. A common feature of these approaches is that they model time effects within individual RCTs; each RCT returns data which has a time component, e.g. dose-response curves, and the goal is to compare these time-varying functions across trials. However, this time component does not address time-based design inconsistencies since treatment effects do not change depending on when the RCT was conducted.

Time-based design inconsistencies could be addressed with standard meta-regression techniques [White et al., 2012]. Salanti et al. [2009], for instance, employs a meta-regression over time in a BNMA to study the effectiveness of oral health interventions: placebo treatments became more effective over time. However, existing meta-regression BNMAs are limited to linear effects. The true pattern of time-varying effects is unknown. If treatments vary non-linearly, these meta-regression techniques will have limited value.

This paper develops a class of BNMA models which can detect time-varying treatment effects: time-varying BNMA (tBNMA). The existence of a latent, unobserved time series for treatment effects is modelled with a Gaussian Process using a combination of white noise, linear, and Matern kernels. In simulations, tBNMA outperforms existing methods when even just one treatment has time-varying effects.

The datasets of Thom et al. [2015], Liu et al. [2016], Guest et al. [2017], Mccool et al. [2017], Li and Xu [2018], Zhang et al. [2019b], Lan et al. [2019], Brown et al. [2021] and Feng et al. [2021] are combined to form one MRSA-cSSSI dataset that includes 58 studies comparing 19 treatments from 2000 to 2019. tBNMA detects

non-linear time trends, finding that vancomycin resistance in MRSA was strongest between 2002 and 2007, but has since decreased. Moreover, tBNMA finds that, while linezolid used to be significantly more effective than vancomycin, the difference is no longer statistically significant in 2019.

4.2 Bayesian Network Meta-Analysis

Often there are many treatment options available for a medical condition. In a given RCT, researchers compare only a subset of those possible treatments. To know whether a given treatment, A, is more or less effective than another treatment, B, then, there is a mix of direct evidence, where A and B are directly compared, and indirect evidence, where the treatment effect is estimated through some joint comparator C. When there are only three treatments with two pairwise comparisons — A compared to B and B compared to C — then analysis is straightforward [Bucher et al., 1997]. However, situations of greater complexity arise and induce a network of comparisons amongst the treatments.

Models developed to estimate the treatment effects are referred to as Network Meta-Analyses (NMA). Frequentist NMA's have been developed in Higgins and Whitehead [1996], Lumley [2002] and Chootrakool and Shi [2008] while Bayesian NMA's have been developed in Ades [2003], Lu and Ades [2004] and Lu and Ades [2006]. The formulation of Dias et al. [2011] for BNMAs with binomial data is followed in this paper.

Let there be I studies comparing (some of) K treatments. If treatment k is used in study i, then the response variable is y_{ik}, the number of successes. Each

y_{ik} has probability of success p_{ik} for n_{ik} subject. Then $y_{ik} \,|\, p_{ik}, n_{ik} \sim \text{Bin}(p_{ik}, n_{ik})$. The probabilities are modelled with a logit-link function: $\text{logit}(p_{ik}) = \mu_i + \delta_{i,b_i,k} 1_{b_i \neq k}$. Here, b_i is the baseline treatment in study i. If possible, all studies would have the same baseline, b, but this usually not the case, so the most common treatment is taken as the baseline. The trial-specific effects of trial i are captured by μ_i. These are nuisance parameters and are modelled as random effects, $\mu_i \sim N(m_\mu, \sigma_\mu^2)$. The μ_i terms allow BNMA to estimate the mean effect of each treatment d_{1k} even when there are unknown confounding effects between studies.

The difference in efficacy between treatment k and treatment b_i in study i is $\delta_{i,b_i,k}$. In a random effects model, it is drawn from a normal distribution, $\delta_{i,b_i,k} \,|\, d_{b_i,k}, \sigma^2 \sim \text{N}(d_{b_i,k}, \sigma^2)$. Homogeneity of variance — that $\sigma^2_{b_i,k} = \sigma^2$ for all b_i and k — is assumed because there is not enough data to learn heterogeneous variance [Higgins and Whitehead, 1996]. In a multi-arm trial, the joint distribution of the $\delta_{i,b_i,k}$ is the following multivariate normal:

$$\begin{bmatrix} \delta_{i,b_i,2} \\ \delta_{i,b_i,3} \\ \cdots \\ \delta_{ib_i,k-1} \end{bmatrix} \sim N\left(\begin{bmatrix} d_{b_i,2} \\ d_{b_i,3} \\ \cdots \\ d_{b_i,k-1} \end{bmatrix}, \begin{bmatrix} \sigma^2 & \frac{\sigma^2}{2} & \ldots & \frac{\sigma^2}{2} \\ \frac{\sigma^2}{2} & \sigma^2 & \ldots & \frac{\sigma^2}{2} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\sigma^2}{2} & \frac{\sigma^2}{2} & \ldots & \sigma^2 \end{bmatrix} \right)$$

It is more efficient to decompose this joint likelihood into a product of conditional

likelihoods:

$$\delta_{i,b_i,k} \mid \delta_{i,b_i,2}, \ldots, \delta_{i,b_i,k-1}, d_{b_i,2}, \ldots, d_{b_i,k-1}, \sigma^2$$
$$\sim \mathrm{N}\left(d_{b_i,k} + \frac{1}{k-1} \sum_{j=1}^{k-1} [\delta_{i,b_i,j} - d_{b_i,j}], \frac{k}{(k-1)} \sigma^2 \right)$$

The relative difference in treatment effect between treatment k and baseline b_i is $d_{b_i,k}$. Under the consistency assumption [Lu and Ades, 2006] (also called coherence in Lumley [2002]), $d_{b_i,k}$ can be split into components in a way analogous to a "differences-in-differences" approach. The difference of b_i and k is equal to the the difference of the difference of k and treatment 1 (which may be taken as the general baseline b) and the difference of treatment b_i and treatment 1. That is, $d_{b_i,k} = d_{1k} - d_{1b_i}$. These baseline differences are drawn from a normal distribution: $d_{1k} \sim N(m_d, \sigma_d^2)$. The $d_{11}, d_{12}, \ldots, d_{1k}$ are called basic parameters while the $d_{b_i,k}$ are called functional parameters.

It remains to choose priors for the hyperparameters. Rosenberger et al. [2021] compares different commonly used prior specifications for variance priors — inverse-gamma, uniform, and half-normal — and found that the prior choice had little effect on point estimates. A vague inverse-gamma prior is thus placed on σ^2, σ_μ^2, and σ_d^2, and a vague normal is placed on the m_μ and m_d. Taken together, the contrast-based

BNMA model with binomial outcomes for each arm is

$$y_{ik} \,|\, p_{ik}, n_{ik} \sim \text{Bin}(p_{ik}, n_{ik}) \qquad \text{logit}(p_{ik}) = \mu_i + \delta_{i,b_i,k} 1_{b_i \neq k}$$

$$\delta_{i,b_i,k} \,|\, \delta_{i,b_i,2}, \ldots, \delta_{i,b_i,k-1}, d_{b_i,2}, \ldots, d_{b_i,k-1}, \sigma^2$$
$$\sim \text{N}\left(d_{b_i,k} + \frac{1}{k-1} \sum_{j=1}^{k-1} [\delta_{i,b_i,j} - d_{b_i,j}], \frac{k}{(k-1)} \sigma^2 \right)$$

$$\mu_i \,|\, m_\mu, sd_\mu \sim N(m_\mu, \sigma_\mu^2) \qquad m_\mu, m_d \sim \text{N}(0, 10000)$$

$$\sigma^2, \sigma_\mu^2, \sigma_d^2 \sim \text{IG}(1,1) \qquad d_{b_i,k} = d_{1k} - d_{1b_i}$$

$$d_{1k} \sim \text{N}(m_d, \sigma_d^2)$$

4.3 Time-Varying Bayesian Network Meta-Analysis

The studies in the dataset are indexed by $i \in \{1, 2, \ldots, I\}$. The time of study i is t_i, so that the list of possibly non-unique timepoints is $t_1, t_2, \ldots, t_I$. Treatment k occurs in I_k studies, and the list of studies it occurs in can be indexed by i_k. The timepoints in which treatment k occurs are indexed by t_{i_k}. If there is a time-based design inconsistency, then $d_{b_i,k} \neq d_{1k} - d_{1b_i}$ for some studies i because the basic parameters d_{1k} cannot capture the time-varying nature of the treatment effect. To remedy this, model a time-specific value of d_{1k}, $d_{1k}^{t_{i_k}}$, at each of the timepoints t_{i_k}. Then redefine the $d_{b_i,k}$: $d_{b_i,k} = d_{1k}^{t_{ik}} - d_{1b_i}^{t_{ib_i}}$. For each k, the $d_{1k}^{t_{ik}}$ correspond to a latent, unobserved, potentially nonstationary time series which could exhibit any of a large number of time-varying trends. To maintain flexibility, the $d_{1k}^{t_{ik}}$ are modelled as arising from a Gaussian Process (GP) kernel [Brahim-Belhouari and Bermak, 2004, Rasmussen and Williams, 2006]. Let $d_{1k}^{t_1} \sim \text{GP}(\boldsymbol{d_{1k}}, K(\cdot,\cdot))$ represent the following

distribution:

$$\begin{bmatrix} d_{1k}^{t_{1_k}} & d_{1k}^{t_{2_k}} & \dots & d_{1k}^{t_{I_{k_k}}} \end{bmatrix}^T \sim N\Bigg(\begin{bmatrix} d_{1k} & d_{1k} & \dots & d_{1k} \end{bmatrix}^T, K(\cdot,\cdot) \Bigg).$$

Decompose the covariance kernel, $K(\cdot,\cdot)$ into three separate kernels (for more on kernel decomposition see, e.g. Corani et al. [2020]): (1) a white noise kernel, (2) a linear kernel, and (3) a Matern covariance kernel. That is

$$K(\cdot,\cdot) = K_{\mathrm{W}}(\cdot,\cdot) + K_{\mathrm{L}}(\cdot,\cdot) + K_{\mathrm{M}}(\cdot,\cdot)$$

The white noise kernel is

$$K_{\mathrm{W}} = \psi^2 \mathbb{I}_{n_k},$$

where $\mathbb{I}_{n_k k}$ is the $n_k \times n_k$ identity matrix. This kernel adds white noise to the covariance terms. The linear covariance kernel is

$$K_{\mathrm{L}}(i,j) = s_{bk}^2 + s_{lk}^2 t_{i_k} t_{j_k},$$

which induces linear functions in the $d_{1k}^{t_{i_k}}$. The Matern covariance kernel, with $\nu = \frac{1}{2}$, is

$$K_{\mathrm{M}}(i,j) = \phi_k^2 \exp(-\rho_k |t_{i_k} - t_{j_k}|)).$$

This last kernel results in functions equivalent to the Ornstein–Ulenbeck process, the continuous time equivalent of an AR(1) model [Roberts et al., 2013]. As BNMA is effective at finding the average values d_{1k} (to be demonstrated below), these are

taken as the mean value for the Gaussian process. Vague priors are placed on all of the hyperparameters. Further, note that not all treatments should be modelled with time-varying effects: some treatments will not vary in time, while others will not have sufficient data to learn time-varying trends. Let $\mathcal{T}_0$ be the set of treatments modelled as constant in time, and let $\mathcal{T}_1$ be the set of treatments modelled as varying in time. The resulting model, termed tBNMA, is

$$y_{ik} \,|\, p_{ik}, n_{ik} \sim \text{Bin}(p_{ik}, n_{ik}) \qquad \text{logit}(p_{ik}) = \mu_i + \delta_{i,b_i,k} 1_{b_i \neq k}$$

$$\delta_{i,b_i,k} \,|\, \delta_{i,b_i,2}, \ldots, \delta_{i,b_i,k-1}, d_{b_i,2}, \ldots, d_{b_i,k-1}, \sigma^2 \sim \text{N}\left(d_{b_i,k} + \frac{1}{k-1} \sum_{j=1}^{k-1} [\delta_{i,b_i,j} - d_{b_i,j}], \frac{k}{(k-1)} \sigma^2 \right)$$

$$\mu_i \,|\, m_\mu, \sigma_\mu \sim N(m_\mu, \sigma_\mu^2) \qquad d_{b_i,k} = d_{1k}^{t_{ik}} - d_{1b_i}^{t_{ib_i}}$$

$$d_{1k}^{t_{ik}} \,|\, k \in \mathcal{T}_1, d_{1k}, \psi, \phi, \rho \sim \text{GP}(d_{1k}, K(\cdot,\cdot))) \qquad d_{1k}^{t_{ik}} \,|\, k \in \mathcal{T}_0, d_{1k} = d_{1k}$$

$$K(i,j) = K_{\text{W}}(i,j) + K_{\text{L}}(i,j) + K_{\text{M}}(i,j) \qquad K_{\text{W}} = \psi^2 \mathbb{I}_{nk}$$

$$K_{\text{L}}(i,j) = s_{bk}^2 + s_{lk}^2 t_{i_k} t_{j_k} \qquad K_{\text{M}}(i,j) = \phi_k^2 \exp(-\rho_k |t_{i_k} - t_{j_k}|))$$

$$\psi, s_{bk}, s_{lk} \sim \text{N}_+(0, 10000) \qquad \sigma^2, \sigma_\mu^2, \sigma_d^2, \phi_k \sim \text{IG}(1,1)$$

$$\rho_k \sim \text{G}(1,1) \qquad d_{1k} \sim \text{N}(m_d, \sigma_d^2)$$

$$m_\mu, m_d \sim \text{N}(0, 10000)$$

A Gibbs sampler is implemented in JAGS.

4.4 Data, Simulations, and Analysis

MRSA-related cSSSI treatments are analyzed using the the combined data from previous studies that employed BNMA. Using the network, treatment arms, and time-

points from these data, data is simulated with time-varying effects on one treatment. The performances of two BNMA methods on this simulated dataset are compared to each other and to tBNMA.

4.4.1 Data

Data from nine reviews employing NMA techniques to study the efficacy of treatments for MRSA-related cSSSIs are used: Thom et al. [2015], Liu et al. [2016], Guest et al. [2017], Mccool et al. [2017], Li and Xu [2018],Zhang et al. [2019b], Lan et al. [2019], Brown et al. [2021], and Feng et al. [2021]. A potential concern with combining datasets from multiple studies is that they will be incompatible — different experimental designs, for instance, may give rise to RCTs implemented on significantly different populations, violating the consistency assumption. The reviews are all conducted according to PRISMA or Cochrane standards, so there is a measure of similarity in how they collected studies. In all of these reviews, the vast majority of studies appeared in at least one other review: this implies transitive consistency. Given the lack of data on MRSA-related cSSSI's [Brown et al., 2021], it is better to be expansive when deciding which studies to include. Moreover, the random effects allow the models to compensate for inconsistencies introduced by combining data from different reviews.

These reviews contribute a total of 58 studies comparing 19 treatments from 2000 to 2019. The earliest date of publication of a study is used — if the day of publication is not available, it is imputed to be the middle of the month. A plot of the network is provided in Figure 8. Four studies have 3 treatment arms; the rest have 2. The most prevalent treatments are vancomycin (VAN), which appears 46 times,

and linezolid (LIN), which appears 27 times. There are 13 direct comparisons of the two. Both vancomycin and linezolid have comparisons with dalbavancin (DAL) and delafloxacin (DEL), but otherwise have no common comparators and the network structure can be thought of as having two poorly connected cliques. Vancomycin has additional comparisons with ceftaroline (CEF1),ceftobiprole (CEF2), oritavancin (ORI), daptomycin (DAP), telavancin (TEL), tigecycline (TIG), iclaprim (ICL), and lefamulin (LEF). Linezolid has additional comparisons with rifampicin (SXT/RIF), teicoplanin (TEI), omadacycline (OMA), a novel fluoroquinolone (JNJ-Q2), fusidic acid (CLEM-102), tedizolid (TED), and oxacillin-dicloxacililn (OXA). Daptomycin and telavancin have one comparison with each other while tigecycline and delafloxin have two. There are no other comparisons in the network.

4.4.2 Simulations

Simulations will show the limitations of existing models in the presence of time-based designed inconsistencies, and demonstrate the ability of tBNMA to solve this problem. The treatment comparisons, timepoints, and network associated with the combined data are used to generate the simulated data.

Three models are compared. The first is standard BNMA, which takes no measures beyond random effects to compensate for time-based design inconsistencies. The second is Meta-BNMA, which runs a meta-regression on time effects by modelling the $d_{tk}^{t_{i_k}}$ as following a linear trend in time for those treatments k which are allowed to vary in time. Meta-BNMA bears similarities to models discussed by Salanti et al. [2009] and White et al. [2012]. The third is tBNMA. There is prior information suggesting that only two treatments present in the study design — vancomycin [Daum, 2007]

and fusidic acid [Brown et al., 2021] — are potentially experiencing time-varying treatment effects. Of these, only vancomycin appears in enough studies for time-varying effects to be detectable. Thus, the two models which account for time-based design inconsistencies, Meta-BNMA and tBNMA, will allow time-varying effects only on vancomycin. Linezolid, the second most common treatment is used as the baseline treatment for all models.

If there are timed-based design inconsistencies, then the $d_{1k}^{t_{i_k}}$ could vary in time according to a large number of curves — but the specific form is unknown for any given treatment k. It is thus desirable to assess the performance of the three models in a number of scenarios. Three datasets, with three different time-varying effects on the vancomycin $d_{1k}^{t_{i_k}}$, are generated. In the first, the $d_{1k}^{t_{i_k}}$ are constant in time; in the second, the $d_{1k}^{t_{i_k}}$ are quadratic in time; in the third, the $d_{1k}^{t_{i_k}}$ are sigmoidal in time.

All three models are run on all three simulated datasets. The 95% credible intervals for the posterior predictive distributions for the $d_{1k}^{t_{i_k}}$ corresponding to the relative treatment effect of vancomycin compared to linezolid over time are plotted in Figure 9 along with the true values of the $d_{1k}^{t_{i_k}}$. When the true curve is constant and there are no time-based design inconsistencies, all three models return approximately constant trends in time. While all models work when there are no time-based design inconsistencies, Meta-BNMA and tBNMA have wider credible intervals than BNMA because they are more complex. When there are time-based design inconsistencies, either quadratic or sigmoidal, there are clear differences between the models. BNMA cannot detect time trends in the $d_{1k}^{t_{i_k}}$, though it can estimate the mean value d_{1k} with considerable accuracy. The time-based design inconsistencies result in elevated

uncertainty compared to the case where the the constant function is true. Meta-BNMA detects significant time trends; however, it is limited to detecting only linear time trends and is thus unable to learn more complicated scenarios. Moreover, it has greatly inflated credible intervals, indicating that it fits the data poorly. In contrast, tBNMA is flexible enough to accurately recover the true time-varying effect no matter the underlying trend.

The better fit found by tBNMA also leads to increased predictive performance. The quantity most of interest is d_{1k}^{T}, the relative treatment effect of treatment k relative to the baseline treatment at time T. As time T corresponds to the end of the study period, it holds the most clinical significance. Point estimates and 95% credible intervals from the posterior predictive distributions for the d_{1k}^{T} are found for all treatments and for all three models on the simulated sigmoidal dataset. The results are plotted in Figure 10 along with the true values. tBNMA consistently produces the best estimates, with the narrowest credible intervals. Since BNMA and Meta-BNMA cannot capture the full effect of the design-based inconsistencies, they compensate by increasing the uncertainty of their predictive posterior distributions, even for treatments which do not have time-varying effects. tBNMA thus outperforms existing methods in the presence of significant time-based design inconsistencies.

4.4.3 Implementation on MRSA Data

BNMA, Meta-BNMA, and tBNMA are run on the agglomerated dataset. As before, linezolid is the baseline treatment for all methods. Meta-BNMA and tBNMA allow for time-varying effects on vancomycin; all other treatment effects are fixed with respect to time. No covariates aside from time are considered because of the lack of

covariate information for most of the RCTs.

Figure 11 shows the 95% credible intervals for the posterior predictive distributions of the treatment effect of each treatment relative to linezolid at the end of the time period, d_{1k}^{T}. The three methods produce similar posterior mean estimates of the d_{1k}^{T}. Meta-BNMA and tBNMA have wider credible intervals because the time-varying effects modelled in the $d_{1k}^{t_{i_k}}$ for vancomycin induce a larger degree of uncertainty in the estimates for the other treatments. The estimate where the models most disagree, however, is that of the treatment effect of vancomycin relative to linezolid. That is, BNMA and Meta-BNMA find at least a 95% chance that vancomycin is less effective than linezolid at treating MRSA at the end of the time period; tBNMA finds only a 75% chance that this is true. At a 95% level, the models lead to different clinical inferences.

Figure 12 plots the 95% credible intervals for the posterior predictive distribution of the $d_{1k}^{t_{i_k}}$ relating the relative efficacy of vancomycin compared to linezolid learned by the tBNMA model. tBNMA discovers significant non-linear trends which are not found by existing methods. For most of the time period, vancomycin and linezolid are indistinguishable from each other at a 95% level; however, vancomycin is significantly less effective from 2002 to 2007. This is consistent with results from the medical literature concerning the overall prevalence of vancomycin-resistant *S. Aureus*. Cosgrove et al. [2004] and Daum [2007] reported the emergence of vancomycin-resistant *S. Aureus* during this period, while Klein et al. [2017] claimed that MRSA prevalence peaked in 2008 and Diekema et al. [2019] reported that there was no increase in vancomycin-resistant *S. Aureus* from 2013 to 2016. The most plausible explana-

tion is that the prevalence of vancomycin-resistant *S. Aureus* was rising in the mid 2000s. Medical experts then designed and implemented a set of medical interventions designed to slow the spread of antibiotic-resistant *S. Aureus* (see, e.g., Liebowitz [2009]) which tBNMA finds to be largely successful.

Previous network-meta analyses conducted to assess various treatments for *S. Aureus* have been divided on whether vancomycin is more effective than linezolid. Zhang et al. [2019b], Li and Xu [2018], Feng et al. [2021], and Mccool et al. [2017] found that linezolid was more effective, while Thom et al. [2015] and Guest et al. [2017] found them to be equivalent. The above results indicate that one reason for this disparity may be time-based design inconsistencies. Standard techniques such as BNMA and Meta-BNMA found linezolid to be significantly more effective than vancomycin at the end of the time period. However, tBNMA finds that, while linezolid used to be more effective than vancomycin at a 95% level, it is not significantly more effective at the end of the time period. Models which do not take the time-varying nature of this comparison into account may predict that there is a significant difference at the end of the time period, rather than in the middle.

4.5 Discussion

A novel model, tBNMA, is proposed which accounts for time-based design inconsistencies in network meta-analyses of RCTs by modelling time-varying effects as a latent, unobserved, time series. A Gaussian Process combining white noise, linear, and Matern kernels is used to model this latent series. tBNMA is fully Bayesian and allow for posterior uncertainty quantification; posterior computation proceeds

through a Gibbs sampler implemented in JAGS. tBNMA substantially outperformed existing methods in simulations in the presence of significant, non-constant, time-varying treatment effects.

Data from a collection of NMA-based review papers on MRSA-related cSSSIs is combined and analyzed using BNMA, Meta-BNMA, and tBNMA. tBNMA finds that MRSA is not more resistant to vancomycin at the end of the period than at the beginning, but there are substantial non-linear effects. Vancomycin resistance in MRSA was strongest between 2002 and 2007, in line with clinical trends, but has since declined. The time-based nature of this disparity may account for the disagreement about whether linezolid is more effective than vancomycin in the literature.

The time-varying methods presented in this paper could be expanded upon. One such extension would follow Jansen [2012] or Phillippo et al. [2020] and employ a meta-regression model to "balance" studies with covariate information to those without. Such methods are data-intensive, however, and care would be needed to employ them simultaneously with the time-varying methods proposed in this paper. Alternate kernels for modelling the time-varying effects could also be explored.

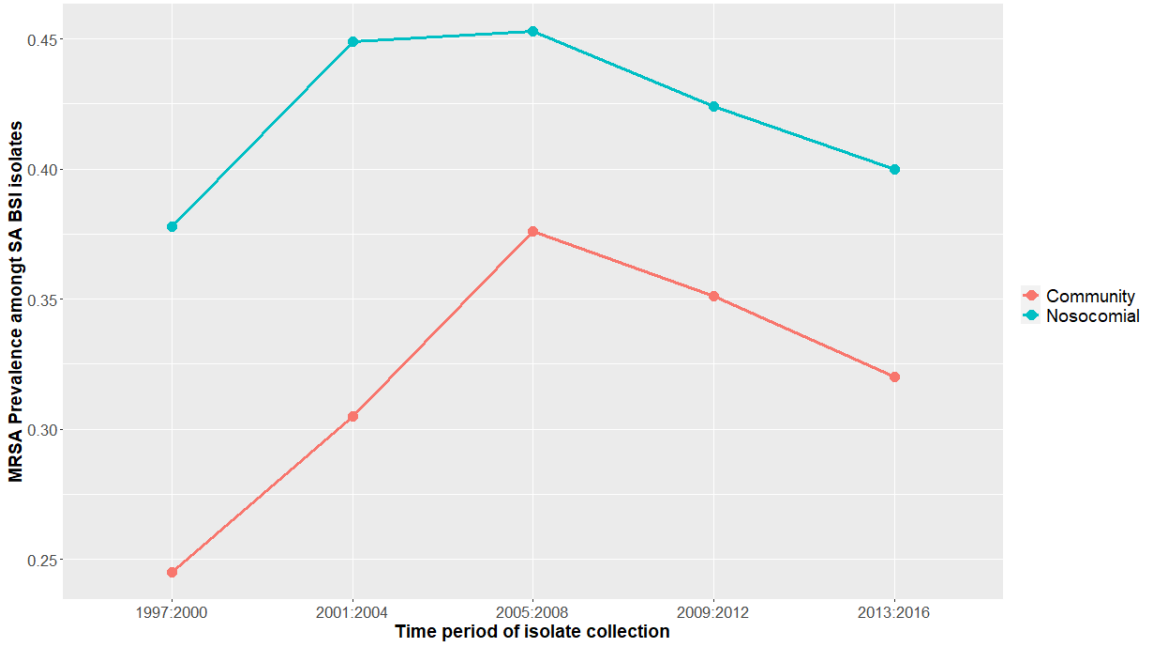


Figure 7: "SENTRY Program 20-year trends in percentage of *Staphylococcus aureus* BSI isolates that are MRSA." [Diekema et al., 2019]

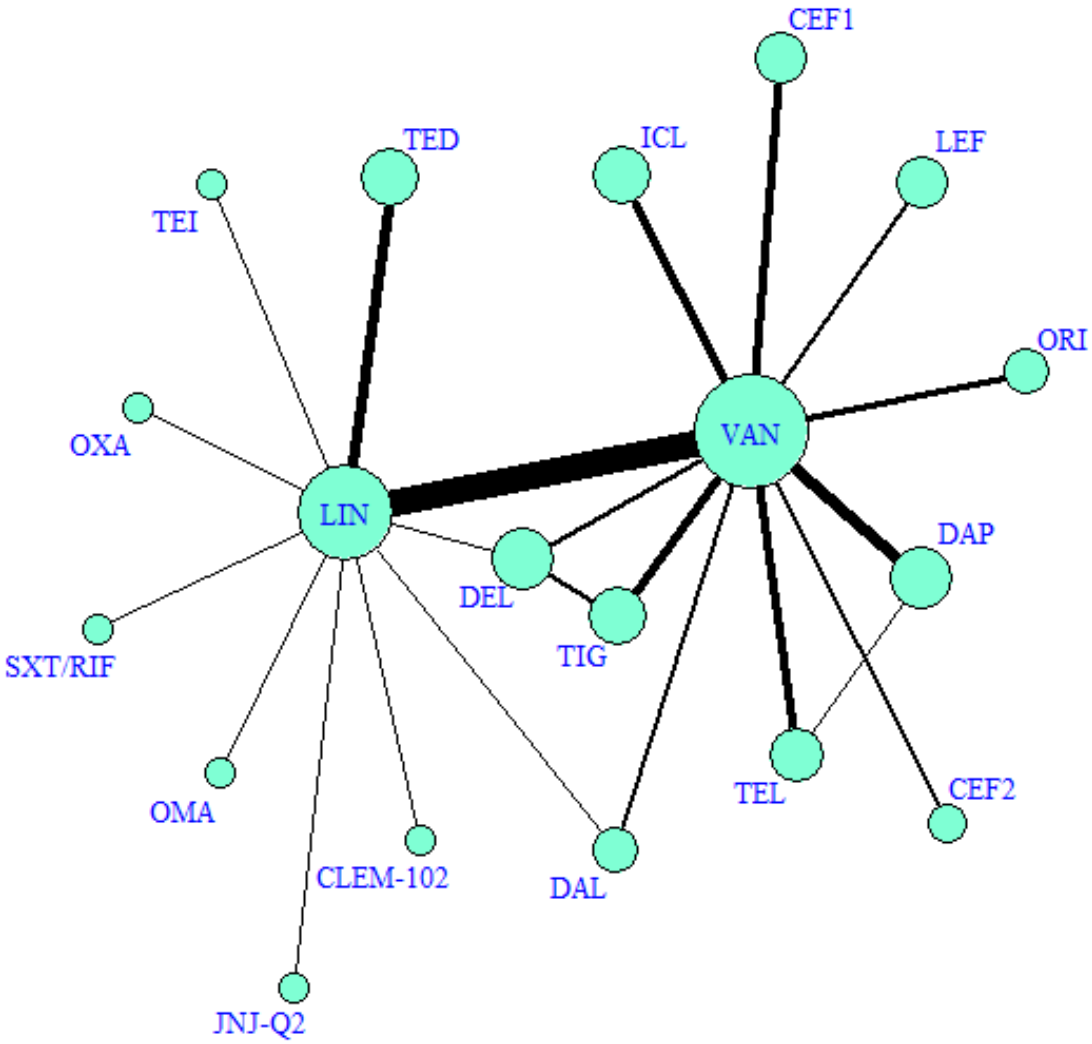


Figure 8: The network of treatments found in the agglomerated dataset. Treatments in larger nodes appeared more often, and the thicker the line between two nodes the more often they were compared.

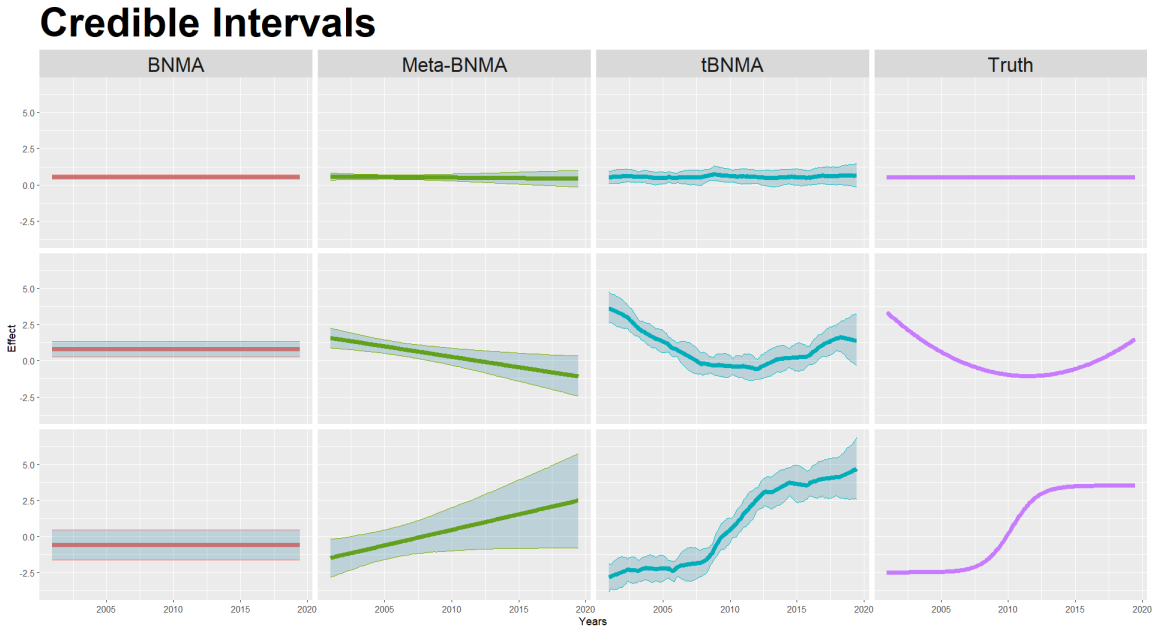


Figure 9: Posterior credible intervals for the $d_{1k}^{t_{i_k}}$ associated with vancomycin in a variety of simulated environments.

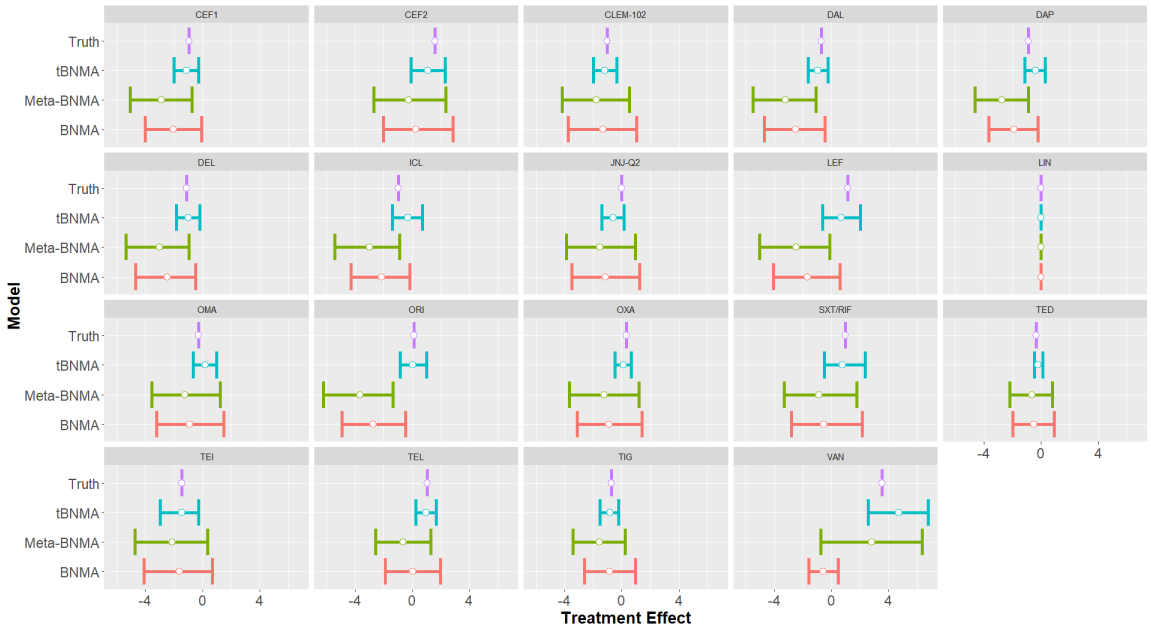


Figure 10: Point estimate and 95% credible interval from the posterior predictive distribution for d_{1k}^{T} by model when there is a sigmoidal time effect on VAN.

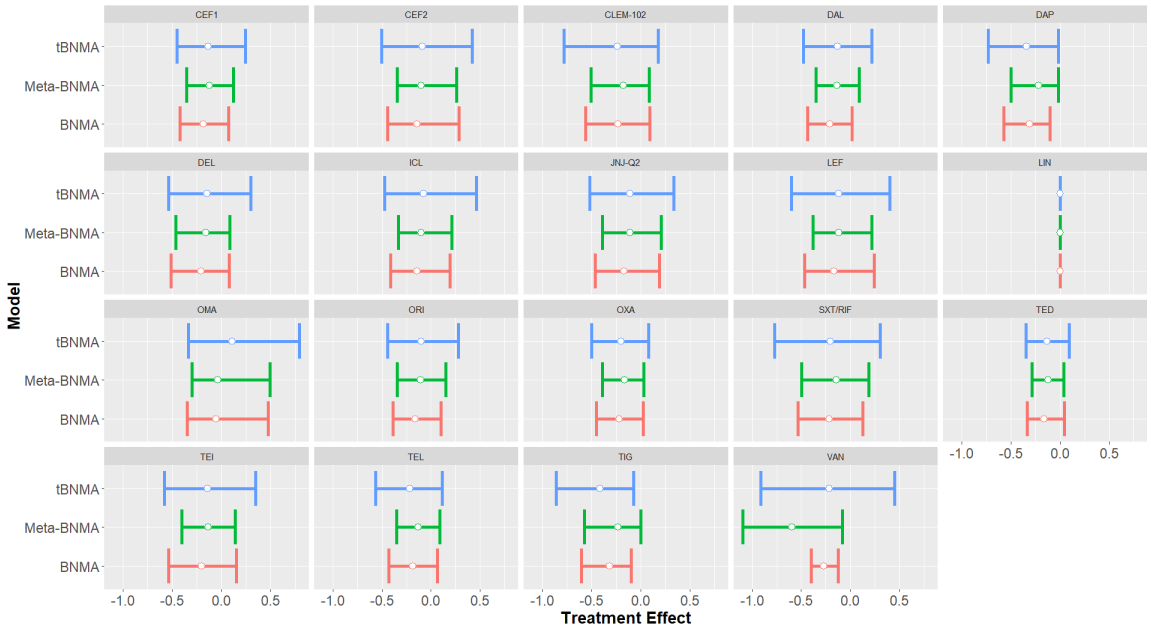


Figure 11: 95% credible intervals and mean estimates for the posterior predictive distribution of d_{1k}^{T} under BNMA, Meta-BNMA, and tBNMA on the agglomerate dataset.

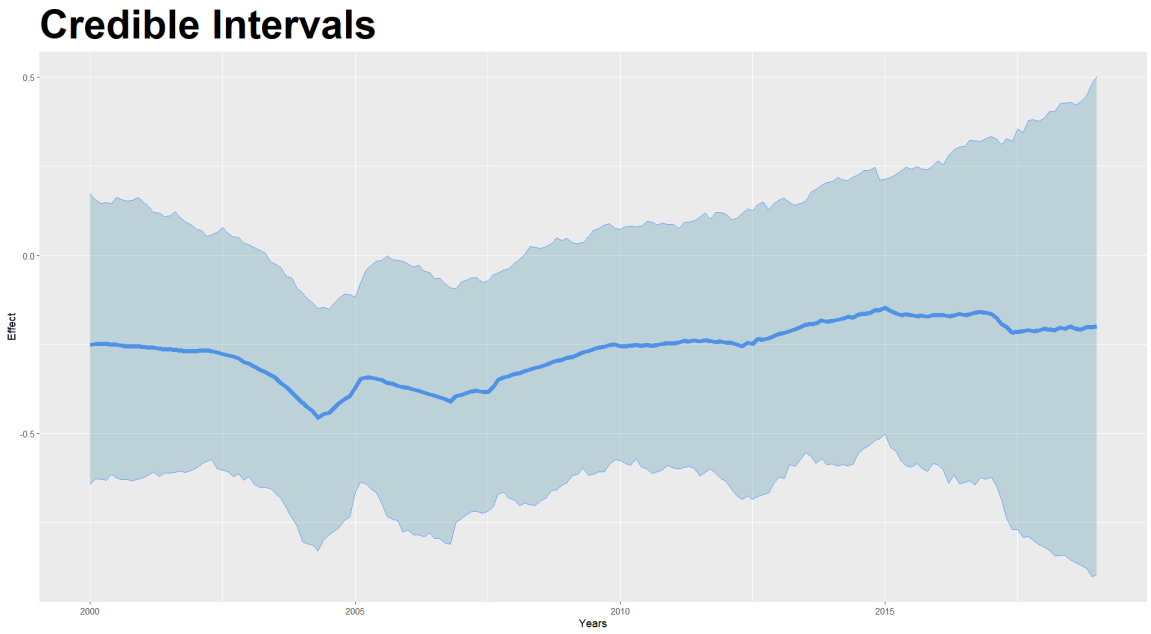


Figure 12: 95% credible intervals for the posterior predictive distribution of the $d_{1k}^{t_{i_k}}$ relating the treatment effect of vancomycin to linezolid under tBNMA on the agglomerated dataset.

5 Concluding Remarks

We investigate three different areas of applied statistics. In Chapter 2, we are interested in appropriately modelling cross-sample heterogeneity in subcommunity composition in microbiome count data. To do so, we propose LTN-LDA: a novel mixed-membership model which models cross-sample heterogeneity using logistic-tree normal distributions on the phylogenetic tree. To make the model conditionally conjugate, we introduce a class of auxiliary Pólya-Gamma variables. The resulting model leads to both improved holistic inference over existing methods and a robustness to overspecifying the number of subcommunities. Though our model is motivated by an application in microbiome data, the methods could be extended to the topic modelling domain. An R package for implementing LTN-LDA is available at https://github.com/PatrickLeBlanc/LTNLDA; data and reproducible code are available at https://github.com/PatrickLeBlanc/ReproduceLTNLDAPaper.

In Chapter 3, we highlight several key findings about recommender systems. Recommender systems are a critically important topic in computational advertising which have heretofore received little attention in the statistical literature. Recently, there has been a combinatorial explosion of recommender system methodologies driven by an increasing number of methods, goals, and settings. Recommender systems have become increasingly bespoke and tailored to individual settings and applications. Much of the ongoing work in the discipline is being performed by computer scientists in industry despite the statistical nature of the underlying models. Thus, there has been a general lack of statistical theory; statisticians can contribute by developing

this theory.

In Chapter 4, we investigate whether MRSA has been developing antibiotic resistance to "gold-standard" treatments such as vancomycin. To do so, we propose tBNMA: a novel Bayesian model which accounts for time-based design inconsistencies in network meta-analyses of randomized controlled trials by modelling time-varying effects as a latent, unobserved, time series drawn from a Gaussian Process combining white noise, linear, and Matern kernels. We combine and analyze data from a collection of NMA-based review papers on MRSA-related cSSSIs using tBNMA. We find that MRSA is not more resistant to vancomycin at the end of the period than at the beginning, but there are substantial non-linear effects. Vancomycin resistance in MRSA was strongest between 2002 and 2007, in line with clinical trends, but has since declined. The time-based nature of this disparity may account for the disagreement about whether linezolid is more effective than vancomycin in the literature. tBNMA could be extended by incorporating a further meta-regression model to "balance" studies with covariate information to those without, as in Jansen [2012] or Phillippo et al. [2020]. Reproducible code and data for Chapter 4 are available at https://github.com/PatrickLeBlanc/tBNMA.

A Supporting Information for Chapter Two

A.1 DTM-LDA

Before developing the LTN-LDA model that we propose in this paper, we had initially attempted to introduce a "DTM-LDA" model, which uses the DTM to model cross-sample variability within topics. However, it turns out that carrying out fully Bayesian inference under DTM-LDA is very computationally demanding for even moderately sized data sets, and this prompted us to seek an alternative solution. We do acknowledge that there may exist alternative strategies outside of fully Bayesian inference such as variational Bayes to achieve scalability approximate inference under DTM-LDA.

We describe the DTM-LDA model that we had initially considered, and demonstrate how the computational difficulties arise. Our full DTM-LDA model is

$$
\begin{aligned}
y_{d,k}(A_l) \,|\, y_{d,k}(A), \theta_{d,k}(A), z_{dn} = k &\propto \text{Bin}(y_{d,k}(A_l) \,|\, y_{d,k}(A), \theta_{d,k}(A)) \\
z_{dn} \,|\, \phi_d &\sim \text{Mult}(1, \phi_d) \\
\phi_d \,|\, \alpha &\sim \text{Dir}(\alpha) \\
\theta_{k,d}(A) \,|\, \theta(A), \tau_k(A) &\sim \text{Beta}(\theta_k(A)\tau_k(A), (1-\theta_k(A))\tau_k(A)) \\
\theta_k(A) \,|\, \theta_0(A), \tau_0(A) &\sim \text{Beta}(\theta_0(A)\tau_0(A), (1-\theta_0(A))\tau_0(A)) \\
\log(\tau_k(A)) &\sim \text{Unif}(1, 7)
\end{aligned}
$$

for $k \in \{1, \ldots, K\}$, $d \in \{1, \ldots, D\}$, $n \in \{1, \ldots, N_d\}$, and $A \in \mathcal{I}$. In this model, we adopted a uniform hyperprior on the log of the per topic dispersion parameter τ_k,

and a Beta hyperprior for the per topic mean parameter θ_k. Note that the computational issues will remain the same no matter which hyperpriors one adopt for these parameters as there are no known conjugate priors for the DT distribution.

Let $\mathbf{z}$ denote a vector encompassing all subcommunity assignments from all samples, $\mathbf{w}$ denote a vector encompassing all sequencing reads from all samples,the superscript $-(d,n)$ indicate that the n^{th} read in the d^{th} sample is excluded, and $\mathcal{P}_{w_{d,n}}$ be a path leading from the root node $\mathcal{R}$ of $\mathcal{T}$ to the leaf corresponding to the sequencing read $w_{d,n}$. Then the form of the full conditional for updating the subcommunity assignments is

$$p(z_{d,n} = k' \,|\, \mathbf{z}^{-(d,n)}, \mathbf{w}) \propto (y_{d,k'}(\mathcal{R})^{-(d,n)} + \alpha_k) \times \frac{\prod_{A \in \mathcal{P}_{w_{d,n}}} \int\int \prod_{d=1}^{D} \left[\frac{B(y_{d,k'}(A_l)+1+\theta_{k'}\tau_{k'}, y_{d,k'}(A_r)+1+(1-\theta_{k'})\tau_{k'})}{B(\theta_{k'}\tau_{k'},(1-\theta_{k'})\tau_{k'})} \right] p(\tau_{k'}) p(\theta_{k'}|\tau_0,\theta_0) d\theta_{k'} d\tau_{k'}}{\prod_{A \in \mathcal{P}_{w_{d,n}}} \int\int \prod_{d=1}^{D} \left[\frac{B(y_{d,k'}(A_l)+\theta_{k'}\tau_{k'}, y_{d,k'}(A_r)+(1-\theta_{k'})\tau_{k'})}{B(\theta_{k'}\tau_{k'},(1-\theta_{k'})\tau_{k'})} \right] p(\tau_{k'}) p(\theta_{k'}|\tau_0,\theta_0) d\theta_{k'} d\tau_{k'}}.$$

There is no closed-form expression for the full conditional, and instead we numerically evaluate the double integral by quadrature. While each individual integral can be computed quickly, for each iteration of the Gibbs sampler we must compute $2 \times \mathcal{P}_{w_{d,n}} \times D \times \bar{N}_d \times K$ of them, where $\bar{N}_d$ is the average number of sequencing reads per document. This results in a Gibbs sampler which is orders of magnitudes slower than the Gibbs sampler for LTN-LDA.

A.2 Block LTN-LDA

We considered more complex covariance priors, but more flexible covariance structures did not lead to improved performance in sumulations and could cause non-identifiability in the model. For demonstration, we implemented the following model, termed Block LTN-LDA, which incorporates a block-diagonal covariance rather than a diagonal covariance in order to maintain identifiability while allowing a bit more flexibility,

$$y_{d,k}(A_l) \,|\, y_{d,k}(A), \psi_{d,k}(A) \overset{\text{ind}}{\sim} \text{Bin}(y_{d,k}(A), \theta_{d,k}(A))$$

$$z_{d,n} \,|\, \boldsymbol{\phi}_d \overset{\text{ind}}{\sim} \text{Mult}(1, \boldsymbol{\phi}_d)$$

$$\boldsymbol{\phi}_d \,|\, \boldsymbol{\alpha} \overset{\text{iid}}{\sim} \text{Dir}(\boldsymbol{\alpha})$$

$$\boldsymbol{\psi}_{d,k} \,|\, \boldsymbol{\mu}_k, \Sigma_k \overset{\text{ind}}{\sim} \text{MVN}(\boldsymbol{\mu}_k, \Sigma_k),$$

$$\boldsymbol{\mu}_k \,|\, \boldsymbol{\mu}_0, \Lambda_0 \overset{\text{iid}}{\sim} \text{MVN}(\boldsymbol{\mu}_0, \Lambda_0)$$

$$\Sigma_k \,|\, \text{G} \overset{\text{iid}}{\sim} \text{G}$$

for $d = 1, \ldots, D$, $k = 1, \ldots, K$, $n = 1, \ldots, N_d$, and $A \in \mathcal{I}$. The form of the model is the same except that G now takes the form of a block covariance prior on Σ_k. That is, let Σ_k^U correspond to the subset of nodes in the upper part of the tree — the set $\{A \in \mathcal{I} || A| \geq C\}$ — and let Σ_k^L correspond to the subset of nodes in the lower part of the tree — the set $\{A \in \mathcal{I} || A| < C\}$. The prior on Σ_k^U we adopt has the form

$$\Sigma_k^U \,|\, \tau_k^U = \text{diag}(\tau_k^U)$$

$$\tau_{i,k}^U \,|\, a^U, b^U \sim \text{IG}(a^U, b^U),$$

as in LTN-LDA. In contrast, we model Σ_k^L as

$$\Omega_k^L = (\Sigma_k^L)^{-1} \,|\, G_k^L \sim \text{GWish}_{G_k^L}(a^L + p^L + 2, b_L + \Phi^L),$$

where p^L is the number of nodes in Σ_k^L and $(a^L, b^L) = (100, 200)$. We draw the precision Ω_k^L of the lower block covariance matrix from a G-Wishart distribution [Lenkoski and Dobra, 2011]. The G-Wishart prior is a suitable covariance prior because it uses a graph to model the dependency structure and so can learn the conditional independence structure of the nodes from the data. Moreover, unlike other Gaussian graphical models such as the Bayesian Graphical Lasso [Wang, 2012], the G-Wishart prior allows us to concentrate the prior anywhere in the real line and control the degree of concentration. This allows us to set the expected level of covariance appropriately while also restraining the posterior values from growing too flexible.

Block LTN-LDA admits a Gibbs sampler similar to LTN-LDA in that every parameter except for Σ_k has the same full conditional. To sample the full conditional for Σ_k^L, we implement the trans-dimensional MCMC sampler described in Mohammadi and Wit [2015] and make use of the direct G-Wishart sampler described in Lenkoski [2013]. However, due to the added complexity in this full conditional, the Gibbs sampler for Block LTN-LDA is significantly slower than the one for LTN-LDA, taking approximately five times as long to complete on datasets of the size used in the paper.

Despite having a more flexible covariance structure, however, Block LTN-LDA did not result in meaningfully better inference than LTN-LDA in our numerical experiments. Specifically, we repeated the analysis in Section 3.1 but simulated from Block

LTN-LDA with a prior probability of $\frac{1}{4}$ that two nodes were dependent; all other parameters remained the same. We then ran LDA, LTN-LDA, and Block LTN-LDA on the dataset for varying K and true C. The results are presented in Figure 13. LDA behaves similarly on a Block LTN-LDA dataset as it does on an LTN-LDA dataset. However, LTN-LDA and Block LTN-LDA predict similar mean posterior ϕ_d as K changes despite Block LTN-LDA having generated the data. We suspect this occurs because LTN-LDA offers a flexible enough covariance structure to capture the cross-sample heterogeneities in the data even though it assumes the nodes are independent: the additional flexibility provided by the G-Wishart priors on the lower block of the covariance matrix did not meaningfully improve inference . This could be related to the strength of the covariance structure and the nature of the Block LTN-LDA prior. There might exist different prior specifications that lead to a more noticeable gap in performance by LTN-LDA and Block LTN-LDA. Moreover, we assume that the covariance structure is related to the nature of the tree—if the nodes were grouped together according to some other specification, a more flexible model such as Block LTN-LDA may be more robust and outperform LTN-LDA. In the context of this assumption and the significant computational burdens induced by Block LTN-LDA, we deem that LTN-LDA has a flexible enough covariance structure to capture heterogeneities across compositions.

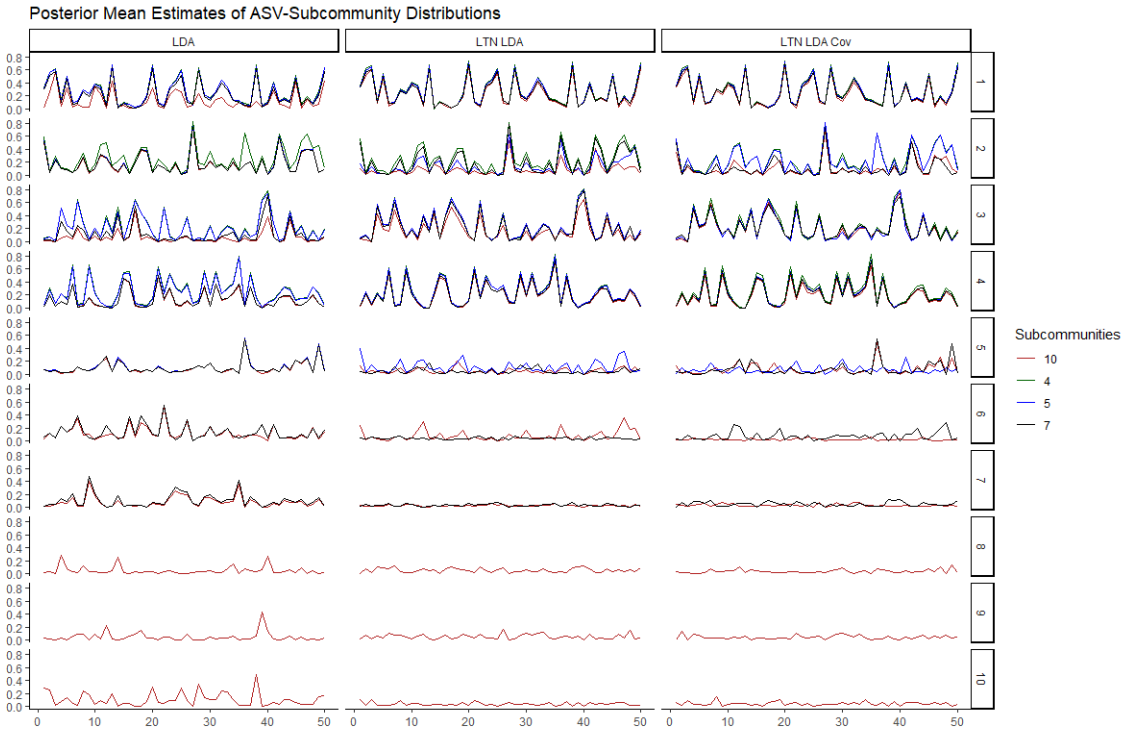


Figure 13: Posterior mean estimates for ϕ as K varies for LDA, LTN-LDA, and Block LTN-LDA.

A.3 Robustness to misspecified trees

The tree structure is vital to the way in which LTN-LDA models cross-sample heterogeneity, and thus it is important to investigate the robustness of the inference to the choice of the tree. To demonstrate this, we generate a dataset as in section 3.1 based on the tree given in Figure 18, and then repeated the analysis comparing LTN-LDA based on this correct tree to LTN-LDA using a misspecified tree as given in Figure 14. The results are presented in Figure 15. The inference provided by the two approaches is similar when $K = 4$, the true value. However, as K increases the mispecified tree's inference deteriorates faster than does the true tree's. Further, we generated a dataset using LTN-LDA with the tree in Figure 18 and ran a perplexity analysis as C varies with the tree in Figure 14 as the tree. The results are in the tree in Figure 16. We can see that the bend in the curve appears to occur before the true value, and so using the mispecified tree can also influence the choice of the tuning parameter. On the other hand, the fitted subcommunity abundances generally maintains the same shape for misspecified K and C, indicating a level of robustness of LTN-LDA with respect to the choice of the tree.

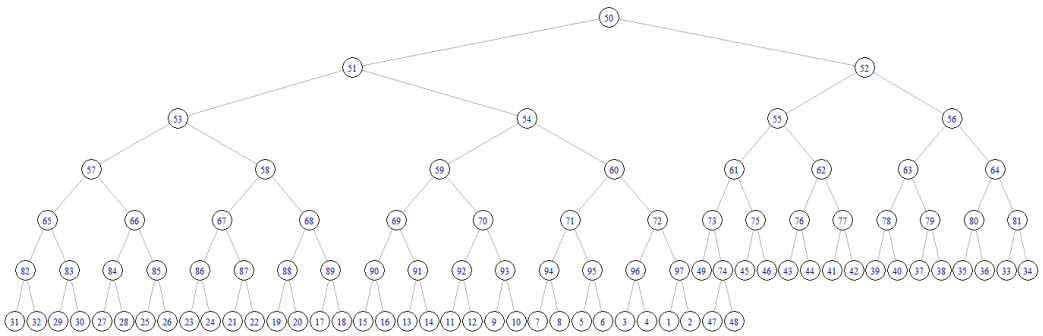


Figure 14: A close to uniform tree constructed from nodes $1, 2, \ldots, 49$.

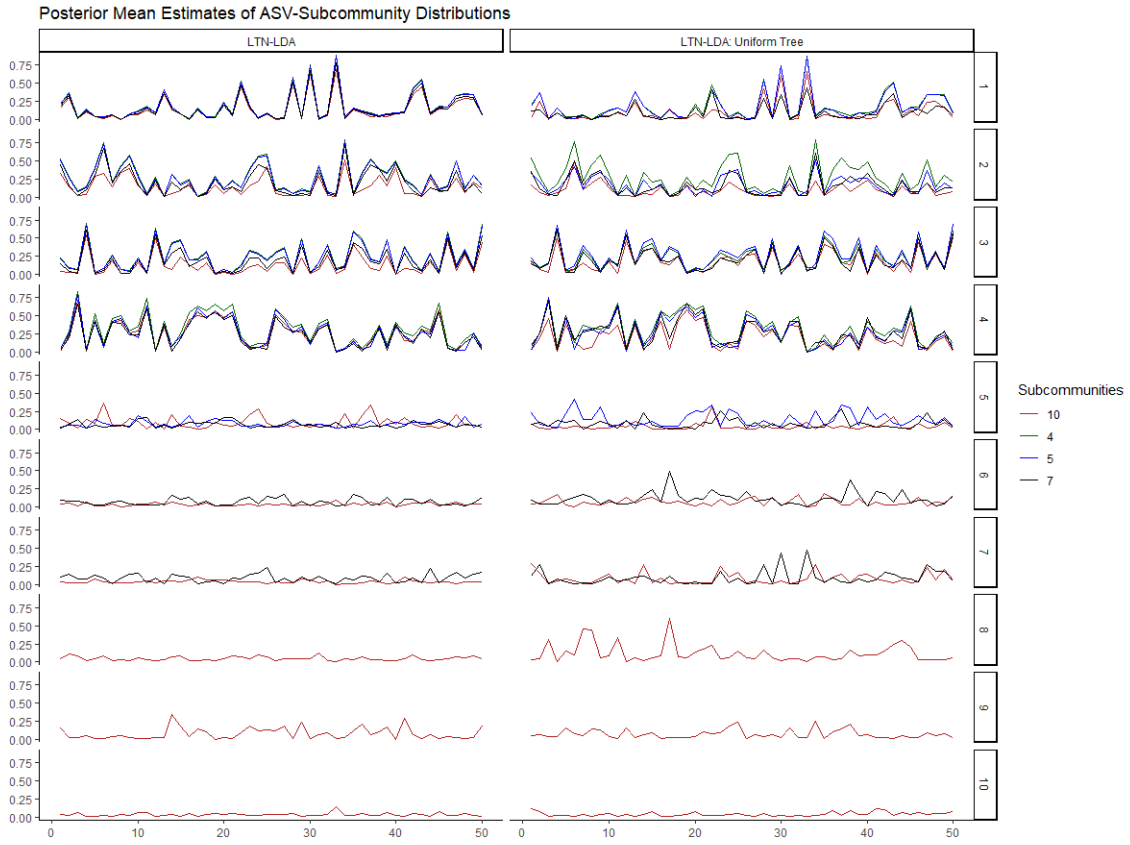


Figure 15: Posterior mean estimates for ϕ as K varies for LTN-LDA using a "true" tree and a uniform tree.

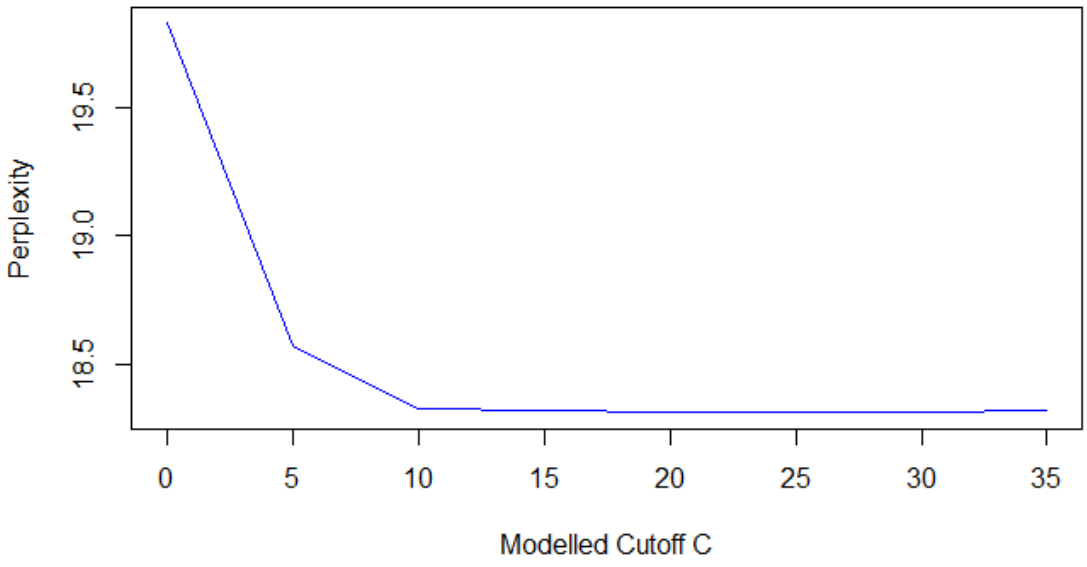


Figure 16: Perplexity results for the mispecified tree as C varies

A.4 Collapsed blocked Gibbs sampler

We integrate the $\boldsymbol{\phi}_d$ out of the sampling model and proceed with a collapsed Gibbs sampler to improve convergence [Griffiths and Steyvers, 2004]. The full conditionals we will sample from are thus

$$(1) \qquad (\boldsymbol{v}_{d,k}, \mathbf{z}_d) \,|\, \boldsymbol{\psi}_{d,k}, \boldsymbol{\mu}_k, \Sigma_k, \Lambda_0, \boldsymbol{\alpha}, a_1, a_2, b \overset{\text{ind}}{\sim} p(\boldsymbol{v_{d,k}}, \mathbf{z}_d | \boldsymbol{\psi}_{d,k}, \boldsymbol{\alpha})$$

$$(2) \qquad \psi_{d,k} \,|\, \boldsymbol{v}_{d,k}, \mathbf{z}_d, \boldsymbol{\mu}_k, \Sigma_k, \Lambda_0, \boldsymbol{\alpha}, a_1, a_2, b \overset{\text{ind}}{\sim} p(\boldsymbol{\psi}_{d,k} | \boldsymbol{v}_{d,k}, \mathbf{z}_d, \boldsymbol{\mu}_k, \Sigma_k)$$

$$(3) \qquad \boldsymbol{\mu}_k \,|\, \boldsymbol{v}_{d,k}, \mathbf{z}_d, \boldsymbol{\psi}_{d,k}, \Sigma_k, \Lambda_0, \boldsymbol{\alpha}, a_1, a_2, b \overset{\text{ind}}{\sim} p(\boldsymbol{\mu}_k | \boldsymbol{\psi}_{d,k}, \Sigma_k, \Lambda_0)$$

$$(4) \qquad \Sigma_k \,|\, \boldsymbol{v}_{d,k}, \mathbf{z}_d, \boldsymbol{\psi}, \boldsymbol{\mu}_k, \Lambda_0, \boldsymbol{\alpha}, a_1, a_2, b \overset{\text{ind}}{\sim} p(\Sigma_k | \boldsymbol{\psi}, \boldsymbol{\mu}_k, \Lambda_0, a_1, a_2, b),$$

The joint full conditional $(\boldsymbol{v_{d,k}}, \mathbf{z}_d)$ is

$$\mathbf{z}_d \,|\, \boldsymbol{\psi}_{d,k}, \boldsymbol{\alpha} \overset{\text{ind}}{\sim} p(\mathbf{z}_d | \boldsymbol{\psi}_{d,k}, \boldsymbol{\alpha})$$

$$\boldsymbol{v_{d,k}} \,|\, \mathbf{z}_d, \boldsymbol{\psi}_{d,k}, \boldsymbol{\alpha} \overset{\text{ind}}{\sim} p(\boldsymbol{v_{d,k}} | \mathbf{z}_d, \boldsymbol{\psi}_{d,k}).$$

To sample the vector $\mathbf{z}_d$ from its full conditional, we sample each subcommunity assignment in order from its multinomial full conditional:

$$p(z_{d,n} = k | \mathbf{z}_d^{-n}, \boldsymbol{\psi}_{d,k}, \boldsymbol{\alpha}) \propto (y_{d,k}(\mathcal{R})^{-n} + \alpha) \times \beta_k^{w_{d,n}},$$

where $\mathbf{z}_d^{-n}$ is the vector of all subcommunity assignments in sample d except for $z_{d,n}$ and $y_{d,k}(\mathcal{R})^{-n}$ is the number of sequencing reads in sample d descended from the root node $\mathcal{R}$ assigned to subcommunity k not counting the n^{th} token. To sample from the

full conditional for $\boldsymbol{v_{d,k}}$, we draw $v_{d,k}(A)$ for each $A \in \mathcal{I}$:

$$v_{d,k}(A) \,|\, y_{d,k}(A), \psi_{d,k}(A) \overset{\text{ind}}{\sim} \text{PG}(y_{d,k}(A), \psi_{d,k}(A)),$$

the conjugate full conditional of a Pólya-Gamma distribution derived in Polson et al. [2013]. However, existing Pólya-Gamma samplers are slow for the current context and so for $y_{d,k}(A) \geq 30$ we use an approximate Pólya-Gamma sampler proposed in Glynn et al. [2019], which uses the Central Limit Theorem to approximate a normal distribution:

$$\text{N}\left(\frac{y_{d,k}(A)^2}{2\psi_{d,k}(A)}\tanh\left(\frac{\psi_{d,k}(A)}{2}\right), \frac{y_{d,k}(A)^2}{4\psi_{d,k}(A)^3}\text{sech}^2\left(\frac{\psi_{d,k}(A)}{2}\right)(\sinh(\psi_{d,k}(A)) - \psi_{d,k}(A))\right).$$

The full conditionals for $\boldsymbol{\mu}_k$ and the $\tau_{i,k}$ follow by conjugate updating:

$$\boldsymbol{\mu}_k \,|\, \boldsymbol{\psi}_{d,k}, \Sigma_k, \Lambda_0 \overset{\text{ind}}{\sim} \text{MVN}\left((\Lambda_0^{-1} + D\Sigma_k^{-1})^{-1}\Sigma_k^{-1}\sum_{d=1}^{D}\psi_{d,k}, (\Lambda_0^{-1} + D\Sigma_k^{-1})^{-1}\right)$$

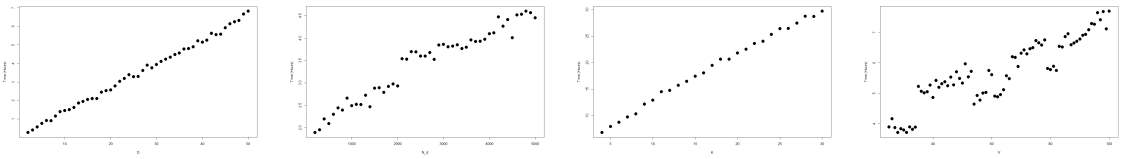
$$\tau_{i,k} \,|\, \boldsymbol{\psi}, \boldsymbol{\mu}_k, , a_1, a_2, b \overset{\text{ind}}{\sim} \text{IG}\left(a_1 + \frac{D}{2}, \frac{2b + \sum_{d=1}^{D}(\psi_{d,k}(A_i) - \mu_k(A_i)^2)}{2}\right) \quad \text{if } |A_i| \geq C$$

$$\tau_{i,k} \,|\, \boldsymbol{\psi}, \boldsymbol{\mu}_k, , a_1, a_2, b \overset{\text{ind}}{\sim} \text{IG}\left(a_2 + \frac{D}{2}, \frac{2b + \sum_{d=1}^{D}(\psi_{d,k}(A_i) - \mu_k(A_i)^2)}{2}\right) \quad \text{if } |A_i| < C.$$

Further, the full conditional for $\boldsymbol{\psi}_{d,k}$ is also normal,

$$\boldsymbol{\psi}_{d,k} \,|\, \mathbf{z}_d, \boldsymbol{v}_{d,k}, \mu_k, \Sigma_k \overset{\text{ind}}{\sim} \text{MVN}\left((\Sigma_k^{-1} + \text{diag}(\boldsymbol{v}_{d,k}))^{-1}(\Sigma_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\kappa}_{d,k}), (\Sigma_k^{-1} + \text{diag}(\boldsymbol{v}_{d,k}))^{-1}\right).$$

The Gibbs sampling algorithm scales linearly with D (Figure 17a), N_d (Fig-



(a) Scaling with D (b) Scaling with N_d (c) Scaling with K (d) Scaling with V

Figure 17: Scalings

ure 17*b*), K (Figure 17*c*), and V (Figure 17*d*). The computation time does not scale with the tree parameters $\mathcal{T}$ and C because of the diagonal covariance structure.

A.5 The phylogenetic tree used in the simulation study

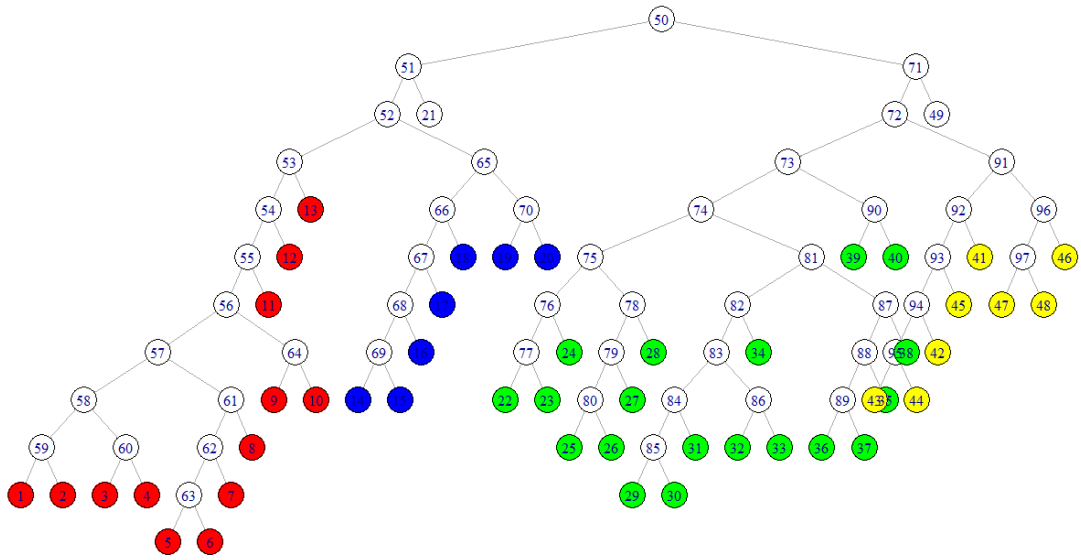


Figure 18: The phylogenetic tree used in simulations

A.6 Perplexity

Perplexity is a transformation of predictive log-likelihood commonly used to assess topic models. If $p(\mathbf{w}_{d^{(te)}}|\mathcal{M})$ is the predictive log-likelihood of a test set sample $d^{(te)}$ given a collection of parameters $\mathcal{M}$, then perplexity is defined as

$$\exp\left(-\frac{\sum_d p(\mathbf{w}_{d^{(te)}}|\mathcal{M})}{\sum_d N_{d^{(te)}}}\right),$$

where $N_{d^{(te)}}$ is the number of sequencing reads in $d^{(te)}$

The document completion method for computing perplexity for LDA is described in section 5 of Wallach et al. [2009]. It involves splitting each sample $d^{(te)}$ in the test set into two halves, $d^{(te),1}$ and $d^{(te),2}$. A modified Gibbs sampler is run on the first half, $d^{(te),1}$ with the value of β_k set equal to the posterior mean of β_k on the training set. The results of this Gibbs sampler are used to develop estimates for ϕ_d and then for perplexity.

We modify this procedure for LTN-LDA. A modified Gibbs is run on $d^{(te),1}$, fixing the values of μ_k and Σ_k at their posterior means from the training set. If there are I iterations in the Gibbs sampler, then the estimate of $\boldsymbol{\phi}_d$ at iterate i is $\boldsymbol{\phi}_d(i)$ and the estimate of $\boldsymbol{\beta}_{d,k}$ of iterate i is $\boldsymbol{\beta}_{d,k}(i)$. We can then take a Monte Carlo estimate over all ASVs observed in $d^{(te),2}$ to estimate the predictive likelihood of d^{te}:

$$\frac{1}{I}\sum_{i=1}^{I}\sum_{n\in\mathbf{w}_{d^{(te),2}}}\log\left(\sum_{k=1}^{K}\phi_d^k(i)\beta_{d,k}^{w_{d,n}}(i)\right).$$

This procedure can be repeated for every sample in the test set, and the resulting set

of predictive likelihood estimates can be transformed into a perplexity estimate.

A.7 Dethlefsen and Relman Tree

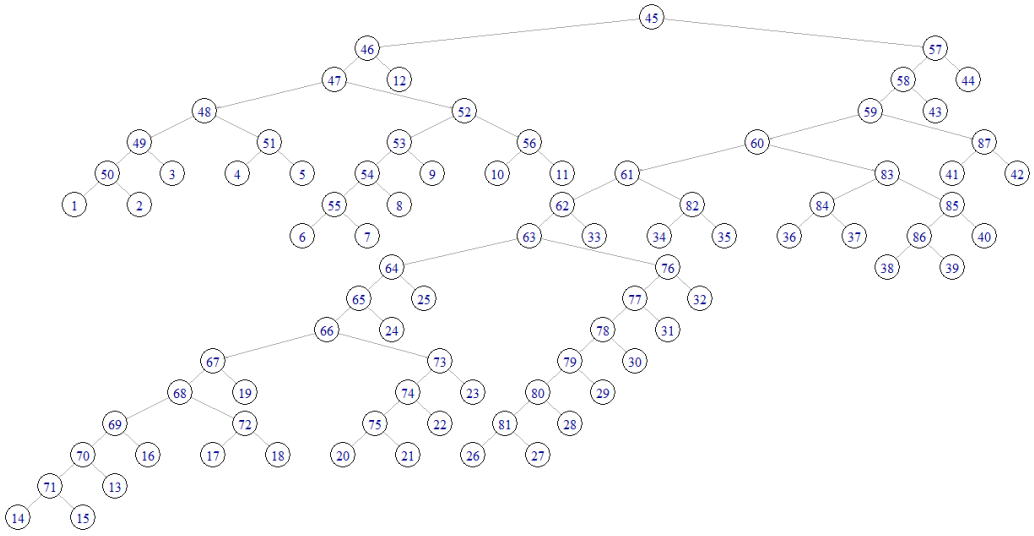


Figure 19: The tree resulting from the dataset of Dethlefsen and Relman [2011]

A.8 Separating the effects of the tree from that of the random effect

LTN-LDA incorporates two new effects: the tree structue and random effects in cross-sample heterogeneity. We note that, in the context of LTN-LDA, using the tree structure alone without allowing random effects does not improve inference in any way.

We provide evidence that the tree structure alone without the random effects does not improve the inference over LDA. We "knock out" the random effects by forcing the sample-specific distributions $\beta_{d,k}$ to not vary from sample-to-sample. Thus, we can approximate this model with an existing Gibbs sampler. We then replicate the results of Section 3.1 of the manuscript but comparing LTN-LDA with the usual

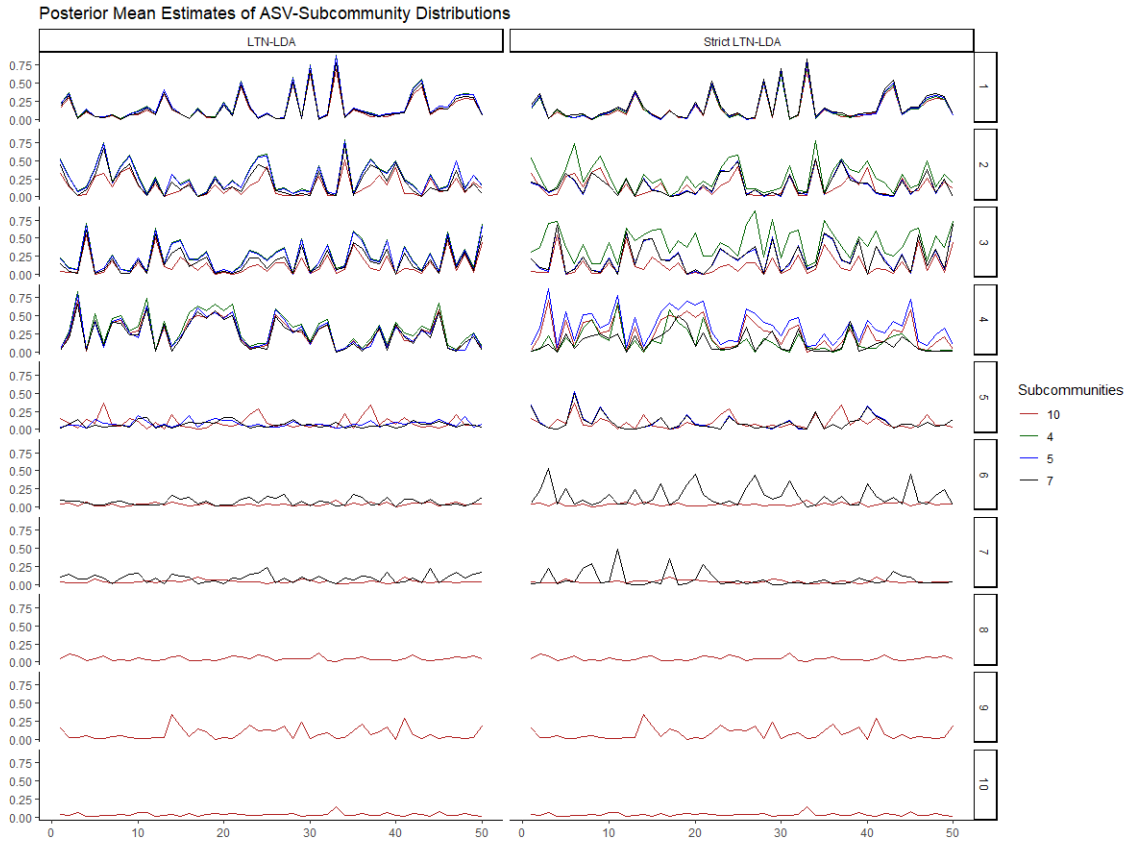


Figure 20: Posterior mean estimates for ϕ as K varies for LTN-LDA with recommended covariance priors vs those with "knock-out" covariance priors.

prior to LTN-LDA with this "knock-out" covariance prior, and present the results in Figure 20. The version with strict priors misestimates subcommunity proportions for $K = 4$, and splits the subcommunities as K grows. We thus deduce that using the tree structure without allowing cross-sample heterogeneity does not reproduce the positive results in the paper.

However, it is difficult to implement a model with just random effects without a tree structures. On the modeling side, without the tree structure, one must induce more complex constraints on the covariance to ensure identifiability. Moreover, without the tree structure, it is unclear how these models can be implemented efficiently

to be applicable to modern microbiome data sets. It would be interesting to see such a comparison so that we can understand to what extent to improvement is due to the tree modeling assumption. However, we know of no such existing implementation of the models suggested. Note that even the seemingly simple Dirichlet random effect model will require a high-dimensional (m-dim) numerical integral where m is the number of taxa within each iteration for the same reason that the DTM requires numerical integration as we mentioned before—there is no known conjugate priors for the parameters in the Dirichlet distribution. For these reasons, we feel that such a comparison goes beyond the scope of this manuscript. Finally, we emphasize that it is indeed the adoption of a tree structure that provides an efficient means to computing. We believe in this regard our use of the tree goes beyond prior works that uses the tree only for modeling purposes, not a computational technique.

References

AE Ades. A chain of evidence with mixed comparisons: Models for multi-parameter synthesis and consistency of evidence. *Statistics in medicine*, 22:2995–3016, 10 2003. doi: 10.1002/sim.1566.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005a. doi: 10.1109/TKDE.2005.99.

Gediminas Adomavicius and Alexander Tuzhilin. Personalization technologies: A process-oriented perspective. *Commun. ACM*, 48(10):83–90, oct 2005b. ISSN 0001-0782. doi: 10.1145/1089107.1089109. URL https://doi.org/10.1145/1089107.1089109.

Charu C. Aggarwal. *Recommender Systems - The Textbook*. Springer, 2016. ISBN 978-3-319-29659-3.

John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177, 1982.

Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. Rate it again: Increasing recommendation accuracy by user re-rating. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, page 173–180, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584355. doi: 10.1145/1639714.1639744. URL https://doi.org/10.1145/1639714.1639744.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 25–32, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553378. URL https://doi.org/10.1145/1553374.1553378.

Poonam B. Thorat, R. Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110:31–36, 01 2015. doi: 10.5120/19308-0760.

M. Balabanovic and Y Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Linas Baltrunas and Xavier Amatriain. Towards time-dependant recommendation based on implicit feedback. *Proceedings of the Third ACM Conference on Recommender Systems*, 01 2009.

Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.*, 52(1):1–37, jun 2019. ISSN 0269-2821. doi: 10.1007/s10462-018-9654-y. URL https:

//doi.org/10.1007/s10462-018-9654-y.

Punam Bedi, Pooja Vashisth, Purnima Khurana, and Preeti. Modeling user preferences in a hybrid recommender system using type-2 fuzzy sets. In *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2013. doi: 10.1109/FUZZ-IEEE.2013.6622471.

Joeran Beel, Marcel Genzmehr, Stefan Langer, Andreas Nürnberger, and Bela Gipp. A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, RepSys '13, page 7–14, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324656. doi: 10.1145/2532508.2532511. URL https://doi.org/10.1145/2532508.2532511.

Robert Bell, Yehuda Koren, Yahoo Research, and Israel Volinsky. The bellkor 2008 solution to the netflix prize. *AT&T Research*, 01 2008.

Xuan Bi, Annie Qu, Junhui Wang, and Xiaotong Shen. A group-specific recommender system. *Journal of the American Statistical Association*, 112, 09 2016. doi: 10.1080/01621459.2016.1219261.

Daniel Billsus, Michael J. Pazzani, and James Chen. A learning agent for wireless news access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, page 33–36, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581131348. doi: 10.1145/325737.325768. URL https://doi.org/10.1145/325737.325768.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowl. Based Syst.*, 46:109–132, 2013.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26:225–238, 2012. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2011.07.021. URL https://www.sciencedirect.com/science/article/pii/S0950705111001882.

Sofiane Brahim-Belhouari and Amine Bermak. Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, 47(4):705–712, 2004. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2004.02.006. URL https://www.sciencedirect.com/science/article/pii/S0167947304000301.

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 43–52, San Francisco, CA,

USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*, 2013.

Nick Brown, Anna Goodman, Carolyne Horner, Abi Jenkins, and Erwin Brown. Treatment of methicillin-resistant staphylococcus aureus (mrsa): updated guidelines from the uk. *JAC-Antimicrobial Resistance*, 3, 01 2021. doi: 10.1093/jacamr/dlaa114.

Heiner C. Bucher, Gordon H. Guyatt, Lauren E. Griffith, and Stephen D. Walter. The results of direct and indirect treatment comparisons in meta-analysis of randomized controlled trials. *Journal of clinical epidemiology*, 50 6:683–91, 1997.

Robin Burke. Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69, 05 2000.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12, 11 2002. doi: 10.1023/A:1021240730564.

Robin Burke. Hybrid web recommender systems. volume 4321, 01 2007. ISBN 978-3-540-72078-2. doi: 10.1007/978-3-540-72079-9_12.

Robin Burke, Kristian Hammond, and Benjamin Young. The findme approach to assisted browsing. *IEEE Expert*, 12:32 – 40, 08 1997. doi: 10.1109/64.608186.

Benjamin J Callahan, Paul J McMurdie, and Susan P Holmes. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *The ISME Journal*, 11:2639–2643, 2017.

Laurent Candillier, Isabelle Tellier, Fabien Torre, and Olivier Bousquet. Ssc : Statistical subspace clustering. 07 2005. ISBN 978-3-540-26923-6. doi: 10.1007/11510888_11.

Laurent Candillier, Frank Meyer, and Marc Boullé. Comparing state of the art collaborative filtering systems. volume 4571, pages 548–562, 07 2007. ISBN 978-3-540-73498-7. doi: 10.1007/978-3-540-73499-4_41.

Erion Cano and Maurizio Morisio. Hybrid recommender systems: A systematic literature review. *Intell. Data Anal.*, 21:1487–1524, 2017.

Li Chen and Pearl Pu. Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22, 04 2012. doi: 10.1007/s11257-011-9108-6.

Jie Cheng and Russell Greiner. Learning bayesian belief network classifiers: Algorithms and system. In *Canadian Conference on AI*, 2001.

Hathaikan Chootrakool and Jian Shi. Meta-analysis of multi-arm trials using empiri-

cal logistic transform. *The open medical informatics journal*, 2:112–6, 02 2008. doi: 10.2174/1874431100802010112.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 815–824, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939746. URL https://doi.org/10.1145/2939672.2939746.

M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, California, 1999. ACM.

C. Cleverdon and M. Kean. Factors determining the performance of indexing systems. 1968.

Giorgio Corani, Alessio Benavoli, and Marco Zaffalon. Time series forecasting with gaussian processes needs priors. *arXiv: Machine Learning*, Sep 2020.

S. E. Cosgrove, K. C. Carroll, and T. M. Perl. Staphylococcus aureus with Reduced Susceptibility to Vancomycin. *Clinical Infectious Diseases*, 39(4):539–545, 08 2004. ISSN 1058-4838. doi: 10.1086/422458. URL https://doi.org/10.1086/422458.

Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci. An evaluation methodology for collaborative recommender systems. pages 224 – 231, 12 2008. doi: 10.1109/AXMEDIS.2008.13.

Nancy Crum, Rachel Lee, Scott Thornton, O Stine, Mark Wallace, Chris Barrozo, Ananda Keefer-Norris, Sharon Judd, and Kevin Russell. Fifteen-year study of the changing epidemiology of methicillin-resistant staphylococcus aureus. *The American journal of medicine*, 119:943–51, 11 2006. doi: 10.1016/j.amjmed.2006.01.004.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 101–109, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362436. doi: 10.1145/3298689.3347058. URL https://doi.org/10.1145/3298689.3347058.

Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and D. Jannach. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, 39:1 – 49, 2021.

Robert Daum. Skin and soft-tissue infections caused by methicillin-resistant staphylococcus aureus. *The New England journal of medicine*, 357:380–90, 08 2007. doi: 10.1056/NEJMcp070747.

Rebecca A Deek and Hongzhe Li. A zero-inflated latent dirichlet allocation model for microbiome studies. *Frontiers in Genetics*, 11:599–614, 2019.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41:391–407, 1990.

Federica Del Chierico, Pamela Vernocchi, Bruno Dallapiccola, and Lorenza Putignani. Mediterranean diet and health: Food effects on gut microbiota and disease control. *International Journal of Molecular Sciences*, 15(7):11678–11699, 2014.

Samuel Y. Dennis III. On the hyper-dirichlet type 1 and hyper-liouville distributions. *Communications in Statistics - Theory and Methods*, 20(12):4069–4081, 1991. doi: 10.1080/03610929108830757. URL https://doi.org/10.1080/03610929108830757.

Les Dethlefsen and David A. Relman. Incomplete recovery and individualized responses of the human distal gut microbiota to repeated antibiotic perturbation. *Proceedings of the National Academy of the Sciences of the United States of America*, 108(Supplement 1):4554–4561, 2011.

Sofia Dias, Nicky Welton, Alex Sutton, and Ae Ades. Nice dsu technical support document 2: A generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials. *National Institute for Health and Clinical Excellence (NICE)*, 01 2011.

Daniel J Diekema, Michael A Pfaller, Dee Shortridge, Marcus Zervos, and Ronald N Jones. Twenty-year trends in antimicrobial susceptibilities among staphylococcus aureus from the sentry antimicrobial surveillance program. *Open Forum Infectious Diseases*, 6(Supplement 1):S47–S53, 03 2019. ISSN 2328-8957. doi: 10.1093/ofid/ofy270. URL https://doi.org/10.1093/ofid/ofy270.

Pedro M. Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 2004.

Finale Doshi-Velez, Byron C Wallace, and Ryan Adams. Graph-sparse lda: A topic model with structured sparsity. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2575–2581. AAAI Press, 2015. ISBN 0262511290.

Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, and John T. Riedl. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 133–140, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306836. doi: 10.1145/2043932.2043958. URL https://doi.org/10.1145/2043932.2043958.

Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

Jingjuan Feng, Feng Xiang, Jian Cheng, Yeli Gou, and Jun Li. Comparative efficacy and safety of vancomycin, linezolid, tedizolid, and daptomycin in treating patients with suspected or proven complicated skin and soft tissue infections: An updated network meta-analysis. *Infectious Diseases and Therapy*, 10, 06 2021. doi: 10.1007/s40121-021-00456-0.

Julia Fukuyama, Kris Sankaran, and Laura Symul. Multiscale analysis of count data through topic alignment. *Biostatistics*, 06 2022. ISSN 1465-4644. doi: 10.1093/biostatistics/kxac018. URL https://doi.org/10.1093/biostatistics/kxac018. In press.

Simon Funk. Netflix update: Try this at home, 2006.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.06.002. URL https://www.sciencedirect.com/science/article/pii/S2666651021000164.

Michael Gasser, Walter Zingg, Alessandro Cassini, and Andreas Kronenberg. Attributable deaths and disability-adjusted life-years caused by infections with antibiotic-resistant bacteria in switzerland. *The Lancet Infectious Diseases*, 19, 11 2019. doi: 10.1016/S1473-3099(18)30708-4.

Chris Glynn, Surya T. Tokdar, Brian Howard, and David L. Banks. Bayesian Analysis of Dynamic Linear Topic Models. *Bayesian Analysis*, 14(1):53 – 80, 2019.

Kenneth Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4:133–151, 07 2001. doi: 10.1023/A:1011419012209.

Ian Gould, Michael David, Silvano Esposito, Javier Garau, Gerard Lina, Teresita Mazzei, and Georg Peters. New insights into meticillin-resistant staphylococcus aureus (mrsa) pathogenesis, treatment and resistance. *International journal of antimicrobial agents*, 39:96–104, 12 2012. doi: 10.1016/j.ijantimicag.2011.09.028.

Neal Grantham, Brian Reich, Elizabeth Borer, and Kevin Gross. Mimix: a bayesian mixed-effects model for microbiome data from designed experiments. *Journal of the American Statistical Association*, 115, 03 2017. doi: 10.1080/01621459.2019.1626242.

Russ Greiner and Wei Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. pages 167–173, 01 2002.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

Julian Guest, Jaime Esteban, Anton Manganelli, Andrea Novelli, Giuliano Rizzardini,

and Miquel Serra-Burriel. Comparative efficacy and safety of antibiotics used to treat acute bacterial skin and skin structure infections: Results of a network meta-analysis. *PLOS ONE*, 12:e0187792, 11 2017. doi: 10.1371/journal.pone.0187792.

Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, dec 2009. ISSN 1532-4435.

Bastiaan W Haak, Jacqueline M Lankelma, Floor Hugenholtz, Clara Belzer, Willem M de Vos, and W Joost Wiersinga. Long-term impact of oral vancomycin, ciprofloxacin and metronidazole on the gut microbiota in healthy humans. *Journal of Antimicrobial Chemotherapy*, 74(3):782–786, 11 2018. ISSN 0305-7453.

Javeria Habib, Shuo Zhang, and Krisztian Balog. Iai moviebot: A conversational movie recommender system. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3405–3408, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3417433. URL https://doi.org/10.1145/3340531.3417433.

Gholamreza Haffari and Yee Whye Teh. Hierarchical dirichlet trees for information retrieval. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, page 173–181, USA, 2009. Association for Computational Linguistics.

Larry Hardesty. The history of amazon's recommendation algorithm: Collaborative filtering and beyond, Nov 2019. URL https://www.amazon.science/the-history-of-amazons-recommendation-algorithm.

Negar Hariri, Bamshad Mobasher, and Robin Burke. Context adaptation in interactive recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, page 41–48, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450326681. doi: 10.1145/2645710.2645753. URL https://doi.org/10.1145/2645710.2645753.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Chen He, Denis Parra, and Katrien Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56:9–27, 2016. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.02.013. URL https://www.sciencedirect.com/science/article/pii/S0957417416300367.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International*

Conference on World Wide Web, WWW '17, page 173–182, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052569. URL https://doi.org/10.1145/3038912.3052569.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, jan 2004. ISSN 1046-8188. doi: 10.1145/963770.963772. URL https://doi.org/10.1145/963770.963772.

Adam Hersh, Henry Chambers, Judith Maselli, and Ralph Gonzales. National trends in ambulatory visits and antibiotic prescribing for skin and soft-tissue infections. *Archives of internal medicine*, 168:1585–91, 08 2008. doi: 10.1001/archinte.168.14.1585.

J. Higgins, D Jackson, Jessica Barrett, Guobing Lu, A. Ades, and I. White. Consistency and inconsistency in network meta-analysis: Concepts and models for multi-arm studies. *Research Synthesis Methods*, 3, 06 2012. doi: 10.1002/jrsm.1044.

Julian P. T. Higgins and Anne Whitehead. Borrowing strength from external trials in a meta-analysis. *Statistics in medicine*, 15 24:2733–49, 1996.

Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, jan 2004. ISSN 1046-8188. doi: 10.1145/963770.963774. URL https://doi.org/10.1145/963770.963774.

Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, page 688–693, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

Ian Holmes, Keith Harris, and Christopher Quince. Dirichlet multinomial mixtures: Generative models for microbial metagenomics. *PloS one*, 7:e30126, 02 2012. doi: 10.1371/journal.pone.0030126.

Zeng-Wei Hong, Rui-Tang Huang, Kai-Yi Chin, Chia-Chi Yen, and Jim-Min Lin. An interactive agent system for supporting knowledge-based recommendation: A case study on an e-novel recommender system. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605588933. doi: 10.1145/2108616.2108681. URL https://doi.org/10.1145/2108616.2108681.

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 193–201, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences

Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052639. URL https://doi.org/10.1145/3038912.3052639.

Andrea Iovine, Fedelucio Narducci, and Giovanni Semeraro. Conversational recommender systems and natural language:: A study through the converse framework. *Decision Support Systems*, 131:113250, 2020. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2020.113250. URL https://www.sciencedirect.com/science/article/pii/S0167923620300051.

D. Jannach and Ahtsham Manzoor. End-to-end learning for conversational recommendation: A long way to go? In *IntRSatRecSys*, 2020.

D. Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54:1 – 36, 2021.

Dietmar Jannach, Lukas Lerche, Fatih Gedikli, and Geoffray Bonnin. What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects. In *International conference on user modeling, adaptation, and personalization*, pages 25–37. Springer, 2013.

Jeroen Jansen. Network meta-analysis of survival data with fractional polynomials. *BMC medical research methodology*, 11:61, 05 2011. doi: 10.1186/1471-2288-11-61.

Jeroen Jansen. Network meta-analysis of individual and aggregate level data. *Research Synthesis Methods*, 3, 06 2012. doi: 10.1002/jrsm.1048.

Jeroen Jansen, M Vieira, and Shannon Cope. Network meta-analysis of longitudinal data using fractional polynomials. *Statistics in medicine*, 34, 04 2015. doi: 10.1002/sim.6492.

Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. Characterisation of explicit feedback in an online music recommendation service. pages 317–320, 09 2010. doi: 10.1145/1864708.1864776.

Pratheepa Jeganathan and Susan P. Holmes. A Statistical Perspective on the Challenges in Molecular Microbial Biology. *Journal of Agricultural, Biological and Environmental Statistics*, 26(2):131–160, June 2021. doi: 10.1007/s13253-021-00447-.

Buhwan Jeong, Jaewook Lee, and Hyunbo Cho. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Information Sciences*, 180(5):602–612, 2010. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2009.10.016. URL https://www.sciencedirect.com/science/article/pii/S0020025509004526.

Eili Klein, Nestor Mojica, Wendi Jiang, Sara Cosgrove, Edward Septimus, Daniel Morgan, and Ramanan Laxminarayan. Trends in methicillin-resistant staphylococcus aureus hospitalizations in the united states, 2010-2014. *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, 65,

07 2017. doi: 10.1093/cid/cix640.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 426–434, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401944. URL https://doi.org/10.1145/1401890.1401944.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

Shao-Huan Lan, Wei-Ting Lin, Shen-Peng Chang, Li-Chin Lu, Chien-Ming Chao, Chih-Cheng Lai, and Jui-Hsiang Wang. Tedizolid versus linezolid for the treatment of acute bacterial skin and skin structure infection: A systematic review and meta-analysis. *Antibiotics*, 8:137, 09 2019. doi: 10.3390/antibiotics8030137.

Ken Lang. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA), 1995. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50048-7. URL https://www.sciencedirect.com/science/article/pii/B9781558603776500487.

Patrick LeBlanc and Li Ma. Microbiome subcommunity learning with logistic-tree normal latent dirichlet allocation. *Biometrics*, n/a(n/a), 2022. doi: https://doi.org/10.1111/biom.13772. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.13772.

Patrick M. LeBlanc and David Banks. Time-varying bayesian network meta-analysis. https://arxiv.org/abs/2211.08312, 2023.

Tong Queue Lee, Young Park, and Yong-Tae Park. A time-based approach to effective recommender systems using implicit feedback. *Expert Systems with Applications*, 34 (4):3055–3062, 2008. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2007.06.031. URL https://www.sciencedirect.com/science/article/pii/S0957417407002357.

Wenqiang Lei, Xiangnan He, Maarten Rijke, and Tat-Seng Chua. Conversational recommendation: Formulation, methods, and evaluation. pages 2425–2428, 07 2020a. doi: 10.1145/3397271.3401419.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2073–2083, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403258. URL https://doi.org/10.1145/3394486.3403258.

Alex Lenkoski. A direct sampler for g-wishart variates. *Stat*, 2, 12 2013. doi: 10.1002/sta4.23.

Alex Lenkoski and Adrian Dobra. Computational aspects related to inference in gaussian graphical models with the g-wishart prior. *Journal of Computational and Graphical Statistics*, 20(1):140–157, 2011. doi: 10.1198/jcgs.2010.08181. URL https://doi.org/10.1198/jcgs.2010.08181.

Hongzhe Li. Microbiome, metagenomics, and high-dimensional compositional data analysis. *Annual Review of Statistics and Its Application*, 2(1):73–94, 2015.

Yan Li and Wei Xu. Efficacy and safety of linezolid compared with other treatments for skin and soft tissue infections: A meta-analysis. *Bioscience Reports*, 38: BSR20171125, 12 2018. doi: 10.1042/BSR20171125.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 689–698, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3186150. URL https://doi.org/10.1145/3178876.3186150.

Lynne D. Liebowitz. Mrsa burden and interventions. *International Journal of Antimicrobial Agents*, 34:S11–S13, 2009. ISSN 0924-8579. doi: https://doi.org/10.1016/S0924-8579(09)70551-5. URL https://www.sciencedirect.com/science/article/pii/S0924857909705515. Focus on Antibiotic Stewardship.

Wey Wen Lim, Peng Wu, Helen Bond, Jessica Y Wong, Kaiwen ni, Wing Hong Seto, Mark Jit, and Benjamin Cowling. Determinants of mrsa prevalence in the asia pacific region: a systematic review and meta-analysis. *Journal of Global Antimicrobial Resistance*, 16, 08 2018. doi: 10.1016/j.jgar.2018.08.014.

Roderick JA Little. A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association*, 83(404):1198–1202, 1988.

Michael Littman, Susan Dumais, and Thomas Landauer. Automatic cross-language information retrieval using latent semantic indexing. *Automatic Cross-Language Information Retrieval Using Latent Semantic Indexing*, 03 1998. doi: 10.1007/978-1-4615-5661-9_5.

Chao Liu, Zhi Mao, Mengmeng Yang, Hongjun Kang, Hui Liu, Liang Pan, Jie Hu, Jun Luo, and Feihu Zhou. Efficacy and safety of daptomycin for skin and soft tissue infections: A systematic review with trial sequential analysis. *Therapeutics and Clinical Risk Management*, Volume 12:1455–1466, 09 2016. doi: 10.2147/TCRM.S115175.

Benedikt Loepp, Tim Hussein, and Jüergen Ziegler. Choice-based preference elicitation for collaborative filtering recommender systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page

3085–3094, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324731. doi: 10.1145/2556288.2557069. URL https://doi.org/10.1145/2556288.2557069.

Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_3. URL https://doi.org/10.1007/978-0-387-85820-3_3.

Guobing Lu and A. Ades. Mixed treatment comparisons: Combination of direct and indirect evidence. statistics in medicine 2004, 23:3105-3124. *Statistics in Medicine*, 10 2004. doi: 10.1002/sim.1875.

Guobing Lu and A.E. Ades. Lu g, ades aassessing evidence inconsistency in mixed treatment comparisons. j am statist assoc 101: 447-459. *Journal of the American Statistical Association*, 101:447–459, 02 2006. doi: 10.1198/016214505000001302.

Thomas Lumley. Lumley tnetwork meta-analysis for indirect treatment comparisons. statist med 21(16): 2313-2324. *Statistics in medicine*, 21:2313–24, 08 2002. doi: 10.1002/sim.1201.

Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014. doi: 10.1109/TII.2014.2308433.

Yuan Ma, Timm Kleemann, and Jürgen Ziegler. Mixed-modality interaction in conversational recommender systems. 09 2021.

Jialiang Mao, Yuhan Chen, and Li Ma. Bayesian graphical compositional regression for microbiome data. *Journal of the American Statistical Association*, 115(530): 610–624, 2020.

David Mawdsley, Meg Bennetts, Sofia Dias, Martin Boucher, and Nicky Welton. Model-based network meta-analysis: A framework for evidence synthesis of clinical trial data. *CPT: pharmacometrics & systems pharmacology*, 5, 08 2016. doi: 10.1002/psp4.12091.

Marianne McCollum, Sonja V. Sorensen, and Larry Z. Liu. A comparison of costs and hospital length of stay associated with intravenous/oral linezolid or intravenous vancomycin treatment of complicated skin and soft-tissue infections caused by suspected or confirmed methicillin-resistant staphylococcus aureus in elderly us patients. *Clinical Therapeutics*, 29(3):469–477, 2007. ISSN 0149-2918. doi: https://doi.org/10.1016/S0149-2918(07)80085-3. URL https://www.sciencedirect.com/science/article/pii/S0149291807800853.

Rachael Mccool, Jacqualyn Eales, T Barata, Mick Arber, M Cikalo, Kelly Fleetwood, Julie Glanville, Ian Gould, and T Kauf. Pin17. systematic review and network meta-

analysis of tedizolid for the treatment of acute bacterial skin and skin structure infection (absssi) due to methicillin-resistant staphylococcus aureus (mrsa). *Value in Health*, 18:A231, 05 2017. doi: 10.1016/j.jval.2015.03.1341.

Lorraine McGinty and Barry Smyth. On the role of diversity in conversational recommender systems. volume 2689, pages 276–290, 06 2003. ISBN 978-3-540-40433-0. doi: 10.1007/3-540-45006-8_23.

Rachana Mehta and Keyur Rana. A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 269–274, 2017. doi: 10.1109/CSCITA.2017.8066567.

Koji Miyahara and Michael J. Pazzani. Collaborative filtering with the simple bayesian classifier. In Riichiro Mizoguchi and John Slaney, editors, *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-44533-3.

Gary Moet, Ronald Jones, Douglas Biedenbach, Matthew Stilwell, and Thomas Fritsche. Contemporary causes of skin and soft tissue infections in north america, latin america, and europe: Report from the sentry antimicrobial surveillance program (1998-2004). *Diagnostic microbiology and infectious disease*, 57:7–13, 02 2007. doi: 10.1016/j.diagmicrobio.2006.05.009.

A. Mohammadi and E. C. Wit. Bayesian Structure Learning in Sparse Gaussian Graphical Models. *Bayesian Analysis*, 10(1):109 – 138, 2015. doi: 10.1214/14-BA889. URL https://doi.org/10.1214/14-BA889.

Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, page 195–204, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 158113231X. doi: 10.1145/336597.336662. URL https://doi.org/10.1145/336597.336662.

F. Morio, H. Jean-Pierre, L. Dubreuil, E. Jumas-Bilak, L. Calvet, G. Mercier, R. Devine, and H. Marchandin1. Antimicrobial susceptibilities and clinical sources of dialister species. *Antimicrobial Agents and Chemotherapy*, 51(12):4498–4501, 2007.

Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. pages 272–281, 08 1994. ISBN 978-3-540-19889-5. doi: 10.1007/978-1-4471-2099-5_28.

Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. An investigation on the user interaction modes of conversational recommender systems for the music domain. *User Modeling and User-Adapted Interaction*, 30, 04 2020. doi: 10.1007/s11257-019-09250-7.

G.P. Nason. *Wavelet Methods in Statistics with R*. Springer Publishing Company, Incorporated, 1 edition, 2008. ISBN 0387759603.

Dilip Nathwani. New antibiotics for the management of complicated skin and soft tissue infections: are they any better? *International Journal of Antimicrobial Agents*, 34:S24–S29, 2009. ISSN 0924-8579. doi: https://doi.org/10.1016/S0924-8579(09)70546-1. URL https://www.sciencedirect.com/science/article/pii/S0924857909705461. New Issues in Skin and Soft Tissue Infections.

David M. Nichols. Implicit rating and filtering. 1998.

Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.

Douglas W. Oard and Jinmook Kim. Implicit feedback for recommender systems. Technical Report AAAI Technical Report WS-98-08, Collegeo f Library and InformationS ervices, University of Maryland,C ollegeP ark, MD20 742, August 1998. URL http://www.aaai.org/Papers/Workshops/1998/WS-98-08/WS98-08-021.pdf.

Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, 2012. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2012.02.038. URL https://www.sciencedirect.com/science/article/pii/S0957417412002825.

Michael J. Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331, 2004.

Hugo Pedder, Sofia Dias, Meg Bennetts, Martin Boucher, and Nicky Welton. Modelling time-course relationships with multiple treatments: Model-based network meta-analysis for continuous summary outcomes. *Research Synthesis Methods*, 10, 04 2019. doi: 10.1002/jrsm.1351.

David Phillippo, Sofia Dias, A.E. Ades, Mark Belger, Alan Brnabic, Zbigniew Kadziola, and Nicky Welton. Multilevel network meta-regression for population-adjusted treatment comparisons. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 183, 03 2020. doi: 10.1111/rssa.12579.

Nicholas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

R. V. V. S. V. Prasad. A categorical review of recommender systems. *International Journal of Distributed and Parallel systems*, 3:73–83, 2012.

Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. Incomplete recovery and individualized responses of the human distal gut microbiota to repeated

antibiotic perturbation. *Genetics*, 155(2):945–959, 2000.

Dimitrios Rafailidis and Alexandros Nanopoulos. Modeling users preference dynamics and side information in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46:782–792, 06 2016. doi: 10.1109/TSMC.2015.2460691.

Al Rashid, Istvan Albert, Dan Cosley, Shyong Lam, Sean McNee, Joseph Konstan, and John Riedl. Getting to know you: Learning new user preferences in recommender systems. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 02 2002. doi: 10.1145/502716.502737.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

James Reilly, Kevin McCarthy, Lorraine Mcginty, and Barry Smyth. Dynamic critiquing. pages 37–50, 11 2004. ISBN 978-3-540-22882-0. doi: 10.1007/978-3-540-28631-8_55.

Boyu Ren, Sergio Bacallado, Stefano Favaro, Tommi Vatanen, Curtis Huttenhower, and Lorenzo Trippa. Bayesian mixed effects models for zero-inflated compositions in microbiome data analysis. *The Annals of Applied Statistics*, 14:494–517, 03 2020a. doi: 10.1214/19-AOAS1295.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. Crsal: Conversational recommender systems with adversarial learning. *ACM Trans. Inf. Syst.*, 38(4), jun 2020b. ISSN 1046-8188. doi: 10.1145/3394592. URL https://doi.org/10.1145/3394592.

Francesco Ricci and Quang Nhat Nguyen. Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent Systems*, 22(3):22–29, 2007. doi: 10.1109/MIS.2007.43.

Stephen J. Roberts, Michael R. Osborne, Mark Ebden, Steven Reece, Neale P. Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371, 2013.

J. J. Rocchio. *Relevance Feedback in Information Retrieval*. Prentice Hall, Englewood, Cliffs, New Jersey, 1971. URL http://www.is.informatik.uni-duisburg.de/bib/docs/Rocchio_71.html.

Kristine Rosenberger, Aiwen Xing, M. Hassan Murad, Haitao Chu, and Lifeng Lin. Prior choices of between-study heterogeneity in contemporary bayesian network meta-analyses: an empirical study. *Journal of General Internal Medicine*, 36, 01 2021. doi: 10.1007/s11606-020-06357-1.

Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. *Active Learning*

in Recommender Systems, pages 809–846. 02 2016. ISBN 978-1-4899-7637-6. doi: 10.1007/978-1-4899-7637-6_24.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1257–1264, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 880–887, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390267. URL https://doi.org/10.1145/1390156.1390267.

Georgia Salanti, Valeria Marinho, and Julian P.T. Higgins. A case study of multiple-treatments meta-analysis demonstrates that covariates should be considered. *Journal of Clinical Epidemiology*, 62(8):857–864, 2009. ISSN 0895-4356. doi: https://doi.org/10.1016/j.jclinepi.2008.10.001. URL https://www.sciencedirect.com/science/article/pii/S089543560800276X.

Kris Sankaran and Susan P Holmes. Latent variable modeling for the microbiome. *Biostatistics*, 20(4):599–614, 2019.

Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Application of dimensionality reduction in recommender system - a case study. 2000.

Andrew Schein, Alexandrin Popescul, Lyle Ungar, and David Pennock. Croc: A new evaluation criterion for recommender systems: World wide web electronic commerce, security and privacy (guest editors: Mary ellen zurko and amy greenwald). *Electronic Commerce Research*, 5, 01 2005. doi: 10.1023/B:ELEC.0000045973.51289.8c.

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 253–260, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135610. doi: 10.1145/564376.564421. URL https://doi.org/10.1145/564376.564421.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 111–112, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334730. doi: 10.1145/2740908.2742726. URL https://doi.org/10.1145/2740908.2742726.

Mahdi Shafiei, Katherine A Dunn, Eva Boon, Shelley M MacDonald, David A Walsh, Hong Gu, and Joseph P Bielawski. Biomico: a supervised bayesian model for

inference of microbial community structure. *Microbiome*, 3(8), 2015.

Bin Shen, Xiaoyuan Su, R. Greiner, P. Musilek, and C. Cheng. Discriminative parameter learning of general bayesian network classifiers. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 296–305, 2003. doi: 10.1109/TAI.2003.1250204.

Andrew F. Shorr. Epidemiology and economic impact of meticillin-resistant staphylococcus aureus. *PharmacoEconomics*, 25:751–768, 2012.

RJ Siezen and M Kleerebezem. The human gut microbiome: are we our enterotypes? *Microb Biotechnology*, 4(5):550–3, 2011.

A.J. Stewardson, N. Gaia, P. Francois, S. Malhotra-Kumar, C. Delemont, B. Martinez de Tejada, J. Schrenzel, S. Harbarth, and V. Lazarevic. Collateral damage from oral ciprofloxacin versus nitrofurantoin in outpatients with urinary tract infections: a culture-free analysis of gut microbiota. *Clinical Microbiology and Infection*, 21 (4):344.e1–344.e11, 2015.

Florian Strub, Romaric Gaudel, and Jérémie Mary. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, page 11–16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347952. doi: 10.1145/2988450.2988456. URL https://doi.org/10.1145/2988450.2988456.

Xiaoyuan Su and Taghi M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 497–504, 2006. doi: 10.1109/ICTAI.2006.41.

Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009, jan 2009. ISSN 1687-7470. doi: 10.1155/2009/421425. URL https://doi.org/10.1155/2009/421425.

Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 235–244, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210002. URL https://doi.org/10.1145/3209978.3210002.

Marta Tallarita, Maria De Iorio, and Gianluca Baio. A comparative review of network meta-analysis models in longitudinal randomized controlled trial. *Statistics in Medicine*, 38, 05 2019. doi: 10.1002/sim.8169.

Yik-Cheung Tam and Tanja Schultz. Correlated latent semantic model for unsupervised lm adaptation. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, 4:IV–41–IV–44, 2007.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 565–573, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

Yunfan Tang, Li Ma, and Dan L. Nicolae. A phylogenetic scan test on a Dirichlet-tree multinomial model for microbiome data. *The Annals of Applied Statistics*, 12(1): 1–26, 2018.

H Thom, J.C. Thompson, David Scott, N Halfpenny, K Sulham, and G.R. Corey. Comparative efficacy of antibiotics for the treatment of acute bacterial skin and skin structure infections (absssi): A systematic review and network meta-analysis. *Current medical research and opinion*, 31:1–34, 06 2015. doi: 10.1185/03007995.2015.1058248.

Amos Tversky and Itamar Simonson. Context-dependent preferences. *Manage. Sci.*, 39(10):1179–1189, oct 1993. ISSN 0025-1909.

Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *NIPS*, 2013.

Robin van Meteren. Using content-based filtering for recommendation. 2000.

Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *J. Artif. Int. Res.*, 27(1):465–503, dec 2006. ISSN 1076-9757.

Jesse Vig, Shilad Sen, and John Riedl. Navigating the tag genome. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, page 93–102, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450304191. doi: 10.1145/1943403.1943418. URL https://doi.org/10.1145/1943403.1943418.

Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Trans. Interact. Intell. Syst.*, 2 (3), sep 2012. ISSN 2160-6455. doi: 10.1145/2362394.2362395. URL https://doi.org/10.1145/2362394.2362395.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. *Proceedings of the 26th Annual International Conference on Machine Learning*, page 1105–1112, 2009.

Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 448–456, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308137. doi: 10.1145/2020408.2020480. URL https://doi.org/10.1145/2020408.2020480.

Hao Wang. Bayesian Graphical Lasso Models and Efficient Posterior Computation. *Bayesian Analysis*, 7(4):867 – 886, 2012. doi: 10.1214/12-BA729. URL https://doi.org/10.1214/12-BA729.

Kebin Wang and Ying Tan. A new collaborative filtering recommendation approach based on naive bayesian method. pages 218–227, 06 2011. doi: 10.1007/978-3-642-21524-7_26.

Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A survey on session-based recommender systems. *ACM Comput. Surv.*, 54(7), jul 2021a. ISSN 0360-0300. doi: 10.1145/3465401. URL https://doi.org/10.1145/3465401.

Tao Wang and Hongyu Zhao. A dirichlet-tree multinomial regression model for associating dietary nutrients with gut microorganisms. *Biometrics*, 73(3):792–801, 2017.

Weiquan Wang and Izak Benbasat. Research note —a contingency approach to investigating the effects of user-system interaction modes of online decision aids. *Information Systems Research*, 24:861–876, 09 2013. doi: 10.1287/isre.1120.0445.

Zhuoqun Wang, Jialiang Mao, and Li Ma. Microbiome compositional analysis with logistic-tree normal models. https://arxiv.org/abs/2106.15051, 2021b.

JL Weber and EW Myers. Human whole-genome shotgun sequencing. *Genome Res*, 7(5):401–9, 1997.

Ian White, Jessica Barrett, Dan Jackson, and Julian Higgins. Consistency and inconsistency in network meta-analysis: Model estimation using multivariate meta-regression. *Research Synthesis Methods*, 3, 06 2012. doi: 10.1002/jrsm.1045.

Mark H. Wilcox. The tide of antimicrobial resistance and selection. *International Journal of Antimicrobial Agents*, 34:S6–S10, 2009. ISSN 0924-8579. doi: https://doi.org/10.1016/S0924-8579(09)70550-3. URL https://www.sciencedirect.com/science/article/pii/S0924857909705503. Focus on Antibiotic Stewardship.

Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, page 153–162, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450337168. doi: 10.1145/2835776.2835837. URL https://doi.org/10.1145/2835776.2835837.

Yanwu Yang, Yinghui Catherine Yang, Bernard J. Jansen, and Mounia Lalmas. Computational advertising: A paradigm shift for advertising and marketing? *IEEE Intelligent Systems*, 32(3):3–6, 2017. doi: 10.1109/MIS.2017.58.

Jingru Zhang and Wei Lin. Scalable estimation and regularization for the logistic

normal multinomial model. *Biometrics*, 75:1098–1108, Dec 2019. doi: 10.1111/biom.13071.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), feb 2019a. ISSN 0360-0300. doi: 10.1145/3285029. URL https://doi.org/10.1145/3285029.

Shuo Zhang and Krisztian Balog. Evaluating conversational recommender systems via user simulation. 06 2020.

Ying Zhang, Yan Wang, Mieke van Driel, Treasure Mcguire, Tao Zhang, Yuzhu Dong, Yang Liu, Leichao Liu, Ruifang Hao, Lu Cao, Jianfeng Xing, and Yalin Dong. Network meta-analysis and pharmacoeconomic evaluation of antibiotics for the treatment of patients infected with complicated skin and soft structure infection and hospital-acquired or ventilator-associated penumonia. *Antimicrobial Resistance and Infection Control*, 8, 05 2019b. doi: 10.1186/s13756-019-0518-2.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 177–186, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271776. URL https://doi.org/10.1145/3269206.3271776.

Qian Zhao, F. Maxwell Harper, Gediminas Adomavicius, and Joseph A. Konstan. Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 1331–1340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450351911. doi: 10.1145/3167132.3167275. URL https://doi.org/10.1145/3167132.3167275.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1006–1014, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403143. URL https://doi.org/10.1145/3394486.3403143.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-rong Wen. Towards topic-guided conversational recommender system. *ArXiv*, abs/2010.04125, 2020b.

Shengyun Zhu, Huiqi Li, Jing Liang, Chaoran Lv, Kai Zhao, Mingshan Niu, Zhenyu Li, Lingyu Zeng, and Kailin Xu. Assessment of oral ciprofloxacin impaired gut barrier integrity on gut bacteria in mice. *International Immunopharmacology*, 83: 106460, 06 2020. doi: 10.1016/j.intimp.2020.106460.

Jie Zou, Yifan Chen, and Evangelos Kanoulas. *Towards Question-Based Recommender Systems*, page 881–890. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450380164. URL https://doi.org/10.1145/3397271.3401180.

B Biography

Patrick M. LeBlanc is from Brielle, New Jersey. He graduated *summa cum laude* from the University of Notre Dame in 2018 with a B.S. in Honors Mathematics and a minor in Philosophy, Politics, and Economics. He was a member of the Hesburgh-Yusko Scholars Program and the Glynn Family Honors Program. His honors thesis was titled *Information Theory: Entropy, Markov Chains, and Huffman Coding* and was supervised by Liviu Nicolaescu. In August, 2018, Patrick began his doctoral studies in the Department of Statistical Science at Duke University. He was a 2020-2021 Integrative Bioinformatics for Investigating and Engineering Microbiomes (IBIEM) Scholar, and was co-advised by David Banks and Li Ma. He plans to graduate with his Ph.D. in Statistical Science in May, 2023.