CataBEEM: Integrating Latent Interaction Categories in Node-wise Community Detection Models for Network Data

Yuhua Zhang [1] **Walter Dempsey** [1]

Abstract

Community detection is a fundamental task in network analysis. Learning underlying network structures has brought deep insights into the understanding of complex systems. While many methods have focused on clustering nodes into blocks, few accounts for the fact that interactions may exhibit edge-level clustering, which we call categories. Real network data often arise via a series of interactions. Interactions in complex systems can often be clustered into different categories and node-level community structures that depend on the category. In this paper, we introduce a category-and-block edge exchangeable model (CataBEEM) to study interaction networks with joint latent interaction-level category and node-level community structures. In particular, the proposed method models the network from the interaction process perspective and allows the incorporation of prior knowledge from auxiliary interaction-wise information. We derive an efficient variational inference algorithm that can be applied to networks consisting of millions of interactions and provide the theoretical bound of the misspecification rate. We demonstrate the effectiveness of our method in various simulation settings and apply the method to TalkLife data, a large-scale online peer-to-peer support network. We show CataBEEM detects more temporally consistent community structures and has better predictions than other methods.

1. Introduction

Network data is everywhere in our daily life. TalkLife, for example, is an online peer support network that focuses on mental health-related issues, where nodes represent users and edges represent interactions among users. Another example is gene co-expression networks where nodes represent genes and edges represent whether two genes are co-expressed. Understanding network structures has become an increasingly important research topic in sociology, biology, and disciplines where the data can be represented as graphs.

Real-world networks often arise from sequences of interactions. Conventional approaches proceed by analyzing constructed graphs via traditional statistical network analysis (Goldenberg et al., 2010; Newman, 2012) that takes a node-centric perspective where the node is the statistical unit. Network data arising from interaction processes, however, benefit from the frameworks where the interaction is the statistical unit. Edge-exchangeable models (Crane & Dempsey, 2018; Dempsey et al., 2021) are built specifically to analyze datasets coming from such complex interaction processes. Compared to conventional approaches, edge-exchangeable models reflect the empirical properties of sparsity and power-law degree distributions observed in real-world network data. While edge exchangeable frameworks are attractive, such models are inadequate to handle latent interaction-level and node-level cluster structures.

In this paper, we focus on a fundamental problem in network science — community detection. Communities are groups of nodes that connect more closely with each other than others in the graph. In social network data, for example, communities can refer to user groups with common interests (Bedi & Sharma, 2016). In the protein-protein interaction networks, communities are likely to be functional modules within the cell (Rives & Galitski, 2003; Chen & Yuan, 2006). The basic task of community detection is to partition nodes into blocks that are more densely connected. Numerous methods have been proposed to address this problem, including stochastic block models (SBM) (Holland et al., 1983) and its extensions (Airoldi et al., 2008; Karrer & Newman, 2011), modularity-based algorithms such as the Newman-Girvan modularity (Newman, 2016), as well as other approaches (Su et al., 2022). These methods are based on a node-centric perspective. Recent work by (Zhang & Dempsey, 2022) addresses node-level community detection within the edge exchangeability framework.

[1]Department of Biostatistics, University of Michigan, Ann Arbor, United States. Correspondence to: Yuhua Zhang <zyuhua@umich.edu>, Walter Dempsey <wdem@umich.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

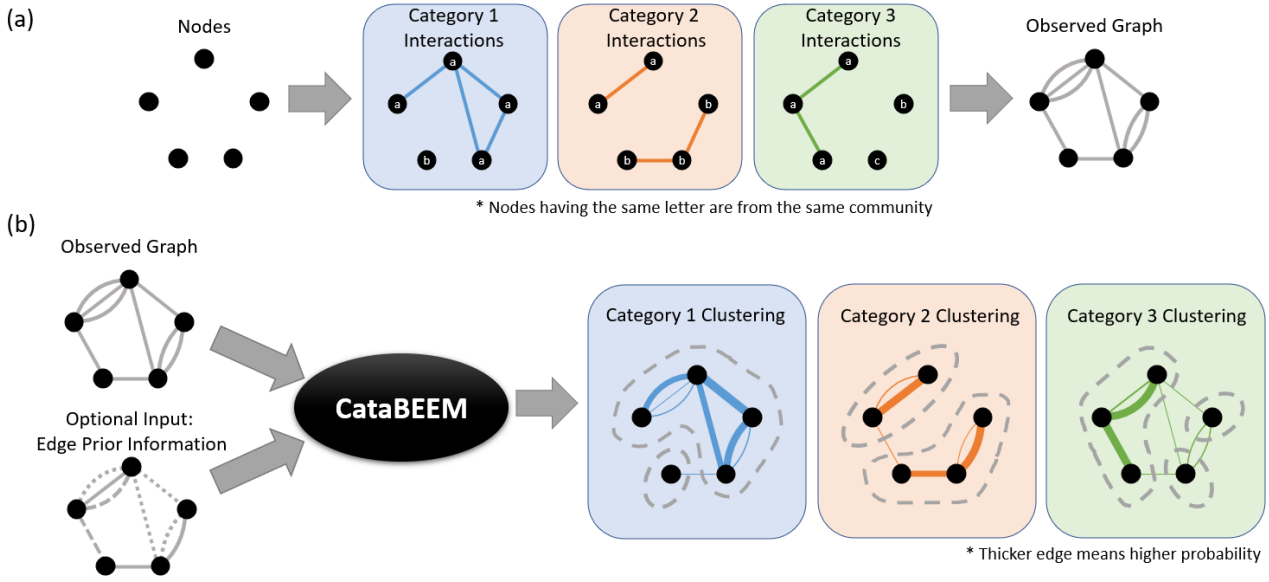

Figure 1. (a) An example generative model with five nodes and three categories. The community structures (labeled by alphabet) are different in different categories (labeled by color). The edge categories and node blocks are latent in the observed graph. (b) The illustration of the algorithm based on the CataBEEM. The inputs are the observed graph and the optional edge prior information. The algorithm infers the underlying interaction categories and the community structures in each category.

While much work has focused on the node-level block structure, few methods focus on interaction-level clustering (Sewell, 2020). On the other hand, community behaviour of nodes can differ by different interaction clusters. In email exchange network, for example, the potential receivers of an email are different given the theme is formal and casual. In this paper we integrate interaction-level cluster structure into the modeling of node-level clustering. To our knowledge, there is no existing approach that jointly models the interaction and node clusters. By analogy, nodes cluster into communities, while interactions cluster into what we term *categories*. Similar to communities, the category of an interaction is latent. A series of interactions has a corresponding series of latent categories that determine the underlying network structure. In gene co-expression networks, for example, categories can be tissue-specific expressions. In citation networks, categories can correspond to research topics. Moreover, node-wise clustering behavior may depend on the category. Figure 1a shows an illustration of our proposed generative model.

A natural question is how learning interaction-level clusters can help researchers go above and beyond node-level community structure. To answer this, consider a hypothetical TalkLife user who is seeking peer support on the mental health issue of depression from other users on the TalkLife platform. In this instance, even if another user is very connected (i.e., lots of prior interactions), they may not discuss this particular issue and therefore it may be more helpful for the platform to help find support from the group of users who have posted on the issue of depression in the past. In this example, the topic of discussion is the latent interaction-level category. With this motivation in mind, we propose a statistical model within the edge-exchangeability framework that allows for joint latent interaction-level category and node-level community structure. We call it category-and-block edge exchangeable model (CataBEEM)[1]. Figure 1b provides a high-level overview of the model.

The main contributions of this paper are as follows:

1. We propose a statistical model that integrates interaction-level clustering (categories), with node-level clustering (communities). Our model provides novel insights into the network structure that can not be captured by prior approaches.

2. Our proposed model can incorporate auxiliary interaction-level information through an interaction-wise prior that provides extra information for learning latent network structure. In TalkLife, for example, post content is used to construct a subjective prior over the latent categories. This prior information is shown to lead to empirical improvements in performance.

3. We derive a scalable variational EM-based algorithm that can be applied to networks consisting of millions of interactions. To our knowledge, variational EM has never been investigated in the community detection scenario where

[1]The code of the corresponding algorithm can be found at: https://github.com/YuhuaZhang1995/CataBEEM

latent categories are present.

4. We provide a theoretical bound on the mis-specification rate of an approximate inference algorithm based on the model, along with support from empirical simulation results.

2. Related Work

Community Detection in Network Data The fundamental task of community detection is to identify the latent labels of each node in the network. We refer to (Fortunato, 2010; Abbe, 2017) for comprehensive reviews on this topic. The two popular approaches are model-based stochastic block models (SBM) (Holland et al., 1983; Rohe et al., 2011; Abbe et al., 2015), and modularity-based methods (Barber, 2007; Newman, 2016). We focus on SBMs, since it is more relevant to our work.

The simplest version of the SBM assumes nodes within the same block have the same probability of forming an edge with other nodes, and within-block interactions are more likely than between-block interactions. Many associated methods have been proposed, such as the degree-corrected SBM (DC-SBM) (Karrer & Newman, 2011; Ball et al., 2011), which allows for degree heterogeneity; the mixed membership model (MMSBM) (Airoldi et al., 2008) which associates each unit of observation with multiple blocks rather than a single one; and other extensions (Peixoto, 2014; Galhotra et al., 2018). Theoretical guarantees of community detection have been well established (Rives & Galitski, 2003; Zhao et al., 2012; Gao et al., 2018).

Recently, Graph Neural Network-based (GNN) community detection methods have been proposed (Scarselli et al., 2008; Kipf & Welling, 2016a;b), which learn network representations regarding each node as a discrete symbol. Many GNN methods are not constructed from an underlying generative model, which makes this approach quite different from model-based approaches such as SBMs. Methods have also been proposed to bridge the gap between SBMs and GNN-based methods (Mehta et al., 2019).

Interaction-based Network Modeling. SBMs are based on the assumption that the statistical units in the network are the nodes. On the other hand, edge exchangeable models have been proposed where the statistical units are the interactions (Crane & Dempsey, 2018). The key modeling assumption is that the observed interaction network is a finite subsequence from an infinite sequence of interactions. Denote the underlying population of nodes by $\mathcal{P}$. Then the network is constructed from an interaction process $\mathcal{I} : \mathbb{N} \mapsto \text{fin}(\mathcal{P})$ which is a correspondence from natural numbers and the finite multisets of $\mathcal{P}$. The most relevant prior work in this direction is the incorporation of node-level community structure into edge exchangeable models (Zhang & Dempsey, 2022). These block edge-exchangeable models (BEEMs) do not account for latent interaction-level clustering structure; moreover, the Gibbs-based inferential algorithm provided is not scalable while the proposed variational algorithm can scale to millions of interactions.

Variational Inference The inference of model parameters relies on the variational method (Blei et al., 2017), which has been used in community detection (Airoldi et al., 2008; Yin et al., 2020). Theoretical and computational guarantees of mean-field variational inference have been thoroughly discussed (Zhang & Zhou, 2020).

3. Algorithm

3.1. Notations and Definitions

Categorized Interaction Process Let $\mathcal{P}$ represent the set of nodes in the underlying population, and T be the total number of categories. For clarity, here we consider 2-way interactions (e.g., a sender and a single receiver). Define a *complete* interaction process E^c: $\mathbb{N} \mapsto [T] \times \mathcal{P} \times \mathcal{P}$. Consider the mth interaction, $E^c(m) = (t, i, j)$ indicates the mth interaction is from category t, and involves node i and j. Each edge has an assigned category according to a distribution parameterized by $\tilde{\pi}$. Denote the category assignment of interaction m by $\tilde{z}_m$. Note that in the observed set of interactions, the category labels are latent. Therefore, we define the *observed* interaction process E: $\mathbb{N} \mapsto \mathcal{P} \times \mathcal{P}$.

Categorized Interaction Process with Block Structure Let the number of blocks in each category be $K_t, t \in [T]$. Note here K_t are different for different ts. The block structure is defined as the mapping: $\mathcal{P} \mapsto [K_t]$, which is category specific. Denote the block assignment of each node (sender or receiver) in category t as z_x^t for $x \in \mathcal{P}$. In the observed interaction process, $\{z_x^t\}_{t\in[T]}$ are latent. Conditional on the category $\tilde{z}_m = t$, the block that initiates the interaction is determined by a distribution $\pi(t)$ over all blocks. The block of the second unit in the interaction is determined by a propensity matrix B_t, with entries in each row summing to 1. This B_t matrix is category specific. Denote the entry of the matrix as $B_t(b, b')$, for $t \in [T], b, b' \in [K_t]$.

Edge Exchangeability The edge-labeled network is constructed from the interaction process by constructing equivalence classes over node labels. Let S be a finite or countable set. An edge-labeled network $\mathcal{Y}$ built from an interaction process among elements in S – formally $I : \mathbb{N} \to \text{fin}(S)$ where $\text{fin}(S)$ are finite multisets of S – is edge exchangeable if $\mathcal{Y}^\sigma =_D \mathcal{Y}$ for all finite permutations $\sigma : \mathbb{N} \to \mathbb{N}$, where $=_D$ means equal in distribution. The definition states that the probability of the network is invariant to the order in which the interactions are observed. Edge exchangeability provides a theoretical foundation for the proposed model to guarantee empirical properties of sparsity and power-law degree distribution in the generative models.

Node Popularity To measure the frequency of a node being observed, denote $\{f_{tb}^i : t \in [T], b \in [K_t]\}$ as the propensity of observing node $i \in \mathcal{P}$ in category t within block b in the network. In our inferential algorithm, we assume propensities form a conditional distribution given category t and block b sum up to 1, i.e.,$\sum_{i\in\mathcal{P}} f_{tb}^i = 1$.

Notations in Observed Data Denote the observed interaction network as Y_M, where we observed a finite number of edges. Denote the total number of edges as M. Let $\mathcal{P}_M$ represent the set of nodes in the observed graph. The total number of observed nodes is $N = |\mathcal{P}_M|$. Consider the mth interaction in the observed graph. Denote $s(m)$ as the sender node, $r(m)$ as the receiver node.

3.2. CataBEEM

Model Description The goal of this work is to infer the underlying network structure. We start by providing network formulation from the generative model perspective. For illustrative purposes, we start with an arbitrary interaction and then generalize it to the entire network.

Consider the mth interaction $E(m) = (i, j)$ in the network. For this specific interaction, the edge category $\tilde{z}_m$, selected from a finite number of categories T, is characterized by:

$$\tilde{z}_m \sim \text{Multinomial}(\tilde{\pi})$$

The block assignment for node i and node j is given by:

$$z_i^t \sim \text{Multinomial}(\pi(t)); z_j^t \sim \text{Multinomial}(B_t(b, \cdot))$$

With the above information, the probability of observing node i and node j is given by:

$$s(m) = i|(z_i^t = b, \tilde{z}_m = t) \propto \text{Multinomial}(f_{tb})$$

$$r(m) = j|(z_j^t = b', \tilde{z}_m = t) \propto \text{Multinomial}(f_{tb'})$$

Combining the above, the probability observing interaction $E(m)$ is:

$$P(E(m) = (i, j)|z, \tilde{z}, \tilde{\pi}, \pi, f, B) =$$

$$\prod_{t=1}^{T} \tilde{\pi}_t^{\tilde{z}_m} \prod_{b=1}^{K_t} \pi_b(t)^{z_i^t} f_{tb}^i \prod_{b'=1}^{K_t} B_t(b, b')^{z_j^t} f_{tb'}^j$$

Denote $\Theta = \{z, \tilde{z}, \tilde{\pi}, \pi, f, B\}$. Consider all interactions in the observed network, the likelihood of the complete model is the product over all interactions:

$$P(Y_M|\Theta) = \prod_{t=1}^{T} \tilde{\pi}_t^{\tilde{L}_t} \prod_{b=1}^{K_t} \pi_b(t)^{L_b(t)} \prod_{s} f_{tb}^{s\ Deg_t(s)} \prod_{b'=1}^{K_t} B_t(b, b')^{W_t(b,b')} \prod_{r\in R(s)} f_{tb'}^{r\ Deg_t(r)} \tag{1}$$

where $\tilde{L}_t$ is the number of interactions that have category t; $L_b(t)$ is the number of interactions that initiate from block b and have category t; $W_t(b, b')$ is the number of interactions that are initiated from block b to b' and have category t; $Deg_t(i)$ represents the degree of $i \in \mathcal{P}_M$.

Hyperparameters For latent parameters $\tilde{\pi}$, $\pi(t)$, and f_{tb}, we specify distributions as follows:

$$\tilde{\pi} \sim \text{Dir}(\tilde{\alpha});\ \pi(t) \sim \text{Dir}(\alpha_t);\ f_{tb} \sim \text{Dir}(\gamma_{tb})$$

where $\tilde{\alpha}$ is of dimension T, and α_t is of dimension K_t. In the observed graph, a finite number of nodes are observed, such that $\sum_{n=1}^{N} f_{tb}^n = 1$. Thus γ_{tb} is of dimension N. If no prior information is provided, each entry in $\tilde{\alpha}$ can take value of $\frac{1}{T}$; each entry in α_t can take value of $\frac{1}{K_t}$; and each entry in γ can take the value of $\frac{1}{N}$.

Auxiliary interaction-level covariates can provide extra information about the underlying category structure, but edge exchangeable models to date have not made use of this additional data. A joint model that includes the auxiliary information can be computationally prohibitive. As a scalable alternative, we incorporate these covariates through an edge-wise prior α^*. For each interaction $m \in [M]$, α_m^* is a prior distribution over T informed by the auxiliary interaction-level covariates. We set the $\tilde{\alpha}$ to be the tuning parameter, such that:

$$\tilde{\pi}_m \sim \text{Dir}(\frac{\tilde{\alpha}}{1+\tilde{\alpha}}\alpha_m^*),\ \forall m \in [M] \tag{2}$$

The interpretation of $\tilde{\alpha}$ is thus the confidence on the input prior, with $\tilde{\alpha}$ being infinity means a complete trust of prior information. Low trust in the prior information will shrink it towards 0.

3.3. Inference

The above model brings latent parameters $\tilde{z}$, z, $\tilde{\pi}$, π and f, as well as model parameter B. Unfortunately, direct maximization of Eq. (1) does not have a closed form and Bayesian MCMC can be computationally prohibitive. Here, we derive a variational EM algorithm as a scalable alternative to estimate the model and latent parameters. The correspondence between variational parameters and latent parameters are shown in Table 1. In the observed network with M interactions and N unique nodes, we specify the following mean-field approximation:

$$q(\tilde{\Theta}) = \prod_{i=1}^{M} q_1(\tilde{z}_i|\psi) q_2(\tilde{\pi}|\tilde{\eta}) q_4(\pi(t)|\eta^t) \prod_{n\in\{s(i),r(i)\}} q_3(z_n^t|\phi^t) q_5(f_{tb}^n|\lambda_{tb}) \tag{3}$$

Given the above approximation, the inference algorithm can be derived. Consider the evidence lower bound (ELBO) of

Table 1. Correspondence between model parameters and designated distributions, variational parameters, and hyperparameters.

LATENT VARIABLE	VARIATIONAL DISTRIBUTION	VARIATIONAL PARAMETER	HYPER-PARAMETER
$\tilde{z}_i$	MULTINOMIAL	ψ_i	–
$z^t_{s(i)}$	MULTINOMIAL	$\phi^t_{s(i)}$	–
$z^t_{r(i)}$	MULTINOMIAL	$\phi^t_{r(i)}$	–
$\pi(t)$	DIRICHLET	η^t	α_t
$\tilde{\pi}$	DIRICHLET	$\tilde{\eta}$	$\tilde{\alpha}$
f_{tb}	DIRICHLET	λ_{tb}	γ_{tb}

the observed network:

$$L(q, \Theta) = \mathbf{E}_q[\log p(Y_M, \Theta)] - \mathbf{E}_q[\log q(\tilde{\Theta})] \quad (4)$$

The ELBO contains two terms. The first term is the expected log joint distribution of the model. The second term is the entropy of the variational distribution. The variational EM algorithm can be implemented by iteratively updating the variational parameters for latent variables and the model parameters, which leads to the maximization of the ELBO, as shown in Algorithm 1. The update formula of variational parameters can be found in Appendix A.2. Given latent parameters, the entries in B matrix can be updated as:

$$B_t(b, b') = \frac{\sum_{i=1}^{M} \psi^t_{s(i)} \psi^t_{r(i)} \phi^{bt}_{s(i)} \phi^{b't}_{r(i)}}{\sum_{i=1}^{M} \sum_{k=1}^{K_t} \psi^t_{s(i)} \psi^t_{r(i)} \phi^{bt}_{s(i)} \phi^{kt}_{r(i)}}$$

The theoretical computational complexity of the algorithm is $O(MTK^2)$. Note here the update of ψ_i are independent for each $i \in [M]$, and the update of ϕ_n are independent for each $n \in [N]$. The optimization of the algorithm through parallel computing is feasible here, which can greatly improve the scalability of the algorithm. The time complexity of the algorithm is discussed in Appendix A.9.

3.4. Consistency

In this section, we provide a theoretical guarantee that the latent block labels can be recovered by maximizing the likelihood shown in Equation 1. Conditional on the latent category of each interaction being known, the misspecification rate of latent block assignments in each category can be bounded under certain conditions. Specifically, the misspecification rate converges to 0 when we focus on high-degree nodes.

Theorem 3.1. *Assume the category of each interaction is known. Let Y_M obey the power-law degree assumption. Let $e_t : \mathcal{P}_M \mapsto [K_t]$ denote the current labeling of the non-isolated nodes after observing M_t interactions in category t. Assume $K_t = 2, \forall t \in [T]$. Let*

$$M_{N_t} = \inf_{\rho' : [K_t] \mapsto [K_t]} \frac{1}{N_t} \sum_{i=1}^{N_t} 1(\hat{z}^t_i \neq \rho' z^t_i)$$

Algorithm 1 Variational EM Algorithm

Input: observed graph Y_M with N unique nodes
Specify: T, K
Initialization: Variational parameters: ψ, ϕ, η, $\tilde{\eta}$, λ; Model parameter: B; Hyperparameters: α, $\tilde{\alpha}$, γ.
repeat
  for $i = 1$ **to** M **do**
    for $t = 1$ to T **do**
      Update ψ_i: $\hat{\psi}^t_i \propto f_1(\phi, \tilde{\eta}, \eta, \lambda, B)$
    end for
  end for
  for $n = 1$**to** N **do**
    for $t = 1$ to T **do**
      for $b = 1$ to K_t **do**
        Update ϕ^{bt}_n: $\hat{\phi}^{bt}_n \propto f_2(\psi, \phi_r, \eta, \lambda, B)$, if n is sender; $\hat{\phi}^{bt}_n \propto f_3(\psi, \phi_s, \lambda, B)$, if n is receiver.
      end for
    end for
  end for
  Update $\tilde{\eta}$: $\hat{\tilde{\eta}} \propto f_4(\psi, \tilde{\alpha})$
  Update η: $\hat{\eta} \propto f_5(\psi, \phi, \alpha)$
  Update λ: $\hat{\lambda} \propto f_6(\psi, \phi, \gamma)$
  Update B matrix
until Converge

(See Appendix A.2 for the full updates formula on f_1 to f_6)

denote the misclassification rate (under potential label switching). As $M_t \to \infty$, for any $u \geq \frac{1}{e}$, under certain conditions on the current labeling provided in Appendix A.1 and the approximate updating rule we have

$$\mathbb{P}[M_{N_t} \geq euP_t] \leq \exp\left[-eN_tP_tu \log u\right]$$

where e is Euler's number, $u > 1/e$ is a constant, and $P_t \to \sum_{d=1}^{\infty} \alpha B(d, \alpha + 1) \exp(-d\mu^2_{\min}/4)$ as $M_t \to \infty$ where $\mu_{\min}$ is a constant depending on the current labeling. Specifically, restricting the misclassification rate $M_{N_t}(D_M)$ to nodes of at least degree D_M, it is possible to construct a sequence D_M, such that

$$\lim_{M_t \to \infty} \mathbb{P}[M_{N_t}(D_M) \geq \epsilon] = 0, \quad \textit{for all } \epsilon > 0.$$

Remark 3.2. The proof of Theorem 3.1 relies on an approximating update rule, which is not equal to the variational inference update rule. While not equivalent, the two rules have similar updates in our empirical studies.

3.5. Model Selection

So far, we have assumed the number of categories and the number of blocks are known. In most real cases, these numbers are not given a priori and need to be decided. Since the emphasis of the paper is not on proposing a novel model selection model, we rely on the existing methods and suggest potential strategies for model selection in our framework.

First, consider the selection of the number of blocks. We focus on two strategies, the model selection based on the maximal marginal posterior (Taddy, 2012), and the BIC criteria (Burnham & Anderson, 2004; Yan, 2016). After fitting a marginal model without considering latent categories, both of these model selection criteria give an accurate estimate of the number of blocks (Appendix A.3). In all scenarios (including where the number of blocks is not the same in different categories), we find the community structures in each category can be recovered by our algorithm after setting a global value on the number of blocks.

The selection of the number of categories is case-dependent. Use TalkLife data as an example. The auxiliary interaction-wise prior knowledge is from the textual information in posts and comments. In this case, we fit the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) to the textual data and get the topic distribution in each post-comment pair. We assume the underlying topics in the LDA model indicates the underlying categories. We select T based on Jaccard Index (between-topic difference) and Coherence (within-topic similarity) (Rehurek & Sojka, 2011; Röder et al., 2015) (See Appendix A.6 for details).

4. Experiments on Synthetic Data

In this section, we demonstrate the effectiveness of the method through simulations. Theorem 3.1 provides the boundary on the misspecification rate of the algorithm. Empirical results to support the conclusion are shown. We start with the generative description of the simulation setup and proceed to evaluating the performance of the algorithm in different scenarios.

4.1. Simulation Set-up

The network data are generated by interaction to mimic the real-world networks. For illustration purposes, suppose m interactions $E_{[m]}$ have been observed along with the block and category assignments. Consider a new interaction E_{m+1}. Suppose $\tilde{\pi}$, $\pi(t), \forall t \in [T]$, and the propensity matrix B are known a priori, such that the category of E_{m+1} and the block assignments of the sender and receiver are sampled according to $\tilde{\pi}_t$, $\pi_b(t)$, and $B_t(b, b')$.

Given the category of the interaction and the block assignments of the nodes, the next step is to select the corresponding sender and receiver nodes. However, a pre-specified f_{tb} is unavailable due to the fact that the network is generated by interaction, and the total number of nodes is a random variable. We use the strategy described in the interaction-framed network (Zhang & Dempsey, 2022) where the node distribution follows a Pitman-Yor process (Pitman et al., 2002). In this way, the sender node $s(m+1)$ can then be drawn according to

$$P(s(m+1) = s|\tilde{z}_{m+1} = t, z^t_{s(m+1)} = b) \sim PY_{tb}(\beta, \theta)$$

where PY_{tb} is the topic-block specific Pitman-Yor Process. The intuition is the frequency of observing a node depends on the observed degree as well as the network properties (characterized by β and θ), We follow a similar procedure to draw the receiver node $r(m+1)$.

The entire network can be generated by repeating the procedures described above. We assume here nodes from different blocks are non-overlapping in the simulation. But the same node can appear in different categories.

4.2. Consistency

We evaluate the algorithm's ability to recover the latent labels in different simulation settings (See Appendix A.4 for details). Experiments are repeated 20 times in each setting. The L2 norm is used to evaluate the performance of the inferential algorithm, which measures the distance between the inferred labels and the truth. It is defined as:

$$L_{\tilde{z}} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\tilde{z}_i - \hat{\tilde{z}}_i)^2};\ L_z = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(z_n^{t^*} - \hat{z}_n^{t^*})^2}$$

where t^* is the category node n contributes most (See Appendix A.5 for discussion of label switching). Figure 2a shows the L_2 norm of block assignments as a function of the node degree cutoff. The degree cutoff means nodes used to calculate the L2 norm have degrees greater or equal to the thresholding value. In all settings, the uncertainty of the inferred latent labels is high with the presence of low-degree nodes. As the degree cutoff increases, the inferred labels gradually converge to the truth, which is consistent with our theoretical conclusion.

Consider the fact that the performance of the algorithm can be affected by the block structure of the network and the similarity of block structures in different categories. In Settings 1 (Easy), 2 (Moderate), and 3 (Hard), the probability of within-block connectivity decreases, and the similarity of the block structure in different categories increases. The convergence rate slows as the difficulty of the learning task goes up. Besides, we also explore the performance of the algorithm in the setting (Setting 5) with an increasing number of blocks and the number of categories, and the setting (Setting 4) where the numbers of blocks are different in different categories.

Let $\delta \in [0, 0.5]$ be the perturbation on the true block label and interaction label, with $\delta = 0$ is the perfect initialization, and $\delta = 0.5$ being a completely random guess. We experiment on the effect of initialization on the recovery of block labels and category labels. Figure 2b shows inferred

block labels can be affected by the initialization. The further the initial value deviates from the underlying truth, the harder it is for the algorithm to identify the correct labels, especially for low-degree nodes. Another interesting phenomenon being observed is that a good initialization can be more helpful when the block structures are more distinct in different categories.

Figure 2c shows the recovery of category assignment in different settings with different initialization strategies. The conclusions are very similar to what's been observed in the recovery of the block labels. Both the initialization and the difficulty of the learning task can have an effect on the performance of the algorithm.

4.3. Auxiliary Interaction-wise Information

We have shown the consistency of the inferred block labels, especially for high-degree nodes. On the other hand, discrepancies exist between the inferred category labels and the ground truth. We show in the following experiments that edge-wise prior can provide extra information for the algorithm to learn the correct label for each interaction.

Recall the hyperparameter design as shown in Eq. 2. We experiment with different weights on the prior information. The $\tilde{\alpha}$ is set to infinity, 1, and 0, which correspond to a complete trust in the prior, down-weighting the prior by half, and no trust in the prior information. Figure 2d shows the L2 norm of the category labels in Setting 3 (Hard). Incorporating prior information for each interaction significantly improves the correctness of the inferred categories labels, especially in the situation where the initialization is poor. In Figure 2e, we show the improvement in correct category assignments also helps improve the accuracy of learned block assignments. The improvement also depends on the initial values. Again the extra prior information benefits most in scenarios with poor initialization.

5. Experiments on Real World Data

5.1. Overview of TalkLife Data and Experiment Design

TalkLife is an online peer-to-peer support network, where users can post contents about their mental health concerns, and/or comment on other users posts. We consider all posts on TalkLife during the Year 2019, which leads to a network of 4,236,829 interactions, and 199,257 users (Poster and Commentator). We apply the proposed method to the TalkLife data. The interaction network can be constructed by linking users who post and users who comment on the same post. Each post consists of a poster and a set of commentators. To make model comparisons with other methods fair, we split interactions between the same poster and each of the commentators into a sequence of poster-commentator pairs.

In this case, there are no ground truth labels for categories or blocks in TalkLife. Therefore, we focus on stability of the detected community structures over time as well as predictive capacity of the proposed model. The more persistent detected community structures are over meaningful time scales, the more helpful it is for the platform to identify the user groups. With this in mind, we partition the data in a consecutive time frame into two halves of comparable sizes, e.g. similar time intervals. We fit the model in the first half of the data (training) and evaluate the performance in the second half of the data (testing). We experiment on data partition of different time frame lengths, e.g., six months of data or one month of data in each part. Note that the presence of new users and the drop-offs of old users will lead to some unpredictable user behaviors. We exclude these users to avoid this effect on model evaluation. Partitioning the data into comparable sizes will give us the largest proportion of user overlap.

5.2. Fit to Data

We select the number of underlying blocks based on the BIC criteria as described in Section 3.5. To identify the number of underlying categories, we apply the LDA model to the annual textual data and identify the number of latent topics (See Appendix A.6). We assume latent topics reflect latent categories in our model, and the number of latent categories is the same as the number of latent topics in LDA model. The probability that each post-comment pair belongs to a topic is used as the edge prior.

We apply the method to the monthly data and the half-year data. The visualization of the detected community structures are shown in Appendix A.7. While the inferred communities reveal the social structures of users, we consider the temporal stability of the detected communities. Note that the blocks in different categories and in different time frames do not have a one-to-one correspondence, and thus are not directly comparable. To overcome this issue, we consider the probability that an arbitrary pair of nodes s and j are in the same block $P_{z_s=z_r}$:

$$P_{z_s=z_r} = \sum_{t=1}^{T} P^t \sum_{b=1}^{K_t} (\hat{z}_s^t \odot \hat{z}_r^t)_b$$

where $P^t \propto P_s^t P_r^t$ is the normalized posterior estimates that node s and r are in an interaction of category t; $\hat{z}_s^t$ and $\hat{z}_r^t$ are the posterior estimates of the block assignments.. To quantify the difference of inferred block structures at different times, we use the L_2 norm:

$$L_2 = \sqrt{\frac{1}{N^2} \sum_{s,r \in \mathcal{P}} |P_{z_s=z_r}^{t_1} - P_{z_s=z_r}^{t_2}|_2}$$

where $P_{z_s=z_r}^{t_1}$ and $P_{z_s=z_r}^{t_2}$ correspond to the estimates of probability in the first and the second part of data (e.g., if t_1

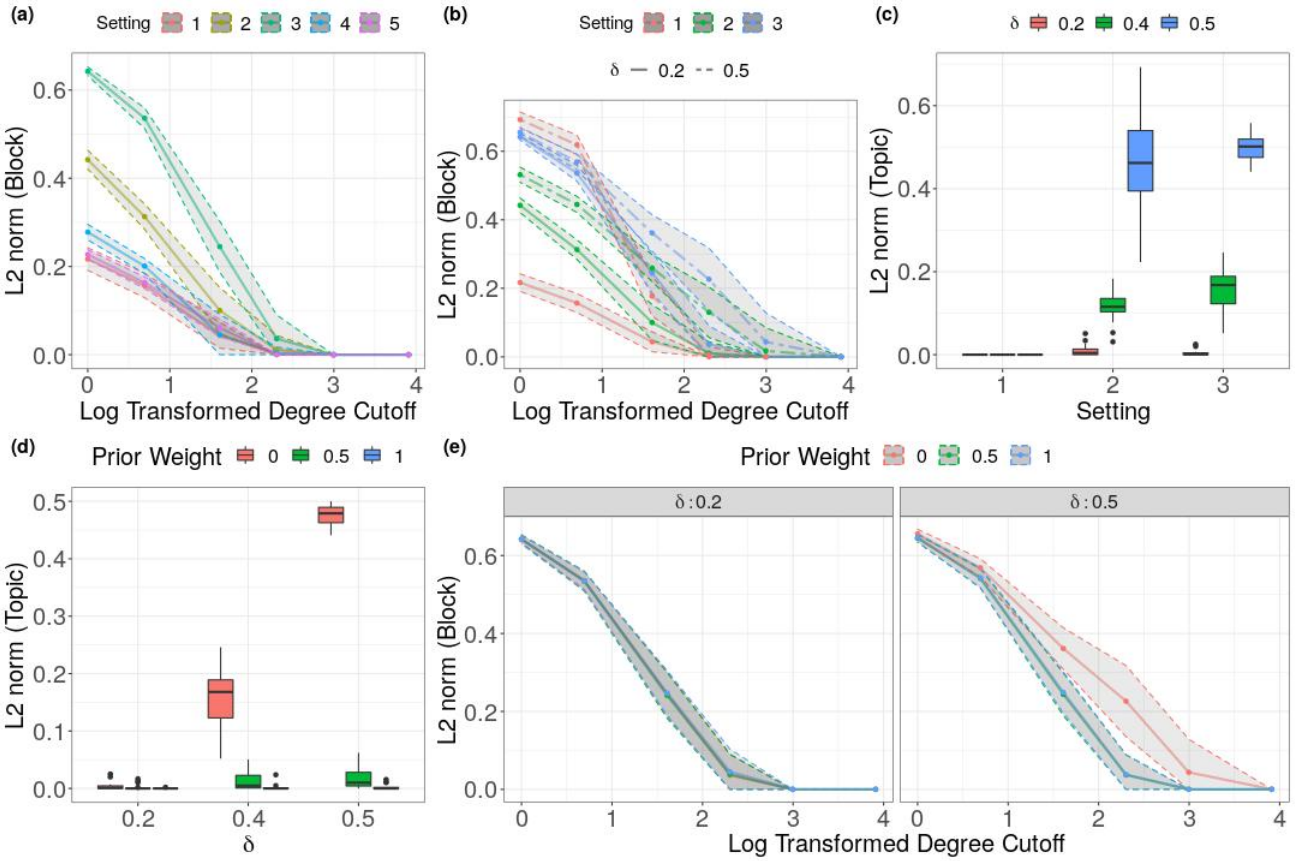

Figure 2. Average L2 norm of block assignments in different settings (a) as a function of node degree, and (b) as a function of node degree with different initialization status. (c) L2 norm of category allocations in different settings varied by different initialization status. (d) Comparison of L2 norm of category allocation with different prior weights given different initialization status. (e) Average L2 norm of block assignments as a function of node degrees with different prior weights.

Table 2. Comparison of the temporal stability of block structures. The cataBEEM has lower L_2 in all data sets, indicating a more persistent block structure over time.

DATA SET	L_2 (CATABEEM)	L_2 (DC-SBM)
SIX MONTHS	0.48	0.70
MONTHLY (SD)	0.38 (0.059)	0.59 (0.075)

is January, then t_2 is February). The rationale is that if the block structures are persistent over time, the probability any arbitrary pair of nodes belongs to the same block should not change, which is reflected by the relatively small value of the L_2 norm. Results are shown in Table 2. There are limited choices of well-implemented community detection methods that can scale up to the size of the data. We compare the results with DC-SBM. In all datasets, our method detects more a stable community structure than the DC-SBM.

5.3. Prediction of Interactions

In this section, we demonstrate the potential translation of the model fitting results into a predictive tool. Consider the probability of observing a certain receiver given the sender. For a specific sender s, we have $P_{r|s}$:

$$P_{r|s} \propto \sum_{t=1}^{T} P^t(\hat{z}_s^t \odot \hat{f}_s^t)\hat{B}_t(\hat{z}_r^t \odot \hat{f}_r^t)'$$

such that $\sum_r P_{r|s} = 1$. We calculate the $P_{r|s}$ in the training data and evaluate the results based on the observed degree in the testing data. It is expected that the higher the $P_{r|s}$ (in training data), the more likely the receiver r is to be observed, and the higher the observed degree in testing data. We focus on the coverage probability of high degree receivers (i.e. $Deg(r|s) \geq d$) of sender s in the testing data: $CP_d^s = \sum_{r,Deg(r|s)\geq d} P_{r|s}$. Define the coverage

Table 3. Coverage Ratio Probability (CRP) calculated by different methods. * indicates the highest value among all methods.

DATA SET	CPR_{10}^{100} (CATABEEM)	CPR_{10}^{100} (DCSBM)	CPR_{10}^{100} (E2)	CPR_{10}^{100} (DGLFRM)
SIX MONTHS	2.11*	2.09	2.06	–[1]
JAN	2.84*	1.61	1.53	1.092
FEB	2.61*	2.27	2.22	1.079
MAR	2.22*	1.98	1.89	1.123
APR	1.81	1.94*	1.91	1.098
MAY	2.77*	2.69	2.68	1.059
JUN	2.56*	1.02	2.54	1.081
JUL	2.27*	2.22	2.23	1.094
AUG	2.64*	2.33	2.28	1.071
SEP	1.61	1.75*	1.72	1.048
OCT	2.72*	2.48	2.42	1.090
NOV	2.78*	2.19	2.11	1.088

[1] NOT ENOUGH MEMORY WHEN APPLYING DGLFRM TO THE SIX-MONTH DATA.

probability ratio of sender s as:

$$CPR_d^s = \frac{CP_d^s}{(\sum_{r,Deg(r)\geq d} P_{ref})}$$

where $P_{ref} = 1/N_{r|s}$. A higher coverage ratio indicates the better chance the interaction between sender s and high-degree receiver nodes of s being predicted. We calculate the average CPR of 100 highest degree senders in the testing set, and set $d = 10$ for receivers, denoted as CPR_{10}^{100} (See Appendix A.8 for comparisons in different settings). Results are shown in Table 2. All CPR_{10}^{100} values are greater than 1, which indicates the effectiveness of cataBEEM to predict high-degree receivers. We compare our method with the original edge exchangeable model (e2), the DC-SBM (Funke & Becker, 2019), as well as DGLFRM – a GNN-based method (Mehta et al., 2019). In most cases (10 out of 12), our method has highest CPR_{10}^{100}.

5.4. Other Data

We compare our method with other methods in two additional datasets, which include the Cora citation networks (McCallum et al., 2000) and Pubmed citation networks (Sen et al., 2008). The Cora citation network is composed of 2,708 documents, and 5,278 interactions. The Pubmed data is composed of 19,717 papers, and 44,324 interactions. The coverage probability ratio is calculated to evaluate the performance of different methods. Results and additional details of model fitting can be found at Appendix A.10.

6. Conclusion

In this paper, we integrate edge-level clustering into community detection within the edge exchangeability framework. The highlights of our proposed method are: (1) the identification of block structures under different categories; (2) the incorporation of the auxiliary interaction-wise prior knowledge; and (3) the scalability of the algorithm, which can handle networks of millions of interactions. We provide the theoretical boundary of node misspecification rate and supportive evidence from synthetic experiments. We demonstrate the method and compare it with several other methods using the TalkLife data. Our method detects more temporally stable community structures and gives better predictions than alternative methods such as DC-SBMs.

7. Acknowledgement

We thank Sebastian Zoellner and Mingjie Gao for helpful discussions.

References

Abbe, E. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Abbe, E., Bandeira, A. S., and Hall, G. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. Mixed membership stochastic blockmodels. *Advances in neural information processing systems*, 21, 2008.

Amini, A. A., Chen, A., Bickel, P. J., and Levina, E. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.

Ball, B., Karrer, B., and Newman, M. E. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.

Barber, M. J. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102, 2007.

Bedi, P. and Sharma, C. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.

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

Burnham, K. P. and Anderson, D. R. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

Chen, J. and Yuan, B. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

Crane, H. and Dempsey, W. Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113(523):1311–1326, 2018.

Dempsey, W., Oselio, B., and Hero, A. Hierarchical network models for exchangeable structured interaction processes. *Journal of the American Statistical Association*, pp. 1–18, 2021.

Fortunato, S. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Funke, T. and Becker, T. Stochastic block models: A comparison of variants and inference methods. *PloS one*, 14 (4):e0215296, 2019.

Galhotra, S., Mazumdar, A., Pal, S., and Saha, B. The geometric block model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. Community detection in degree-corrected block models. *The Annals of Statistics*, 46(5):2153–2185, 2018.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., Airoldi, E. M., et al. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2): 129–233, 2010.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Karrer, B. and Newman, M. E. Stochastic blockmodels and community structure in networks. *Physical review E*, 83 (1):016107, 2011.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.

Mehta, N., Duke, L. C., and Rai, P. Stochastic blockmodels meet graph neural networks. In *International Conference on Machine Learning*, pp. 4466–4474. PMLR, 2019.

Newman, M. E. Communities, modules and large-scale structure in networks. *Nature physics*, 8(1):25–31, 2012.

Newman, M. E. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, 2016.

Peixoto, T. P. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4(1):011047, 2014.

Pitman, J. et al. Combinatorial stochastic processes. *Lecture notes in mathematics*, 1875:7–24, 2002.

Rehurek, R. and Sojka, P. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2):2, 2011.

Rives, A. W. and Galitski, T. Modular organization of cellular networks. *Proceedings of the national Academy of sciences*, 100(3):1128–1133, 2003.

Röder, M., Both, A., and Hinneburg, A. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pp. 399–408, 2015.

Rohe, K., Chatterjee, S., and Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Sewell, D. K. Model-based edge clustering. *Journal of Computational and Graphical Statistics*, 30(2):390–405, 2020.

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., et al. A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Taddy, M. On estimation and selection for topic models. In *Artificial Intelligence and Statistics*, pp. 1184–1193. PMLR, 2012.

Yan, X. Bayesian model selection of stochastic block models. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 323–328. IEEE, 2016.

Yin, M., Wang, Y. R., and Sarkar, P. A theoretical case study of structured variational inference for community detection. In *International Conference on Artificial Intelligence and Statistics*, pp. 3750–3761. PMLR, 2020.

Zhang, A. Y. and Zhou, H. H. Theoretical and computational guarantees of mean field variational inference for community detection. *The Annals of Statistics*, 48(5): 2575–2598, 2020.

Zhang, Y. and Dempsey, W. Node-level community detection within edge exchangeable models for interaction processes. *arXiv preprint arXiv:2208.08539*, 2022.

Zhao, Y., Levina, E., and Zhu, J. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

A. Appendix

A.1. Proof of Theorem 3.1

We follow a similar proof logic as shown in (Zhang & Dempsey, 2022). Suppose the ground truth category label of edge is known. Consider an arbitrary category $t \in [T]$. Given initial labeling $e_t : \mathcal{P}_M \to [K_t]$ of all observed nodes in the network $\mathbf{Y_M^t}$ in category t, our goal is to show the misclassification rate (accounting for label switching) is bounded after a single iteration of an updating algorithm that approximately maximizes the likelihood given initial labeling. Specifically, we will show the misclassification rate decays rapidly and is therefore small for high-degree nodes.

Conditional on the model parameters Θ, the likelihood of observing the interaction networks that composed of edges in category t is given by:

$$P(Y_M^t|\Theta) = \prod_{b=1}^{K_t} \pi_b(t)^{L_b(t)} \prod_s f_{tb}^{s\ Deg_t(s)} \prod_{b'=1}^{K_t} B_t(b,b')^{W_t(b,b')} \prod_{r\in R(s)} f_{tb'}^{r\ Deg_t(r)} \tag{5}$$

Assumption A.1. In the proof below, we consider the setting where $K_t = 2$, $a = B_t(1,1) = B_t(2,2)$, $b = B_t(1,2) = B_t(2,1)$, $f_{tb}^x = f_{tb'}^x, \forall x \in \mathcal{P}_M, b, b' \in [K_t]$, $\pi_1(t) = \pi_2(t)$, and $a > b$. We call this a *balanced setting* since the likelihood of initiating an interaction and propensity of observing an arbitrary node are the same across different blocks.

Directed Graph Setting Note in the cataBEEM we consider sender and receiver nodes independently, which fit into the directed graph assumptions by definition. Consider node i as the sender node. Under Assumption A.1, the difference of log likelihood of assigning node to block 1 and block 2 is:

$$l_{i,1} - l_{i,2} = (\log a - \log b)(Deg_t(i,1) - Deg_t(i,2))$$

where $Deg_t(i,1)$ and $Deg_t(i,2)$ refer to the degree of node i being connected to node in block 1 and block 2 under label e_t. Consider node i as the receiver will give us a similar expression:

$$l_{i,1} - l_{i,2} = (\log a - \log b)(Deg_t(1,i) - Deg_t(2,i))$$

For simplicity, denote $Deg_t(i,1)$ and $Deg_t(i,2)$ as the degree of node i being connected to nodes from block 1 and 2 correspondingly, regardless of the direction. A mis-specification can happen if either the sender or the receiver are not correctly labeled. Therefore, a natural updating rule is: $e_t^i = 1$ if $Deg_t(i,1) > Deg_t(i,2)$. Let $\xi_j(e_t)$ for $j \in \mathcal{P}_M$ such that $\xi_j(e_t) = -1$ if $e_t^j = 1$ and $\xi_j(e_t) = 1$ if $e_t^j = 2$. Then define

$$\epsilon_i := \sum_j D_t(i,j)\xi_j(e_t)$$

where $D_t(i,j)$ is the number of interactions (i,j) in Y_M^t. Then the updating rule leads to correct specification of the node (i.e., the assigned label $e_t^i = 1$ equals the true label $z_i^t = 1$) if $\epsilon_i < 0$. Thus bounding the probability of misclassifying node i is equivalent to bounding $P(\epsilon_i > 0)$. Let $\{J_1\}$ be the set of nodes that match to the truth under e_t, and $\{J_2\}$ be the set of nodes that do not match to the truth under e_t, we have:

$$\begin{aligned}
\mathbb{E}(\epsilon_i) &= -Deg_t(i)\left(a\sum_{j\in\{J1\}} f_{t1}^j + b\sum_{j\in\{J2\}} f_{t2}^j - b\sum_{j\in\{J1\}} f_{t2}^j - a\sum_{j\in\{J2\}} f_{t1}^j\right) \\
&= -Deg_t(i)\left[a\left(\sum_{j\in\{J1\}} f_{t1}^j - \sum_{j\in\{J2\}} f_{t1}^j\right) - b\left(\sum_{j\in\{J1\}} f_{t2}^j - \sum_{j\in\{J2\}} f_{t2}^j\right)\right].
\end{aligned}$$

Let $\gamma_b = \sum_{j\in J1} f_{tb}^j \in [0,1]$ denote the weighted fraction of nodes that are correctly specified and note that $\sum_{j\in J1} f_{tb}^j + \sum_{j\in J2} f_{tb}^j = 1$. Let $\mu_{1,2} := a(2\gamma_1 - 1) - b(2\gamma_2 - 1)$ then

$$Var(\epsilon_i) = 4Deg_t(i)[\mu_{1,2}(1-\mu_{1,2})] \le Deg_t(i)$$

Recall $\epsilon_i = \sum_j D_t(i,j)\xi_j(e_t)$ which consists of independent random variables with mean μ and variance bound by 1. Then, by Bernstein's inequality, we have

$$P\left(\epsilon_i > \mathbb{E}(\epsilon_i) + c\right) \leq \exp\left(-\frac{c^2}{2(Deg_t(i) + 2c/3)}\right) \tag{6}$$

for any $c \geq 0$. Note the Bernstein inequality requires $c = -\mathbb{E}(\epsilon_i) = Deg_t(i)\mu_{1,2}$ to be greater or equal to 0, and thus $\mu_{1,2} > 0$. When considering node i in block 2, this implies a positivity condition $\mu_{2,1} > 0$. We use the same set of design on γ_b as in the original paper, which states that:

Assumption A.2 (Positivity)**.** Assume (a) $\gamma_b \in (1/2, 1]$ for $b = 1, 2$, i.e., the weighted fraction of nodes that are correctly specified is greater than $1/2$; and (b) that $\{\gamma_1, \gamma_2\}$ satisfy

$$\mu_{1,2} \wedge \mu_{2,1} =: \mu_{\min} = 2 \cdot a\left(2\frac{\gamma_{\min} + \gamma_{\max}}{2} - 1\right) - (2\gamma_{\max} - 1) > 0 \tag{7}$$

where $\gamma_{\min} = \gamma_1 \wedge \gamma_2$ and $\gamma_{\max} = \gamma_1 \vee \gamma_2$.

Assumption A.2(b) guarantees positivity of both $\mu_{1,2}$ and $\mu_{2,1}$ which is necessary for the labelling e_t to guarantee a bound via Bernstein's inequality.

Lemma A.3. *For each $i \in \mathcal{P}_M$, given $Deg_t(i)$ and propensities $\{f^i_{tb}\}$, then under Assumption A.1 and A.2 we have*

$$P(\epsilon_i > 0) \leq \exp\left(-\frac{Deg_t(i)\mu^2_{\min}}{4}\right)$$

Bounding misclassification rate. The misclassification rate under label e_t is:

$$M_{N_t}(e_t) = \inf_{\rho':[K_t]\to[K_t]} \frac{1}{N_t}\sum_{i=1}^{N_t} 1[e^i_t = \rho' z^t_i]$$

where ρ' indicates the potential label switching. Let $N(\xi(e_t)) = \sum_{i=1}^{N_t} 1(\epsilon_i \geq 0)$. Then the misspecification rate is bounded by :

$$M_{N_t}(e_t) \leq \frac{N(\xi(e_t))}{N_t} \tag{8}$$

The inequality is due to treating the ambiguous case $\epsilon_i = 0$ as an error. To bound the RHS of Eq 8, we assume node degree follows the Yule-Simon distribution parameterized by α such that the following Lemma from (Zhang & Dempsey, 2022) can be hold:

Lemma A.4. *Given the degree sequence $\{Deg_t(i)\}_{i\in\mathcal{P}_M}$ and propensities $\{f^j_{tb}\}$, then*

$$P_t := \frac{1}{N_t}\sum_{i=1}^{N_t} P(\epsilon_i > 0) \overset{a.s.}{\to} \sum_{d=1}^{\infty} \alpha B(d, \alpha+1)\exp(-d\mu^2_{\min}/4)$$

Bounding the RHS of (8). We next show $\frac{1}{N_t}N(\xi(e_t))$ is bounded. The proof relies on the following Lemma 5 from (Zhang & Dempsey, 2022; Amini et al., 2013):

Lemma A.5. *For independent Bernoulli R.V. X_i, $i \in [n]$ and any $u > \frac{1}{e}$, where e is the Eular's number,*

$$P\left(\overline{X} \geq eu\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i)\right) \leq \exp\left(-e\left(\sum_{i=1}^{n}\mathbb{E}(X_i)\right)u\log u\right)$$

Note that $1(\epsilon_i \geq 0)$ is a Bernoulli random variable. Given $\{f^i_{tb}\}$, $1(\epsilon_i > 0)$ are independent random variables for $i \in \mathcal{P}_M$. Then by Lemma A.5:

$$P\left[\frac{1}{N_t}N(\xi(e_t)) \geq euP_t\right] \leq \exp\left(-eN_tP_tu\log u\right)$$

which guarantees that as $M_t \to \infty$ the misclassification rate can be bounded by $u \cdot e \cdot P_t$. By Lemma A.4, P_t converges to a constant. Now consider the high-degree nodes:

Lemma A.6. *Let N_t^D denote the set of nodes with degree greater than D. Then define*

$$P_D = \frac{1}{N_t^D}\sum_{i=1}^{N_t^D} P(\epsilon_i > 0) \to C_\alpha^{-1}\sum_{d>D}\alpha B(d,\alpha+1)\exp(-d\mu_{min}^2)/4)$$

where $C_\alpha = \sum_{d>D}\alpha B(d,\alpha+1) \le 1$. Then one can construct a sequence D_M, such that as $M_t \to \infty$, $P_{D_M} \to 0$, and $N_t^{D_M} P_{D_M} \to \infty$. Any such sequence guarantees

$$\lim_{M_t\to\infty} P\left(\frac{1}{N_t^{D_M}} N(\xi(e_t)) > 0\right) = 0.$$

From approximating updates to variational inference Note that all above derivation relies on the approximating update rule that $e_t^i = 1$ if $Deg_t(i,1) > Deg_t(i,2)$. Consider the variational update for latent parameter ϕ_i^t for node $i \in \mathcal{P}_M$ (either sender or receiver). Assume $\lambda_{tb}^x = \lambda_{tb'}^x, \forall b, b' \in [K_t]$ and are known. With some algebra, it can be shown that:

$$\log\phi_{t1}^i - \log\phi_{t2}^i = \sum_j (\phi_{t1}^j - \phi_{t2}^j)(\log a - \log b)$$

where $j \in \mathcal{P}_M$ are the nodes that have interaction with node i. As a comparison:

$$l_{i,1} - l_{i,2} = (Deg_t(i,1) - Deg_t(i,2))(\log a - \log b)$$

There's a correspondence between $\sum_j \phi_{t1}^j$ and $Deg_t(i,1)$; $\sum_j \phi_{t2}^j$ and $Deg_t(i,2)$. The difference is that variational inference updates allow the uncertainty in the block assignments. This will introduce more variance as compared to the approximating update rules which only allows binary values. The vairance depends on the node degrees and the current block labeling status in the variational inference.

A.2. Algorithm Derivation

We provide a complete derivation of the variational inference algorithm in the section. Consider the ELBO as shown in Eq.3. The expectation of log of joint likelihood is:

$$\mathbf{E}_q[\log p(Y_M,\Theta)] = \underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{b=1}^{K_t}\prod_{i=1}^M P(z_{s(i)}^t = b|\pi_b(t))P(z_{r(i)}^t = b|\pi_b(t))]}_{(a)} + \underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{s(i),i=1}^M P(\pi(t)|\alpha^t)]}_{(b)}$$

$$+\underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{b=1}^{K_t}\prod_{i=1}^M P(s(i)|f,z,\tilde{z})P(r(i)|f,z,\tilde{z})]}_{(c)} + \underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{i=1}^M P(\tilde{z}_i = t|\tilde{\pi}_t)]}_{(d)}$$

$$+\underbrace{\mathbf{E}_q[\log\prod_{i=1}^M P(\tilde{\pi}|\tilde{\alpha})]}_{(e)} + \underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{b=1}^{K_t} P(f_{bt}|\gamma_{bt})]}_{(f)} + \underbrace{\mathbf{E}_q[\log\prod_{t=1}^T\prod_{b=1}^{K_t}\prod_{b'=1}^{K_t}\prod_{i=1}^M P_4(E(i)|B,s(i),r(i),\tilde{z},z)]}_{(g)}$$

Next, consider the entropy:

$$\mathbf{E}_q[\log q(\tilde{\Theta})] = \underbrace{\mathbf{E}_q[\log q(\pi|\eta)]}_{(1)} - \underbrace{\mathbf{E}_q[\log q(z|\phi)]}_{(2)} - \underbrace{\mathbf{E}_q[\log q(f|\lambda)]}_{(3)} - \underbrace{\mathbf{E}_q[\log q(\tilde{z}|\psi)]}_{(4)} - \underbrace{\mathbf{E}_q[\log q(\tilde{\pi}|\tilde{\eta})]}_{(5)}$$

Write out each term in its explicit form gives us:

$$(a) = \sum_{t=1}^{T}\sum_{b=1}^{K_t}\sum_{i=1}^{M}\psi_i^t\phi_{s(i)}^{bt}[\Psi(\eta^{tb}) - \Psi(\sum_{b=1}^{K_t}\eta^{tb})] + \sum_{t=1}^{T}\sum_{b=1}^{K_t}\sum_{i=1}^{M}\psi_i^t\phi_{r(i)}^{bt}[\Psi(\eta^{tb}) - \Psi(\sum_{b=1}^{K_t}\eta^{tb})]$$

$$(b) = \sum_{s(i),i=1}^{M}\sum_{t=1}^{T}\mathbf{E}_q[\log\frac{\Gamma(\sum_b\alpha^{tb})}{\prod_b\Gamma(\alpha^{tb})}\prod_b\pi_b(t)^{\alpha^{tb}-1}] = \sum_{s(i),i=1}^{M}\sum_{t=1}^{T}\log\frac{\Gamma(\sum_b\alpha^{tb})}{\prod_b\Gamma(\alpha^{tb})} + \sum_b\mathbf{E}_q[\log\pi_b(t)^{\alpha^{tb}-1}]$$

$$= \sum_{s(i),i=1}^{M}\sum_{t=1}^{T}\log\Gamma(\sum_b\alpha^{tb}) - \sum_b\log\Gamma(\alpha^{tb}) + \sum_b(\alpha^{tb}-1)[\Psi(\eta^{bt}) - \Psi(\sum_b\eta^{bt})]$$

$$(c) = \sum_{t=1}^{T}\sum_{b=1}^{K_t}\sum_{i=1}^{M}\psi_{s(i)}^t\phi_{s(i)}^{bt}[\Psi(\lambda_{bt}^{s(i)}) - \Psi(\sum_{n=1}^{N}\lambda_{bt}^n)] + \psi_i^t\phi_{r(i)}^{bt}[\Psi(\lambda_{bt}^{r(i)}) - \Psi(\sum_{n=1}^{N}\lambda_{bt}^n)]$$

$$(d) = \sum_{t=1}^{T}\sum_{i=1}^{M}\psi_i^t[\Psi(\tilde{\eta}^t) - \Psi(\sum_t\tilde{\eta}^t)]$$

$$(e) = \sum_{i=1}^{M}\mathbf{E}_q[\log\frac{\Gamma(\sum_t\tilde{\alpha}^t)}{\prod_t\Gamma(\tilde{\alpha}^t)}\prod_t\tilde{\pi}_t^{\tilde{\alpha}^t-1}] = \sum_{i=1}^{M}[\log\Gamma(\sum_t\tilde{\alpha}^t) - \sum_t\log\Gamma(\tilde{\alpha}^t) + \sum_t(\tilde{\alpha}^t-1)[\Psi(\tilde{\eta}^t) - \Psi(\sum_t\tilde{\eta}^t)]]$$

$$(f) = \mathbf{E}_q[\prod_t\prod_b\log\frac{\Gamma(\sum_{n=1}^{N}\gamma_{tb}^n)}{\prod_{n=1}^{N}\Gamma(\gamma_{tb}^n)}\prod_{n=1}^{N}(\lambda_{bt}^n)^{\gamma_n-1}]$$

$$= \sum_t\sum_b[\log\Gamma(\sum_{n=1}^{N}\gamma_{tb}^n) - \sum_{n=1}^{N}\log\Gamma(\gamma_{tb}^n) + \sum_{n=1}^{N}(\gamma_{tb}^n-1)[\Psi(\lambda_{bt}^n) - \Psi(\sum_{n=1}^{N}\lambda_{bt}^n)]]$$

$$(g) = \sum_{i=1}^{M}\sum_t\sum_b\sum_{b'}\mathbf{E}_q[\log B_t(b,b')^{z_{s(i)^t}=b, z_{r(i)}^t=b'}]$$

$$= \sum_i\sum_t\sum_b\sum_{b'}\psi_i^t\phi_{s(i)}^{bt}\phi_{r(i)}^{b't}[E(i)\log B_t(b,b') + (1-E(i))\log(1-B_t(b,b'))]$$

$$(1) = \sum_t\mathbf{E}_q[\log(\frac{\Gamma(\sum_b\eta^{tb})}{\prod_b\Gamma(\eta^{tb})})\prod_b\pi_t(b)^{\eta^{tb}-1}]$$

$$= \sum_t\log\Gamma(\sum_b\eta^{tb}) - \sum_b\log\Gamma(\eta^{tb}) + \sum_b(\eta^{tb}-1)[\Psi(\eta^{tb}) - \Psi(\sum_b\eta^{tb})]$$

$$(2) = \sum_i\sum_t\sum_b\phi_{s(i)}^{bt}\log\phi_{s(i)}^{bt} + \sum_i\sum_t\sum_b\phi_{r(i)}^{bt}\log\phi_{r(i)}^{bt}$$

$$(3) = \sum_b\sum_t[\log\Gamma(\sum_n\lambda_{bt}^n) - \sum_n\log\Gamma(\lambda_{bt}^n) + \sum_n(\lambda_{bt}^n-1)[\Psi(\lambda_{bt}^n) - \Psi(\sum_n\lambda_{bt}^n)]]$$

$$(4) = \sum_i\sum_t\psi_i^t\log\psi_i^t$$

$$(5) = [\log\Gamma(\sum_t\tilde{\eta}^t) - \sum_t\log\Gamma(\tilde{\eta}^t) + \sum_t(\tilde{\eta}^t-1)[\Psi(\tilde{\eta}^t) - \Psi(\sum_t\tilde{\eta}^t)]]$$

Maximize the above equation regarding each latent parameter leads to the following updates:

$$\hat{\psi}_i^t \propto \exp\{[\Psi(\tilde{\eta}^t) - \Psi(\sum_t \tilde{\eta}^t)]\} \times \exp\{\sum_{b=1}^{K} \phi_{s(i)}^{bt}[\Psi(\eta^{bt}) - \Psi(\sum_b \eta^{bt})]\}$$
$$\times \exp\{\sum_{b=1}^{K} \phi_{s(i)}^{bt}[\Psi(\lambda_{bt}^s) - \Psi(\sum_{n=1}^{N} \lambda_{bt}^n)]\} \times \exp\{\sum_{b=1}^{K} \phi_{r(i)}^{bt}[\Psi(\lambda_{bt}^{r(i)}) - \Psi(\sum_{n=1}^{N} \lambda_{bt}^n)]\}$$
$$\times \exp\{\sum_{b=1}^{K}\sum_{b'=1}^{K} \phi_{s(i)}^{bt}\phi_{r(i)}^{b't} \log B_t(b,b')\}$$

$$\hat{\phi}_s^{bt} \propto \exp\{\sum_{i=1,s(i)=s}^{M} \psi_i^t[\Psi(\eta^{bt}) - \Psi(\sum_b \eta^{bt})]\} \times \exp\{\sum_{i=1,s(i)=s}^{M} \psi_i^t[\Psi(\lambda_{bt}^s) - \Psi(\sum_{n=1}^{N} \lambda_{bt}^n)]\}$$
$$\times \exp\{\sum_{i=1,s(i)=s}^{M}\sum_{b'=1}^{K} \psi_i^t\phi_{r(i)}^{b't} \log B_t(b,b')\}$$

$$\hat{\phi}_r^{bt} \propto \exp\{\sum_{i=1,r(i)=r}^{M} \psi_i^t[\Psi(\lambda_{bt}^r) - \Psi(\sum_{n=1}^{N} \lambda_{bt}^n)]\} \times \exp\{\sum_{i=1,r(i)=r}^{M}\sum_{b'=1}^{K} \psi_i^t\phi_{s(i)}^{b't} \log B_t(b',b)\}$$

$$\hat{\tilde{\eta}}^t = \sum_{i=1}^{M} \psi_i^t + \tilde{\alpha}^t$$

$$\hat{\eta}^{tb} = \sum_{i=1}^{M} \psi_i^t\phi_{s(i)}^{bt} + \alpha^{tb}$$

$$\hat{\lambda}_{bt}^n = \sum_{i=1,s(i)=n}^{M} \psi_i^t\phi_{s(i)}^{bt} + \sum_{i=1,r(i)=n}^{M} \psi_i^t\phi_{r(i)}^{bt} + \gamma_{tb}^n$$

where Ψ is the digamma function. Note that by definition we need to normalize ϕ, ψ, such that: $\sum_b \hat{\phi}_{bt}^s = 1$, $\sum_{b'} \hat{\phi}_{b't}^r = 1$, and $\sum_t \hat{\psi}_t^i = 1$. Note here $\hat{\psi}_t^i$ corresponds to f_1, $\hat{\phi}_s^{bt}$ corresponds to f_2, $\hat{\phi}_r^{bt}$ corresponds to f_3, $\hat{\eta}^t$ corresponds to f_4, $\hat{\tilde{\eta}}$ corresponds to f_5, and $\hat{\lambda}_{bt}^n$ corresponds to f_6 in Algorithm 1.

A.3. Model Selection

We experiment on the model selection criteria mentioned in Section 3.5 using synthetic data. We simulate three datasets according to the steps as described in Section 4.1. We assume the Pitman-Yor process parameters are the same in all datasets. That is $\beta = 0.5$ and $\theta = 5$. Specifically, we have:

Dataset 1: The network contains 1000 interactions. Let $T = 2$ and $K = 4$. Assume $\tilde{\pi}_1 = \tilde{\pi}_2 = 0.5$; $\pi_1(1) = \pi_2(1) = \pi_3(2) = \pi_4(2) = 0.45$, while $\pi_1(2) = \pi_2(2) = \pi_3(1) = \pi_4(1) = 0.05$. The propensity matrix B is:

$$B_t = \begin{bmatrix} 0.9 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.9 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.9 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.9 \end{bmatrix}, \forall t \in [2]$$

Dataset 2: The network contains 2000 interactions. Let $T = 2$ and $K = 8$. Assume $\tilde{\pi}_1 = \tilde{\pi}_2 = 0.5$; $\pi_1(1) = \pi_2(1) = \pi_3(1) = \pi_4(1) = \pi_5(2) = \pi_6(2) = \pi_7(2) = \pi_8(2) = 0.45$, while $\pi_1(2) = \pi_2(2) = \pi_3(2) = \pi_4(2) = \pi_5(1) = \pi_6(1) =$

$\pi_7(1) = \pi_8(1) = 0.05$. The propensity matrix B is:

$$B_t = \begin{bmatrix} 0.8 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.8 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.8 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.8 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.8 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.8 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.8 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.8 \end{bmatrix}, \forall t \in [2]$$

Dataset 3: The network contains 1000 interactions. Let T=2 and K=4. Assume $\tilde{\pi}_1 = \tilde{\pi}_2 = 0.5$; $\pi_1(1) = \pi_2(1) = 0.45$, $\pi_3(1) = \pi_4(1) = 0.05$, while $\pi_1(2) = \pi_2(2) = 0.5$, $\pi_3(2) = \pi_4(2) = 0$. The propensity matrix takes the value:

$$B_1 = \begin{bmatrix} 0.9 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.9 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.9 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.9 \end{bmatrix}, B_2 = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.9 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We repeat the simulation steps 20 times in each dataset. First consider the marginal posterior maximization, we fit the BEEM model (Zhang & Dempsey, 2022) to the simulated datasets over a range of K, and calculate the marginal likelihood based on posterior estimates of the BEEM. The average log-likelihood are shown in Figure 3(a)(b)(c). In all datasets, the average marginal likelihood maximizes at the true value of K.

Next, we consider the BIC criteria. This time, we fit the DC-SBM to the simulated datasets over a range of K values, and calculate the BIC based on (Yan, 2016):

$$BIC = -\ln P(Y_M|\hat{\Theta}) + \frac{K^2}{2} \ln \Theta(N^3)$$

The results are shown in Figure 3(d)(e)(f). The average BIC maximizes at the true value in dataset 1 and 3. In dataset 2, the BIC are very similar over a range of K values. Note that BIC has a wider range of variation over the repeats. On the other hand, methods that are suitable to BIC criteria such as DC-SBM are much more computationally efficient.

Figure 3(g) shows the L_2 norm of block assignments as a function of degree cutoff. In particular, the algorithm is able to recover the true labels in the scenario when a global K is given but the number of blocks are different in different categories.

A.4. Simulation Details

We follow the generative process as described in Section 4.1 to simulate the network data. We assume the interactions have equal probability to be from any of the category t in all settings. That is, for an arbitrary interaction i, $\tilde{\pi}_i^t = 1/T, \forall t \in [T]$. We set parameters in the Pitman-Yor process as $\beta = 0.5$ and $\theta = 5$ in all categories and blocks. Specifically, we have:

Setting 1: The network contains 1,000 interactions. Let T=2 and K=4. Assume $\pi_1(1) = \pi_2(1) = \pi_3(2) = \pi_4(2) = 0.5$, while $\pi_1(2) = \pi_2(2) = \pi_3(1) = \pi_4(1) = 0$. That is, nodes from block 1 and 2 only involve category 1; nodes from block 3 and 4 involve only in category 2. We further assume the propensity matrix takes the value:

$$B_t = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.9 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}, \forall t \in [2]$$

Setting 2: The network contains 1,000 interactions. Let T=2 and K=4. Assume $\pi_1(1) = \pi_2(1) = \pi_3(2) = \pi_4(2) = 0.4$, while $\pi_1(2) = \pi_2(2) = \pi_3(1) = \pi_4(1) = 0.1$. The propensity matrix takes the value:

$$B_t = \begin{bmatrix} 0.8 & 0.07 & 0.07 & 0.07 \\ 0.07 & 0.8 & 0.07 & 0.07 \\ 0.07 & 0.07 & 0.8 & 0.07 \\ 0.07 & 0.07 & 0.07 & 0.8 \end{bmatrix}, \forall t \in [2]$$

Setting 3: The network contains 1,000 interactions. Let T=2 and K=4. Assume $\pi_1(1) = \pi_2(1) = \pi_3(2) = \pi_4(2) = 0.3$, while $\pi_1(2) = \pi_2(2) = \pi_3(1) = \pi_4(1) = 0.2$. The propensity matrix takes the value:

$$B_t = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}, \forall t \in [2]$$

Setting 4: The network contains 4,000 interactions. Let T=4 and K=16. Assume $\pi_1(1) = \pi_2(1) = \pi_3(1) = \pi_4(1) = 0.25$, while $\pi_5(2) = \pi_6(2) = \pi_7(2) = \pi_8(2) = 0.25$, $\pi_9(3) = \pi_{10}(3) = \pi_{11}(3) = \pi_{12}(3) = 0.25$, and $\pi_{13}(4) = \pi_{14}(4) = \pi_{15}(4) = \pi_{16}(4) = 0.25$. That is, nodes from block 1, 2, 3, and 4 only involve category 1; nodes from block 5, 6, 7, and 8 involve only in category 2, nodes from block 9, 10, 11, and 12 only involve category 3; nodes from block 13, 14, 15, and 16 involve only in category 4. Assume a block diagonal structure of the propensity matrix, such that:

$$B(sub) = \begin{bmatrix} 0.9 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.9 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.9 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.9 \end{bmatrix}; B_t = \begin{bmatrix} B(sub) & 0 & 0 & 0 \\ 0 & B(sub) & 0 & 0 \\ 0 & 0 & B(sub) & 0 \\ 0 & 0 & 0 & B(sub) \end{bmatrix}, \forall t \in [4]$$

Setting 5: The network contains 1,000 interactions. Let T=2 and K=4. Assume $\pi_1(1) = \pi_2(1) = \pi_3(1) = \pi_4(1) = 0.25$, while $\pi_1(2) = \pi_2(2) = 0.5$, $\pi_3(2) = \pi_4(2) = 0$. The propensity matrix takes the value:

$$B_1 = \begin{bmatrix} 0.9 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.9 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.9 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.9 \end{bmatrix}, B_2 = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.9 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A.5. Discussion on Label Switching in Simulated Data

In this paper, label switching refers to the category labels and the block labels. First consider the category labels on interactions. We use the following strategy to match the inferred labels with the ground truth:

$$\arg \min_{\rho_t:[T]\mapsto[T]} \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\tilde{z}_i - \hat{\tilde{z}}_i)^2}$$

This will give us the matching ρ_t that minimize the distance between the inferred category labels and the truth. Conditional on the category being matched, the block labels in each category can be matched according to:

$$\arg \min_{\rho_b^t:[K_t]\mapsto[K_t]} \sqrt{\frac{1}{N}\sum_{n=1}^{N}(z_n^t - \hat{z}_n^t)^2}$$

This will give us the matching ρ_b^t that minimize the distance between the inferred block labels and the truth in each category.

A.6. Exploratory Data Analysis and Model Selection in TalkLife Data

Figure 4 shows the degree distribution for posts in (a) annual data, (b) six months data, and (c) one month data (e.g. January). The power-law degree distribution is apparent in the overall network data of Year 2019, as well as in the two subsets of the annual data. This fit into the underlying assumption of our model.

Regarding the selection of the number of underlying blocks K. We utilize the BIC criteria as mentioned in Section 3.5. We apply DC-SBM to the TalkLife data. Figure 5 shows the BIC curve in different datasets. We pick the K if it gives the maximal BIC in each dataset, as shown in Table 7.

Next, we consider the selection of the number of underlying categories T. In TalkLife data, we assume the underlying topics indicates the categories. Therefore, we focus on the textual data. The distribution of the word count in each of the

post-comment pair are shown in Figure 6. Note that the text exchange between users are mainly in the form of short snippets. The average length of word count in the post-comment pairs is 17. The left skewed distribution also indicates this fact. We apply the LDA model to the texts from all post-comment pairs in Year 2019. We assume the text exchange between the poster and the commentator in a single post is one document when fitting the LDA model.

Before running the LDA model, we clean the textual data based on the following steps. We first remove the numbers, urls, stopping words, and other special characters from the corpora. Next, we filter out words whose lengths are less than 3. Followed by the step to lemmatize and stem the words. We feed the LDA model with the cleaned dataset.

Next, we consider the selection of topic T. The selection is based on the Jaccard Index and the coherence score. Jaccard Index is used to measure the similarity between two sets. In this example, each topic is composed of a set of words. Denote the set of word as $T_t, \forall t \in [T^*]$, where T^* is an arbitrary number that is set to be the number of topics in LDA model for now. Jaccard Index is defined as:

$$JI_{t1t2} = \frac{|T_{t1} \cap T_{t2}|}{|T_{t1} \cup T_{t2}|}, \forall t1, t2 \in [T^*]$$

The average value of Jaccard Index over all pair-wise topic combinations can be used to measure the between-topic difference. Meanwhile, we consider the coherence score, which is defined as:

$$C = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum j = i + 1^N PMI(w_i, w_j);\ PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

The average coherence score measures the co-occurence rate of words, which can be used to measure the within topic similarity. We select the number of topics as the one that maximize the within topic similarity and the between topic difference. The value is given when the difference between the coherence score and the Jaccard Index is maximized, as shown in Figure 7. Note here due to the computational concern, the number of topics is limited to be less or equal to 10. With this restriction, the number of topic is selected as $T = 6$. A visualization of the word distribution in each topic is shown in Figure 8.

A.7. Community Structures in TalkLife

We construct the relationship matrix between users in TalkLife data based on the estimates of the parameters. Consider two users in the network s and r. The probability of observing an interaction between them in category t based on the posterior estimates is given by: $\vec{z}_s^t B_t \vec{z}_r^{t\prime}$. The visualization of the example relationship matrices are shown in Figure 9, Figure 10, Figure 11, and Figure 12.

A.8. Coverage Probability Rate in Different Settings

In Table 2, we only provide the average value of the L_2 norm. Complete results is shown in Table 7. In addition, we consider the coverage probability rate in different settings that correspond to different cutoffs on the receivers' degrees, and different cutoffs on the senders' degrees. We experiment on setting receiver nodes' degree cutoff to be 5 and 20 (Table 8), and selecting the sender nodes based on the degree cutoff, e.g. $Deg(s) \geq 200$ and $Deg(s) \geq 500$ (Table 9). In all settings, our method gives the highest CPR in most datasets.

A.9. Time Complexity of the Algorithms

The theoretical computational complexity is $O(m)$, where m is the total number of interactions in the network. Shown in Table 4 is the computational time as a function of number of interactions in synthetic data. All experiments are conducted on Intel(R) Xeon(R) Platinum 8176 CPU at 2.10GHz.

We also compare the CataBEEM with (1) DGLFRM – a SBM-based GNN model; and (2) LSEC – model-based edge clustering in two real-world datasets. Results are summarized in Table 5

A.10. Additional Real-world Data

In addition to the TalkLife data, we include two more data sets to compare the performance of different methods. Namely, the Cora citation networks and the Pubmed citation networks. The Cora citation network is composed of 2,708 documents, and 5,278 interactions. The Pubmed dataset is composed of 19,717 papers, and 44,324 interactions. We apply all methods in

Table 4. Time complexity of the algorithm. Experiments are repeated 20 times in each setting.

INTERACTIONS	AVERAGE TIME IN SEC
1,000	24.55 (6.62)
5,000	29.40 (12.50)
10,000	40.80 (17.45)
50,000	136.70 (28.09)
100,000	232.85 (63.21)

Table 5. Comparison of time complexity (sec) of different algorithms in real world datasets.

	CORA	PUBMED
CATABEEM	459	865
DGLFRM	174	9,366
LSEC	660	52,524

comparison to both datasets. The coverage probability ratio is calculated to evaluate the performance of different methods. In Cora data, we randomly split the interactions into two parts. We calculate the probability certain paper being cited in the first half of the data, and evaluate the results in the second half of the data. Due to the fact that papers being cited by the same paper do not overlap in the two parts of the Cora data, and that the network is very sparse, we consider the coverage probability of all papers being cited in the second half of the data given the paper has cited more than 20 papers. For instance, consider a specific paper s^* (the degree of which is greater than 20), we calculate $P(r|s^*)$ for all potential papers in the citation network based on the first half of data. Denote the set of papers cited by s^* in the second half of data as $\mathcal{R}$,the coverage probability of s^* is then defined as $\sum_{r\in\mathcal{R}} P(r|s^*)$. Similar process is done on Pubmed data. Results are shown in Table 6.

Table 6. Coverage Probability Ratio of different algorithms in Cora and Topmed data.

DATA	CATABEEM	DC-SBM	E2	DGLFRM
CORA	3.87	1.00	1.02	0.96
PUBMED	2.21	2.75	2.75	1.61

A.11. Supplementary Figures and Tables

Table 7. Number of underlying blocks and L_2 norms of block labels in different time frames.

DATA SET	K	L_2 (CATABEEM)	L_2 (DC-SBM)
SIX MONTHS	2	0.48	0.70
JAN	4	0.42	0.61
FEB	2	0.47	0.71
MAR	6	0.34	0.52
APR	4	0.42	0.61
MAY	2	0.48	0.70
JUN	4	0.39	0.61
JUL	4	0.39	0.61
AUG	6	0.33	0.52
SEP	6	0.34	0.53
OCT	4	0.39	0.61
NOV	8	0.29	0.47

Table 8. Coverage Ratio Probability (CRP) calculated by different methods in different settings with varying receiver degree cutoffs. * indicates the highest value among all methods.

DATA SET	CPR_5^{100} (CATABEEM)	CPR_5^{100} (SBM)	CPR_5^{100} (E2)	CPR_{20}^{100} (CATABEEM)	CPR_{20}^{100} (SBM)	CPR_{20}^{100} (E2)
SIX MONTHS	2.12	2.48*	2.47	2.39*	2.34	2.28
JAN	1.97*	1.72	1.78	3.22*	1.82	1.61
FEB	2.21*	2.18	2.16	3.12*	2.48	2.43
MAR	2.02*	1.94	1.90	2.29*	2.09	2.0
APR	1.83	2.02	2.04*	2.0	2.07*	2.02
MAY	1.94	2.08*	2.07	2.90*	2.60	2.69
JUN	2.19*	1.02	2.17	2.96*	1.01	2.75
JUL	2.11	2.27*	2.23	1.90	2.62*	2.57
AUG	2.80*	2.11	2.09	2.93*	2.22	2.14
SEP	1.79	1.82*	1.81	2.59*	2.08	2.06
OCT	2.55*	2.19	2.16	2.13	2.38*	2.34
NOV	2.15*	2.10	2.11	4.95*	2.82	2.68

Table 9. Coverage Ratio Probability (CRP) calculated by different methods in different settings with varying sender degree cutoffs. * indicates the highest value among all methods.

DATA SET	CPR_{10}^{D200} (CATABEEM)	CPR_{10}^{D200} (SBM)	CPR_{10}^{D200} (E2)	CPR_{10}^{D500} (CATABEEM)	CPR_{10}^{D500} (SBM)	CPR_{10}^{D500} (E2)
SIX MONTHS	2.32*	1.68	1.65	2.23	2.39	2.40*
JAN	3.75*	1.72	1.64	2.89*	2.08	2.04
FEB	2.76*	2.43	2.38	3.05*	2.61	2.59
MAR	2.31*	2.03	1.92	1.96	2.46*	2.42
APR	1.78	1.92*	1.88	2.58*	2.33	2.29
MAY	2.33	2.61	2.62*	2.79*	2.48	2.46
JUN	2.57*	1.01	2.39	2.03*	1.00	2.02
JUL	2.22	2.42*	2.38	1.96	2.65	2.66*
AUG	3.16*	2.21	2.13	3.08*	2.29	2.24
SEP	1.71	1.77*	1.73	1.65	2.50*	2.44
OCT	2.85*	2.48	2.42	2.69*	2.68	2.62
NOV	2.95*	2.23	2.16	3.45*	2.71	2.63

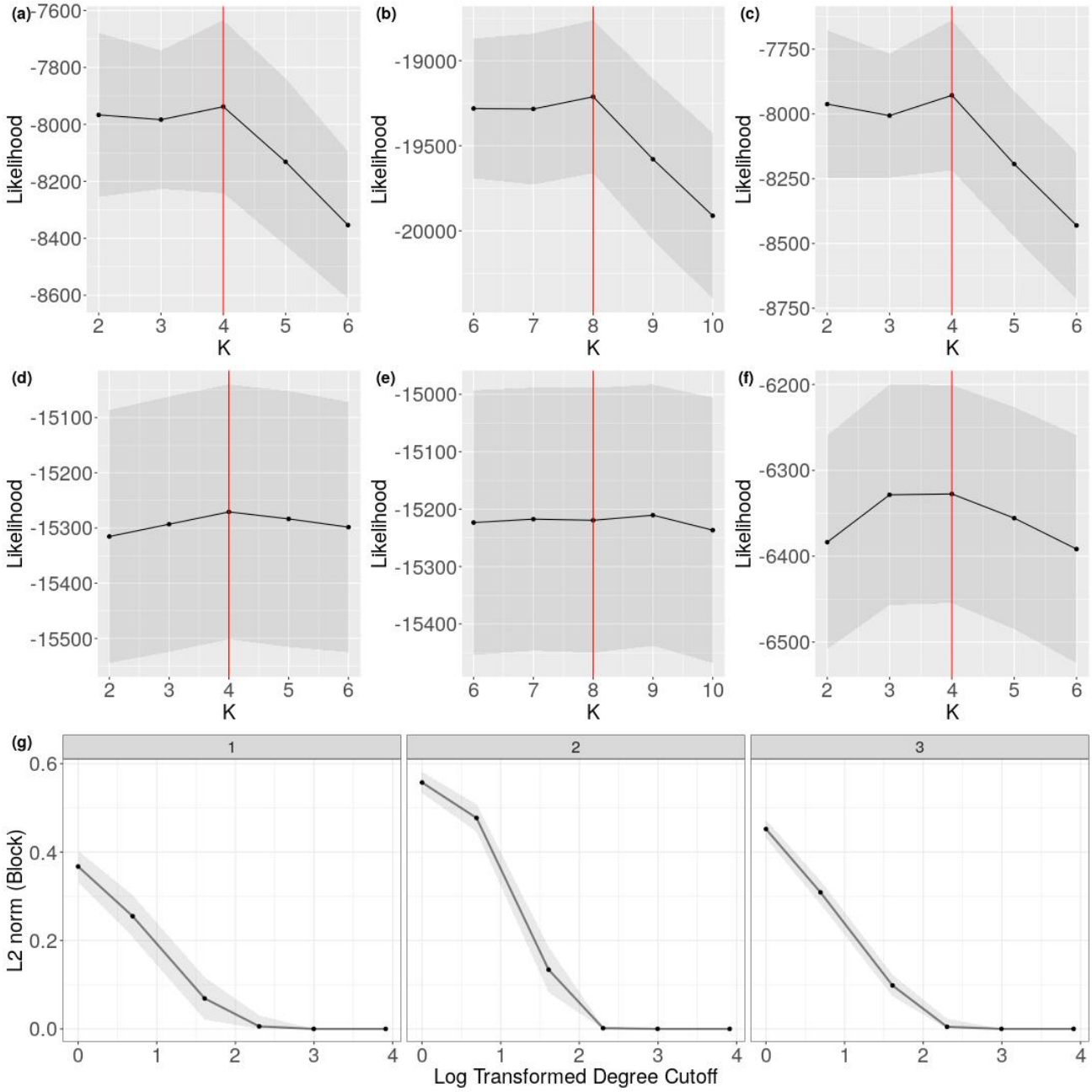

Figure 3. (a)(b)(c) The average marginal log-likelihood given by posterior estimates from BEEM that corresponds to dataset1, dataset2, and dataset3. (d)(e)(f) The average BIC given by the estimates from DC-SBM that corresponds to dataset1, dataset2, and dataset3. The black line is the mean value, and the grey area is the confidence interval. The underlying truth is indicated by the red vertical line. (g) L2 norm of block assignments as a function of degree cutoff in dataset 1, 2, and 3.

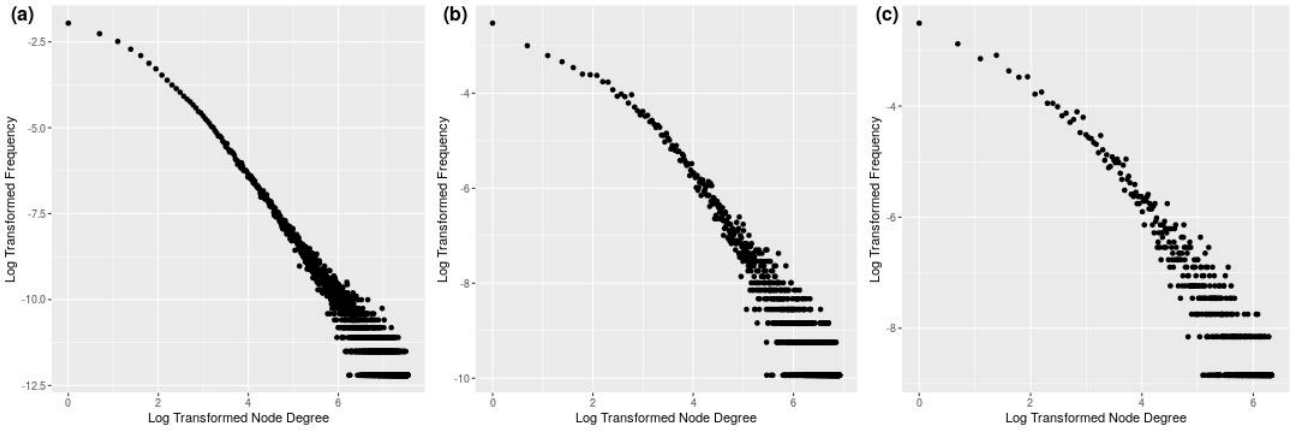

Figure 4. Node degree distribution in TalkLife data that corresponds to (a) annual data; (b) six months data; (c) one month data.

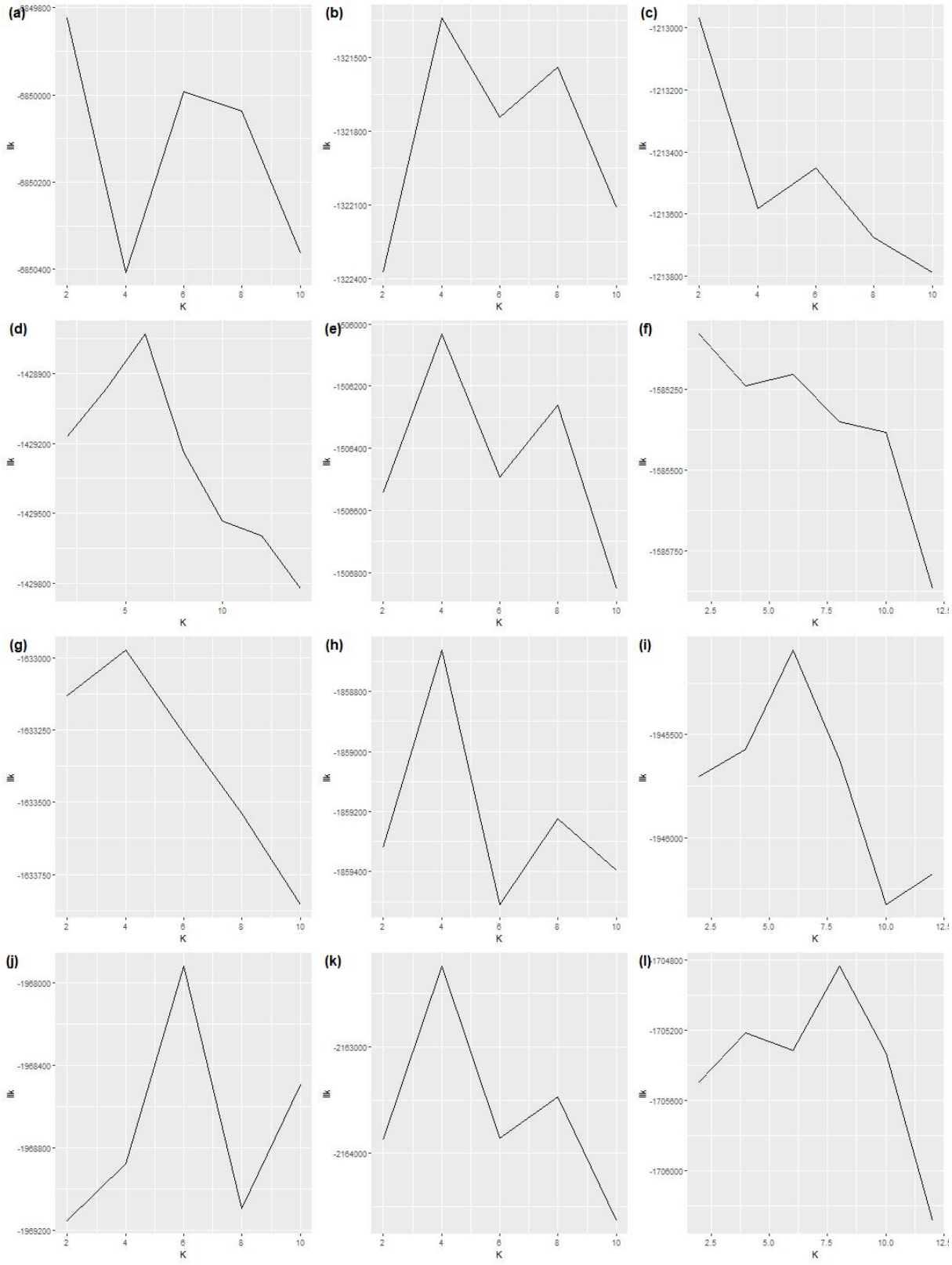

Figure 5. BIC curve after applying to DC-SBM to (a) Jan to Jun, (b) January, (c) February, (d) March, (e) April, (f) May, (g) June, (h) July, (i) August, (j) September, (k) October, (l) November data.

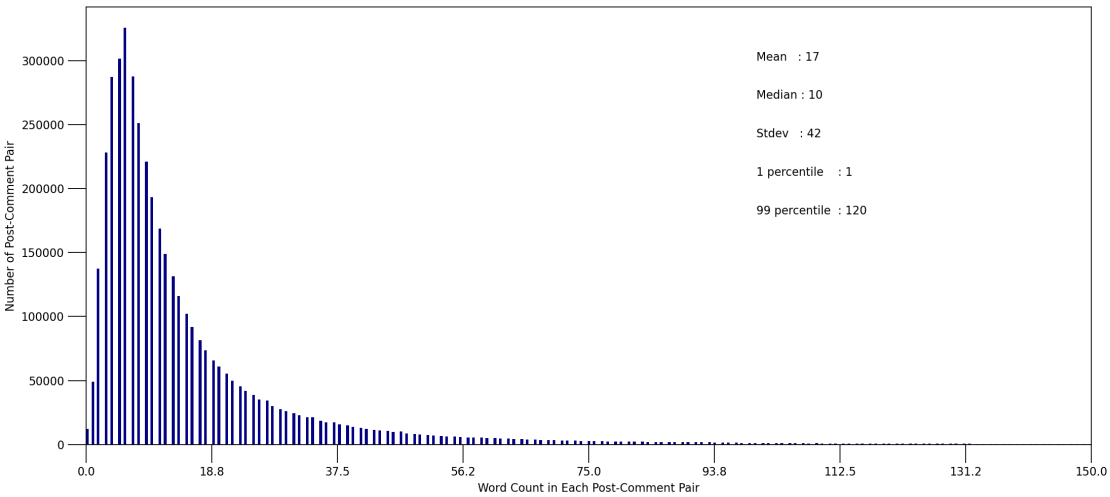

Figure 6. Distribution of word counts in post-comment pairs in TalkLife data.

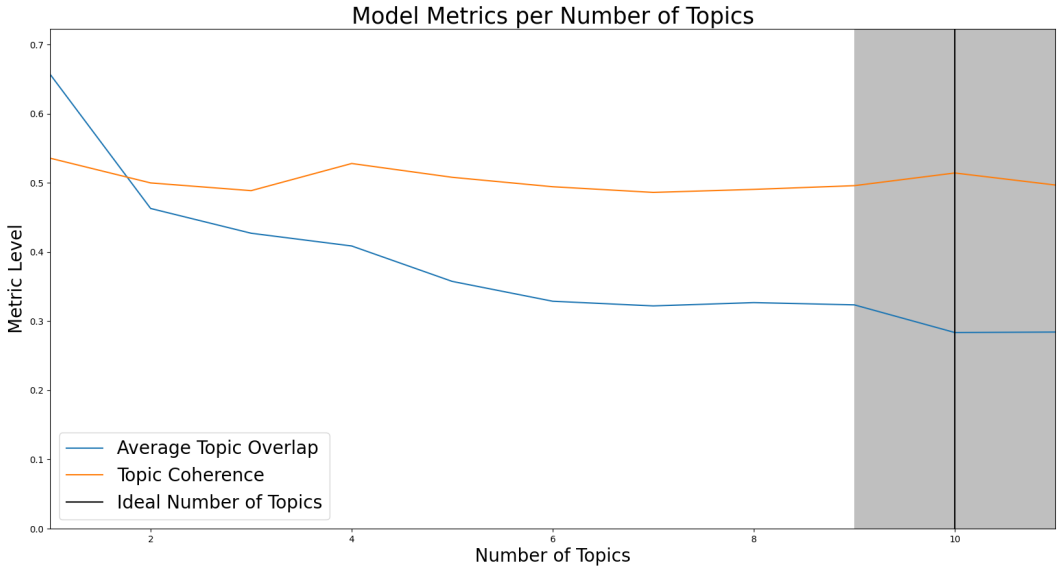

Figure 7. Jaccard Index (blue) and Coherence Score (orange) as a function of number of topics.

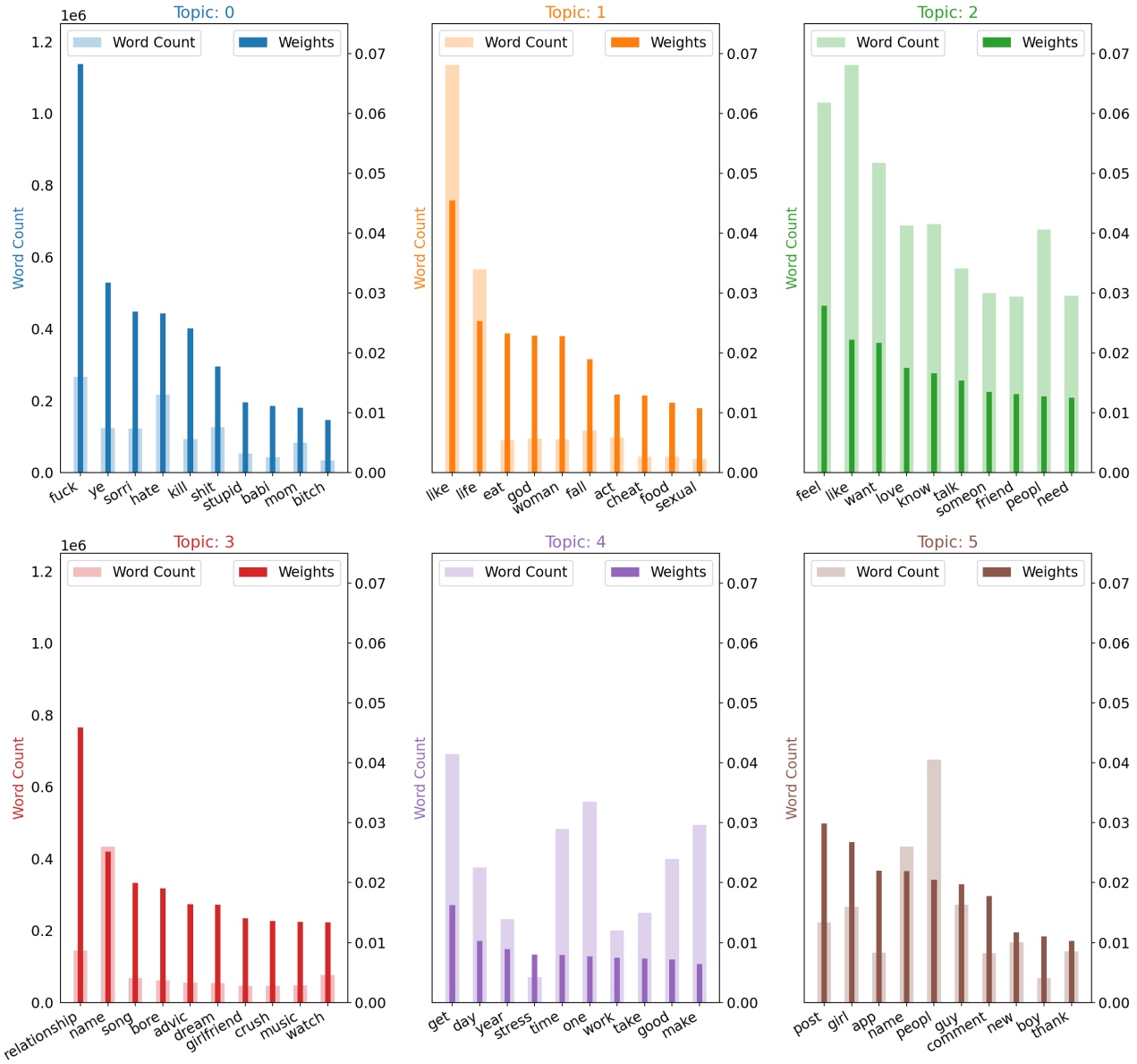

Figure 8. Visualization of the words' weight in each topic and their overall frequency in the corpora.

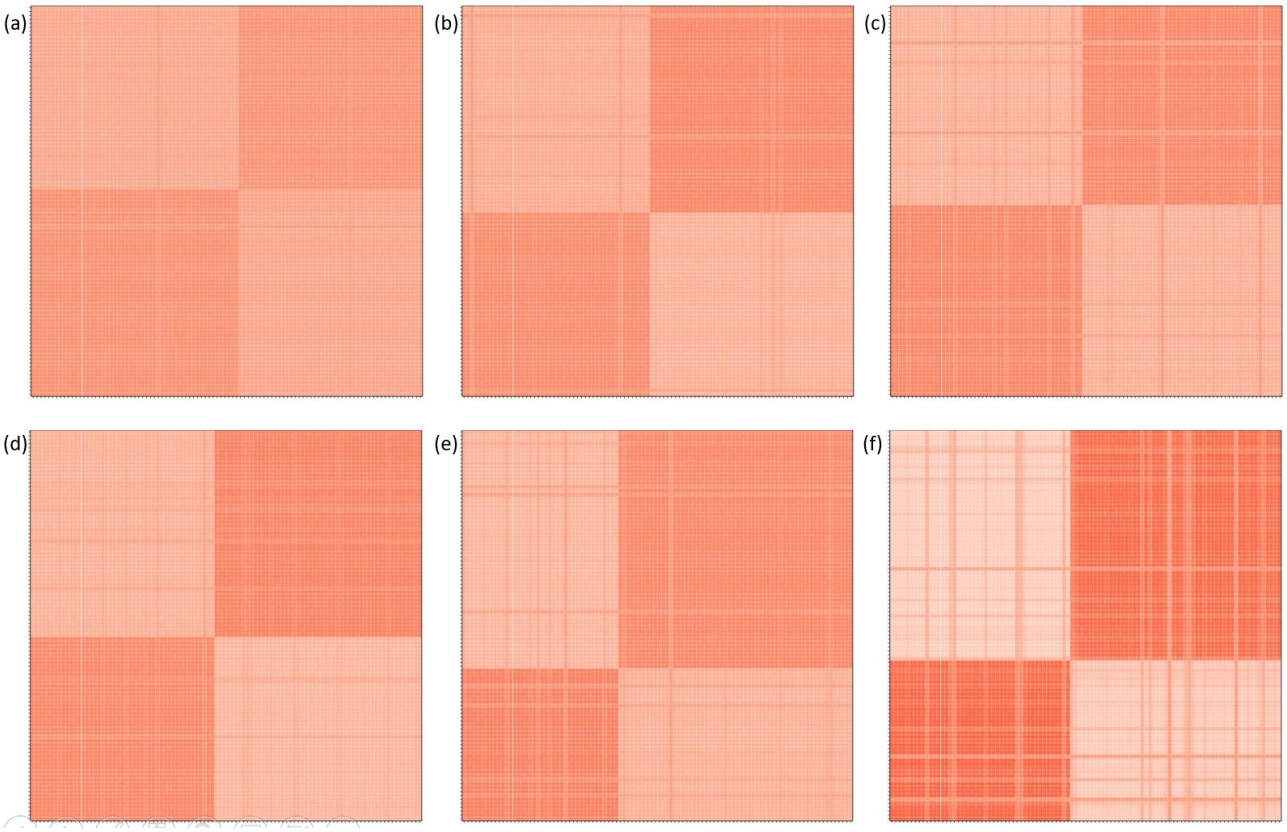

Figure 9. Inferred Community Structures in six months (Jan-Jun) data. x-axis and y-axis are ordered by block labels.

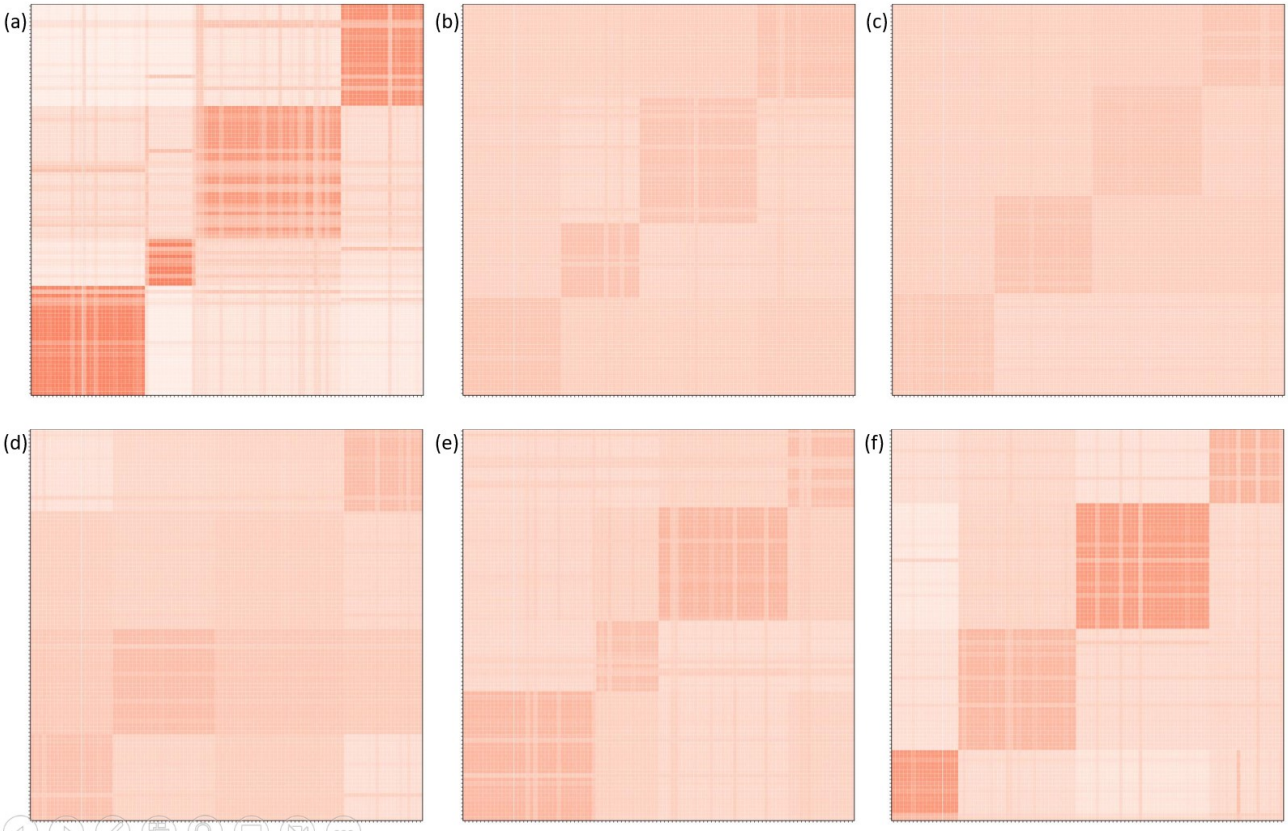

Figure 10. Inferred Community Structures in June data. x-axis and y-axis are ordered by block labels.

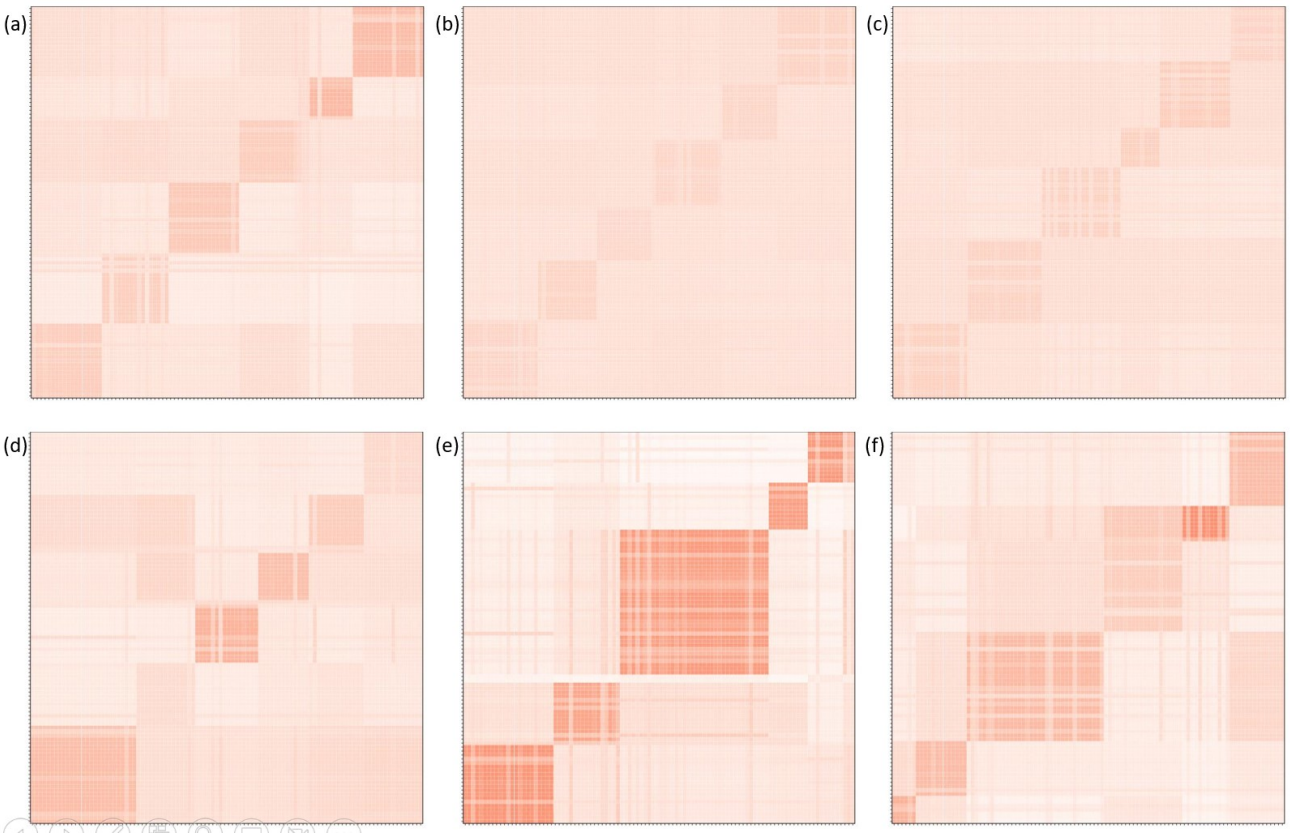

Figure 11. Inferred Community Structures in March data. x-axis and y-axis are ordered by block labels.

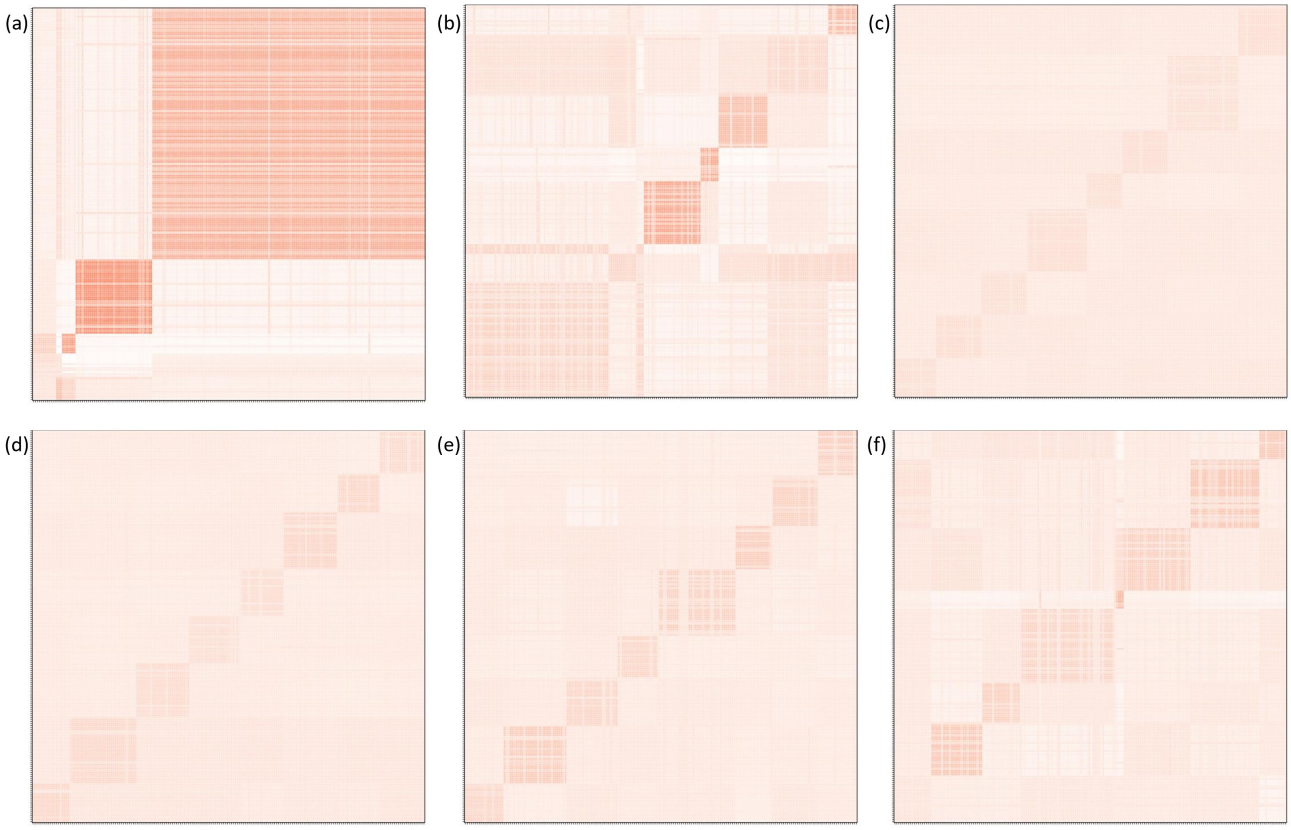

Figure 12. Inferred Community Structures in November data. x-axis and y-axis are ordered by block labels.