

# Community detection within edge exchangeable models for interaction processes

Yuhua Zhang, Walter Dempsey

September 30, 2021

## Abstract

Health scientists are increasingly interested in discovering community structure from peer-to-peer support networks. While many methods have been proposed for finding community structure, few account for the fact that these modern networks arise from processes of interactions in the population such as user post and comment exchanges. We introduce cluster-wise edge exchangeable models for the study of interaction networks with latent community structure. In particular, we introduce the cluster-wise vertex components model as a canonical example. Several theoretical and practical advantages over traditional vertex-centric approaches are highlighted. In particular, cluster-wise edge exchangeable models allow for sparse degree structure and power law degree distributions within communities. Our theoretical analysis bounds the misspecification rate of cluster assignments, while supporting simulations show the properties of the network can be recovered. A computationally tractable Gibbs algorithm is derived. We demonstrate the proposed model using post-comment interaction data from Talklife, a large-scale online peer-to-peer support network.

**Keywords**— Large-scale Sparse Network, Edge Exchangeable Model, Community Detection, Power-law Degree Distribution

# 1 Introduction

The WorldWideWeb, social media, and technological innovation allow people from around the world to easily interact across geographical, cultural and economic boundaries. Among its many benefits, increased connectivity enhances information flow, promotes community building, and facilitates the formation of support systems for individuals struggling with illness and addiction. In this paper, we focus on network data arising from sequences of interactions collected on social media platforms such as peer-to-peer support networks. In suicide research, for example, social support has been shown to be a preventative factor for future suicidal ideation [12]. To further these goals, this paper focuses on discovery of latent community structure.

An increasingly popular class of models for community detection are the stochastic block models (SBMs) [10]. The basic version of the SBM assumes vertices within the same block have the same probability of forming an edge (i.e., interact) with other vertices, and within-cluster interactions are more likely than between-cluster interactions. Many associated methods have been proposed. These include but are not limited to spectral clustering based methods ([18], [3]), Bayesian methods ([15], [13]), and pseudo-likelihood based methods ([2], [20]). The stochastic block model was later generalized to the degree-corrected stochastic block model (DCSBM) [11], which allows for vertex degree heterogeneity. Theoretical guarantees of community recovery have been well established [7, 2, 28].

As mentioned in the previous literature[1], the power law degree distributions are observed in many communication and social networks. That is, the network contains a few high degree nodes and many low degree nodes. This feature of network can be exploited when searching for the underlying community structures of the network. Efforts have been made to adapt the SBM to the power-law distributions [17]. Though much progress has been made, SBM and its extensions do not account for the fact that modern networks arise from processes of interactions. While the previously proposed edge exchangeable model has well addressed this problem [4]. Meanwhile, it also incorporated the power-law structure and the sparsity of the network into the estimation of the network properties.

The model proposed in this paper is motivated by an important fact: most common complex net-

works constructed from interaction data are clustered. The edge exchangeable model focus mainly on the inference of the statistics of the network, but doesn't consider the community structure. Here, we propose a new model which extends the edge exchangeable model to allow community structure while permitting power-law degree and sparsity. Our goal is to have a better understanding of the network structure through the identification of the latent communities formed by nodes in the network, as well as provide the accurate estimates of the power-law parameters of the corresponding sub-networks.

## 1.1 Outline and main contributions

The main contribution of this paper is organized in the following way. We start by formally defining the network data and notations in Definition 2.2. Followed by the generative process of the interaction data being described in Section 2.3. A computationally tractable Gibbs sampling inferential algorithm is derived in Section 3.1, followed by the simulation results in Section 3.2. We show the application of our method in the real world health-related network data in Section 4. Finally, We close the paper with the summary of the proposed methods and its implication in the discussion section.

# 2 Cluster-wise Edge Exchangeable Model

## 2.1 Motivating Example

Our motivating example is TalkLife, a large-scale online peer support network for mental health. The users of TalkLife post short snippets of text to which other users react and comments. Each post consists of a poster and a set of commentators. All the posts are of at least one of the health-related topics, such as Anxiety/ Panic/ Fear suspected, Body Image/ Eating Disorders suspected as so on. The platform has collected millions of interactions among its users. A primary purpose of TalkLife is to strengthen its peer support networks and improve mental health outcome for its users. But the scale of the network constructed from all users' data is too large to handle. Besides, the network is of complex structures, and viewing the entire dataset as a whole network will lead

to the loss of many features of the interactions among the users. Therefore, a natural question that arise is to break the entire dataset into smaller groups and identify certain groups of people from the interaction data.

## 2.2 Notation and Data

We start by defining cluster-structured interaction data. Inspired by the previous work [4], we extend the definition of interaction data to the setting where there exists an underlying cluster structure. To start, we use a simple example based on Talklife. Recall each Talklife post consists of a poster  $s$  and a set of commentators  $\{r_1, \dots, r_k\}$ . Then a post on Talklife can be summarized by the ordered pair  $\{\{s\}, \{r_1, \dots, r_k\}\}$ . See Figure 1 for a visualization taking in the Talklife data and constructing the ordered pairs for a set of posts. Here, posters and commentators are drawn from the same underlying population, which we denote  $\mathcal{P} \subset \mathbb{N}$ . Definition 2.1 formalizes this example into a general definition of interaction data.

**Definition 2.1 (Interaction data)** For a set  $\mathcal{P}$ , let  $fin(\mathcal{P})$  be the multisets of all finite subsets of  $\mathcal{P}$ . The interaction process of  $\mathcal{P}$  is the correspondence  $\mathcal{I} : \mathbb{N} \rightarrow fin(\mathcal{P})$  between the natural numbers and finite subsets of  $\mathcal{P}$ .

Additional structure can be incorporated into Definition 2.1. In Talklife, for example, there is a single poster and multiple commentators. Definition 2.1 can capture this by replacing  $fin(\mathcal{P})$  with  $fin_1(\mathcal{P}) \times fin(\mathcal{P})$  where  $fin_1(\mathcal{P})$  are sets of size one from  $\mathcal{P}$ . See [5] for additional examples of these extensions. Note that the nodes are labeled in the order of appearance. If a new individual is chosen, that person is given the next label in the order of appearance. For example, if they are the 10th person to be observed, they are labelled as individual 10.

Using Figure 1 as an example, each interaction is represented by the user who posts and the users who comment. The users, regardless of being posters or commentators, are drawn from the population  $\mathcal{P}$  set equal to  $\mathbb{N}$ . The interaction process is a correspondence  $\mathcal{I} : \mathbb{N} \rightarrow fin_1(\mathbb{N}) \times fin(\mathbb{N})$ . For Post 1,  $\mathcal{I}(1) = (\{1\}, \{3, 4, 5\})$  representing the poster posted by User 1 being commented by User 3, 4, and 5. The second Post  $\mathcal{I}(2) = (\{2\}, \{6, 7\})$  represents User 2 posts and User 6 and 7 comment.

Under our model assumption, individuals who are from the same cluster have more frequent interactions with the users from the same cluster as compared to other users. In Figure 1, for example, User 3, 4, and 5 are in the same cluster, which leads to a higher likelihood of observing these individuals commenting on a post by User 1. User 8 and User 9 are in a separate cluster, which explains the second interaction. Note, however, that users in distinct clusters may comment on each other's posts, i.e., User 7 comments on a post by User 2. We formalize community structure in interaction data in Definition 2.2.

**Definition 2.2 (Cluster structured interaction data)** A  $K$ -cluster structure is defined as the mapping  $B : \mathcal{P} \rightarrow [K]$ , such that for all  $x \in \mathcal{P}$ ,  $B(x) \in [K]$  is the cluster that  $x$  is assigned. The interaction process induced by  $B$  is called the *cluster interaction process*, which is the correspondence  $\mathcal{I}_B : \mathbb{N} \rightarrow \text{fin}([K])$ .

In our running example shown in Figure 1,  $K = 2$  and the interaction process induced by  $B$  is  $\mathcal{I}_B(1) = (\{1\}, \{1, 1, 1\})$  for Post 1, and  $\mathcal{I}_B(2) = (\{1\}, \{1, 2\})$  for Post 2, and  $\mathcal{I}_B(3) = (\{2\}, \{2, 2\})$  for Post 3. Note that for the pre-image  $B^{-1}$ ,  $B^{-1}(i) \cap B^{-1}(j) = \emptyset$  if  $i \neq j$ , and  $B^{-1}(i) = B^{-1}(j)$  if  $i = j$  for  $i, j \in [K]$ . That is,  $B$  partitions  $\mathcal{P}$  into  $K$  non-overlapping sets.

In the remainder of the paper, we write  $\mathcal{Y}_m$  to indicate the observed network with edges labeled in  $m \in \mathbb{N}$ , and  $E_1, E_2, \dots, E_m$  as the interactions observed, where  $E_j = (\{S_j, C_j^{(s)}\}, \{R_j, C_j^{(r)}\})$ ,  $j \in [m]$ , that is, the interaction  $j$  is initiated from the sender  $S_j$  of cluster  $C_j^{(s)}$  and points to the receiver  $R_j$  of cluster  $C_j^{(r)}$ . Let  $v(\mathcal{Y}_m)$  be the number of vertices in the network, and  $m(\mathcal{Y}_m)$  be the total degree of the network. Note that  $m(\mathcal{Y}_m) = 2m$ . We subscript  $c \in [K]$  to indicate the cluster specific variables. For example,  $m(\mathcal{Y}_c)$  is the total degree of cluster  $c$ . We now give the definition of the cluster edge exchangeable network.

**Definition 2.3 (Cluster edge exchangeable network)** The network constructed from interactions within the same cluster  $\mathcal{Y}_c \subset \mathcal{Y}_m$ ,  $c \in [K]$ , is exchangeable if  $\mathcal{Y}_c^\sigma =_D \mathcal{Y}_c$ , for all permutations  $\sigma : \mathcal{P} \rightarrow \mathcal{P}$  where  $=_D$  denotes equality in distribution. We say an interaction process is cluster edge exchangeable if this definition holds for all  $c \in [K]$ .

Under our model assumption, given the nodes  $S_j$  and  $R_j$ , and the corresponding cluster labels, the interaction  $E_j$  is a Bernoulli random variable, with  $\mathbb{E}(E_j)$  being specified by  $C_j^{(s)}$  and  $C_j^{(r)}$ ,

denoted as  $\mathbb{E}(E_j) = \mathcal{B}(C_j^{(s)}, C_j^{(r)})$ , where  $\mathcal{B}$  is a  $K \times K$  propensity matrix. In the remainder of this paper,  $\mathcal{B}(c, )$  is used to indicate the  $c$ th row of  $\mathcal{B}$ ,  $c \in [K]$ .

## 2.3 Sequential description of the C-VCM model

We next provide a sequential description of a particular family of relatively exchangeable interaction processes. For ease of comprehension, the process is described in the context of TalkLife posts assuming a single commentator. Suppose  $m$  posts  $E_{[m]} := \{E_1, \dots, E_m\}$  have been observed along with the cluster assignments for each observed individual. That is, the post  $j \in [m]$  is given by  $E_j = (\{S_j, C_j^{(s)}\}, \{R_j, C_j^{(r)}\})$  where  $S_j$  is the sender and  $R_j$  is the commentator,  $S_j, R_j \in \mathbb{N}$ , and the cluster assignments are given by  $C_j^{(s)}$  and  $C_j^{(r)}$ , where  $C_j^{(s)}$  and  $C_j^{(r)} \in [K]$ . Let  $\mathcal{H}_m^{(c)}$  denote the set of unique individuals in cluster  $c \in [K]$  that have either posted or commented on a post in the first  $m$  interactions.

The next post,  $E_{m+1}$ , conditional on  $E_{[m]}$  and the associated cluster assignments are constructed by first drawing a cluster, denoted  $C_{m+1}^{(s)}$ , according to

$$P(C_{m+1}^{(s)} = c | \pi_1, \dots, \pi_K) = \pi_c, \text{ for } c \in [K] \quad (1)$$

where  $\pi_c$  is the pre-specified propensity of an interaction initiated from the cluster  $c$ , such that  $\sum_{c=1}^K \pi_c = 1$ . Note that each individual per cluster can be involved in multiple interactions, either as a poster or commentator. Let  $D_m(i, c)$  be the number of times individual  $i \in \mathcal{H}_m^{(c)}$  in cluster  $c$  has been observed (either has poster or commentator) in the first  $m$  interactions. Similar to the Chinese Restaurant Process [16], given parameters  $0 < \alpha_c < 1$  and  $\theta_c > -\alpha_c$  for each  $c \in [K]$ , poster  $S_{m+1}$  can be either chosen from one of the previously observed nodes  $i$  in cluster  $c$ , or an unobserved node in cluster  $c$  with probability:

$$P(S_{m+1} = s | \mathcal{H}_m, C_{m+1}^{(s)} = c) \propto \begin{cases} D_m(i, c) - \alpha_c, & \text{if } s = i \in \mathcal{H}_m^{(c)} \\ \theta_c + \alpha_c \sum_{i \in \mathcal{H}_m^{(c)}} D_m(i, c), & \text{if } s \notin \mathcal{H}_m^{(c)} \end{cases} \quad (2)$$

Given the sender, a similar process is used to choose the  $(m+1)$ st commentator. Suppose

the within and between cluster propensity of link is known. That is, the entries in the propensity matrix  $\mathcal{B}$  are given. The cluster assignment for  $R_{m+1}$ , denoted  $c' = C_{m+1}^{(r)} \in [K]$ , is given by:

$$P(C_{m+1}^{(r)} = c' | C_{m+1}^{(s)} = c, \mathcal{B}) = \mathcal{B}(c, c')$$

Let the set of previously observed nodes in cluster  $c'$  in the first  $m$  interactions be  $\mathcal{H}_m^{(c')}$ . Note that if  $c' = c$ ,  $\mathcal{H}_m^{(c')} = \mathcal{H}_m^{(c)} \cup \{s\}$ , and  $D_m(s, c') = D_m(s, c) + 1$ . Let  $R_{m+1}$  be the commentator of the  $(m+1)$ st interaction. Similar to the process of selecting the sender  $S_{m+1}$ , conditional on  $\mathcal{H}_m^{(c')}$ , the receiver node  $R_{m+1}$  is selected with probability whose explicit form is similar to that of Eq. 2

Let  $\mathbf{C}_m$  denote the cluster assignments of all observed nodes in the first  $m$  interactions. Based on the above sequential description, given the total number of observed interactions being  $m$ , the proposed cluster exchangeable interaction process (C-VCM) takes the explicit form:

$$P(\mathcal{Y}_m = \mathbf{y}_m | \{\alpha_c, \theta_c\}, \mathbf{C}_m) = \underbrace{\prod_{c=1}^K \pi_c^{L_c}}_{(I)} \times \underbrace{\frac{[\alpha_c + \theta_c]_{\alpha_c}^{v(\mathbf{y}_c)-1}}{[\theta_c + 1]_1^{m(\mathbf{y}_c)-1}} \prod_{j \in \mathcal{H}_m^{(c)}} [1 - \alpha_c]_1^{D_m(j,c)-1}}_{(II)} \times \underbrace{\prod_{c'=1}^K \mathcal{B}(c, c')^{W_m(c, c')}}_{(III)} \quad (3)$$

where  $L_c$  is the number of observed interactions that are initiated from cluster  $c$ ;  $W_m(c, c')$  is the number of observed interactions from cluster  $c$  to  $c'$ . Part (I) of Eq. 3 is the probability of an interaction initiating from cluster  $c$ . Part (II) of Eq. 3 is the joint distribution of all the nodes within the same cluster  $\mathcal{H}_m^{(c)}$  which can be expressed explicitly as [16]:  $P(\mathcal{H}_m^{(c)} | \alpha_c, \theta_c, \{\mathbf{C}_m\}) = \frac{[\alpha_c + \theta_c]_{\alpha_c}^{v(\mathbf{y}_c)-1}}{[\theta_c + 1]_1^{m(\mathbf{y}_c)-1}} \prod_{j \in \mathcal{H}_m^{(c)}} [1 - \alpha_c]_1^{D_m(j,c)-1}$ , where for real number  $x$  and  $a$ , and non-negative integer  $N$ , the operation on  $x$ ,  $[x]_a^N = x(x+a)\dots(x+(N-1)a)$ . Part (III) of Eq. 3 is the propensity of connection between cluster  $c$  and  $c'$ .

## 2.4 Statistical Properties

Given the sparsity of the network is highlighted in our model, one of the key issues to be addressed is whether the sparsity holds in both the sub-networks and the entire network. With the presence of the latent community structure, we argue that for the mixture of sub-networks following the

power-law distributions parameterized by different values, the overall sparsity still holds in the entire network with minor restrictions on the propensity matrix.

**Theorem 2.4 (Sparsity)** *Let  $K$  be the number of clusters in the graph, such that  $\forall c \in [K]$ :*

$$\limsup_{m(\mathcal{Y}_c) \rightarrow \infty} \frac{m(\mathcal{Y}_c)}{v(\mathcal{Y}_c)^2} = 0$$

*That is, the sparsity in each sub-network. Let  $a$  and  $b$  denote the diagonal and off-diagonal elements of the propensity matrix. Assume  $p$  and  $q$  are the same across clusters, and  $a = \mu Kb$  ( $a > b$ ), where  $\mu$  is a normalizing constant that doesn't change with respect to the number of clusters, then*

$$\limsup_{m(\mathcal{Y}_m) \rightarrow \infty} \frac{m(\mathcal{Y}_m)}{v(\mathcal{Y}_m)^2} = 0$$

*That is, the overall sparsity of the entire network.*

Another important question in the community detection is whether the latent community structure can be recovered. The theoretical guarantee of the various implementations of SBM and DCSBM has been well established. For example, the general theory for checking consistency of community detection under DCSBM and SBM and the comparison of several criteria are shown in [28]; the minimax optimal rate of community detection is discussed in ([7], [6]). Here, we follow the logic of proof as shown in the pseudo-likelihood methods for community detection [2] and show that as the total degrees of network increases, the mis-specification rate will be bounded.

**Theorem 2.5 (Consistency)** *Given the power-law parameters  $\alpha_c$  and  $\theta_c$  for all  $c \in [K]$ , let  $v(\mathcal{Y}_m)$  be the number of nodes in the network;  $m(\mathcal{Y}_m)$  be the total degree of the graph; and  $\gamma m(\mathcal{Y}_m)$  be the exactly number of node in community 1 under the labeling  $e$  of all the possible permutations. Assume there are two underlying clusters in the network and  $\alpha_1 = \alpha_2$  and  $\theta_1 = \theta_2$ .*

*Let  $h(\gamma)$  be the binary entropy function of  $\gamma$  and  $\kappa_\gamma(n) = \frac{1}{n} [\log \frac{n}{4\pi\gamma(1-\gamma)} + \frac{1}{3n}]$ . Denote  $M_{v(\mathcal{Y}_m)} = \frac{1}{v(\mathcal{Y}_c)} \sum_{i=1}^{v(\mathcal{Y}_c)} 1(\hat{C}^{(i)} \neq C^{(i)})$ . Then, as  $m(\mathcal{Y}_m) \rightarrow \infty$ , for a sequence  $\mu_n \geq \frac{1}{e}$ , we have:*



$$\mathbb{P}[\sup_e M_{v(\mathcal{Y}_m)} \geq \frac{4h(\gamma)}{\log \mu_n}] \leq \exp[-2v(\mathcal{Y}_m)(h(\gamma) - \kappa_\gamma(2v(\mathcal{Y}_m)))]$$

where  $\hat{C}^{(i)}$  and  $C^{(i)}$  stands for the inferred cluster assignment of the node  $i$  in the network and the underlying truth.

### 3 Inference

In this section, we are going to demonstrate the efficacy of our algorithm regarding the inference of network structures. We start with a brief introduction of the inference algorithm, and proceed into the accurate estimate of the power-law parameters and the recoverability of the cluster assignment. At the end of this section, we propose one potential model selection criteria based on the marginal log likelihood.

#### 3.1 A Gibbs-sampling based inferential algorithm

Unfortunately, the likelihood shown in Eq.3 does not have closed form solutions for the parameters  $\{\alpha_c\}$ ,  $\{\theta_c\}$ , and  $\{\mathbf{C}_m\}$ . Here, we present a Bayesian approach to infer these model parameters. We start with the selection of the priors. First, consider the cluster assignment  $\mathbf{C}_m$ . Recall that the probability of an interaction initiating from a specific cluster is given by  $\pi$ . Here, we assume a Dirichlet prior on  $\pi$ , such that  $\pi \sim \text{Dirichlet}(\eta)$ . Without strong prior information, we'd suggest  $\eta$  being a symmetric Dirichlet.

As of the connection between and within the cluster, a node has to be connected to another node either in the same cluster or in the other cluster. Hence, the entries in the same row of propensity matrix  $\mathcal{B}$  should sum up to 1. Without forcing the propensity matrix to be symmetric, we proposed here a row-wise Dirichlet prior, such that  $P(\mathbf{B}(c, \cdot)) \sim \text{Dirichlet}(\nu)$ ,  $c \in [K]$

In terms of the priors of power-law parameters  $\{\alpha_c\}$  and  $\{\theta_c\}$ , let  $\alpha_c \sim P(\mu)$ ,  $\theta_c \sim P(\delta)$ ,  $c \in [K]$ . A reasonable choice could be the Beta distribution and the Gamma distribution correspondingly such that the posterior distribution will have the close form to sample from.

Given the specified priors, we introduce the algorithm built upon Gibbs sampling to infer the

model parameters. Based on the description of our model, the parameters to be updated are the power-law parameters  $\{\alpha_c\}$ ,  $\{\theta_c\}$ , and the latent clustering assignment indicator  $\{\mathbf{C}_m\}$ . The description of the algorithm is shown as follows:

---

**Algorithm 1** Pseudo-code of Gibbs-sampling based algorithm

---

```

Specify  $\eta, \nu, \sigma, \mu, K$ 
Initiate  $\mathcal{B}, \{\alpha_c\}, \{\theta_c\}, \{\mathbf{C}_m\}$ 
for iterations do
  for  $s \in v(\mathcal{Y}_m)$  do
    Update  $C_m^{(s)}$  according to Eq (4)
  end for
  for  $c \in [K]$  do
    Update  $\alpha_c$  according to Eq (6)
    Update  $\theta_c$  according to Eq (5)
  end for
  for  $c \in [K]$  do
    Update  $\mathcal{B}(c, \cdot)$  from the Dirichlet distribution
  end for
end for

```

---

The first parameter to be updated is  $\{\mathbf{C}_m\}$ . The conditional updates of latent cluster indicator  $\{\mathbf{C}_m\}$  is based on the following derivations. Suppose the algorithm is assigning a specific node  $s \in \{\mathcal{H}_m\}$  to a specific cluster  $c \in \{1, \dots, K\}$ . The corresponding probability  $p_c$  is proportional to:

$$p_c \propto \underbrace{P(\{C_m^{(s)}\} = c | \eta)}_{(I)} \times \underbrace{P(\mathcal{B}(C_m^{(s)} = c, \cdot) | \nu)}_{(II)} \times \underbrace{P(\{\mathcal{H}_m^{(c)}\} | \alpha_c, \theta_c, \delta, \mu)}_{(III)}$$

The first term (I) can be achieved through integrating the latent parameter  $\pi$  from  $P(\{C_m^{(s)}\} | \pi, \eta)$  in a way that

$$P(\{C_m^{(s)}\} | \eta) = \int_{\pi} P(\{C_m^{(s)}\} | \pi) P(\pi | \eta) d\pi = \frac{\mathcal{B}(\eta + L)}{\mathcal{B}(\eta)}$$

where  $L = \{L_1, \dots, L_K\}$  is the counts of the interactions initiated from cluster 1 to K, and  $\mathcal{B}(\cdot)$  is the Beta function.

The second term II can be simplified by considering only the nodes that have connections with the node  $s$  that is updated in the current iteration. Let  $D_m(s, c)$  be the degree of node  $s$ , and  $D_m(s, 1)$  to  $D_m(s, K)$  be the number of interactions node  $s$  has with nodes in cluster 1 to K

correspondingly:

$$P(\mathcal{B}(C_m^{(s)} = c, )|\nu) \propto \frac{D_m(s, c)!}{D_m(s, 1)! \dots, D_m(s, K)!} \prod_{c'=1}^K \mathcal{B}(c, c')^{D_m(s, c')}$$

The third term *III* takes the explicit form as shown in Eq (4). Normalizing over different  $p_c$  over  $c \in [K]$ , we get the conditional probability of assigning the node to cluster  $c$ :

$$P(C_m^{(s)} = c|\alpha_c, \theta_c, B, \eta, \mu, \delta) \sim \text{Multinomial}(\frac{p_c}{\sum_{k=1}^K p_k}) \quad (4)$$

The conditional updates of  $\alpha_c$  and  $\theta_c$  is shown as follows. Given the cluster allocation  $\{\mathbf{C}_m\}$ , the values of  $\alpha_c$  and  $\theta_c$  can be sampled from the posterior distribution. First, consider the  $\theta_c$ :

$$P(\theta_c|\{\mathbf{C}_m\}, \delta) \propto \frac{[\theta_c + \alpha_c]_{\alpha_c}^{v(\mathbf{y}_c)-1}}{[\theta_c + 1]_1^{m(\mathbf{y}_c)-1}} P(\theta_c|\delta)$$

Here, we utilize a similar idea of sampling auxiliary variable methods [23] that was also introduced in the hierarchical e2 model [5]. Let  $\{\alpha_c^*\}$  and  $\{\theta_c^*\}$  be the values of the power-law parameters from the previous Gibbs updating iteration, and  $\{a, b, c, d\}$  be the pre-specified constants. Then, the denominator of the above equation can be written as  $([\theta_c + 1]^{m(\mathbf{y}_c)-1})^{-1} = \Gamma(m(\mathbf{y}_c) - 1)^{-1} \int_0^1 x^{\theta_c} (1-x)^{m(\mathbf{y}_c)-2} dx$ , where  $x$  is an intermediate variable that can be sampled given  $\{\theta_c^*\}$ , such that  $x \sim \text{Beta}(\theta_c^* + 1, m(\mathbf{y}_c) - 1)$ . Similarly, the nominator  $[\theta_c + \alpha_c]_{\alpha_c}^{v(\mathbf{y}_c)-1} = \prod_{i=1}^{v(\mathbf{y}_c)-1} \sum_{y_i \in [0,1]} \theta_c^{y_i} (\alpha_c * i)^{1-y_i}$  where  $y_i \sim \text{Bernoulli}(\frac{\theta_c^*}{\theta_c^* + \alpha_c^* * i})$ . Eventually, the posterior of  $\theta_c$  is given by

$$\theta_c|C_m \sim \text{Gamma}(\sum_{i=1}^{v(\mathbf{y}_c)-1} y_i + a, b - \log x) \quad (5)$$

Similarly, the posterior of  $\alpha_c$  is given by

$$\alpha_c|C_m \sim \text{Beta}(c + \sum_{i=1}^{v(\mathbf{y}_c)-1} (1 - y_i), d + \sum_{s_c} \sum_{j=1}^{D_m(c,s)-1} (1 - z_{s_c,j})) \quad (6)$$

where  $y_i \sim \text{Bernoulli}(\frac{\theta_c}{\theta_c + \alpha_c * i})$  and  $z_{s_c,j} \sim \text{Bernoulli}(\frac{j-1}{j-\alpha_c^*})$ .

To summarize, the algorithm first draws the values of intermediate variables  $x$ ,  $y$ , and  $z$  based

on the previous round values of  $\alpha_c^*$  and  $\theta_c^*$ . Given the values of these variables, we draw the values of the updated  $\alpha_c$  and  $\theta_c$  according to Eq.5 and Eq.6.

Last but not least, given the cluster allocation  $c \in [K]$ , the row-wise update of the propensity matrix will be from the Dirichlet distribution given the prior.

### 3.2 Power-law parameter inference

An important question is whether the posterior distributions adequately concentrate around the true cluster-specific parameters  $\{\alpha_c\}_{c \in [K]}$  which controls the within-cluster sparsity and power-law structure. Here, a generative model as described in Section 2.3 is used to simulate various networks to assess our ability to accurately estimate these parameters.

Here we focus on simulations where the number of clusters  $K$  is equal to 2. In this setting, we choose  $\{\alpha_1, \alpha_2\}$  from  $\{\{0.1, 0.9\}, \{0.2, 0.8\}, \{0.3, 0.7\}, \{0.4, 0.6\}\}$  and set  $\theta_1 = \theta_2 = 5$ . The within-group interaction propensity is set to 0.9 (i.e.,  $a := \mathcal{B}(1, 1) = \mathcal{B}(2, 2) = 0.9$ ), and between-group is set to 0.1 (i.e.,  $b := \mathcal{B}(1, 2) = \mathcal{B}(2, 1) = 0.1$ ). Clusters are assumed equally likely to initiate an interaction (i.e.,  $\{\pi_1, \pi_2\} = \{0.5, 0.5\}$ ). Networks of size 1,000, 10,000 and 100,000 were simulated, repeated 20 times. Given the priors specified in the previous section, the means of the posterior of the power-law parameters and the entries in propensity matrix are shown in Table 1. To assess uncertainty due to latent communities, we also infer power-law parameters given true cluster assignments, denoted  $\tilde{\alpha}$ . The results are shown in Table 1.

### 3.3 Recovering Cluster Assignment Labels

We show in Theorem 2.5 the mis-specification rate of C-VCM can be bounded. Here, we show in simulation that the posterior concentrates on the correct cluster assignment labels. The same simulation settings are inherited from the previous section, and are used to empirically confirm the ability of our algorithm to recover the underlying network structure when only a finite number of interactions are observed. The cross entropy loss is used to measure the misclassification loss. We

start by defining the cross entropy loss for a specific cluster  $c \in [K]$ :

$$\mathcal{L}_c = \sum_{j \in \mathcal{H}_m^{(c)}} -p(C_j = c) \log q(\hat{C}_j = c)$$

where the  $p(\cdot)$  is the true cluster assignment, and  $q(\cdot)$  is the posterior mean of the inferred cluster assignment. Further define the average per node entropy loss  $\mathcal{L}_s = \frac{1}{v(y_m)} \sum_{c=1}^K \mathcal{L}_c$ ,

The results are shown in Table ??.

### 3.4 The selection of K

So far, we have assumed K is known and set to be 2. It remains to be addressed the selection of K. The well-acknowledged solutions include but are not limited to the likelihood ratio test based methods ([25], [26]), the cross validation based methods ([9],[8]), and the Bayesian selection criteria ([27],[22]). Though proven to be a big success in many cases, these methods cannot fit into our settings well. The likelihood ratio test based method requires a good distributional approximation, which is hard to find in our setting; and cross validation based methods bring the even more complex sampling issues and the lack of the interpretation; and the implementation of Bayesian selection criteria can be very computationally expensive.

Here, we consider a model selection criteria based on the marginal posterior maximization that is similar to the one used in the topic models[21]. Based on this criteria, the number of clusters is determined by choosing the K that gives the largest marginal likelihood  $P(\mathbf{y}_m | \{\alpha_c, \theta_c\}, \mathbf{C}_m, K)$ . Simulation results show solid support to our proposed method. Future work is required for a formal study of the model selection criteria in the cluster edge exchangeable model.

## 4 Case Study

### 4.1 Data overview

In this section, we apply the C-VCM to the TalkLife dataset which consists of millions of user posts and comments. Here we consider all non-deleted posts on TalkLife during the year 2019, consisting

of 1,508,895 posts with an average of 2.82 comments per post, leading to 4,258,792 post-comment pairs. TalkLife deploys a series of classification algorithms to flag posts if they include language related to one of several mental health topics. For example, a post can be flagged as *Anxiety Panic Fear Suspected* if the corresponding classifier was triggered by the text of the post. There are a total of 33 classifiers applied to the majority of posts. Figure 2 shows the global degree distribution for posts, and the degree distribution for posts flagged by 4 out of 33 different classifiers

## 4.2 Inferring Cluster labels

In this section, we aim to identify the latent community structure of TalkLife users. In the running example shown in Section 2.2, the model only considers pairs where each user only compose a single poster. In TalkLife, however, the data consists of multiple posters posted by a single user. Here, we make an independence assumption of different posters posted by the same user.

In the following analysis, we focus on users that pay attention to posts of similar themes. We subset the data based on these machine learning generated tags. Given the network built from posts with a particular machine learning tag, we run the Gibbs-sampling algorithm to learn the latent community structure. We consider a varying number of clusters and infer the cluster assignments of each user over a range of  $K$  values.

We further investigate the within and the between cluster connectivity of the inferred communities. Using Alcohol and Substance Abuse network as an example, the count of the interactions when  $K$  is set to be 2, 6, and 10 is shown in Figure 3. It is consistently witnessed that the within community connectivity is greater than the between community connectivity based on the results of our method, which indicates the existence of the clustering structures of the users.

## 4.3 Comparison with Spectral Clustering

In the section, we further compare our method with the spectral clustering method ([24],[14]). To implement the spectral clustering method, we constructed a binary graph. The connectivity of the communities detected by the spectral clustering method is shown in Figure 3. The spectral clustering method gives different community structures as compared to our method. The overlapping

of the nodes in the largest group between our method and Spec Clus range from 36% to 48% over different  $K$  values in Alcohol and Substance Abuse network, for example. Also note there is little variations in the nodes clustered to the largest two groups by spectral clustering. This makes the communities detected by the spectral clustering method less informative than our method.

To further assess the quality of the inferred latent community structure, we check the consistency of the clusters inferred by our algorithm over time. We split the data into two halves based on whether the post occurred before June 30th, and infer the latent community structure within each time window using our proposed approach and spectral clustering. The Hellinger distance is used to quantify the difference between the cluster assignments, which is defined as:

$$HL = \frac{1}{v(\mathbb{Y}_m)} \sum_{i=1}^{v(\mathbb{Y}_m)} \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^K (\sqrt{p_{ki}} - \sqrt{q_{ki}})^2}$$

where  $p$  and  $q$  correspond to the probability of a node belonging to one of the  $K$ th clusters given by the Gibbs iterations and the underlying truth. To address the label switching problem, we take the minimal Hellinger distance as the final one.

For illustrative purposes, we use the *Alcohol and Substance Abuse Suspected* and *Behavioral Symptoms Suspected* networks as our two running examples. The results are shown in [Figure 4](#). In the Behavioral Symptoms Suspected network, the marginal likelihood  $P(\mathbf{Y}|m|K, \{\alpha_c\}, \{\theta_c\}, \mathbb{C}_m)$  hits the largest value when  $K = 4$ , which indicates setting the underlying communities to be 4 is the most reasonable grouping of users. In the both networks, the Hellinger Distance of clustering assignments of our model is consistent with the marginal likelihoods. It serves to prove the efficacy of the heuristic strategy mentioned in [Section 3.4](#) in real data.

We also compare our method with the spectral clustering method. Due to uncertainty in  $K$ , we look at a range of values. The minimal Hellinger Distance of the cluster assignment is smaller in our model than in the spectral clustering, which indicates the communities detected by our method are more consistent over time than the communities detected by the spectral clustering method. That is, the trend holds true over time that users tend to have more connections with people from the same inferred communities than people outside the inferred communities.

## 5 Discussion

In this paper, we have proposed a new method that can be applied to cluster the nodes and identify the underlying community structures of real world networks. Specifically, our method features incorporating the power-law property which is observed in the large sparse networks. We demonstrate the efficacy of our algorithm through simulation and finalize this paper by applying our method to the TalkLife data to identify the potential online user communities.

The contribution of our work is to account for the community structure when modeling the interaction process. Using TalkLife data as an example, our method can correctly identify the users' groups so that the platform will know to whom they can boost the visibility of the posts from certain users. Besides, our method provides the estimates of the power-law parameters for each of the group. The consistency of the communities detected by our method guarantees the effect of the interventions provided by the platform over time. The revelation of the network properties can also help the platform make proper decisions on what kind of interventions to make. For example, a smaller power-law parameter indicates a looser connection between users. The platform will need extra efforts to ensure certain users within the community getting enough attention and social supports. Though we demonstrate our method using the TalkLife data, it can be well applied to any sparse interaction data sets [19].



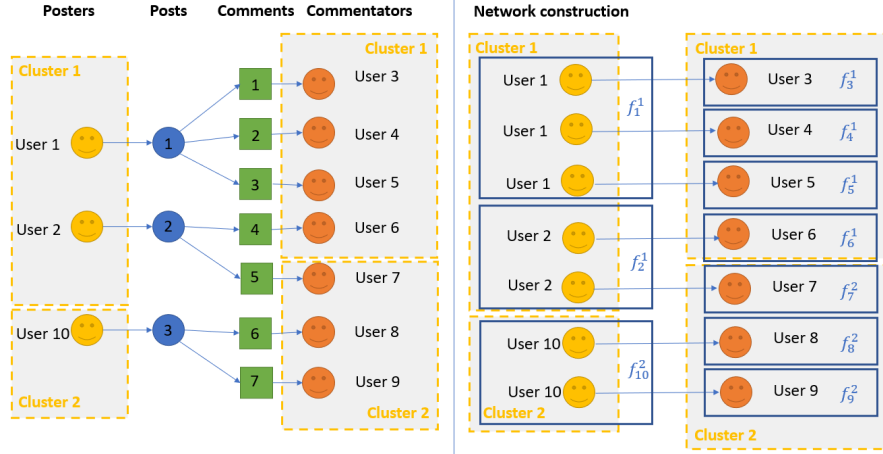
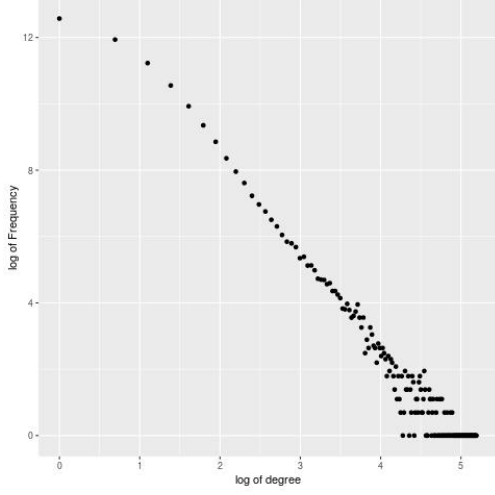
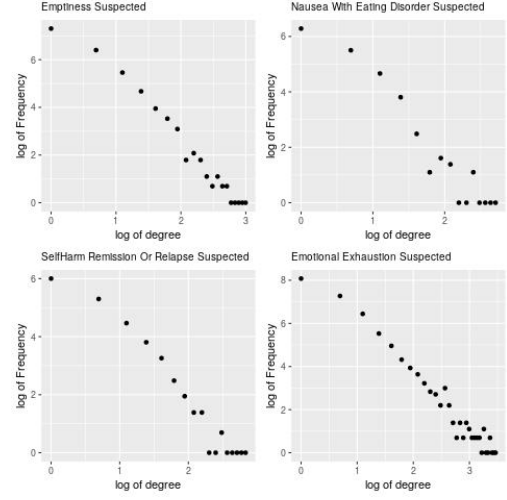


Figure 1: Interaction data collected on TalkLife (Left), and the corresponding network (Right). In this example, there are 3 posts and 10 users involved. Post 1 posted by User 1, is commented by User 3, 4, and 5; Post 2 posted by User 2 is commented by User 6 and 7; Post 3 posted by User 10 is commented by User 8 and 9. The network is constructed by connecting User 1 to User 3, 4, and 5; User 2 to User 6 and 7; User 10 to User 8 and 9. Note that User 1, 2, 3, 4, 5, 6 are from cluster 1, and the rest users are from cluster 2. Therefore, 4 of the interactions are from cluster 1 to cluster 1, 1 of them is from cluster 1 to cluster 2, the rest are from cluster 2 to cluster 2.

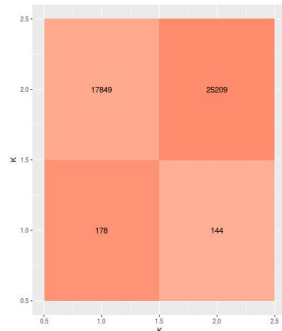


(a) Overall degree plot of 2019 senders



(b) Degree plot of 2019 senders by different tags

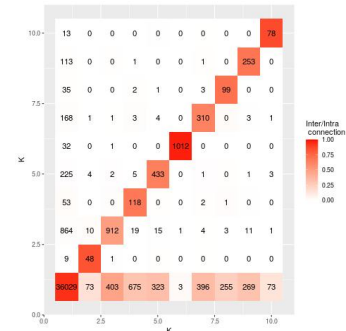
Figure 2: (a) Overview of the degree distribution of 2019 senders; (b) Degree distribution of the 2019 senders in sub-networks. The power-law degree distribution is observed in all the networks.



(a)



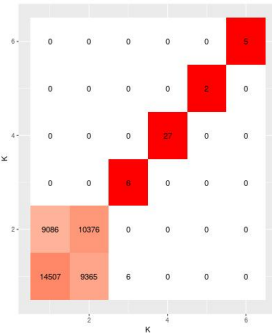
(b)



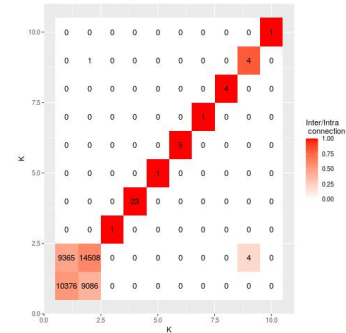
(c)



(d)



(e)



(f)

Figure 3: The inter/intra connectivity of the communities detected by our algorithm (the first line) spectral clustering (the second line), ranging from 0 to 1, indicating the proportion of the interactions that initiated from one cluster to the other. The number within each cell is count of the directed interactions.

<b>1,000 interactions</b>	Parameters	$\alpha = \{0.1, 0.9\}$	$\alpha = \{0.2, 0.8\}$	$\alpha = \{0.3, 0.7\}$	$\alpha = \{0.4, 0.6\}$
$\mathcal{B} = \{0.1, 0.9\}$	$\alpha_1$	<b>0.301 (0.182)</b>	<b>0.296 (0.116)</b>	<b>0.374 (0.08)</b>	<b>0.449 (0.083)</b>
	$\tilde{\alpha}_1$	0.200 (0.100)	0.274 (0.084)	0.354 (0.068)	0.460 (0.086)
	$\alpha_2$	<b>0.886 (0.079)</b>	<b>0.804 (0.022)</b>	<b>0.712 (0.037)</b>	<b>0.612 (0.056)</b>
	$\tilde{\alpha}_2$	0.904 (0.015)	0.805 (0.021)	0.714 (0.037)	0.604 (0.057)
	Diagonal	0.907 (0.020)	0.905 (0.019)	0.903 (0.019)	0.898 (0.018)
$\mathcal{B} = \{0.3, 0.7\}$	$\alpha_1$	<b>0.335 (0.196)</b>	<b>0.335 (0.139)</b>	<b>0.412 (0.143)</b>	<b>0.501 (0.099)</b>
	$\tilde{\alpha}_1$	0.226 (0.094)	0.256 (0.095)	0.384 (0.113)	0.470 (0.100)
	$\alpha_2$	<b>0.886 (0.069)</b>	<b>0.801 (0.029)</b>	<b>0.704 (0.061)</b>	<b>0.604 (0.064)</b>
	$\tilde{\alpha}_2$	0.901 (0.015)	0.806 (0.025)	0.712 (0.061)	0.617 (0.058)
	Diagonal	0.709 (0.031)	0.710 (0.028)	0.708 (0.030)	0.708 (0.034)
$\mathcal{B} = \{0.5, 0.5\}$	$\alpha_1$	<b>0.627 (0.162)</b>	<b>0.624 (0.185)</b>	<b>0.613 (0.115)</b>	<b>0.523 (0.124)</b>
	$\tilde{\alpha}_1$	0.212 (0.096)	0.290 (0.079)	0.368 (0.078)	0.479 (0.105)
	$\alpha_2$	<b>0.800 (0.267)</b>	<b>0.726 (0.125)</b>	<b>0.604 (0.112)</b>	<b>0.555 (0.106)</b>
	$\tilde{\alpha}_2$	0.898 (0.017)	0.804 (0.025)	0.710 (0.030)	0.601 (0.064)
	Diagonal	0.521 (0.042)	0.532 (0.051)	0.540 (0.053)	0.526 (0.050)
<b>10,000 interactions</b>	Parameters	$\alpha = \{0.1, 0.9\}$	$\alpha = \{0.2, 0.8\}$	$\alpha = \{0.3, 0.7\}$	$\alpha = \{0.4, 0.6\}$
$\mathcal{B} = \{0.1, 0.9\}$	$\alpha_1$	<b>0.201 (0.089)</b>	<b>0.273 (0.083)</b>	<b>0.336 (0.051)</b>	<b>0.426 (0.042)</b>
	$\tilde{\alpha}_1$	0.175 (0.063)	0.244 (0.063)	0.336 (0.043)	0.420 (0.039)
	$\alpha_2$	<b>0.899 (0.006)</b>	<b>0.799 (0.011)</b>	<b>0.700 (0.017)</b>	<b>0.605 (0.020)</b>
	$\tilde{\alpha}_2$	0.900 (0.006)	0.800 (0.011)	0.701 (0.017)	0.607 (0.019)
	Diagonal	0.900 (0.006)	0.900 (0.006)	0.901 (0.006)	0.900 (0.006)
<b>100,000 interactions</b>	Parameters	$\alpha = \{0.1, 0.9\}$	$\alpha = \{0.2, 0.8\}$	$\alpha = \{0.3, 0.7\}$	$\alpha = \{0.4, 0.6\}$
$\mathcal{B} = \{0.1, 0.9\}$	$\alpha_1$	<b>0.157 (0.063)</b>	<b>0.242 (0.054)</b>	<b>0.322 (0.034)</b>	<b>0.413 (0.025)</b>
	$\tilde{\alpha}_1$	0.136 (0.045)	0.220 (0.040)	0.319 (0.027)	0.412 (0.022)
	$\alpha_2$	<b>0.900 (0.002)</b>	<b>0.799 (0.004)</b>	<b>0.700 (0.006)</b>	<b>0.604 (0.012)</b>
	$\tilde{\alpha}_2$	0.900 (0.002)	0.799 (0.004)	0.700 (0.006)	0.605 (0.011)
	Diagonal	0.900 (0.002)	0.900 (0.002)	0.900 (0.002)	0.900 (0.002)

Table 1: Estimates of network parameters over different settings. First, estimation accuracy is higher when the networks are more loosely connected. For example, in the 1000 interactions setting where  $\{\alpha_1, \alpha_2\} = \{0.3, 0.7\}$  (highlighted in Table 1 in red), the power-law estimates when the truth is equal to 0.7 are more accurate than the estimates when the truth is equal to 0.3. This conclusion holds regardless of whether the community labels are known or unknown. Second, power-law estimation accuracy degrades as the inter-community connection rate increases. For example, in the 1000 interaction setting where  $\{\alpha_1, \alpha_2\} = \{0.2, 0.8\}$ , the power-law estimate decreases in accuracy as the likelihood of inter-community connections goes up from 0.1 to 0.5 (highlighted in Table 1 in yellow). Third, estimation accuracy increases as the number of interactions increase. For example, in the case where  $\alpha_1 = 0.1$ , and  $\{a, b\} = \{0.9, 0.1\}$ , the power-law estimate becomes more accurate as the sample size grows from  $N = 1,000$  to 100,000.

	Interactions	$\alpha = \{0.1, 0.9\}$	$\alpha = \{0.2, 0.8\}$	$\alpha = \{0.3, 0.7\}$	$\alpha = \{0.4, 0.6\}$
$\mathcal{B} = \{0.1, 0.9\}$	1,000	0.056 (0.014)	0.095 (0.028)	0.152 (0.028)	0.212 (0.040)
	10,000	0.014 (0.004)	0.039 (0.013)	0.094 (0.012)	0.170 (0.032)
	100,000	0.0027 (0.0006)	0.013 (0.004)	0.054 (0.012)	0.134 (0.018)
$\mathcal{B} = \{0.2, 0.8\}$	1,000	0.084 (0.020)	0.151 (0.032)	0.239 (0.065)	0.343 (0.050)
	10,000	0.019 (0.005)	0.055 (0.015)	0.157 (0.084)	0.298 (0.074)
	100,000	0.013 (0.033)	0.018 (0.004)	0.097 (0.040)	0.223 (0.029)
$\mathcal{B} = \{0.3, 0.7\}$	1,000	0.102 (0.152)	0.182 (0.050)	0.346 (0.100)	0.445 (0.061)
	10,000	0.052 (0.118)	0.078 (0.020)	0.210 (0.089)	0.411 (0.123)
	100,000	0.039 (0.086)	0.048 (0.060)	0.099 (0.016)	0.342 (0.124)
$\mathcal{B} = \{0.4, 0.6\}$	1,000	0.161 (0.082)	0.345 (0.117)	0.489 (0.097)	0.625 (0.057)
	10,000	0.120 (0.129)	0.279 (0.186)	0.311 (0.120)	0.546 (0.098)
	100,000	0.291 (0.202)	0.120 (0.125)	0.216 (0.118)	0.508 (0.103)
$\mathcal{B} = \{0.5, 0.5\}$	1,000	0.311 (0.202)	0.517 (0.110)	0.621 (0.052)	0.658 (0.067)
	10,000	0.254 (0.158)	0.291 (0.140)	0.434 (0.115)	0.711 (0.136)
	100,000	0.422 (0.107)	0.283 (0.146)	0.332 (0.140)	0.604 (0.049)

Table 2: Cross Entropy Loss over different settings. For fixed  $\alpha$ , the higher the intra-community connection probability, the better the recovery of the cluster assignments. Second, as the number of observations increases, the cluster assignments become more accurate in almost all settings. Third, fixing the number of interactions and inter-community connection rate, as the difference in power-law parameters decreases between the two clusters, the cluster assignment accuracy decreases. We do observe some instability when the propensity of connection is close to 0.5 (highlighted in blue)

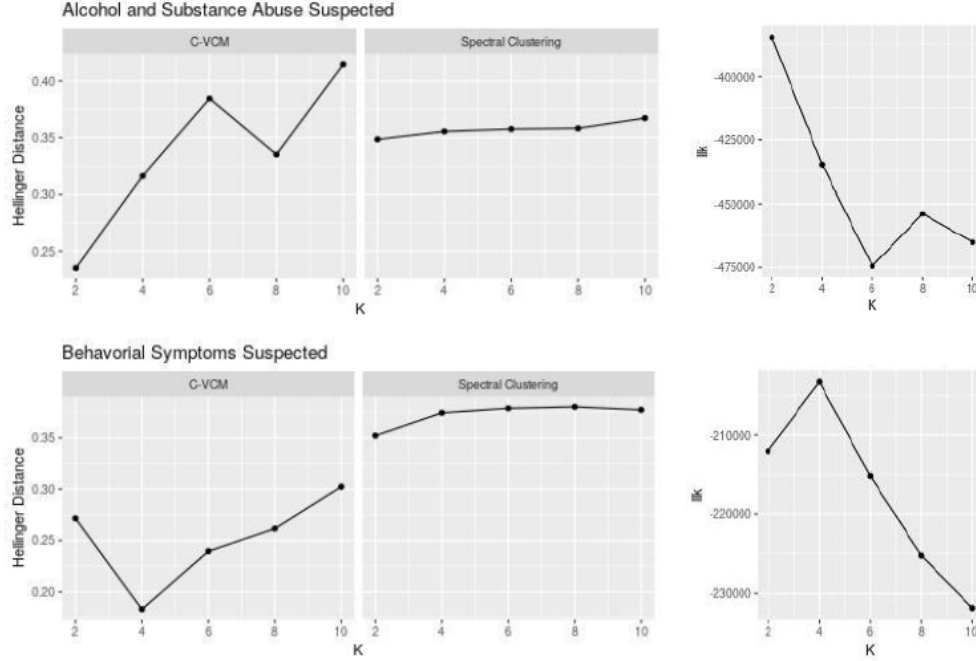


Figure 4: The Hellinger distances of the cluster assignment between the first half and the second half of the 2019 data of Alcohol and Substance Abuse suspected network and Behavioral Symptoms Suspected network.

## References

- [1] Lada A Adamic et al. “Search in power-law networks”. In: *Physical review E* 64.4 (2001), p. 046135.
- [2] Arash A Amini et al. “Pseudo-likelihood methods for community detection in large sparse networks”. In: *The Annals of Statistics* 41.4 (2013), pp. 2097–2122.
- [3] Peter Chin, Anup Rao, and Van Vu. “Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery”. In: *Conference on Learning Theory*. PMLR. 2015, pp. 391–423.
- [4] Harry Crane and Walter Dempsey. “Edge exchangeable models for network data”. In: *arXiv preprint arXiv:1603.04571* (2016).

- [5] Walter Dempsey, Brandon Oselio, and Alfred Hero. “Hierarchical network models for exchangeable structured interaction processes”. In: *Journal of the American Statistical Association* (2021), pp. 1–18.
- [6] Chao Gao, Yu Lu, and Harrison H Zhou. “Rate-optimal graphon estimation”. In: *The Annals of Statistics* 43.6 (2015), pp. 2624–2652.
- [7] Chao Gao et al. “Community detection in degree-corrected block models”. In: *The Annals of Statistics* 46.5 (2018), pp. 2153–2185.
- [8] Justin Grimmer. “A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases”. In: *Political Analysis* 18.1 (2010), pp. 1–35.
- [9] Bettina Grün and Kurt Hornik. “topicmodels: An R package for fitting topic models”. In: *Journal of statistical software* 40.1 (2011), pp. 1–30.
- [10] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. “Stochastic blockmodels: First steps”. In: *Social networks* 5.2 (1983), pp. 109–137.
- [11] Brian Karrer and Mark EJ Newman. “Stochastic blockmodels and community structure in networks”. In: *Physical review E* 83.1 (2011), p. 016107.
- [12] Evan M Kleiman and Richard T Liu. “Social support as a protective factor in suicide: Findings from two nationally representative samples”. In: *Journal of affective disorders* 150.2 (2013), pp. 540–545.
- [13] Morten Mørup and Mikkel N Schmidt. “Bayesian community detection”. In: *Neural computation* 24.9 (2012), pp. 2434–2456.
- [14] Andrew Y Ng, Michael I Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems*. 2002, pp. 849–856.

- [15] SL van der Pas and AW38078661407 van der Vaart. “Bayesian community detection”. In: *Bayesian Analysis* 13.3 (2018), pp. 767–796.
- [16] Jim Pitman et al. *Combinatorial stochastic processes*. Tech. rep. Technical Report 621, Dept. Statistics, UC Berkeley, 2002. Lecture notes for ..., 2002.
- [17] Maoying Qiao et al. “Adapting stochastic block models to power-law degree distributions”. In: *IEEE transactions on cybernetics* 49.2 (2018), pp. 626–637.
- [18] Karl Rohe, Sourav Chatterjee, and Bin Yu. “Spectral clustering and the high-dimensional stochastic blockmodel”. In: *The Annals of Statistics* 39.4 (2011), pp. 1878–1915.
- [19] Piotr Sapiezynski et al. “Interaction data from the copenhagen networks study”. In: *Scientific Data* 6.1 (2019), pp. 1–10.
- [20] David Strauss and Michael Ikeda. “Pseudolikelihood estimation for social networks”. In: *Journal of the American statistical association* 85.409 (1990), pp. 204–212.
- [21] Matt Taddy. “On estimation and selection for topic models”. In: *Artificial Intelligence and Statistics*. PMLR. 2012, pp. 1184–1193.
- [22] Yee W Teh, David Newman, and Max Welling. *A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation*. Tech. rep. CALIFORNIA UNIV IRVINE SCHOOL OF INFORMATION and COMPUTER SCIENCE, 2007.
- [23] Yee Whye Teh. “A Bayesian interpretation of interpolated Kneser-Ney”. In: (2006).
- [24] Hadrien Van Lierde, Tommy WS Chow, and Guanrong Chen. “Scalable spectral clustering for overlapping community detection in large-scale networks”. In: *IEEE Transactions on Knowledge and Data Engineering* 32.4 (2019), pp. 754–767.
- [25] Quang H Vuong. “Likelihood ratio tests for model selection and non-nested hypotheses”. In: *Econometrica: Journal of the Econometric Society* (1989), pp. 307–333.
- [26] YX Rachel Wang and Peter J Bickel. “Likelihood-based model selection for stochastic block models”. In: *The Annals of Statistics* 45.2 (2017), pp. 500–528.

- [27] Xiaoran Yan. “Bayesian model selection of stochastic block models”. In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2016, pp. 323–328.
- [28] Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. “Consistency of community detection in networks under degree-corrected stochastic block models”. In: *The Annals of Statistics* 40.4 (2012), pp. 2266–2292.