Multiscale Community Blockmodel for Network Exploration

Qirong Ho

School of Computer Science

Carnegie Mellon University, Pittsburgh, PA 15217

email: qho@cs.cmu.edu

Ankur P. Parikh

School of Computer Science

Carnegie Mellon University, Pittsburgh, PA 15217

email: apparikh@cs.cmu.edu

Eric P. Xing

School of Computer Science

Carnegie Mellon University, Pittsburgh, PA 15217

email: epxing@cs.cmu.edu

January 9, 2012

Abstract

Real world networks exhibit a complex set of phenomena such as underlying hierarchical organization, multiscale interaction, and varying topologies of communities. Most existing methods do not adequately capture the intrinsic interplay among such

phenomena. We propose a nonparametric Multiscale Community Blockmodel (MSCB) to model the generation of hierarchies in social communities, selective membership of actors to subsets of these communities, and the resultant networks due to within- and cross- community interactions. By using the nested Chinese Restaurant Process, our model automatically infers the hierarchy structure from the data. We develop a collapsed Gibbs sampling algorithm for posterior inference, conduct extensive validation using synthetic networks, and demonstrate the utility of our model in real-world datasets such as predator-prey networks and citation networks.

KEYWORDS: Hierarchical network analysis, Latent space model, Bayesian nonparametrics, Gibbs sampler

1. INTRODUCTION

How do complex networks and their self-organization arise from coordinated interactions and information sharing among the actors? One way to tap into this question is to understand the latent structures over actors which lead to the formation and organization of these networks. In particular, we are interested in uncovering the functional/sociological communities of network actors, and their influence on network connections. We consider a community to be a group of actors that share a common theme, like a clique of football fans in a social network, or an ecosystem of dependent organisms in a biological food web. Our objective is to gain a deeper understanding of the relationships within and among these communities, so as to shed insight into the network topology.

More specifically, we seek to address three critical aspects of network modeling and community discovery:

1. **Hierarchy** — not all communities are equal: a community can contain sub-communities, or be contained by super-communities. This is a natural way to structure the latent space of actors.

2. **Multiscale Granularity** — we must distinguish between coarse or generic associa-

tions that may occur in a large super-community, as opposed to fine grained interactions that occur within or among small, closely-interconnected sub-communities.

3. **Assortativity/Disassortativity** — some communities have strong within-community interactions and weak cross-community interactions (*assortativity*), yet others may exhibit the reverse (*disassortativity*).

These aspects are not independent, but are strongly interrelated. As an example, consider an oceanic food web (Figure 1), a directed network with species as actors and predator-prey relationships as edges. This network exhibits *hierarchy*: *cold-blooded animals* and *mammals* are large super-communities that can be sub-divided into smaller sub-communities, such as *sharks* and *squid*, or *toothed whales* and *pinnipeds*. These sub-communities can in turn be divided into even smaller communities (not shown). The ideas of hierarchy and network should *not* be confused with each other. The hierarchy is an organization of actors in some latent space learned from the observed network.

Next, the predator-prey relationships in the ocean are *multiscale*. Consider a sperm whale: it occasionally eats fish, which are common prey for many oceanic animals. Hence, this "sperm whale, fish" interaction is *generic*. Moreover, sperm whales usually eat giant squid, which are prey specific to them (making this interaction *fine-grained*). It is important to differentiate between such interactions of different scale.

Finally, the toothed whale sub-community demonstrates both *assortative* and *disassortative* behavior. Many toothed whales feed on small fish and seals, which are cross-community interactions. However, whales such as orcas feed on other whales, which are within-community interactions.

We propose a nonparametric Multiscale Community Blockmodel (MSCB) that presents a unified approach to address these three concerns. Using the nested Chinese Restaurant Process (Blei, Griffiths and Jordan 2010) as a nonparametric structural prior, our model infers the appropriate hierarchy from the data, without requiring the user to pre-specify the branching factor at each node. Moreover, by exploiting latent space ideas from Blei *et al.*

(2003) and Airoldi *et al.* (2008), we uncover the coarse/fine-grained interactions that underlie the network. Finally, our model builds upon the blockmodel concept (Wang and Wong 1987; Airoldi, Blei, Fienberg and Xing 2008) to integrate assortativity and disassortativity into our hierarchy. In order to use our model, we develop an MCMC algorithm for posterior inference and hyperparameter estimation, and study its performance on simulated and real datasets.

In particular, our qualitative studies are centered on two networks: a 75-species food web of grass-feeding wasps and their parasites (Dawah, Hawkins and Claridge 1995), and a subset of the arXiv High-Energy Physics (Theory) citation network (KDD 2010). The latter network originally contains 27,770 papers, but due to algorithmic limitations, we focus on a 1,000-paper subset published from Jan 2002 through May 2003[1]. Using our MCMC algorithm, we uncover hierarchies from both networks, and analyze these in the context of nodal attributes such as species' trophic levels (i.e. parasite, herbivore or plant) and journal paper titles. On the food web, we recover a hierarchy that, at a high level, follows intuitive trophic divisions: parasites are grouped together, and similarly for herbivores or plants. This is in contrast to the hierarchy-finding model of Clauset *et al.* (2008), whose recovered hierarchy is centered around highly-connected species rather than trophic levels. On the arXiv citation network, our recovered hierarchy splits the papers into specific research topics, which corresponds to the perception that most scientific research is conducted by highly-specialized communities of researchers. Finally, we support our qualitative studies with simulations that reveal conditions under which our algorithm outperforms other statistical and non-statistical network models.

1.1 Comparison to Existing Work

Existing methods for graph clustering and inferring community structure may address one or two of the aspects we have described, yet none capture *all three* aspects simultaneously.

[1]Due to the short window of time, this 1,000-paper subnetwork has a lower citation density than the original network. We acknowledge that this subnetwork is not ideal for hierarchy learning, since papers that share only older citations will have no network paths between them. Nevertheless, this subnetwork retains enough structure for our algorithm to recover a 2-level hierarchy, which we report in our experiments.

To begin, methods such as Girvan and Newman (2002), Hoff *et al.* (2002), Handcock *et al.* (2007), Krause *et al.* (2003) and Guimera and Amaral (2005) cannot discover *disassortative* communities characterized by weak within-community and strong cross-community interactions. Furthermore, they do not explicitly model organizational structure — and by extension, multiscale granularity of interactions. These methods do not meet any of our criteria, and are therefore unsuited for our purposes.

A common strategy for learning hierarchies is *divisive* or top-down: begin by learning first level clusters using a non-hierarchical network clustering algorithm, and then recursively apply the same algorithm to the subgraphs corresponding to each cluster. Spectral methods (Chung 1997) are a popular choice for top-down network clustering, not least because they approximate a graph cut objective function, making them a natural fit for networks. However, the biggest issue with divisive strategies is that deeper divisions cannot retroactively influence shallower ones, leading to situations in which poor choices made early in the algorithm doom the final outcome. In contrast, our hierarchical model specifies a probability distribution over the space of all possible K-level hierarchies, making it immune to such issues in principle — though its effectiveness in practice will admittedly depend on the particular inference algorithm used.

The counterpart to divisive clustering is *agglomerative* or bottom-up clustering, in which network entities are repeatedly merged to form larger an larger clusters. One popular version of this strategy, as adopted in the software program Pajek (Batagelj and Mrvar 1998), is to generate a dissimilarity matrix from the network, such as the corrected Euclidean-like dissimilarity of Batagelj *et al.* (1992), and then perform agglomerative clustering using Ward's criterion (Ward 1963). As with top-down clustering, bottom-up clustering suffers from an inability to retroactively apply information gleaned from later merges, highlighting the need for probabilistic models that consider all possible hierarchies at once. In our experiments, we shall compare our method to the top-down spectral clustering strategy, as well as the bottom-up Pajek strategy.

The Mixed Membership Stochastic Blockmodel (MMSB) (Airoldi et al. 2008) is a "mix-

ture of features" model, in that it aims to discover the multiple latent "roles" played by each actor in the network; additionally, it employs a blockmodel to accommodate both disassortative and assortative types of interactions. While the multi-role memberships discovered by MMSB are similar to our notion of coarse/fine-grained interactions, they are not identical; furthermore, MMSB does not induce a hierarchical structure over the actors. These considerations prevent MMSB from modeling the organized network phenomena that our model is designed to explore. Another "mixture of features" latent space model is that of Miller *et al.* (2009), which allows each actor to take on multiple binary features in an infinite-dimensional space. Like MMSB, this model does not learn a structure over its latent space, and therefore cannot replicate our model's ability to discover community hierarchies.

Finally, methods such as Clauset *et al.* (2004), Radicchi *et al.* (2004) and Kemp *et al.* (2008) explicitly model some form of organizational structure, but do not permit actors to have multiple kinds of interactions, which precludes them from learning the kind of multiscale interactions we have described. Our MSCB model is perhaps most closely related to the Infinite Relational Model (IRM) (Kemp, Tenenbaum, Griffiths, Yamada and Ueda 2006), which is a special case of MSCB. More specifically, the IRM is equivalent to MSCB with a hierarchy depth of one (i.e. a flat hierarchy with no multiscale membership), an unsurprising fact given that the IRM is a nonparametric generalization of the stochastic blockmodel (Wang and Wong 1987), from which MSCB inherits some of its key attributes. Roy *et al.* (2007) have generalized the IRM in a different way for hierarchical group discovery, and further extended their work to the nonparametric setting with Mondrian Processes (Teh and Roy 2009). However, these two models are limited to *binary* hierarchies, and furthermore, they adopt a notion of multi-community membership that is not directly comparable to ours. In contrast, our model assumes no limit on the hierarchy's branching factor, which is more realistic for certain networks.

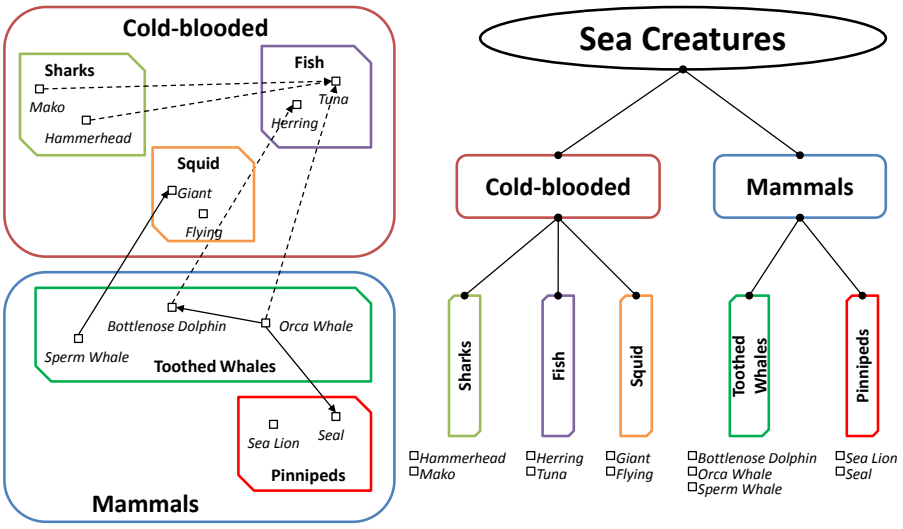

Figure 1: Illustration of an oceanic food web as a set of nested communities **(Left)**, and the corresponding hierarchy of communities **(Right)**. Vertices in the network represent individual species, and directed edges represent predator-prey relationships (not all shown). Solid edges are *fine-grained*, specific interactions, while dashed edges are *coarse-grained* and generic to a large community.

2. MULTISCALE COMMUNITY BLOCKMODEL (MSCB)

We now describe aspects of our model, beginning with the hierarchy and ending with network generation. Throughout, we adopt an oceanic food web as a running example (Figure 1).

2.1 The Community Hierarchy

Our model organizes network actors into a depth-K tree-shaped hierarchy, where each node in the tree represents a community. The tree contains a single root node at level 0, followed by K levels of nodes; parent nodes can have any number of child nodes. Nodes closer to the root represent large super-communities, (e.g. the "cold-blooded animals" and "mammals" in Figure 1), while nodes closer to the terminal leaves represent finer-grained sub-communities (e.g. "toothed whales" or "sharks").

The most important aspect of our hierarchy is that each actor is associated with not just one community, but with an entire *path* of super- through sub-communities, starting from a level-1 node and ending with a level-K terminal leaf. In our food web example, the *sperm*

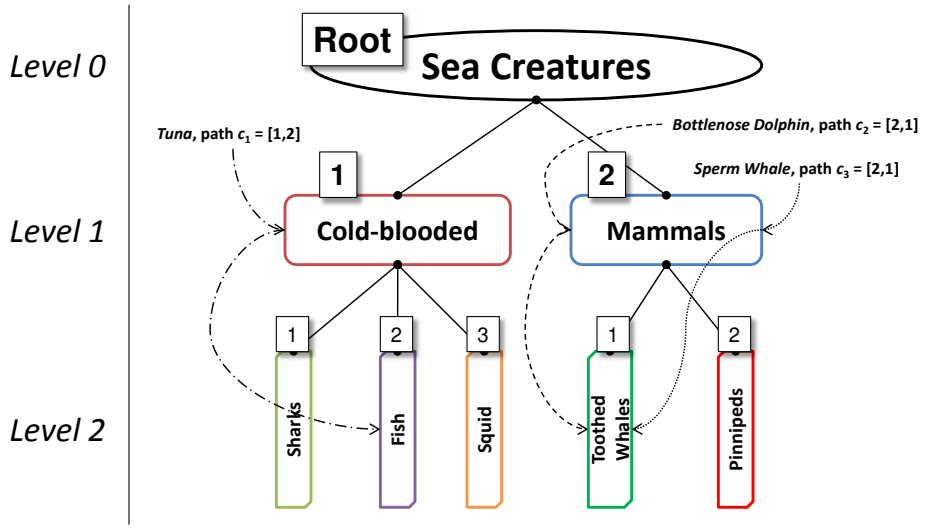

Figure 2: Oceanic food web from Figure 1, annotated with three species' path vectors c_i.

whale follows the path [mammal, toothed whale]. More formally, we represent actor i's path by a random-valued, K-length *path vector* c_i, where the k-th element of c_i represents the *branch choice* taken at level k. Referring to our oceanic food web in Figure 2, we see that "mammals" are the 2nd branch from the root "sea creatures", while "toothed whales" are the 1st branch from "mammals" — hence the sperm whale has a path vector of $c_3 = [2, 1]$ in this hierarchy. Henceforth, we shall use the term *path* to refer to path vectors c_i.

It is important to note that we never represent our hierarchy explicity; it is implicit from the set of all actor paths $\{c_1, \ldots, c_N\}$. For example, given $c_1 = [1, 1]$, $c_2 = [1, 2]$, and $c_3 = [2, 1]$, we can recover a hierarchy with two level-1 nodes, carrying 2 and 1 child nodes respectively at level 2. This implict representation is convenient for probabilistic inference, as we shall demonstrate later.

2.2 A Measure over Hierarchies

Because the number of actor paths N is finite, the space of meaningful hierarchies is also finite. Unfortunately, this space is still very large, which makes model selection (i.e. hierarchy selection) an inherently difficult task. In order to solve this problem, we make use of a measure over sets of actor paths $\{c_1, \ldots, c_N\}$ (recall that these paths implicitly define a

depth-K hierarchy), the Nested Chinese Restaurant Process (nCRP) (Blei et al. 2010). As its name suggests, the nCRP is a recursive version of the Chinese Restaurant Process (CRP) (Aldous 1985; Teh, Jordan, Beal and Blei 2006), the latter being an exchangeable distribution over *partitions* of a finite set of objects (i.e. ways to divide up the objects).

To clearly present the nCRP, a brief overview of the CRP and the Dirichlet Process (DP) (Ferguson 1973) is in order. In Bayesian nonparametric mixture modeling, the DP is employed as a prior over a countably infinite number of mixture components (Escobar and West 1995; MacEachern and Müller 1998). However, the DP can only model flat (i.e. without hierarchy) nonparametric mixture models, which represent an extreme case of our model when the hierarchy has only $K = 1$ level. One particularly useful view of the DP is given by the Pólya urn scheme (Blackwell and MacQueen 1973), which provides a closed form for the i-th data point's mixture component, given the mixture components of the previous $i - 1$ data points. If we disregard the *locations* of the mixture components and simply focus on their *identities*, the Pólya urn scheme becomes the CRP. Recall that the CRP is an exchangeable distribution over partitions of objects: in the context of the DP, the objects are data points, which are placed into parts (i.e. divisions of the partition) that represent mixture components.

In essence, the CRP allows us to separate the identities of the DP mixture components from their locations in the data domain (as drawn from a base measure). By exploiting this separation, Blei *et al.* (2010) generalize the CRP to the nCRP, transforming the flatly-organized mixture components of the former into the tree-shaped hierarchy of the latter.

Nested Chinese Restaurant Process The nCRP, as defined by Blei *et al.* (2010), is a measure over sets of tree paths $\{c_1, \ldots, c_N\}$ of *infinite* length, though for our purposes we shall restrict the nCRP to paths of length K. The nCRP is most intuitive when described generatively: beginning with actor 1, each actor chooses its tree path in turn, conditioned on the existing paths chosen by previous actors. Consider the i-th actor: he begins at the root, and needs to choose which level-1 branch to take. With probability $n^{(1)}_{x,i-1}/(i-1+\gamma)$ he

selects branch x already in the tree, or with probability $\gamma/(i-1+\gamma)$ he starts a completely new branch. Here, $n_{x,i-1}^{(1)}$ is the number of actors before i that chose branch x at level 1, and γ is a hyperparameter dictating the probability that an actor will start a new branch.

Actor i continues this process as he descends the tree. When picking a branch at level k, with probability $n_{y,i-1}^{(k)}/(n_{i-1}^{(k-1)}+\gamma)$ he selects branch y, and with probability $\gamma/(n_{i-1}^{(k-1)}+\gamma)$ he starts a new branch. Here, $n_{i-1}^{(k-1)}$ counts the number of actors $1,\ldots,i-1$ having the same path as actor i up to (and including) level $k-1$. Out of these actors, $n_{y,i-1}^{(k)}$ is the number that picked branch y (at level k). This sequence of branch choices defines the path c_i of actor i, and as we have explained previously, the set of all actor paths implicitly defines the hierarchy. Note that our model limits the hierarchy to a maximum depth of K.

We now provide a more formal definition of the nCRP. In order to do so, we must first introduce the (original) Chinese Restaurant Process (CRP), an exchangeable distribution over *partitions* (of a set of objects) (Aldous 1985; Teh et al. 2006). For concreteness, we shall represent an object's assigned part by a positive integer — as an example, suppose there are three random variables X_1, X_2, X_3 corresponding to a partition of three objects, then $X_1 = 1, X_2 = 1, X_3 = 2$ represents a partition where objects 1 and 2 are in one part, and object 3 is in another part. Note that this scheme allows every partition to be represented by infinitely many assignments, e.g. $X_1 = 3, X_2 = 3, X_3 = 2$ represents the same partition as the earlier assignment. Despite this non-identifiability, the positive-integer representation turns out to be convenient for describing and implementing our Gibbs sampling inference procedure.

Let c_{ik} denote the k-th element of the i-th actor path, i.e. the branch taken at level k by actor i. In our model, the collection of all actor path first-level branch choices, which we denote as $\{c_{i1}\}$, forms a partition distributed according to a CRP prior:

$$\mathbb{P}(c_{i1} = x \mid c_{1:(i-1),1}) = \begin{cases} \frac{|\{j<i \,|\, c_{j1}=x\}|}{i-1+\gamma} & x \in \{c_{1:(i-1),1}\} \\ \frac{\gamma}{i-1+\gamma} & x \text{ is the smallest positive integer not in } \{c_{1:(i-1),1}\} \end{cases} \tag{1}$$

where $\gamma > 0$ is a "concentration" parameter that controls the probability of drawing new

integers, and $c_{1:(i-1),1} = (c_{11}, \ldots, c_{(i-1)1})$ denotes the set of 1st-level elements from paths c_1 through c_{i-1}. Again, we stress that different assignments to $\{c_{i1}\}$ can correspond to the same partition. High values of γ imply that partitions with more parts are more likely.

The nCRP (Blei et al. 2010) extends the CRP to *recursively nested partitions*. The nCRP can be thought of as a hierarchy of CRPs, beginning with a single CRP at the top level. To each unique integer x seen at the top-level prior, we associate a child CRP with $|\{i \,|\, c_{i1} = x\}|$ observations, resulting in a two-level tree of CRPs. We can repeat this process *ad infinitum* on the newly-created child CRPs, resulting in an infinite-level tree of CRPs; however, we only use a K-level nCRP (denoted as nCRP_K) as we have limited the maximum hierarchy depth to K. In our model, all child CRPs of the nCRP share the same concentration parameter γ, and high values of γ make "branchier" trees more likely. We note that one could easily have different γ's for different tree levels, but do not discuss this modification.

Our model treats the collection of all actor paths $\{c_i\}$ as a recursively nested partition (of depth K), distributed according to an $\text{nCRP}_K(\gamma)$ prior:

$$\mathbb{P}(c_{ik} = x \;\mid\; c_{1:(i-1)} \,,\; c_{i,1:(k-1)}) = \tag{2}$$
$$\begin{cases} \frac{\left|\{j<i \,|\, c_{j,1:(k-1)}=c_{i,1:(k-1)} \,,\, c_{jk}=x\}\right|}{\left|\{j<i \,|\, c_{j,1:(k-1)}=c_{i,1:(k-1)}\}\right|+\gamma} & x \in \{c_{jk} \;\mid\; (j<i) \,,\, c_{j,1:(k-1)} = c_{i,1:(k-1)}\} \\ \frac{\gamma}{\left|\{j<i \,|\, c_{j,1:(k-1)}=c_{i,1:(k-1)}\}\right|+\gamma} & x \text{ is the smallest positive integer not in the above set,} \end{cases}$$

where in the first line, $c_{1:(i-1)} = \{c_1, \ldots, c_{i-1}\}$ is the set of *complete* paths c_1 through c_{i-1}, and $c_{i,1:(k-1)} = \{c_{i,1}, \ldots, c_{i,k-1}\}$ contains the 1st through $(k-1)$-th elements of path c_i. Each actor path c_i is represented by a length-K vector of positive integers, and the distribution above enables us to draw their elements one at a time: $c_{11}, \ldots, c_{1K}, c_{21}, \ldots, c_{2K}$, and so on. As with the original CRP, different assignments to $\{c_i\}$ can correspond to the same set of (nested) partitions.

2.3 Multiscale Membership

In conjunction with the actor paths, MSCB's notion of *Multiscale Membership* (MM) distinguishes it from other hierarchical clustering methods. Briefly, each actor i is endowed with

a probability distribution over the hierarchy nodes in its path. We denote this distribution by a Multiscale Membership (MM) vector θ_i, a K-dimensional multinomial that encodes an actor's tendencies to interact as a member of the different super- and sub-communities along its K-length path. Specifically, θ_{ik}, the k-th element of θ_i, is the probability that actor i will interact as a member of the k-th community along its path from the root. Through MM, our model is able to accomodate multiscale granularity on interactions, i.e. the notion that some interactions are finer or coarser than others.

To use our food web example (Figure 2), consider two species of toothed whales: dolphins and sperm whales. Although both share the same hierarchy path, [mammal, toothed whale], they behave quite dissimilarly. A dolphin's diet consists mainly of fish, which are common prey for many mammals. Thus, we would say it typically interacts as a member of the mammal super-community, though it occasionally chooses prey that are more specific to its species. On the other hand, a sperm whale barely eats fish, so it rarely interacts as a member of its super-community; instead, it prefers giant squid, a more specific prey uncommon to most mammals. Thus, the sperm whale has a higher probability of participating in fine-grained interactions (say, $\theta = [0.1, 0.9]$) unlike the dolphin, which is more likely to pursue coarse-grained interactions (for example, $\theta = [0.8, 0.2]$).

At this juncture, it is worth comparing our Multiscale Membership to the Mixed-Membership Stochastic Blockmodel (MMSB) (Airoldi et al. 2008). The latter model endows each actor with a distribution over latent roles, just as our Multiscale Membership vector provides each actor with a distribution over communities. There is a key difference, however: MMSB's Mixed Membership vectors permit each actor to have a distribution over *all latent roles*, whereas our model's Multiscale Membership vectors constrain each actor to distributions only over super- and sub-communities on its path. This restriction is crucial: if actors were allowed to have distributions over all hierarchy communities, the hierarchy could be rendered virtually meaningless — for instance, we could say that dolphins are simultaneously members of the shark *and* toothed whale communities, which is plainly unrealistic.

Just as with the paths c_i, we place a suitable prior on the MM vectors θ_i, a truncated

two-parameter stick breaking[2] process (Blei et al. 2010), denoted by $\text{Stick}_K(m, \pi)$ where $0 < m < 1$ and $\pi > 0$. This prior is conjugate to the K-dimensional multinomial distribution, which will be important when deriving our Gibbs sampler inference algorithm. Furthermore, unlike the Dirichlet distribution (which is also conjugate to the multinomial), the stick breaking process makes it easy to bias the posterior of θ_i towards coarser interactions (i.e. elements θ_{ik} have more probability mass for k closer to 1) or finer interactions (more mass for k closer to K). In contrast, a single-parameter Dirichlet prior cannot accomodate coarse/fine biases, while a full K-parameter Dirichlet prior may be too expressive — for instance, we do not anticipate needing a bimodal prior on θ_{ik} for our applications.

Truncated Two-parameter Stick Breaking Process As its name suggests, the $\text{Stick}_K(m, \pi)$ distribution generates multinomial parameters θ_i via the analogy of breaking sticks into pieces. Beginning with a stick of length 1, draw the first stick fraction $V_{i1} \sim \text{Beta}(m\pi, (1-m)\pi)$ and let the first stick length be $\theta_{i1} = V_{i1}$, leaving $1 - \theta_{i1}$ as the remainder of the stick. To get the second stick length θ_{i2}, draw $V_{i2} \sim \text{Beta}(m\pi, (1-m)\pi)$ and break this fraction off from the remainder, giving $\theta_{i2} = V_{i2}(1 - V_{i1})$. We repeat this process until we have K sticks, after which we discard the remainder and renormalize the sticks to get θ_i.

Formally, let V_{ik} be the k-th fraction to break off from the stick's remainder, and let θ_{ik} be the length of the k-th stick. To draw $\theta_i \sim \text{Stick}_K(m, \pi)$, we first draw $V_{ik} \sim \text{Beta}(m\pi, (1-m)\pi)$ for $k \in \{1, \ldots, K\}$ and define θ_{ik} to be

$$\theta_{ik} \propto V_{ik} \prod_{u=1}^{k-1} (1 - V_{iu}) \tag{3}$$

with normalization factor $\sum_{k=1}^{K} \theta_{ik}$. We note that Blei *et al.* (2010) called this distribution a "two-parameter GEM distribution", and let $K \to \infty$ to get an infinte-dimensional prior.

Intuitively, the parameter $0 < m < 1$ influences the posterior mean of θ_i; $m \to 1$ results in elements θ_{ik} having more mass for k closer to 1, while $m \to 0$ results in more mass for k

[2]Note that our use of the stick-breaking process is unrelated to the stick-breaking construction for the Dirichlet Process. We use the stick-breaking process to produce a mixture over the mixture components induced by the nCRP, not to define the mixture components themselves.

closer to K. The other parameter $\pi > 0$ indicates our confidence in the stick breaking prior; $\pi \to \infty$ indicates more confidence, causing the posterior mean of θ_i to approach the prior mean, and the posterior variance of θ_i to approach zero.

2.4 Network Edge Generation

We now explain how the paths c_i and MM vectors θ_i generate edges in the network. At this point, we must introduce additional notation: let E be the $N \times N$ adjacency matrix of observed network edges, where element E_{ij} corresponds to the *directed edge* or interaction/relationship from actor i to j. In the context of our food web, the actors are sea creatures such as dolphins and sperm whales, and the edges represent predator-prey interactions. A value of $E_{ij} = 1$ indicates that the interaction is present, while $E_{ij} = 0$ indicates absence, and we ignore self-edges E_{ii}. Because we are modeling a directed network, E is not necessarily symmetric. If we need to model symmetric relationships such as friendship, we can let $E_{ij} = E_{ji}$ for all i, j.

To begin, we shall introduce MSCB's *generative process* for network edges:

- For each actor $i \in \{1, \ldots, N\}$:
  - Sample i's hierarchy path $c_i \sim \text{nCRP}_K(\gamma)$.
  - Sample i's Multiscale Membership (MM) vector $\theta_i \sim \text{Stick}_K(m, \pi)$. Note that θ_i and c_i are drawn independently.
- To generate the network, for each possible directed edge E_{ij}:
  - Sample donor level $z_{\to ij} \sim \text{Multinomial}(\theta_i)$, and let $h = c_i[z_{\to ij}]$. Formally, h represents the community at level $z_{\to ij}$ on path c_i, i.e. the $z_{\to ij}$-th element of c_i.
  - Sample receiver level $z_{\leftarrow ij} \sim \text{Multinomial}(\theta_j)$, and let $h' = c_j[z_{\leftarrow ij}]$.
  - Sample the edge $E_{ij} \sim \text{Bernoulli}(\text{S}_B(h, h'))$. We shall define $\text{S}_B()$ later.

The basic idea is as follows: for every directed edge E_{ij}, both actor i (the donor) and actor j (the receiver) pick communities h and h' from their respective paths c_i, c_j, according to the levels drawn from their MM vectors θ_i, θ_j. The communities h, h' are then used to select a *community compatibility* parameter via the function $\text{S}_B(h, h')$, which in turn generates

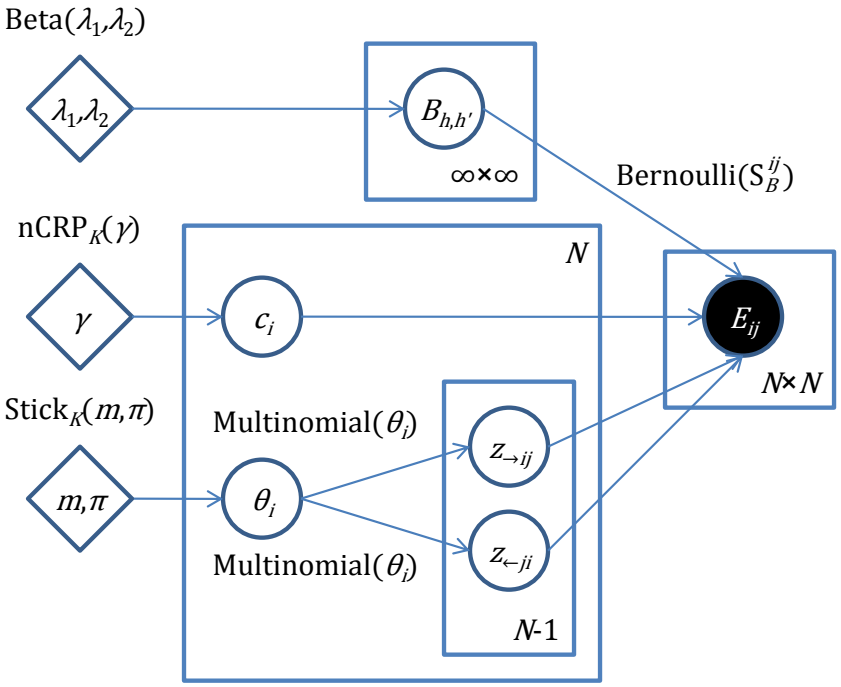

Figure 3: Graphical model representation of our MSCB model. Diamonds represent model parameters, hollow circles represent hidden random variables, and filled circles represent observed random variables. Variables inside a numbered rectangle are duplicated by that number, e.g. there are N variables θ_i, numbered from θ_1 through θ_N. Arrows denote probabilistic dependencies, and these are annotated with the probability distribution they represent: for instance, $\theta_i \sim \text{Stick}_K(m, \pi)$, $z_{\rightarrow ij} \sim \text{Multinomial}(\theta_i)$, etc..

$E_{ij} \sim \text{Bernoulli}(\text{S}_B(h, h'))$. Note that the arrow in $z_{\rightarrow ij}$ or $z_{\leftarrow ij}$ indicates whether the variable belongs to the donor actor (i for $z_{\rightarrow ij}$) or receiver actor (j for $z_{\leftarrow ij}$), with respect to the edge E_{ij} from i to j. The arrow does *not* indicate the edge direction between i and j.

2.5 Community Compatibility Matrices $\mathbf{B}$

We turn our discussion to the *community compatibility* parameters that define edge probabilities between communities, as well as how the $\text{S}_B()$ function selects them during edge generation. Intuitively, the *compatibility* from community h to h' is high if actors from h often interact with actors from h'. Conversely, a low compatibility indicates that actors from h rarely interact with actors from h'. Thus, it is natural that we define compatibility to be a Bernoulli parameter in $[0, 1]$, where 1 indicates perfect compatibility. This notion of compat-

ibility is what allows our model to account for both assortative and disassortative behavior, in similar fashion to stochastic blockmodels (Wang and Wong 1987) — for example, strongly assortative networks correspond to high compatibility parameters when $h = h'$, i.e. when the source and destination communities are the same.

There are many ways to associate compatibility parameters with pairs of communities h, h', and the challenge is to find a scheme that tightly integrates compatibility parameters with the hierarchy and multiscale interactions over communities. A first attempt might be to ignore the hierarchy, and place a full $H \times H$ compatibility matrix over all community pairs h, h' (where H is the total number of nodes in the hierarchy); this is analogous to the role-compatibility matrix used in MMSB (Airoldi et al. 2008). However, this formulation fails to capture the multiscale nature of interactions, because there is simply no connection between the compatibility parameter for h, h' and those communities' levels in the hierarchy.

In order to connect the community parameters with the hierarchy levels, we must *restrict* them in some meaningful way. First, we need to define the notion of a *sibling group*. Formally, a sibling group is *a largest possible set of communities* such that (1) all communities are at the same level, and (2) have the same immediate parent community. To put it another way, every parent (including the level-0 root) contains a sibling group comprised of its *immediate* children (but not grandchildren, or great-grandchildren, etc.). Hence, if there are P parent nodes, there are P sibling groups, and all sibling groups are disjoint from one another. To give a concrete example, if we have three paths, $c_1 = [1, 1]$, $c_2 = [1, 2]$, and $c_3 = [2, 1]$, then the hierarchy contains 3 sibling groups: one at level 1 with communities $\{[1], [2]\}$, and two at level 2 with communities $\{[1, 1], [1, 2]\}$ and $\{[2, 1]\}$ respectively.

Each sibling group is associated with its own compatibility matrix $\mathbf{B}$, which contains the compatibility parameters for every pair of communities within that sibling group; refer to Figure 4 for an example illustration. This scheme restricts the community parameters — notice that communities from different sibling groups do not have explicit community parameters between them; we shall discuss how this affects edge generation shortly. Also, since MSCB infers the number of hierarchy nodes from the data by way of the nCRP prior,

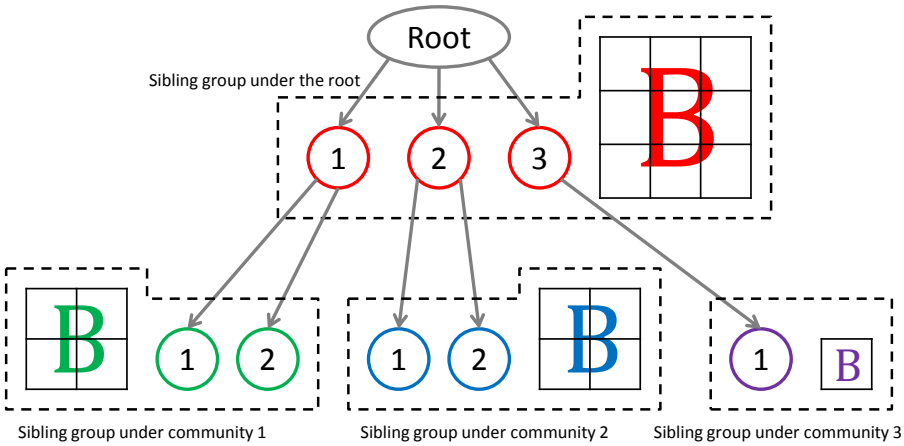

Figure 4: Four sibling groups in an example hierarchy, and the sizes of their compatibility matrices $\mathbf{B}$.

we cannot know *a priori* the number and sizes of the sibling group compatibility matrices. We will return to this issue when we discuss our inference procedure.

Now that we have defined the sibling group compatibility matrices, we can describe how network edges are generated from them. Recall that the edge generative process picks two interacting communities h, h' according to paths c_i, c_j and MM vectors θ_i, θ_j. When h, h' are at the same level and share the same immediate parent, we simply pick their community compatibility parameter[3] $\mathbf{B}_{h,h'}$ from their associated sibling group matrix, and draw the edge $E_{ij} \sim \text{Bernoulli}(\mathbf{B}_{h,h'})$. However, if h, h' do not share the same parent, then we invoke the following *coarsening* procedure:

1. Recall that $h = c_i[z_{\rightarrow ij}]$ and $h' = c_j[z_{\leftarrow ij}]$.

2. Set $z_{min} := \min(z_{\rightarrow ij}, z_{\leftarrow ij})$, i.e. the smaller interaction level from h or h'.

3. If the *coarsened* communities $h_{coarse} = c_i[z_{min}]$ and $h'_{coarse} = c_j[z_{min}]$ are in the same sibling group, then we look up their compatibility matrix entry $\mathbf{B}_{h_{coarse},h'_{coarse}}$. We then generate

[3] A word on notation: $\mathbf{B}_{h,h'}$ is used as shorthand to (1) first select the community-compatibility matrix $\mathbf{B}$ associated with the parent of communities h, h', and then (2) select the element of that $\mathbf{B}$ corresponding to h interacting with h'. Because our inference procedure integrates out the $\mathbf{B}$ matrices, a precise, long-form notation for them is unnecessary.

$E_{ij} \sim \text{Bernoulli}(\mathbf{B}_{h_{coarse},h'_{coarse}})$.

4. Otherwise, set $z_{min} := z_{min} - 1$ and repeat step 3. This effectively sets h_{coarse} to its immediate parent (and likewise for h'_{coarse}).

To put it another way, the coarsening procedure finds the deepest common hierarchy node (which could be the root node) shared by the paths c_i and c_j and whose level is strictly shallower than $\min(z_{\rightarrow ij}, z_{\leftarrow ij})$. Supposing that this deepest common node has level $z_{min} - 1$, we then draw the compatibility parameter $\mathbf{B}_{c_i[z_{min}],c_j[z_{min}]}$ and generate E_{ij} according to it. Notice that the hierarchy nodes $c_i[z_{min}], c_j[z_{min}]$ are part of the same sibling group by construction, thus the compatibility parameter $\mathbf{B}_{c_i[z_{min}],c_j[z_{min}]}$ always exists. Now, we can formally define our $\mathrm{S}_B()$ function from the previous section:

$$\mathrm{S}_B(h, h') = \mathbf{B}_{h_{coarse},h'_{coarse}}, \qquad \text{where} \qquad h = c_i[z_{\rightarrow ij}], \qquad h' = c_j[z_{\leftarrow ij}], \tag{4}$$
$$h_{coarse} = c_i[z_{min}], \qquad h'_{coarse} = c_j[z_{min}], \qquad z_{min} = 1 + \max_{z:\, 0 \leq z < \min(z_{\rightarrow ij}, z_{\leftarrow ij}),\, c_i[z] = c_j[z]} z,$$

and where we adopt the convention $c_i[0] = c_j[0]$, to represent the fact that all paths implicitly share the root node. In future, we shall employ the shorthand $\mathrm{S}_B^{ij} := \mathrm{S}_B(h, h')$ for brevity.

Finally, in keeping with our use of Bayesian inference, we place a $\text{Beta}(\lambda_1, \lambda_2)$ prior over every element $\mathbf{B}_{x,y}$ of every sibling group compatibility matrix. This adds the following step to our generative process:

- For each element x, y of each sibling group compatibility matrix:
  - Sample $\mathbf{B}_{x,y} \sim \text{Beta}(\lambda_1, \lambda_2)$.

This step comes after generating paths c_i, but before generating edges E_{ij}. A graphical model representation of our full generative process can be found in Figure 3. In summary, our use of sibling group compatibility matrices enforces the role of the hierarchy during edge generation, thus distinguishing MSCB from "flat" blockmodels such as Wang *et al.* (1987) and Airoldi *et al.* (2008).

3. A DISCUSSION ON MODEL IDENTIFIABILITY

In general, MSCB is not identifiable for hierarchy depths $K > 1$; the lack of identifiability implies that, for a given observed network E_{ij}, there will be multiple hierarchies that produce the same model likelihood. However, the degree to which MSCB is non-identifiable can be compared to two other models, the Infinite Relational Model (IRM) (Kemp et al. 2006) and the Mixed Membership Stochastic Blockmodel (MMSB) (Airoldi et al. 2008).

There are two ways in which MSCB is non-identifiable, and it is important to distinguish between them. First, observe that the communities can be permuted without changing the model likelihood. This issue is common to practically all clustering models and algorithms; even the regular stochastic blockmodel (Wang and Wong 1987) suffers from this. Nevertheless, this type of non-identifiability is rarely a problem: in most clustering applications, the clusters are meant to be inspected by humans, or else further criteria can be applied post-hoc to select clusters of interest. The second, and more critical form of non-identifiability, arises because the MM vectors θ_i are mixtures. We shall discuss this further when we compare MSCB to the MMSB.

Observe that when $K = 1$ or $m \to 1$, the MM vectors θ_i are reduced to point masses at the first element θ_{i1}, which eliminates the second non-identifiability (mixture non-identifiability). In this form, MSCB reduces to a nonparametric stochastic blockmodel (by nonparametric, we mean that the number of roles/clusters is automatically selected), identical to the IRM, and with only permutation non-identifiability to worry about. Although these limiting cases do not produce meaningful hierarchies and are thus uninteresting from an application perspective, they remain useful as a kind of lower bound on MSCB's non-identifiability.

For the general case in which the MM vectors θ_i are nonzero in more than one dimension, we can gain insight by comparing MSCB to the MMSB. The latter is essentially a stochastic blockmodel with L communities/roles, but unlike the stochastic blockmodel, network entities are not restricted to just one role. Instead, each network entity i has a distribution ϕ_i over all L roles, from which every edge-touching entity i draws its own (possibly different)

role assignment; this process is similar to how our MSCB allows edges to draw different level assignments. Because of the entity role mixtures ϕ_i, MMSB suffers from mixture non-identifiability, though it has been successfully applied to various datasets (Airoldi et al. 2008). Formally, the MMSB marginal likelihood of $E_{ij} = 1$ conditioned on the role mixtures (but not the edge role assignments) is $\phi_i^\top B \phi_j$, where B is the blockmatrix. Observe that for any orthonormal matrix U, we have $\phi_i^\top (UU^\top) B (UU^\top) \phi_j = \phi_i^\top B \phi_j$. Assuming a permutation-unbiased prior on the ϕ_i, this implies that the MMSB model likelihood remains the same if we transform B to $U^\top BU$ and all ϕ_i to $U^\top \phi_i$. In short, MMSB is non-identifiable up to orthonormal transformations.

To see how MSCB relates to MMSB, imagine that the MSCB sibling group matrices are sub-matrices along the diagonal of some large $H \times H$ blockmatrix B, where H is the total number of hierarchy nodes. Elements of this blockmatrix that do not correspond to some sibling group matrix represent community-community interaction parameters eliminated by coarsening, and thus are absent from the model. In this new representation, the MM vectors θ_i can be represented as H-dimensional vectors ψ_i, but with support on at most K elements, where K is the hierarchy depth. If we disregard coarsening, the marginal likelihood of $E_{ij} = 1$ is simply $\psi_i^\top B \psi_j$, just like MMSB. The difference is that the vectors ψ_i have support on at most $K \ll H$ elements, making MSCB significantly more constrained, and thus more identifiable, than a H-role MMSB. Specifically, we can only transform B and ψ_i using orthonormal matrices U that preserve ψ_i having at most K elements of support, otherwise the resulting model no longer corresponds to a K-level MSCB. As for coarsening, its effect is to force parameter sharing — think of it as remapping the "eliminated" elements of B onto elements corresponding to some sibling group matrix. This further constrains MSCB, making it in fact more identifiable than our comparison to MMSB implies.

In summary, the fixed-depth hierarchy and coarsening rules make MSCB significantly more identifiable than the closely-related MMSB. However, as we increase the maximum hierarchy depth K, the degree of model non-identifiability also increases. We recommend using a maximum hierarchy depth of $K = 2$ or 3, noting that for a fixed depth K, MSCB can instead

learn wider, highly-branched hiearchies without increasing the model non-identifiability.

4. COLLAPSED GIBBS SAMPLER INFERENCE

Given a directed network adjacency matrix E, our model's primary purpose is to estimate (1) a hierarchy over the network actors (implictly given by the paths $\mathbf{c}$), and (2) the actors' Multiscale Membership (MM) vectors $\boldsymbol{\theta}$ indicating their propensities to interact at different levels of the hierarchy. As secondary goals, we might also estimate (3) the sibling group compatibility matrices $\mathbf{B}$ so as to learn the communities' propensities to interact with one another, or (4) the donor/receiver link interaction levels $\mathbf{z}$ in order to discover the granularity of each interaction. To be precise, since our model is Bayesian, we seek the *posterior distributions* of $\mathbf{c}$, $\boldsymbol{\theta}$, $\mathbf{B}$ and $\mathbf{z}$ given E.

Unfortunately, finding the exact posteriors is infeasible — the number of possible assignments to the discrete random variables $\mathbf{c}$ and $\mathbf{z}$ is exponentially large, while analytic solutions to the posteriors of the continuous random variables $\boldsymbol{\theta}$ and $\mathbf{B}$ do not even exist. We must therefore resort to approximating the posteriors, via some approximate inference algorithm. Two general strategies exist for approximate inference: Markov Chain Monte Carlo (MCMC) methods (Robert and Casella 2004) that take random samples from the posterior, and variational approximations (Jordan, Ghahramani, Jaakkola and Saul 1999; Wainwright and Jordan 2008) that find the closest approximation to the posterior in some space of "simpler", analytically tractable distributions.

The literature contains many examples of variational approximations being applied to variants of the stochastic blockmodel (Airoldi et al. 2008; Xing, Fu and Song 2010; Nallapati, Ahmed, Xing and Cohen 2008). Variational approximations tend to be computationally faster than comparable MCMC methods, but lack guarantees on the quality of their approximation. In addition, nonparametric priors like the Nested Chinese Restaurant Process can have an unbounded number of parameters, hence they pose special difficulties for variational inference. This unbounded parameter issue can be solved by truncating the variational distribution (Blei and Jordan 2004), at the cost of introducing a new parameter that specifies

the degree of truncation. Wang and Blei (2009) have developed a variational approximation specifically for the nCRP, in which special techniques are introducted to handle truncation adaptively. In general however, variational inference in the nonparametric case remains more difficult to derive and implement than in the parametric case.

We have chosen an MCMC method for posterior inference in our model: specifically, a collapsed Gibbs sampling scheme (Liu 1994), in which the continuous random variables $\mathbf{B}$ and $\boldsymbol{\theta}$ are integrated out with the goal of simplifying the sampling equations. Gibbs sampling on discrete random variables is guaranteed to converge to the true posterior, though there are no general guarantees on how many samples are required for convergence. Unlike nonparametric variational inference, the unbounded number of parameters is not an issue for collapsed Gibbs samplers, hence we need not worry about truncation issues. In addition, the MCMC literature for nonparametric priors is mature; Gibbs sampling schemes have been derived for a variety of priors (Teh et al. 2006), including the nCRP itself (Blei et al. 2010).

4.1 Gibbs Sampling Equations

Approximate posterior inference in our model is conducted via a collapsed Gibbs sampling scheme (Liu 1994), in which some of the random variables are integrated out. Specifically, we integrate out two types of random variables: the first type are the community-compatibility matrices $\mathbf{B}$, which we integrate by exploiting conjugacy between the Beta and Bernoulli distributions. This adds conditional dependencies among interactions E_{ij} — specifically, all E_{ij} that use the same compatibility parameter $\mathbf{B}_{h_{coarse},h'_{coarse}}$ (by way of the levels $\mathbf{z}$ and paths $\mathbf{c}$) become dependent on each other when conditioned on $\mathbf{z}, \mathbf{c}$. However, E_{ij} that use different compatibility parameters remain conditionally independent given all other variables. The second type of random variable that we integrate are the MM vectors θ_i, by exploiting conjugacy between the multinomial distribution and the truncated stick breaking process; note that this adds dependence between levels $\mathbf{z}$ that share the same θ_i.

The point of integrating $\mathbf{B}, \boldsymbol{\theta}$ is that it may lead to faster convergence of the Gibbs sampler, and this technique is widely used in the latent space modelling community (Blei

et al. 2010; Mimno and McCallum 2007; Newman, Chemudugunta and Smyth 2006). Moreover, the resulting sampler is easier to implement because the remaining latent and observed variables $\mathbf{z}, \mathbf{c}, \mathbf{E}$ are all discrete. The reader might ask why we do not integrate $\mathbf{z}, \mathbf{c}$; our answer is that there are no known techniques for integrating these variables, and to the best of our knowledge, there is no evidence suggesting this will have any benefit over integrating $\mathbf{B}, \boldsymbol{\theta}$. Note that, given a sample of the remaining hidden variables $\mathbf{c}, \mathbf{z}$, the posterior distributions of $\mathbf{B}, \boldsymbol{\theta}$ are easily recovered using Bayes' Rule.

With $\mathbf{B}, \boldsymbol{\theta}$ integrated out, the only variables that need to be explicitly sampled are the levels $\mathbf{z}$ and the paths $\mathbf{c}$. We shall derive their Gibbs sampling equations below.

Sampling Levels z To get the collapsed level sampling equations, we begin with the joint distribution of $z_{\rightarrow ij}, \mathbf{B}, \boldsymbol{\theta}$ conditioned on all other variables, and then integrate out $\mathbf{B}, \boldsymbol{\theta}$:

$$
\begin{aligned}
&\int\int \mathbb{P}(z_{\rightarrow ij}, \mathbf{B}, \boldsymbol{\theta} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}; \gamma, m, \pi, \lambda_1, \lambda_2)\, d\mathbf{B} d\boldsymbol{\theta} \\
&= \mathbb{P}(z_{\rightarrow ij} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}; \gamma, m, \pi, \lambda_1, \lambda_2) \qquad \text{(integration)} \\
&= \frac{\mathbb{P}(E_{ij}, z_{\rightarrow ij} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}_{-(ij)}; \gamma, m, \pi, \lambda_1, \lambda_2)}{\mathbb{P}(E_{ij} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}_{-(ij)}; \gamma, m, \pi, \lambda_1, \lambda_2)} \qquad \text{(conditional probability definition)} \\
&\propto \mathbb{P}(E_{ij}, z_{\rightarrow ij} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}_{-(ij)}; \gamma, m, \pi, \lambda_1, \lambda_2) \qquad \text{(denominator does not depend on } z_{\rightarrow ij}) \\
&= \mathbb{P}(E_{ij} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(ij)}; \gamma, m, \pi, \lambda_1, \lambda_2)\, \mathbb{P}(z_{\rightarrow ij} \mid \mathbf{c}, \mathbf{z}_{-(\rightarrow ij)}, \mathbf{E}_{-(ij)}; \gamma, m, \pi, \lambda_1, \lambda_2) \qquad \text{(chain rule)} \\
&= \mathbb{P}(E_{ij} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(ij)}; \lambda_1, \lambda_2)\, \mathbb{P}(z_{\rightarrow ij} \mid \mathbf{z}_{i,(-j)}; m, \pi) \qquad \text{(conditional independence)} \qquad (5)
\end{aligned}
$$

where $\mathbf{E}_{-(ij)}$ is the set of all edges $\mathbf{E}$ except E_{ij}, and $\mathbf{z}_{-(\rightarrow ij)}$ is the set of all level indicators $\mathbf{z}$ except $z_{\rightarrow ij}$, and finally $\mathbf{z}_{i,(-j)} = \{z_{\rightarrow i\cdot}, z_{\leftarrow \cdot i}\} \setminus z_{\rightarrow ij}$ is the set of all $\mathbf{z}$'s that are drawn from θ_i except $z_{\rightarrow ij}$.

Two aspects of Eq. (5) are worth explaining. First, our goal is to Gibbs sample from the conditional distribution of $z_{\rightarrow ij}$ (with $\mathbf{B}, \boldsymbol{\theta}$ integrated out), i.e. line 2. Consequently, we can discard proportionality factors that do not depend on $z_{\rightarrow ij}$, such as the denominator $\mathbb{P}(E_{ij} \mid \dots)$ from line 3 to 4. Second, from line 5 to 6, we can change the 2nd $\mathbb{P}()$ term's conditioning variables from $\mathbf{z}_{-(\rightarrow ij)}$ to $\mathbf{z}_{i,(-j)}$ because (1) we are *not* conditioning on $\mathbf{E}_{ij}$,

and (2) we are conditioning on m, π. Hence, $z_{\rightarrow ij}$ is d-separated and thus conditionally independent from all $\mathbf{z}$'s *not* drawn from θ_i.

Moving on, let us now expand the first $\mathbb{P}()$ term:

$$\mathbb{P}(E_{ij} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(ij)}; \lambda_1, \lambda_2) = \frac{\Gamma(a+b+\lambda_1+\lambda_2)}{\Gamma(a+\lambda_1)\Gamma(b+\lambda_2)} \cdot \frac{\Gamma(a+E_{ij}+\lambda_1)\Gamma(b+(1-E_{ij})+\lambda_2)}{\Gamma(a+b+1+\lambda_1+\lambda_2)}, \tag{6}$$

$$a = \left|\left\{(x,y) \mid (x,y) \neq (i,j), \mathrm{S}_B^{xy} = \mathrm{S}_B^{ij}, E_{xy} = 1\right\}\right|, \quad b = \left|\left\{(x,y) \mid (x,y) \neq (i,j), \mathrm{S}_B^{xy} = \mathrm{S}_B^{ij}, E_{xy} = 0\right\}\right|.$$

Eq. (6) results from integrating the compatibility matrices $\mathbf{B}$ via Beta-Bernoulli conjugacy; notice that it is a function of $z_{\rightarrow ij}$ through the condition $\mathrm{S}_B^{xy} = \mathrm{S}_B^{ij}$ within the sub-expressions a, b. Because we have integrated $\mathbf{B}$, the random variable $z_{\rightarrow ij}$ is now dependent on all interactions E_{xy} that, *for the current sample value of* $\mathbf{z}, \mathbf{c}$, use the same compatibility parameter S_B^{ij} as E_{ij} (by way of $\mathbf{z}, \mathbf{c}$ and coarsening).

The second $\mathbb{P}()$ term can be computed using the law of total expectation, conditioned on the stick-breaking lengths $V_1, \ldots, V_K$ associated with $z_{\rightarrow ij}$:

$$\begin{aligned}
\mathbb{P}(z_{\rightarrow ij} = k \mid \mathbf{z}_{i,(-j)}, m, \pi) \;&=\; \mathbb{E}\left[\mathbb{I}(z_{\rightarrow ij} = k) \mid \mathbf{z}_{i,(-j)}, m, \pi\right] \;=\; \mathbb{E}\left[\mathbb{E}\left[\mathbb{I}(z_{\rightarrow ij} = k) \mid V_{i1}, \ldots, V_{ik}\right] \mid \mathbf{z}_{i,(-j)}, m, \pi\right] \\
&=\; \mathbb{E}\left[V_{ik} \prod_{u=1}^{k-1} (1 - V_{iu}) \mid \mathbf{z}_{i,(-j)}, m, \pi\right] \;=\; \mathbb{E}[V_{ik} \mid \mathbf{z}_{i,(-j)}, m, \pi] \prod_{u=1}^{k-1} \mathbb{E}[(1 - V_{iu}) \mid \mathbf{z}_{i,(-j)}, m, \pi] \\
&\propto\; \frac{m\pi + \#[\mathbf{z}_{i,(-j)} = k]}{\pi + \#[\mathbf{z}_{i,(-j)} \geq k]} \prod_{u=1}^{k-1} \frac{(1-m)\pi + \#[\mathbf{z}_{i,(-j)} > u]}{\pi + \#[\mathbf{z}_{i,(-j)} \geq u]},
\end{aligned} \tag{7}$$

where $\#[\mathbf{A} = x]$ is the number of elements in set $\mathbf{A}$ equal to x. The proportionality factor arises from truncating the stick-breaking process at level K, and is equal to $\sum_{k=1}^{K} \mathbb{P}(z_{\rightarrow ij} = k \mid \mathbf{z}_{i,(-j)}, m, \pi)$. Overall, Eq. (7) is a consequence of integrating out θ_i using multinomial-stick-breaking conjugacy, which makes $z_{\rightarrow ij}$ dependent on $\mathbf{z}_{i,(-j)}$.

For brevity, we omit the sampling equations for $z_{\leftarrow ij}$, as they are derived in similar fashion. The computational complexity of sampling a single z_{ij} is $\mathcal{O}(K)$, where K is the (fixed) depth of our hierarchy. Hence the total runtime required to sample all $\mathbf{z}$ is $\mathcal{O}(N^2 K)$.

Sampling Paths c As with the levels, we start with the joint distribution of $c_i, \mathbf{B}, \boldsymbol{\theta}$ conditioned on all other variables:

$$\begin{aligned}
\int\int \mathbb{P}(c_i \mid \mathbf{c}_{-i}, \mathbf{z}, \mathbf{E}; \gamma, m, \pi, \lambda_1, \lambda_2)\, d\mathbf{B} d\boldsymbol{\theta} \;&=\; \mathbb{P}(c_i \mid \mathbf{c}_{-i}, \mathbf{z}, \mathbf{E}; \gamma, m, \pi, \lambda_1, \lambda_2) \qquad \text{(integration)} \\
&\propto\; \mathbb{P}(c_i, \mathbf{E}_{(i\cdot),(\cdot i)} \mid \mathbf{c}_{-i}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \gamma, m, \pi, \lambda_1, \lambda_2) \qquad \text{(cond. probability; discarding denominator)} \\
&=\; \mathbb{P}(\mathbf{E}_{(i,\cdot),(\cdot,i)} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \gamma, m, \pi, \lambda_1, \lambda_2)\, \mathbb{P}(c_i \mid \mathbf{c}_{-i}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \gamma, m, \pi, \lambda_1, \lambda_2) \quad \text{(chain rule)} \\
&=\; \mathbb{P}(\mathbf{E}_{(i\cdot),(\cdot i)} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \lambda_1, \lambda_2)\, \mathbb{P}(c_i \mid \mathbf{c}_{-i}; \gamma) \qquad \text{(conditional independence)} \qquad (8)
\end{aligned}$$

where $\mathbf{E}_{(i\cdot),(\cdot i)} = \{E_{xy} \mid x{=}i \text{ or } y{=}i\}$ is the set of incoming and outgoing edges from entity i, and $\mathbf{E}_{-(i\cdot),-(\cdot i)}$ is its complement. In line 2, we have implicitly discarded the denominator $\mathbb{P}(\mathbf{E}_{(i\cdot),(\cdot i)} \mid \dots)$ as it does not depend on c_i.

The first term $\mathbb{P}(\mathbf{E}_{(i\cdot),(\cdot i)} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \lambda_1, \lambda_2)$, as a function of c_i, is

$$\text{1st term} = \prod_{B \in \mathbb{B}_{(i,\cdot),(\cdot,i)}} \frac{\Gamma(g_B + h_B + \lambda_1 + \lambda_2)}{\Gamma(g_B + \lambda_1)\Gamma(h_B + \lambda_2)} \cdot \frac{\Gamma(g_B + r_B + \lambda_1)\Gamma(h_B + s_B + \lambda_2)}{\Gamma(g_B + h_B + r_B + s_B + \lambda_1 + \lambda_2)}, \qquad (9)$$

$$\mathbb{B}_{(i\cdot),(\cdot i)} = \{\mathbf{B}_{h,h'} \mid \exists (i,j)[E_{ij} \in \mathbf{E}_{(i\cdot),(\cdot i)}, \mathrm{S}_B^{ij} = \mathbf{B}_{h,h'}]\},$$

$$g_B = \left|\left\{(x,y) \mid E_{xy} \in \mathbf{E}_{-(i\cdot),-(\cdot i)},\, \mathrm{S}_B^{xy} = B, E_{xy} = 1\right\}\right|, \quad h_B = \left|\left\{(x,y) \mid E_{xy} \in \mathbf{E}_{-(i\cdot),-(\cdot i)},\, \mathrm{S}_B^{xy} = B, E_{xy} = 0\right\}\right|,$$

$$r_B = \left|\left\{(x,y) \mid E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)},\, \mathrm{S}_B^{xy} = B, E_{xy} = 1\right\}\right|, \quad s_B = \left|\left\{(x,y) \mid E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)},\, \mathrm{S}_B^{xy} = B, E_{xy} = 0\right\}\right|.$$

Similar to Eq. (6), Eq. (9) is a consequence of integrating out $\mathbf{B}$ for all interactions E associated with actor i. In brief, the set $\mathbb{B}_{(i\cdot),(\cdot i)}$ contains all sibling group community compatibility matrix elements $\mathbf{B}_{h,h'}$ that are associated with some edge in $\mathbf{E}_{(i\cdot),(\cdot i)}$; note that these elements may not necessarily be from the same sibling group matrix. More precisely, $\mathbb{B}_{(i\cdot),(\cdot i)}$ is constructed as follows: (1) for each edge $E_{xy} \in \mathbf{E}_{(i\cdot),(\cdot i)}$, find E_{xy}'s corresponding sibling group matrix element $\mathbf{B}_{h,h'}$ by applying the coarsening procedure to $c_x, c_y, z_{\to xy}, z_{\leftarrow xy}$, and then (2) take the union over all matrix elements found this way. Because the sibling group matrices $\mathbf{B}$ have been integrated out, the set $\mathbb{B}_{(i\cdot),(\cdot i)}$ is only used to reference their sufficient statistics, through the $\mathrm{S}_B()$ function defined in Eq. (4). In particular, the four terms g_B, h_B, r_B, s_B are functions of B from the product inside $\mathbb{P}(\mathbf{E}_{(i\cdot),(\cdot i)} \mid \mathbf{c}, \mathbf{z}, \mathbf{E}_{-(i\cdot),-(\cdot i)}; \lambda_1, \lambda_2)$, and they represent counts of 0/1 edges associated with each $B \in \mathbb{B}_{(i\cdot),(\cdot i)}$.

As for the second term $\mathbb{P}(c_i \mid \mathbf{c}_{-i}; \gamma)$, it can be directly computed using the nCRP

definition Eq. (2). The computational complexity for a single c_i is $\mathcal{O}(NH)$, where H is the number of hierarchy nodes — hence the time to sample all $\mathbf{c}$ is $\mathcal{O}(N^2H)$. Note that $H = \mathcal{O}(NK)$, so the complexity of sampling all $\mathbf{c}$ is $\mathcal{O}(N^3K)$.

4.2 Hyperparameter Selection Using Metropolis-Hastings

The MSCB model contains 6 hyperparameters that need to be set: the hierarchy depth K, as well as the 5 prior hyperparameters $\gamma, m, \pi, \lambda_1, \lambda_2$. We will not discuss selection of K, expecting that the user knows how deep a hierarchy he or she needs — bearing in mind that model non-identifiability increases with increasing hierarchy depth. On the other hand, selecting all 5 prior hyperparameters is not a trivial affair and requires some attention. One could perform a gridsearch using the marginal likelihood of the network $\mathbf{E}$ as the objective function, but the search would be over 5 dimensions and thus impractical. Moreover, we would have to approximate the marginal likelihood as no analytic formula exists for computing it.

Clearly, we need a different strategy for selecting prior hyperparameters $\gamma, m, \pi, \lambda_1, \lambda_2$, and we choose to place hyper-priors on the hyperparameters, a common Bayesian technique for data-driven hyperparameter selection:

$$\gamma \sim \text{Exponential}(\eta_1), \quad m \sim \text{Beta}(\eta_2, \eta_3), \quad \pi \sim \text{Exponential}(\eta_4), \quad \lambda_1, \lambda_2 \sim \text{Exponential}(\eta_5).$$

Although this introduces 5 new hyper-hyperparameters $\eta_1, \ldots, \eta_5$, models with hyper-priors are typically less sensitive to the choice of hyper-hyperparameters than the original models were to the choice of hyperparameters (Bernardo, Smith and Berliner 2000). Thus, by setting the hyper-hyperparameters to reasonable values — all our experiments use $\eta_1, \ldots, \eta_5 = 1$ — we allow the model to decide the best values for the hyperparameters.

All that remains is to derive an inference algorithm for the model with hyper-priors. Because the hyper-priors are not conjugate to the other distributions in the model, we cannot derive Gibbs sampling equations for the hyperparameters $\gamma, m, \pi, \lambda_1, \lambda_2$. We overcome this via a general MCMC strategy, in which we alternate between sampling sweeps over all model latent variables using our collapsed Gibbs sampling equations from Section 4.1,

and sampling each hyperparameter using Independence Chain Metropolis-Hastings with the hyper-prior distributions as the proposals. This new inference algorithm is still a valid Markov Chain, although it may take longer to converge than our original model's Gibbs sampling equations (Robert and Casella 2004). In particular, using the hyper-priors as proposal distributions greatly simplifies the Metropolis-Hastings acceptance probabilities, reducing runtime complexity and making the algorithm easier to implement. The simplified acceptance probabilities are:

$$\mathbb{P}_a(\gamma_{new};\gamma_{old}) = \frac{\mathbb{P}(\mathbf{c};\gamma_{new})}{\mathbb{P}(\mathbf{c};\gamma_{old})}, \qquad \mathbb{P}_a(m_{new},\pi_{new};m_{old},\pi_{old}) = \frac{\mathbb{P}(\mathbf{z};m_{new},\pi_{new})}{\mathbb{P}(\mathbf{z};m_{old},\pi_{old})},$$
$$\mathbb{P}_a(\lambda_{1,new},\lambda_{2,new};\lambda_{1,old},\lambda_{2,old}) = \frac{\mathbb{P}(\mathbf{E}\mid\mathbf{c},\mathbf{z};\lambda_{1,new},\lambda_{2,new})}{\mathbb{P}(\mathbf{E}\mid\mathbf{c},\mathbf{z};\lambda_{1,old},\lambda_{2,old})}, \tag{10}$$

noting that we sample m jointly with π and sample λ_1 jointly with λ_2, and where

$$\mathbb{P}(\mathbf{c};\gamma) \;=\; \prod_{i=1}^{N}\prod_{k=1}^{K}\mathbb{P}(c_{ik}\mid c_{1:(i-1)},c_{i,1:(k-1)};\gamma), \tag{11}$$

$$\mathbb{P}(\mathbf{z};m,\pi) \;=\; \prod_{i=1}^{N}\prod_{j\neq i}^{N}\mathbb{P}(z_{\rightarrow i,j}\mid z_{\rightarrow i,1:j-1};m,\pi)\mathbb{P}(z_{\leftarrow j,i}\mid z_{\rightarrow i,1:N},z_{\leftarrow 1:j-1,i};m,\pi), \tag{12}$$

$$\mathbb{P}(\mathbf{E}\mid\mathbf{c},\mathbf{z};\lambda_1,\lambda_2) = \prod_{B\in\mathbb{B}}\frac{\Gamma(\lambda_1+\lambda_2)}{\Gamma(\lambda_1)\Gamma(\lambda_2)}\cdot\frac{\Gamma(a_B+\lambda_1)\Gamma(b_B+\lambda_2)}{\Gamma(a_B+b_B+\lambda_1+\lambda_2)}, \tag{13}$$

$$a_B = \left|\{(x,y)\mid \mathrm{S}_B^{xy}=B, E_{xy}=1\}\right|, \quad b_B = \left|\{(x,y)\mid \mathrm{S}_B^{xy}=B, E_{xy}=0\}\right|,$$

with $\mathbb{B}$ being the set of all sibling group compatibility matrix elements. Eq. (13) is similar to Eq. (9), except that we now consider all network edges instead of just those incident to some network entity i. As for Eq. (11,12), they result from applying the chain rule to exchangeable distributions; in the case of Eq. (11), we have applied it to the nCRP distribution over all paths $\mathbf{c}$, whereas in Eq. (12), we have applied it to N compound-stick-breaking-multinomial distributions, each corresponding to the set of level indicators $\mathbf{z}_i$ associated with some network entity i. The product terms in Eq. (11) are computed using the nCRP definition Eq. (2), while the terms in Eq. (12) are:

$$\mathbb{P}(z_{i,x} \;=\; k\mid \mathbf{z}_{i,subset};m,\pi) \;\propto\; \frac{m\pi+\#[\mathbf{z}_{i,subset}=k]}{\pi+\#[\mathbf{z}_{i,subset}\geq k]}\prod_{u=1}^{k-1}\frac{(1-m)\pi+\#[\mathbf{z}_{i,subset}>u]}{\pi+\#[\mathbf{z}_{i,subset}\geq u]},$$

where $z_{i,x}$ and $\mathbf{z}_{i,subset}$ denote an element and a subset respectively of $\mathbf{z}_i=\{z_{\rightarrow i\cdot},z_{\leftarrow\cdot i}\}$, and where the normalization factor is $\sum_{k=1}^{K}\mathbb{P}(z_{i,x} \;=\; k\mid \mathbf{z}_{i,subset};m,\pi)$.

By adopting this MCMC algorithm for simultaneous hyperparameter selection and posterior inference, we have reduced user input to just the hierarchy depth K. This not only makes our model easier to use, but also provides a data-driven means of choosing the hyperparameters $\gamma, m, \pi, \lambda_1, \lambda_2$. Finally, the computational complexity of drawing all 5 hyperparameters and testing their acceptance probabilities is $\mathcal{O}(N^2K)$, which is asymptotically less than a single $\mathcal{O}(N^3K)$ Gibbs sampler sweep over all latent variables. In other words, using our Metropolis-Hastings strategy for hyperparameter selection does not incur a significant computational cost over Gibbs sampling the latent variables.

To initialize our MCMC algorithm, we first set the hyperparameters $m, \pi, \gamma, \lambda_1, \lambda_2$ to user-provided initial values, and then randomly draw the latent variables $\mathbf{c}, \mathbf{z}$ according to the MSCB generative process (Section 2.4). Once done, we alternate between sampling the latent variables and the hyperparameters, as described earlier.

5. SIMULATION

We now evaluate our inference algorithm's ability to recover hierarchies from data simulated from our model. Our goal is to examine how MSCB's ability to model both assortative (within-community) interactions and disassortative (cross-community) interactions differentiates it from standard hierarchical clustering algorithms.

For all simulations, the number of actors N was 300. For $K = 2$, $\theta = (0.25, 0.75)$ for all actors, meaning that actors interact at level 1 25% of the time and level 2 75% of the time. For $K = 3$, $\theta = (0.1, 0.3, 0.6)$ for all actors.

Our experiments explore the effect of different compatibility matrices $\mathbf{B}$. We first explore networks generated from "on-diagonal" $\mathbf{B}$s, where the diagonal elements are much larger than the off-diagonal elements (strong assortative interactions). We also investigate "off-diagonal" $\mathbf{B}$s, where the off-diagonal elements are larger (strong disassortative interactions). "Low noise" means the on/off diagonal element values are far apart, while "high noise" means they are closer together. Specifically, the types of $\mathbf{B}$s explored are:

1. **K=2, on-diagonal, low noise** - $B_{on-diagonal} = (0.4, 0.8)$, $B_{off-diagonal} = (0.02, 0.02)$;

2. **K=2, off-diagonal, low noise** - $B_{on-diagonal} = (0.02, 0.02)$, $B_{off-diagonal} = (0.4, 0.8)$;

3. **K=2, on-diagonal, high noise** - $B_{on-diagonal} = (0.3, 0.6)$, $B_{off-diagonal} = (0.1, 0.2)$;

4. **K=2, off-diagonal, high noise** - $B_{on-diagonal} = (0.1, 0.2)$, $B_{off-diagonal} = (0.3, 0.6)$;

5. **K=3, on-diagonal, low noise** - $B_{on-diagonal} = (0.5, 0.7, 0.9)$, $B_{off-diagonal} = (0.02, 0.02, 0.02)$;

6. **K=3, off-diagonal, low noise** - $B_{on-diagonal} = (0.02, 0.02, 0.02)$, $B_{off-diagonal} = (0.5, 0.7, 0.9)$.

7. **K=3, on-diagonal, high noise** - $B_{on-diagonal} = (0.4, 0.6, 0.8)$, $B_{off-diagonal} = (0.1, 0.1, 0.2)$;

8. **K=3, off-diagonal, high noise** - $B_{on-diagonal} = (0.1, 0.1, 0.2)$, $B_{off-diagonal} = (0.4, 0.6, 0.8)$.

$B_{on-diagonal} = (a, b)$ means that actors interacting in the same level-1 community do so with probability a, while actors interacting in the same level 2 community do so with probability b (and analogously for $B_{off-diagonal}$).

We compare our approach to two baselines. The first is hierarchical spectral clustering, an adaptation of spectral clustering (Chung 1997) to top-down hierarchical clustering. Because spectral clustering does not specify how to select the number of clusters at each hierarchy node, we shall explore two variants that represent worst- and best-case scenarios respectively. The first variant, *Spectral-Binary*, does binary splits at every level. For the second variant, *Spectral-Oracle*, **we give it the number of 1st level branches as an advantage**, and then perform binary splits at deeper levels. We also compare to agglomerative clustering with Ward's criterion (Ward 1963) with the dissimilarity measure used in Pajek (Batagelj and Mrvar 1998), (Batagelj and Ferligoj Patrick 1992). Like spectral clustering, we also have two variants: *Ward-Binary)* and *Ward-Oracle*. *Ward-Binary* does binary splits at all levels, while *Ward-Oracle* is given the true number of first level clusters as an advantage, but does binary splits for deeper levels.

For our approach, we intialize $m = \pi = \lambda_1 = \lambda_2 = .5$ and $\gamma = 0.1$. We run our collapsed gibbs sampler for 10,000 burn-in iterations, and then draw 10 samples with 100

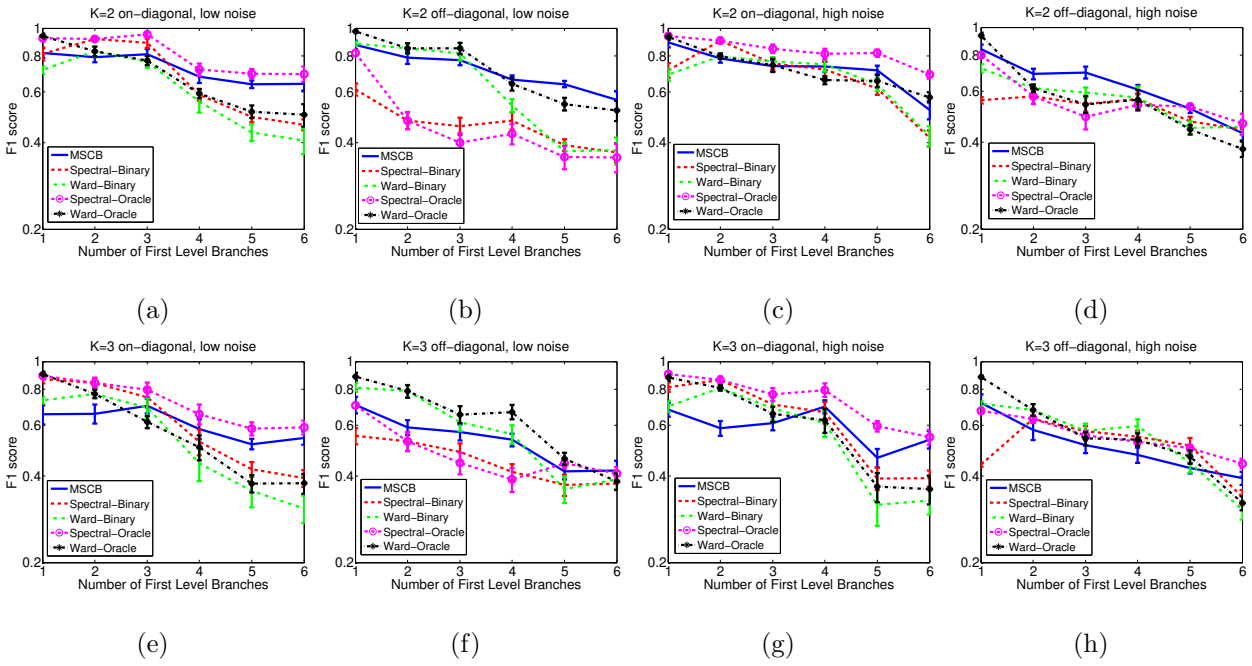

Figure 5: Simulation Results for on/off diagonal compatibility matrices in low/high noise settings for depth of 2 and 3. We compare our method, MSCB (blue) against both hierarchical spectral clustering (red/pink, see text for details), and Ward's method as implemented in Pajek (green/black) (Batagelj and Mrvar 1998).

lag in-between. We calculate the F1 score at each level k, $\text{F1}_k = \frac{2*Precision*Recall}{Recall+Precision}$ where $Recall = \frac{TP}{TP+FN}$, and $Precision = \frac{TP}{TP+FP}$. TP is true positive count (actors that should be in the same cluster, up to depth k), FP is false positive count, TN is true negative count, and FN is false negative count. The total F1 score is computed by averaging the F1_k scores for each level. For our approach, we average the F1 score over all the samples.

Figure 5 shows the F1 scores for all algorithms, as a function of the number of size $\geq$ 10 branches at the true hierarchy's 1st level. To ensure a fair comparison, this was also the number of 1st level branches given to the *Oracle* variants of both baselines. From Figure 5(a), 5(c), 5(e), and 5(g), we see that when $\mathbf{B}$ is strongly on-diagonal, our method generally performs better than or comparable to all other methods when there are more than 3 first level branches, (except *Spectral-Oracle*), demonstrating its ability to determine the correct number of clusters from the data. In these on-diagonal experiments, Ward's method tends to perform the worst while MSCB and Spectral perform better.

However, once $\mathbf{B}$ is strongly off-diagonal (implying strong cross-community interactions), Spectral performs poorly. This is to be expected — by formulation, spectral clustering cannot recover disassortative communities. On the other hand, our method continues to yield good results (Figures 5(b), 5(d), 5(f), and 5(h)) comparable to the on-diagonal $\mathbf{B}$ case. Ward's method can also recover dissassortative communities, but for K=2, we outperform the binary variant and perform comparably to the oracle variant. For the K=3 off-diagonal results (Figures 5(f) and 5(h)), Ward's tends to perform a little better until the number of branches gets very large. However, Ward's does not perform as well on the assortative simulations.

As a result, our method performs consistently well in all settings, and we note that for larger numbers of branches it outperforms both the binary variants of Ward and Spectral in most scenarios. The oracle variants sometimes perform better, but these require a priori knowledge of the number of first level branches, which our method does not.

6. HELD-OUT EVALUATION

Previously, we evaluated our MCMC algorithm's performance against non-probabilistic hierarchical clustering schemes. One might then ask how our algorithm compares to probabilistic network models that do not model hierarchical structure. In particular, does MSCB's hierarchy assumption permit a better statistical fit to real-world networks, in terms of the network marginal log-likelihood $\mathbb{P}(\mathbf{E})$?

In this section, we compare our MSCB inference algorithm to the inference algorithms of two other probabilistic models: the Infinite Relational Model (IRM) (Kemp et al. 2006), essentially a nonparametric version of the stochastic blockmodel (Wang and Wong 1987), and the Mixed Membership Stochastic Blockmodel (Airoldi et al. 2008), a variant of the stochastic blockmodel that permits network entities to come from multiple communities (which Airoldi *et al.* call "roles"). Since the IRM is a special case of MSCB when $K = 1$, we reused our full MCMC algorithm (Gibbs sampler plus Metropolis-Hastings) for posterior inference and hyperparameter selection. As for MMSB, we used the variational EM algorithm

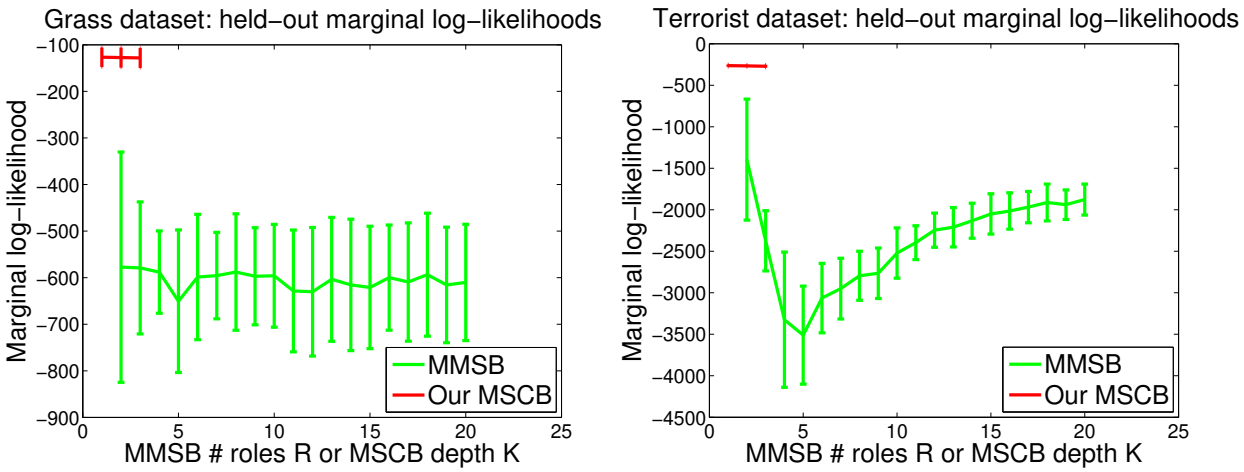

Figure 6: Average held-out marginal likelihoods $\mathbb{P}(E_{test}; \hat{\Theta})$ and standard deviations for our MSCB inference algorithm with hierarchy depth $K = 1$ (equivalent to IRM), 2 and 3, versus the MMSB variational inference algorithm for $R \in \{2, \ldots, 20\}$ roles.

in Airoldi *et al.* (2008), which also performs both posterior inference and hyperparameter selection. In Section 3, we discussed how MSCB relates to IRM and MMSB: recall that the IRM is a special case of MSCB with hierarchy depth $K = 1$, while MSCB in turn is a highly-constrained version of MMSB. Particularly, we noted that MMSB is highly non-identifiable, and that MSCB, while also non-identifiable, is much less so than MMSB. On this point, we expect our MSCB inference/selection algorithm to perform better than MMSB's.

Our experiments use two real-world datasets: a 75-species food web of grass-feeding wasps (Dawah et al. 1995; Clauset, Moore and Newman 2008), and the 62-actor September 11th, 2001 hijacker terrorist network (Krebs 2002; Clauset et al. 2008). These networks reflect common real-world modes of interaction: edges in the food web denote predator-prey relationships, while edges in the terrorist network reflect social cohesion. The food web could be represented as a hierarchy where different branches reflect different trophic levels (e.g. parasite, predator or prey), while the terrorist network could be interpreted as an organization chart.

At a high level, we conducted our held-out evaluation as follows: for each model, we (1)

used the corresponding inference/selection algorithm to estimate model hyperparameters $\hat{\Theta}$ for a training network $\mathbf{E}_{train}$ (ignoring the latent variable posteriors), and then (2) estimated the test network marginal log-likelihood $\mathbb{P}(\mathbf{E}_{test}; \hat{\Theta})$, so as to evaluate how well each inference/selection algorithm estimates parameters for its model. More specifically, for both datasets, we generated 5 pairs of training and test subgraphs; each pair was obtained by randomly partitioning the actors into two equal sets, and keeping only the edges within each partition. Then, for each of the 3 models on each of the 5 training graphs, we selected the model hyperparameters $\hat{\Theta}$ using the appropriate inference/selection algorithm[4]. Finally, we estimated the corresponding test network's marginal log-likelihood $\mathbb{P}(\mathbf{E}_{test}; \hat{\Theta})$ using 10,000 importance samples, and averaged the results over all 5 train-test network pairs. The initial hyperparameter values for this experiment were $m, \pi = 0.5$ and $\gamma, \lambda_1, \lambda_2 = 1$.

Figure 6 displays the results for this experiment. On both datasets, observe that our MSCB algorithm achieves greater held-out marginal log-likelihoods $\mathbb{P}(E_{test}; \hat{\Theta})$ than MMSB, regardless of the MSCB hierarchy depth K or MMSB number of roles R. We believe this is related to MSCB being significantly more constrained than MMSB, and thus more identifiable. Moreover, MMSB's likelihood peaks at $R = 2$ on both datasets, suggesting that we should choose just 2 roles, too few to provide anything but an extremely coarse network analysis. In contrast, our MSCB inference algorithm uses model hyperparameters to automatically select a suitable size and shape for its hierarchy — for example, the grass dataset training network posteriors for $K = 2$ had an average of 10.0 hierarchy nodes, a reasonable number considering that the training networks have 38 actors each. This illustrates one of the advantages of nonparametric models like MSCB and IRM over parametric models

[4]Algorithm details: for the IRM/MSCB MCMC algorithm, we took 10,000 samples as burn-in, and then estimated each hyperparameter $\gamma, m, \pi, \lambda_1, \lambda_2$ by its average over the next 1,000 samples. This was repeated for hierarchy depths $K = 1$ (i.e. IRM), 2 and 3. For the MMSB variational EM algorithm, we ran 100 random restarts until convergence, and then took hyperparameter estimates from the restart with the highest variational lower bound (with respect to the true likelihood). Because MMSB requires the number of latent roles R to be specified, we repeated this experiment for each $2 \leq R \leq 20$.

such as MMSB. Finally, we note that the differences of heldout likelihoods between $K = 1$ (IRM), 2 and 3 for MSCB are negligible and within error, suggesting that the increased non-identifiability from larger K has minimal negative impact on model fit.

7. EFFECTS OF HYPERPARAMETER INITIALIZATION

Apart from latent variable inference for $\mathbf{c}, \mathbf{z}$, our MCMC algorithm also performs hyperparameter selection for $\gamma, m, \pi, \lambda_1, \lambda_2$. Given that the final hyperparameter estimates may depend on their initial settings, it is only natural that we study how the former changes with the latter. In this section, we conducted experiments on the grass-feeding wasps and terrorist networks, repeating our MCMC algorithm over different initial hyperparameter values. We studied one out of the five hyperparameters at a time, while initializing the four remaining hyperparameters to the following default values: $m = 0.5$ and $\pi, \gamma, \lambda_1, \lambda_2 = 1$. The hierarchy depth was fixed to $K = 2$ throughout. For the hyperparameter under study, we ran our algorithm for five different initializations of that hyperparameter: $m \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ or $\pi, \gamma, \lambda_1, \lambda_2 \in \{0.01, 0.1, 1, 10, 100\}$. Thus, we studied 25 different hyperparameter settings in total: 5 for each of the five hyperparameters. For each of these 25 settings, we took 5 repeat trials of 1,000 samples, with 10,000 iterations of burn-in prior to taking samples. In each trial, we estimated the hyperparameter under study by its mean value over the 1,000 samples, and we present each estimate's average and standard deviation over the 5 trials in Tables 1 and 2.

For either dataset, our MCMC algorithm's final estimates of λ_1, λ_2 have low variance, and are highly invariant to the choice of initial values. Since λ_1, λ_2 influence the posterior of the community-compatibility matrices $\mathbf{B}$, this suggests that our MCMC algorithm reliably estimates the posterior of $\mathbf{B}$. The low values of λ_1 compared to λ_2 reflect the fact that both networks have more 0-edges than 1-edges.

The situation for m, π, is more nuanced. Recall that these hyperparameters respectively control the posterior mean and variance of the Multiscale Membership vectors $\boldsymbol{\theta}$. In Section 3, we argued that the MM vectors lead to some non-identifiability in the MSCB model, and

Table 1: Grass-feeding wasps network: Final hyperparameter estimates under different initializations.

Initial m	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
Final m	0.878 ± 0.126	0.833 ± 0.111	0.842 ± 0.130	0.832 ± 0.171	0.849 ± 0.0454
Initial $\pi, \gamma, \lambda_1, \lambda_2$	**0.01**	**0.1**	**1**	**10**	**100**
Final π	2.22 ± 2.08	2.11 ± 0.953	1.25 ± 0.927	9.75 ± 2.09	100 ± 0
Final γ	1.64 ± 0.395	1.66 ± 0.587	2.67 ± 0.950	3.08 ± 2.29	21.8 ± 43.8
Final λ_1	0.0869 ± 0.0338	0.0641 ± 0.0172	0.0516 ± 0.0149	0.0645 ± 0.0366	0.0656 ± 0.0218
Final λ_2	1.08 ± 0.394	1.26 ± 0.453	1.19 ± 0.465	0.840 ± 0.265	1.24 ± 0.532

Table 2: Terrorist network: Final hyperparameter estimates under different initializations.

Initial m	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
Final m	0.0801 ± 0.0692	0.0982 ± 0.0518	0.144 ± 0.0708	0.0695 ± 0.0227	0.116 ± 0.126
Initial $\pi, \gamma, \lambda_1, \lambda_2$	**0.01**	**0.1**	**1**	**10**	**100**
Final π	0.406 ± 0.296	0.937 ± 0.545	1.64 ± 1.01	6.31 ± 5.06	100 ± 0
Final γ	2.15 ± 0.683	2.71 ± 1.97	2.63 ± 1.83	2.75 ± 0.678	3.49 ± 0.785
Final λ_1	0.0656 ± 0.0140	0.0805 ± 0.0239	0.0776 ± 0.0381	0.0877 ± 0.0188	0.0921 ± 0.0143
Final λ_2	0.252 ± 0.00766	0.221 ± 0.0931	0.292 ± 0.0602	0.287 ± 0.0843	0.248 ± 0.0348

it is therefore not surprising that m, π are more difficult to estimate reliably. On either dataset, the estimates of $m \in (0, 1)$ generally have low variance and are reasonably constant across initial values, implying that the posterior mean estimates of $\boldsymbol{\theta}$ are fairly reliable. We can even interpret these estimates: for the grass dataset, the estimates of $m \approx 0.85$ imply that the MM vectors $\boldsymbol{\theta}$ place almost all mass at the first level of the hierarchy, whereas the terrorist dataset estimates of $m \approx 0.1$ imply that the MM vector mass is evenly distributed over the first and second levels. On the other hand, the estimates of π tend to vary with initialization (particularly when $\pi = 100$), and exhibit high variance over trials with the same initial value. Despite this, we note that π only controls the posterior variance of the

MM vectors $\boldsymbol{\theta}$. Hence, if we only desire good mean estimates of $\boldsymbol{\theta}$, we may not necessarily need accurate estimates of π.

Lastly, we observe that γ's final estimate tends to increase with its initial setting. This is a consequence of our MCMC algorithm, which initializes the actor paths $\mathbf{c}$ using the MSCB generative model. A high initial γ creates a starting tree with many branches, and the MCMC algorithm is unable to merge[5] all the superfluous branches within the 10,000 burn-in iteration limit, causing the final estimate of γ to be higher than it would have been otherwise. Thus, care should be taken when choosing an initial value for γ, for high values will lead to more highly-branched hierarchies.

8. REAL-DATA QUALITATIVE ANALYSIS

In this section, we apply the MSCB inference/selection algorithm to interpret two real-world networks: (1) a 75-species food web of grass-feeding wasps and their parasitoids, and (2) a citation network containing 1,000 papers from the High-Energy Physics (Theory) portion of arXiv, an open-access archive of scientific paper preprints. Our objective is to study how the three network aspects we seek — hierarchy, multiscale granularity, and assortativity/disassortativity — manifest in real-world networks.

For both networks, we ran our algorithm with $K = 2$ levels, and initialized the hyperparameters to $m, \pi = 0.5$ and $\gamma, \lambda_1, \lambda_2 = 1$. We discarded 10,000 iterations as burn-in, and then took 1,000 samples to estimate the posterior distribution over actor levels $\mathbf{z}$ and paths $\mathbf{c}$. One issue we faced was the lack of an obvious way to visualize the posterior hierarchy; the posterior mean or mode of $\mathbf{c}$ do not represent intelligible hierarchies because of permutation non-identifiability[6]. In order to represent the posterior hierarchy meaningfully,

[5] One could address this by a Metropolis-Hastings scheme that splits and merges hierarchy branches, but the development of such a scheme is out of the scope of this work.

[6] This issue is not unique to MSCB, but occurs when naively applying MCMC methods to models with permutation non-identifiability, such as stochastic blockmodels or models meant for clustering. In many such models, permuting the communities/clusters has no effect on the model likelihood, so the posterior mean must be an average over all permutations — which is usually uninterpretable.

we constructed a "consensus" hierarchy from the samples: for each level $1 \leq k \leq K$ and each pair of actors i, j, we computed the fraction of samples S_{ijk} in which i, j had the same level-k hierarchy position. If i, j shared the same level-k position in at least τ of all samples, i.e. $S_{ijk} \geq \tau$, then we assigned them identical level-k positions in the consensus. Higher threshold values τ produce wider, more detailed consensus hierarchies, whereas lower values give rise to simpler hierarchies. We use $\tau = 0.35$ in our analyses, as it provides a good middle ground between detail and interpretability.[7]

As for interaction levels $\mathbf{z}$, we represent each $z_{\rightarrow ij}$'s or $z_{\leftarrow ij}$'s posterior by taking its K-bin histogram over all its samples (recall that the $\mathbf{z}$ are discrete with K possibilities). Note that our ultimate goal is really the MM vectors $\boldsymbol{\theta}$; we obtain the posterior mean of actor i's MM θ_i by averaging the histograms of all zs that, according to the generative model, are drawn from θ_i (that is to say, $\{z_{\rightarrow i\cdot}, z_{\leftarrow \cdot i}\}$).

8.1 Grass-Feeding Wasp Parasitoids Food Web

We first consider the $N = 75$ species grass-feeding wasps food web. In this network, the actors are species in a food web, and the 113 directed edges represent predator-prey relationships. Each species in the network is annotated with its position or "trophic level" in the food web: grass, herbivore, parasitoid, hyper-parasitoid (parasites that prey on other parasites), or hyper-hyper parasitoid. We stress that these trophic levels are *not* the hierarchy levels in our model, but are nodal attributes of the species.

The inference/selection algorithm took 9 minutes on a 2.83GHz Intel Core 2 processor, and the average hyperparameters were $\gamma = 1.49, m = 0.915, \pi = 2.30, \lambda_1 = 0.0774, \lambda_2 = 1.50$. The high value of m suggests that most interactions were shallow (occurring at level 1), while the large ratio λ_2/λ_1 is expected since the number of edges is $\ll N^2$. We report the posterior "consensus" hierarchy and mean Multiscale Membership (MM) vectors in Figure

[7] A more thorough analysis would include consensus hierarchies over multiple values of $\tau \in [0, 1]$, so as to present a fuller complete picture of hierarchy posterior variation. Alternatively, one could analyze the "closeness" of pairs of network actors by their number of shared path samples.

7. In the same Figure, we also show the original network, where each interaction $E_{ij} = 1$ has been colored according to the 2nd-level communities involved (missing links $E_{ij} = 0$ are not shown). The trophic level annotations are shown in the network by node shapes, and summarized as counts in the hierarchy.

Generally speaking, the first-level super-communities separate the trophic levels. For instance, all grass species are found in community 3, while community 2 contains all but one of the herbivores, and community 1 contains most of the parasitoids. Notice that the trophic levels form a set of *disassortative* communities, e.g. we see that herbivores feed on grasses, but not on other herbivores. We contrast our results with those of Clauset *et al.* (2008), who did not recover this structure because their method assumes all communities are assortative. On the other hand, our model is able to learn disassortative network structures by virtue of its stochastic blockmodel assumption.

Let us analyze the sub-communities in detail. We begin with super-community 4, which is separated from the rest of the network by just one edge — from species 20 (*Tetramesa petiolata*) to 6 (*Deschampsia cespitosa*) — and therefore represents an isolated sub-web. Observe that species 20, a herbivore, is unexpectedly found in sub-community 4.2 rather than 2.1. We explain this by noting that species 20 is the only herbivore that the parasitoids in sub-communities 4.1, 4.3 and 4.4 prey on. Additionally, the sub-communities within super-community 4 are topologically ordered, which reflects their food web trophic levels. To be precise, the species in 4.1 prey on $4.2-4$, while 4.4 preys only on $4.2-3$, and 4.3 preys only on 4.2. Next, consider super-communities 1 and 5. While the bulk of the parasitoids are in super-community 1, the apex parasitoids that prey on them are all in super-community 5. The distinction between apex parasitoids in sub-communities 5.1 and 5.2 appears to be driven by the number of parasitoids they prey upon: species 67 (*Macroneura vesicularis*) from 5.2 preys on more parasitoids (specifically, 8), whereas species 65 (*Eupelmus atropurpureus*) and 75 (*Mesopolopus graminum*) from 5.1 prey on fewer (4 and 5 parasitoids respectively).

Finally, we inspect the posterior mean of the MM vectors $\boldsymbol{\theta}$, shown in Figure 7. Intuitively, the MM vector θ_i shows how often species i interacts at the generic super-community level,

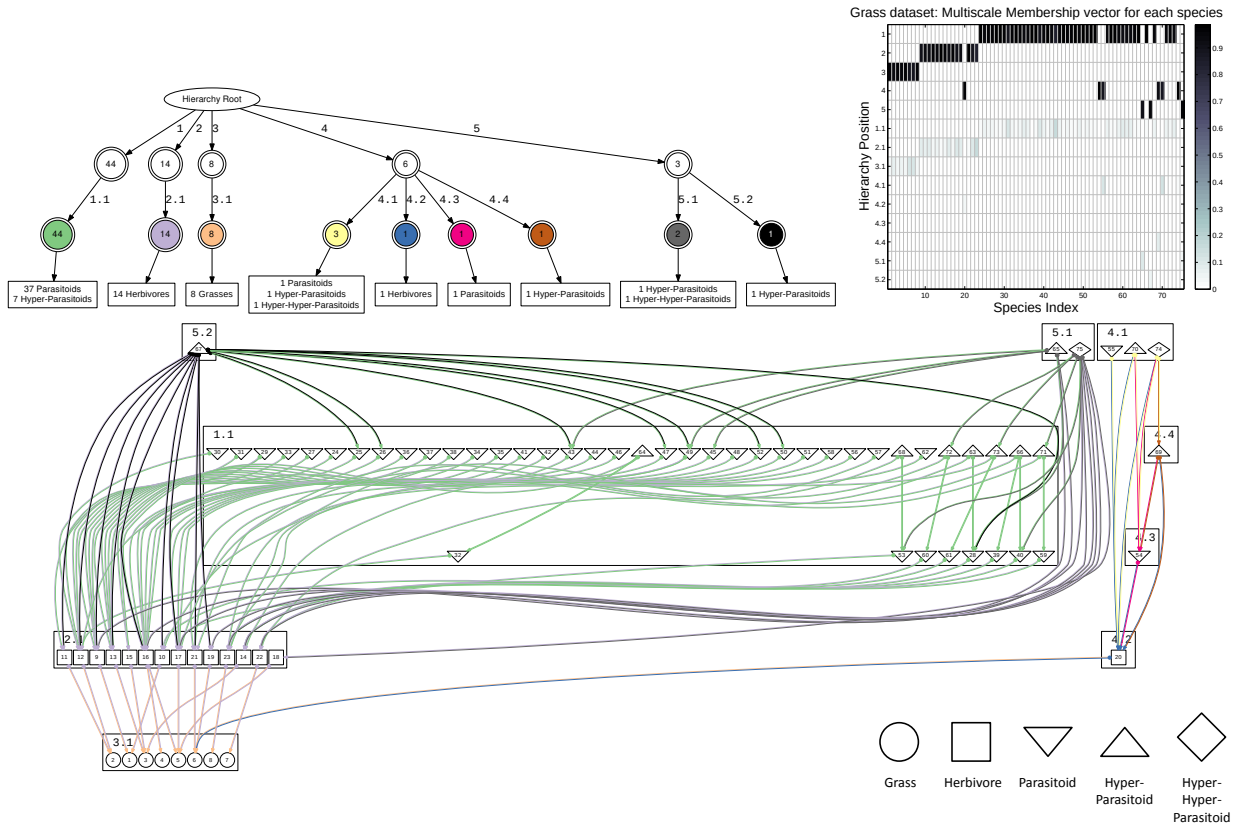

Figure 7: Grass-feeding wasps network: **Top Left:** Inferred hierarchy of communities, with community trophic level counts at the bottom. **Top Right:** Multiscale Membership vectors. Columns correspond to actor MM vectors θ_i, while rows correspond to hierarchy positions. Note that each MM vector θ_i can be nonzero only at the $K = 2$ hierarchy positions along its corresponding actor path c_i. Cell intensities reflect the value of θ_i at a particular hierarchy position — black and white correspond to 1 and 0 respectively, while shades of blue correspond to values between 0 and 1. **Bottom:** Original network. Each edge is drawn with two colors, representing its 2nd-level level donor/receiver communities as inferred by our algorithm. Node shapes represent annotated trophic levels (see legend in bottom right).

versus the more specific sub-community level. In this network, all species interact primarily at the super-community level, with occasional interaction at the sub-community level. This suggests that the network structure is mostly explained by the hierarchy's first level, while the second level is responsible for local structural details. Most second level interactions are found in super-communities 1, 2 and 3, corresponding to the majority of parasitoids, herbivores and grass species respectively. Because second-level interactions are used by the MSCB model to account for atypical behavior within super-communities, species with many such interactions are likely to have specialized roles in the food web, and thus make good targets for further investigation.

8.2 High-Energy Physics Citation Network

For our final experiment, we consider an $N = 1,000$-paper subgraph of the arXiv high-energy physics citation network, taken from the 2003 KDD Cup (2010). This subgraph was constructed by subsampling papers involved in citations from Jan 2002 through May 2003. The average hyperparameters from the inference/selection algorithm were $\gamma = 7.76, m = 0.851, \pi = 0.492, \lambda_1 = 0.0169, \lambda_2 = 0.976$, and the algorithm took 12.4 days to run on a 2.83GHz Intel Core 2 processor. This high runtime is a consequence of two things: the $\mathcal{O}(N^3K)$ runtime complexity per Gibbs sampler sweep, as well as the large value of γ, which reflects a high branching factor in the posterior hierarchy.

Figure 8 shows part of the posterior consensus hierarchy, where each displayed community has been annotated with the number of papers associated with it, as well as the most frequent title words from those papers[8]. We stress that the hierarchy is learnt only from the citation network adjacency matrix, without knowledge of the paper texts. In addition, we reveal the network community structure in Figure 9, by permuting the original adjacency matrix to match the order of inferred communities. The same figure also shows the posterior mean of

[8]We note that this output is reminiscent of text models from Natural Language Processing, particularly the Latent Dirichlet Allocation (Blei, Ng and Jordan 2003). However, we stress that MSCB is *not* a text model; the title words are determined post hoc, after hierarchy inference.

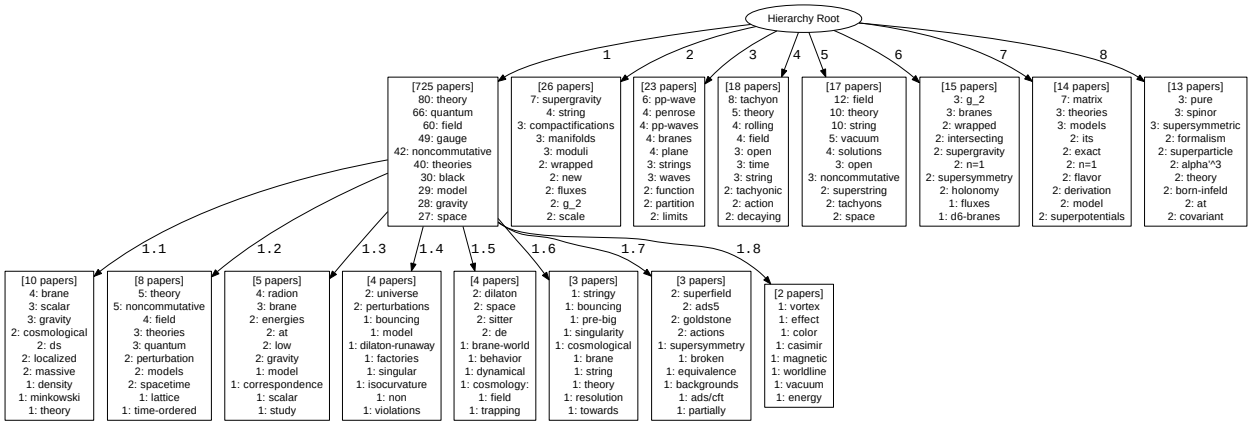

Figure 8: HEP network: Inferred community hierarchy. Each hierarchy node is annotated with its number of associated papers, and the most common title words from those papers. Due to space contraints, we only show the largest 8 1st-level communities, as well as the largest 8 2nd-level sub-communities under 1st-level community 1. Regarding the other 1st-level communities, all their sub-communities contain only one document, thus they are uninteresting and have been hidden to save space.

the MM vectors, as a histogram over 2nd-level elements $\theta_{i,2}$.

The consensus hierarchy reflects strong assortative organization, and we expect most communities to correspond to specific areas of study. The giant community 1 contains 725 papers, and is clearly visible as the sparse diagonal block convering indices 1 to 725 in the adjacency matrix of Figure 9. Its top title words are general physics terms such as "theory", "quantum" and "field", implying that, as a whole, this community does not represent any specific subfield of theoretical physics. This observation is further supported by the fact that papers in community 1 have few citations among themselves or to the rest of the network.

The remaining 1st-level communities exhibit denser within-community citations than community 1. We hypothesize that they are associated with small groups of researchers that work on specific subfields, and who are highly likely to cite each others' work. For instance, community 2 (26 papers) corresponds to the dense diagonal block between indices 726 and 751 in Figure 9, and its top title keywords suggest it is predominantly about research in supergravity and string theory. In similar fashion, observe that community 3 (23 papers) corresponds to indices 752 through 774, and is focused on pp-waves and Penrose limits. The

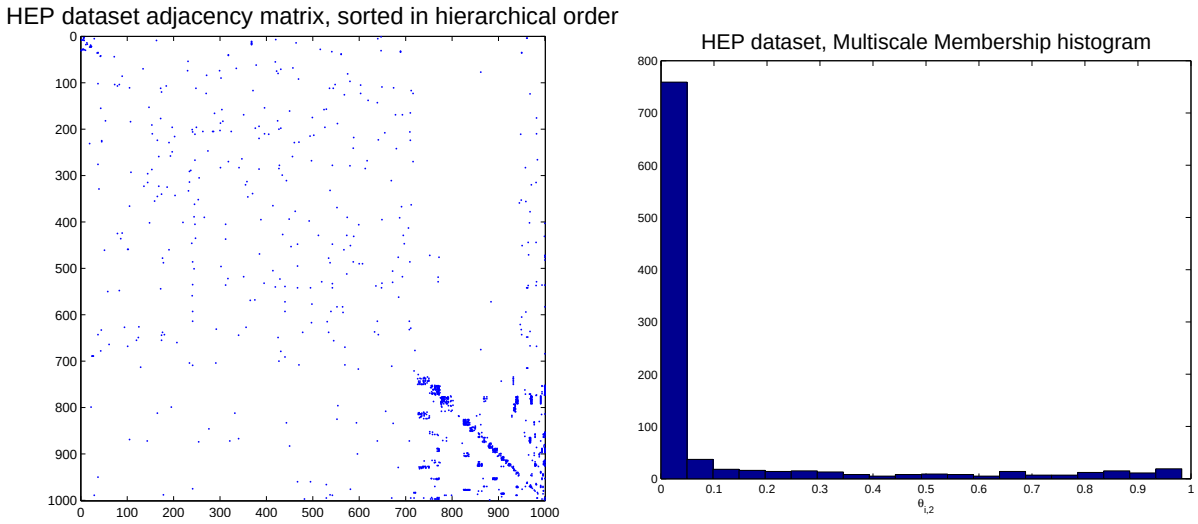

Figure 9: HEP network: **Left:** Adjacency matrix, permuted to match the order of communities in Figure 8. The blue dots represent citations from row i to column j. **Right:** Histogram of Multiscale-Membership vectors. The x-axis represents the value of $\theta_{i,2}$, i.e. the 2nd-level component of the MM vectors.

remaining 45 1st-level communities contain < 20 papers each, but still feature the dense within-community citations and specific jargon characteristic of specialized research.

At the 2nd level, the consensus hierarchy contains mostly singleton (one-member) sub-communities. This indicates that, past the 1st hierarchy level, our MCMC algorithm found little evidence to further group the papers. There are only 10 sub-communities with size > 1, all of which are found in super-community 1. The largest 8 are shown in Figure 8, and correspond to the diagonal block from index 1 through 39 in the adjacency matrix of Figure 9. These sub-communities contain either fewer or more within/between-community citations than is average for super-community 1, hence they are justified under a blockmodel assumption. For example, sub-community 1.3 contains 7 citations between 5 papers, while 4 of those papers contain the word "radion" (a hypothetical particle) in their titles. This suggests a tightly-knit community of researchers focused on a specific object.

Finally, the Multiscale Membership distribution (Figure 9) tells a similar story to the grass-feeding wasps network: most interactions occur at the 1st level; over 75% of MM vectors θ_i have less than 0.05 of their mass at the 2nd level. This observation, coupled with

the small number of meaningful 2nd-level sub-communities, suggests that the 1st hierarchy level suffices to explain most of the network. Because of this, the few non-singleton 2nd-level sub-communities are in fact significant, and merit further investigation.

8.3 Posterior Sample Analysis for Both Datasets

To complete our analysis, we need to inspect the quality of the posterior samples returned by our MSCB algorithm. It is well-known that adjacent samples in an MCMC sequence tend to be highly correlated, which inflates the variance of the samples by decreasing the effective sample size. Consequently, any estimator will require more samples to achieve the same level of precision, compared to an uncorrelated sequence of samples.

We quantify the degree of sample correlation in MSCB via the *autocorrelation function* $R_X(k)$. For a particular random variable X, this is defined as

$$R_X(k) = \frac{\sum_{t=1}^{n-k} \left(X_t - \bar{X}\right)\left(X_{t+k} - \bar{X}\right)}{\sum_{t=1}^{n-k} \left(X_t - \bar{X}\right)^2},$$

where n is the total number of samples, X_t is the t-th sample of X, and $\bar{X}$ is the sample mean of X. The question then, is which random variable to inspect. Recall that the MSCB algorithm samples the discrete random variables $\mathbf{c}$ (paths) and $\mathbf{z}$ (level indicators); however, the autocorrelation function is not well-defined for discrete domains. As a proxy, we shall instead consider $\ell = \log\left(\mathbb{P}(\mathbf{E}, \mathbf{c}, \mathbf{z})\right)$, the complete log-likelihood of a particular sample of $\mathbf{c}, \mathbf{z}$ (after integrating out $\mathbf{B}, \boldsymbol{\theta}$). We expect the autocorrelation $R_\ell(k)$ of ℓ to provide a good picture of sample correlation in $\mathbf{c}, \mathbf{z}$.

Figure 10 shows the log-likelihood autocorrelation $R_\ell(k)$ for the grass-feeding wasps and High Energy Physics networks, as computed on the 1,000 posterior samples. Using first-order autoregressive process theory, we can compute the posterior's *sample size inflation factor* or SSIF $s = (1+\rho)/(1-\rho)$, where $\rho = R_\ell(1)$ is the first-order autocorrelation. Intuitively, if we have n samples, then the effective sample size is given by n/s, because the autocorrelation ρ increases the standard error of the mean of ℓ by a factor of $\sqrt{s}$. For the grass-feeding wasps network, $\rho = 0.985$ with SSIF $s = 135$, while for the High Energy Physics network,

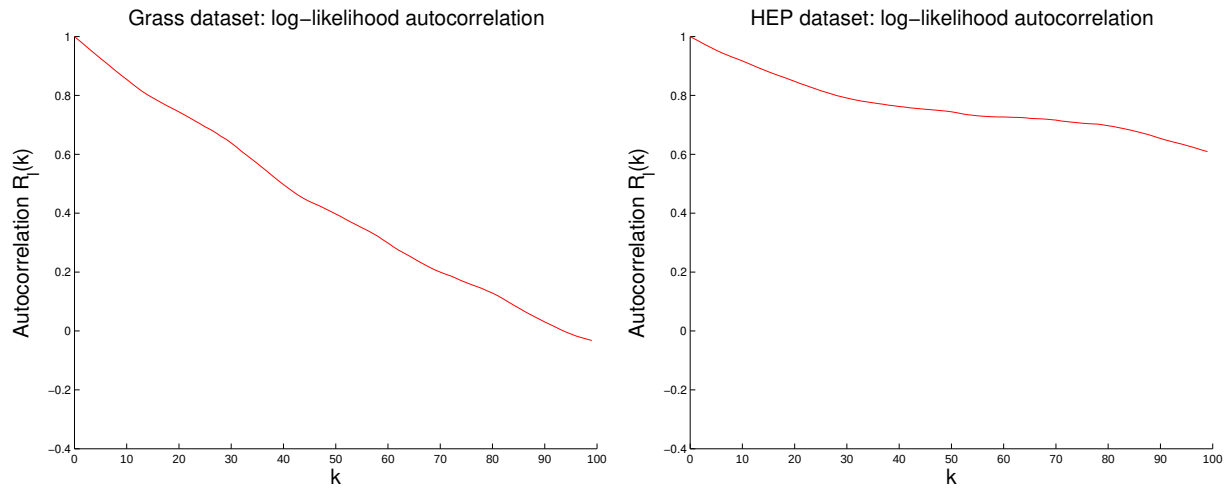

Figure 10: Autocorrelation function of the complete log-likelihood from the 1,000 posterior samples, for both the grass-feeding wasp parasitoids food web (**Left**) and the High-Energy Physics citation network (**Right**).

$\rho = 0.991$ with SSIF $s = 210$. This implies an effective posterior sample size of 7.4 and 4.8 for the grass-feeding wasps and High Energy Physics networks respectively.

Regarding the grass-feeding wasps network, we note that the MSCB algorithm took only 9 minutes to perform burn-in (10,000 samples) and to take 1,000 samples from the posterior, for a total of 11,000 samples. Hence, it is computationally feasible to increase the effective sample size by taking 10,000 or even 100,000 samples from the posterior, and keeping only every 10th or 100th sample to save memory. However, the same cannot be said of the High-Energy Physics network, on which the MSCB algorithm took nearly 2 weeks to obtain the same number of samples (11,000 including burn-in). This dramatic increase in runtime directly results from MSCB's $\mathcal{O}(N^3K)$ computational complexity, which makes it impractical to take many samples from larger networks.

An alternative to taking more samples is to supplement the MSCB Gibbs sampling algorithm with Metropolis-Hastings moves, so as to decrease the autocorrelation between adjacent samples. Our inspection reveals that the MSCB Gibbs sampler has difficulty when trying to change many actor paths $\mathbf{c}$ quickly. To illustrate this problem, suppose the sampler

is initialized with 100 actors in the same path, but the true posterior requires these actors to be split 50-50 in two different paths. The Gibbs sampler must first split off a single actor to form a new path, and then move the remaining 49 actors to the new path. However, because of the nCRP prior's "rich-get-richer" property, the latter events tend to have low Gibbs sampler probabilities, and it may take many samples before the true 50-50 split is established. One solution to this problem is a *split-merge* strategy, similar to that used by Jain and Neal (2004) for the vanilla Chinese Restaurant Process. Briefly, a split-merge strategy for the nCRP will involve Metropolis-Hastings proposal moves that (1) split actors in a single path into two paths, and (2) merge actors from two paths into a single path. By interleaving these moves with the regular Gibbs sampler, we can make large jumps in the state space of $\mathbf{c}$, and thus reduce autocorrelation. We expect this split-merge strategy to be a fairly simple extension of Jain and Neal (2004), though its implementation and derivation are out of the scope of this paper.

9. CONCLUSION

We have developed a nonparametric Multiscale Community Blockmodel (MSCB) that models networks in terms of hierarchical community memberships, which actors selectively undertake during interactions. To apply our model, we derived an MCMC algorithm that combines collapsed Gibbs sampling for latent variable posterior inference, and Metropolis-Hastings proposals for hyperparameter learning. Our algorithm automatically infers the structure of the hierarchy while simultaneously recovering the Multiscale Memberships of every actor, setting it apart from hierarchy-discovering methods that are restricted to binary hierarchies and/or single-community-memberships for actors. Moreover, because MSCB integrates aspects of stochastic blockmodels, it is expressive enough to account for both assortative (within-community) and disassortative (cross-community) interactions, as our simulation and real data experiments have demonstrated. These aspects of MSCB allow us to explore hierarchical network phenomena in a principled, statistical manner.

Acknowledgements This paper is based on work supported by NSF IIS-0713379, NSF DBI-0546594 (Career), ONR N000140910758, AFOSR FA9550010247, NIH 1R01GM093156, and an Alfred P. Sloan Research Fellowship to Eric P. Xing. Qirong Ho is supported by a graduate fellowship from the Agency for Science, Technology And Research, Singapore. We thank Dr. Le Song for his help and discussion during this project.

REFERENCES

Airoldi, E., Blei, D., Fienberg, S., and Xing, E. (2008), "Mixed membership stochastic blockmodels," *The Journal of Machine Learning Research*, 9, 1981–2014.

Aldous, D. (1985), "Exchangeability and related topics," *École d'Été de Probabilités de Saint-Flour XIII1983*, pp. 1–198.

Batagelj, V., and Ferligoj Patrick, A. (1992), "Direct and indirect methods for structural equivalence* 1," *Social Networks*, 14(1-2), 63–90.

Batagelj, V., and Mrvar, A. (1998), "Pajek-program for large network analysis," *Connections*, 21(2), 47–57.

Bernardo, J., Smith, A., and Berliner, M. (2000), *Bayesian theory* Wiley New York, New York, USA.

Blackwell, D., and MacQueen, J. (1973), "Ferguson distributions via Pólya urn schemes," *The annals of statistics*, pp. 353–355.

Blei, D., Griffiths, T., and Jordan, M. (2010), "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies," *Journal of the ACM (JACM)*, 57(2), 1–30.

Blei, D., and Jordan, M. (2004), Variational methods for the Dirichlet process,, in *Proceedings of the twenty-first international conference on Machine learning*, ACM, p. 12.

Blei, D., Ng, A., and Jordan, M. (2003), "Latent dirichlet allocation," *The Journal of Machine Learning Research*, 3, 993–1022.

Chung, F. (1997), Spectral Graph Theory,, in *Regional Conference Series in Mathematics American Mathematical Society*, Vol. 92, American Mathematical Society, pp. 1–212.

Clauset, A., Moore, C., and Newman, M. (2008), "Hierarchical structure and the prediction of missing links in networks," *Nature*, 453(7191), 98–101.

Clauset, A., Newman, M., and Moore, C. (2004), "Finding community structure in very large networks," *Physical Review E*, 70(6), 66111.

Dawah, H., Hawkins, B., and Claridge, M. (1995), "Structure of the parasitoid communities of grass-feeding chalcid wasps," *Journal of animal ecology*, 64(6), 708–720.

Escobar, M., and West, M. (1995), "Bayesian density estimation and inference using mixtures," *Journal of the american statistical association*, pp. 577–588.

Ferguson, T. (1973), "A Bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230.

Girvan, M., and Newman, M. (2002), "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, 99(12), 7821.

Guimera, R., and Amaral, L. (2005), "Functional cartography of complex metabolic networks," *Nature*, 433, 895–900.

Handcock, M., Raftery, A., and Tantrum, J. (2007), "Model-based clustering for social networks," *Journal of the Royal Statistical Society-Series A*, 170(2), 301–354.

Hoff, P., Raftery, A., and Handcock, M. (2002), "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, 97, 1090–1098.

Jain, S., and Neal, R. (2004), "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model," *Journal of Computational and Graphical Statistics*, 13(1), 158–182.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999), "An introduction to variational methods for graphical models," *Machine learning*, 37(2), 183–233.

KDD (2010), "KDD Cup 2003 - Datasets,", `http://www.cs.cornell.edu/projects/kddcup/datasets.html`.

Kemp, C., and Tenenbaum, J. (2008), "The discovery of structural form," *Proceedings of the National Academy of Sciences*, 105(31), 10687.

Kemp, C., Tenenbaum, J., Griffiths, T., Yamada, T., and Ueda, N. (2006), Learning systems of concepts with an infinite relational model,, in *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 381.

Krause, A., Frank, K., Mason, D., Ulanowicz, R., and Taylor, W. (2003), "Compartments revealed in food-web structure," *Nature*, 426(6964), 282–285.

Krebs, V. (2002), "Mapping networks of terrorist cells," *Connections*, 24(3), 43–52.

Liu, J. (1994), "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem.," *Journal of the American Statistical Association*, 89(427).

MacEachern, S., and Müller, P. (1998), "Estimating mixture of Dirichlet process models," *Journal of Computational and Graphical Statistics*, pp. 223–238.

Miller, K., Griffiths, T., and Jordan, M. (2009), "Nonparametric Latent Feature Models for Link Prediction," *Advances in Neural Information Processing Systems (NIPS)*, .

Mimno, D., and McCallum, A. (2007), Organizing the OCA: Learning faceted subjects from a library of digital books,, in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, ACM, pp. 376–385.

Nallapati, R., Ahmed, A., Xing, E., and Cohen, W. (2008), Joint latent topic models for text and citations,, in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 542–550.

Newman, D., Chemudugunta, C., and Smyth, P. (2006), Statistical entity-topic models,, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 680–686.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004), "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, 101(9), 2658.

Robert, C., and Casella, G. (2004), *Monte Carlo statistical methods* Springer Verlag.

Roy, D., Kemp, C., Mansinghka, V., and Tenenbaum, J. (2007), "Learning annotated hierarchies from relational data," *Advances in neural information processing systems*, 19, 1185.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006), "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, 101(476), 1566–1581.

Teh, Y., and Roy, D. (2009), "The Mondrian Process," *Advances in neural information processing systems*, .

Wainwright, M., and Jordan, M. (2008), "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, 1(1-2), 1–305.

Wang, C., and Blei, D. (2009), "Variational inference for the nested Chinese restaurant process," *Advances in Neural Information Processing Systems*, 22, 1990–1998.

Wang, Y., and Wong, G. (1987), "Stochastic blockmodels for directed graphs," *Journal of the American Statistical Association*, 82(397), 8–19.

Ward, J. (1963), "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, 58(301), 236–244.

Xing, E. P., Fu, W., and Song, L. (2010), "A State-Space Mixed Membership Blockmodel for Dynamic Network Tomography," *Annals of Applied Statistics*, 4(2), 535–566.