Model-based edge clustering

Daniel K. Sewell
Department of Biostatistics, University of Iowa

June 12, 2020

Abstract

Relational data can be studied using network analytic techniques which define the network as a set of actors and a set of edges connecting these actors. One important facet of network analysis that receives significant attention is community detection. However, while most community detection algorithms focus on clustering the actors of the network, it is very intuitive to cluster the edges. Connections exist because they were formed within some latent environment such as, in the case of a social network, a workplace or religious group, and hence by clustering the edges of a network we may gain some insight into these latent environments. We propose a model-based approach to clustering the edges of a network using a latent space model describing the features of both actors and latent environments. We derive a generalized EM algorithm for estimation and gradient-based Monte Carlo algorithms, and we demonstrate that the computational cost grows linearly in the number of actors for sparse networks rather than quadratically. We demonstrate the potential impact of our proposed approach on a patient transfer network, verifying these results by running simple epidemic simulations, and on a real friendship network amongst faculty members at a university in the United Kingdom.

Keywords: Community detection; Latent space models; Network analysis; Social networks

1 Introduction

Network analysis is an ever expanding analytical toolbox providing researchers the ability to study relational data to better understand and predict natural phenomena. An important facet of network data analysis is community detection, the unsupervised clustering of the actors in the network based on their relations, or edges. This topic is one of the most widely researched in the field of network analysis. An excellent introduction and survey of community detection is given in Fortunato (2010). This is by no means the only survey of community detection, and in fact there is even an increasing number of surveys of subtopics within community detection such as evolutionary algorithms (Cai et al., 2016; Pizzuti, 2018), dynamic networks (Enugala et al., 2015), online social networks (Dhumal and Kamde, 2015), and directed networks (Malliaros and Vazirgiannis, 2013).

While many algorithms and statistical models assume that actors belong to only one community, more sophisticated methods acknowledge this unrealistic constraint by allowing overlapping communities to exist. That is, an actor in the network may belong to multiple communities. While Xie et al. (2013) provides a survey of such overlapping community detection methods, the primary model-based overlapping community detection method is the mixed-membership stochastic blockmodel (MMSB) (Airoldi et al., 2008). It is easy to argue that the MMSB and other overlapping community detection methods present a much more realistic representation of observed networks than those approaches assigning a single cluster to each actor. Despite this, some authors and practitioners find difficulties in interpreting the output from such algorithms, which typically assign a "belonging factor" to each actor, i.e., a compositional vector to each actor describing the proportion to which that actor belongs to each community. For example, Ahn et al. (2010) (pg. S9) states,

> It is, however, often more natural to consider each node as a full member of its communities. A person's family would be disappointed if anyone proclaimed that he or she was only $1/5^{th}$ of a member of it; in the metabolic network, it would also be strange to say that H_2O was only $1/200^{th}$ a member of a given pathway.

This paper focuses not on clustering the actors of a network but on clustering the edges.

A simple method of clustering the edges would be to first cluster the actors and then group the edges according to the clusters of the incident actors. This, however, still relies on the premise that each actor belongs to a single community. In contrast, we believe that in many or perhaps most cases it can be assumed that if two actors form an edge it is because they have engaged or participated in a common activity or group. For example, an individual might be friends with one person because they work at the same company, while friends with another person because they both belong to the same sailing club. Kim and Jeong (2011) states (pg. 026110-1),

> From the theoretical point of view, the community of link could be more intuitive than the community of node in some real-world networks, because the link is more likely to have a unique identity while the node tends to have multiple identities.

In addition, it is very natural to think of temporary contexts in which multiple edges may form, e.g., several friendships which form at some one time event, whereas such an edge-inducing event does not have as clear an interpretation within frameworks which cluster the actors into communities.

Despite edge clustering being a very intuitive way of understanding the community structure of the network, significantly less work has been devoted to the area of edge clustering in comparison with traditional actor clustering approaches. Almost all of the extant edge clustering methods are algorithmic rather than model-based. One of the earliest approaches to clustering edges came from Pereira-Leal et al. (2004), in which a method was given to first construct a line graph[1] and then perform community detection on that line graph by optimizing the modularity. Similar approaches have been taken by constructing a line graph and applying standard (non-overlapping) community detection algorithms on it (Evans and Lambiotte, 2009, 2010; Wu et al., 2010). Ahn et al. (2010) first developed a somewhat different approach in which rather than building a line graph, a distance matrix is constructed on the edges of the network and standard clustering techniques are then

[1] A line graph can be thought of as reversing the roles of edges and actors; edges are (typically) connected in a line graph if they are incident on a common actor.

applied; genetic algorithms were applied to Ahn et al.'s approach by Shi et al. (2013) and Bello-Orgaz et al. (2018). Similar approaches were taken by Kim and Kim (2015); Deng et al. (2017); Zhang et al. (2018).

In contrast with algorithmic approaches, model-based community detection is more satisfactory and allows one to make future predictions, infer missing data, quantify the uncertainty of one's estimates, and use likelihood-based information criteria to choose the number of clusters. To the author's knowledge, there is only one model-based approach to edge clustering, namely that given in Ball et al. (2011). This approach uses a Poisson model for undirected and unweighted networks, modeling each actor's propensity to form edges of a certain type. The authors ignore some inherent constraints in the data; specifically they allow self-loops and multiple edges between a pair of actors, and in so doing gain considerable computational speed.

This paper develops statistical methods for model-based edge clustering. Detecting clusters of edges allows one to identify either systems of flows in the network or latent groups that facilitate edges. The latter case corresponds to scenarios in which there are latent activities, groups, places, or some other set of contexts in which edges form, as tends to be the case for, e.g., social networks. The former case corresponds to contexts in which information, disease transmission, or some other flow is of interest. In such a case, a network having community structure with respect to the edges can be viewed as a collection of systems of flows, where each system acts as a set of pipes (edges) which tend to feed into each other. To understand this analogy of pipes, one can imagine a set of pipes with water flowing through them, finding an access point in one of these pipes and pouring a dye inside. The idea is that the pipes belonging to the same system of flows as that of the initially accessed pipe will all soon be transporting the dyed water, whereas it would be much later before pipes belonging to different systems of flows would carry dyed water. That is, information/disease/etc. cascades quickly through edges within a system of flows due to the interconnectedness of the edges, while this cascade takes longer before reaching and flowing through edges in other systems. For graphical illustrations of these two edge clustering scenarios, see Figure 1.

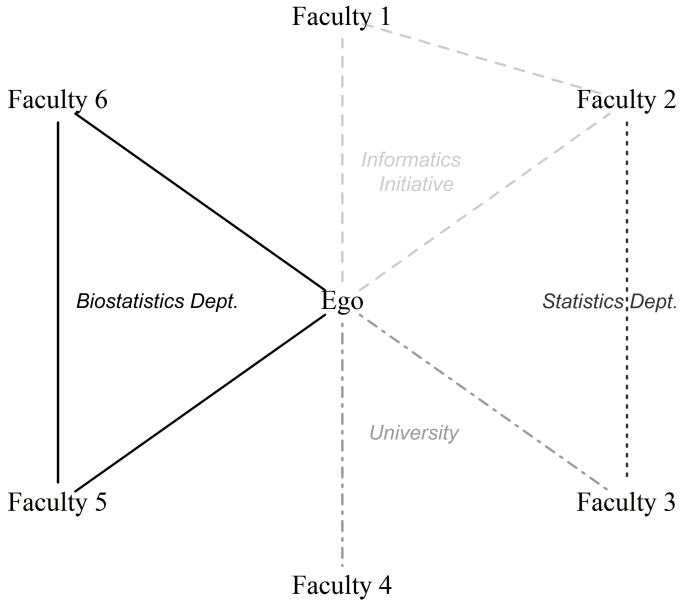
The contribution of this paper lies in the development of a network model designed to cluster edges and a scalable estimation algorithm[2]. We use a latent space approach, which has an intuitive motivation. Although there has been some promising work on scalability for latent space models (LSMs) (e.g., Raftery et al., 2012; Salter-Townshend and Murphy, 2013; O'Connor et al., 2015; Rastelli et al., 2018), LSMs are notoriously slow, often requiring long MCMC runs where the computational cost of each iteration is $\mathcal{O}(n^2)$, where n denotes the number of actors in the network. In contrast, the computational cost of the likelihood of our proposed model is linear in the number of edges, although care must be taken during estimation to keep the computational cost at this low level. We propose a fast generalized EM (GEM) algorithm, as well as guidance on gradient-based Monte Carlo algorithms, to perform estimation of the edge clustering model.

We describe our model, the model's motivation, and estimation in Section 2. In Section 3 we provide a simulation study to test various information criteria on selecting the dimension of the latent space and the number of clusters simultaneously, as well as to evaluate the overall accuracy of our estimation algorithm. In Section 4 we analyze two real datasets in order to illustrate the practical usefulness of our approach. The first is a patient transfer network among hospitals in California where we identify systems of flows of patients. We then run an epidemic simulation based on this empirical network demonstrating how identifying such systems of flows can help understand how an epidemic might spread. The second is a friendship network consisting of faculty members at a university in the United Kingdom demonstrating how our approach can be used to identify the environments in which friendships form. We then end with a discussion in Section 5.

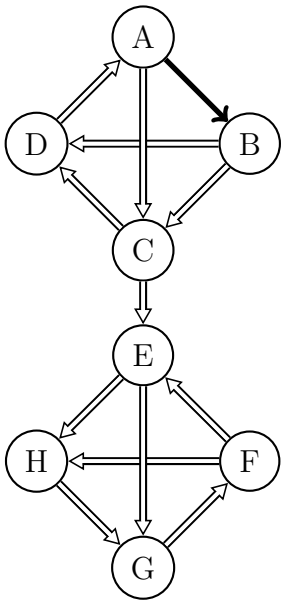
2 Method

A network is a mathematical representation of relational data consisting of a set of n actors $\mathcal{A} = \{1, 2, \ldots, n\}$ and a set of M edges $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$. Implicit in all statistical network models is that the likelihood implicitly conditions on the number of actors n. In our proposed model, we also implicitly condition on the number of edges M. This makes sense in the

[2]R code provided at `https://myweb.uiowa.edu/dksewell/Code.html`



(a) Illustration of latent contexts via an ego-network for a hypothetical Biostatistics faculty member at a research institution. The ego has collaborations with others in her department, with others due to participation in a university-wide informatics research initiative, and with still others due to belonging to the same university.



(b) Illustration of systems of flows. For example, if disease were to flow from A to B, we might soon see the disease continue to flow through, e.g., the edges (A,C), (B,C), (B,D), (C,D), and (D,A), but not through, e.g., (F,H).

Figure 1: Pedagogical illustrations of edge clustering.

context of edge clustering, as we only care to cluster those edges which were observed. In what follows, we assume that self-loops are invalid, i.e., $\mathcal{E} \subseteq \mathcal{S}_{\mathcal{A}} := \{(i,j) : i,j \in \mathcal{A}, i \neq j\}$. We also assume that the network is *directed*, i.e., the order of each element $\boldsymbol{e} = (i,j) \in \mathcal{E}$ is non-ignorable. That is, the relations have an inherent direction to them. Our model and approach could be easily adapted to handle undirected networks.

2.1 Model

Most statistical network models are *actor centric*, modeling the joint distribution of $\{A_{ij}\}_{(i,j)\in\mathcal{S}_{\mathcal{A}}}$, where A_{ij} equals one if $(i,j) \in \mathcal{E}$ and zero otherwise, i.e., the focus is on asking "given actors i and j, what is the probability that there is an edge between them?" In contrast, our approach is *edge centric*, modeling the joint distribution of $\{\boldsymbol{e}_m\}_{m=1}^{M}$, i.e., the focus is on asking "given we have an edge, what is the probability that it connects actors i and j?" The fundamental difference between these two paradigms is that which is viewed as the unit of observation. The actor centric view treats the dyads as the units of observation, leading to $n(n-1)$ observations of $\mathcal{S}_{\mathcal{A}}$, whereas the edge centric view treats the edges as the units of observation, leading to M observations of $\mathcal{E}$. Interestingly, a prominent and groundbreaking LSM paper using an actor centric statistical framework advocated for the idea of treating M as the number of observations in computing information criteria (Handcock et al., 2007). Most often networks are sparse, implying that $M << n(n-1)$, and in fact if we suppose that the average degree does not grow with n, then $M = \mathcal{O}(n)$. This immediately implies that the cost of evaluating the likelihood for an edge centric model will in many (or perhaps most) cases be linear in n rather than quadratic as is the case with actor centric models.

The main idea behind the model formulation is this: an edge taking place in a particular environment will most likely connect actors which share some features with that environment. As alluded to in the introduction, these environments may take two forms. First, the edges form systems of flows, where within each system the edges have a common set of features. These edges then connect actors which share these features in some way. For example, in a patient transfer network where the actors are the hospitals and the edges are the patients being shared, features of a system of flows may correspond to a geospatial region where patients live, a common health insurer, or a set of patient morbidities. Hospitals are likely to be linked through edges in a system due to belonging to that geospatial region, holding a contract with a common health insurer (i.e., being "in network" for a particular insurer), or sending/receiving patients due to an absence/presence of medical specialization. Second, a set of edges form due to the participants engaging in a common

latent group. For example, within a friendship network a hiking club may attract members who are environmentalists, enjoy exercise, etc. and through this hiking club many friendships may form amongst its members.

Let the edges be denoted as $\boldsymbol{e}_m = (\boldsymbol{e}_{m1}, \boldsymbol{e}_{m2})$ so that $\mathcal{E} = \{\boldsymbol{e}_m\}_{m=1}^M$. Let K be the number of latent classes of edges. Let $\boldsymbol{Z}$ be the $M \times K$ matrix such that $\boldsymbol{Z}_{mk}$ equals one if the m^{th} edge belongs to class k and zero otherwise. Let $\boldsymbol{U}$ be the $n \times p$ matrix such that the i^{th} row $\boldsymbol{U}_i$ is the i^{th} actor's latent sending features, similarly let $\boldsymbol{V}$ be the $n \times p$ matrix such that the i^{th} row $\boldsymbol{V}_i$ is the i^{th} actor's latent receiving features, and let $\boldsymbol{W}$ be the $K \times p$ matrix such that the k^{th} row $\boldsymbol{W}_k$ is the k^{th} edge class' latent features, where p is the number of latent features. Finally, let $\boldsymbol{S} = (S_1, \ldots, S_n)$ and $\boldsymbol{R} = (R_1, \ldots, R_n)$ be the actor specific overall propensities to send and receive edges respectively, thereby accounting for degree heterogeneity. Then the latent space edge clustering (LSEC) model is given by

$$
\begin{aligned}
\pi(\mathcal{E}|\boldsymbol{Z}) &= \prod_{m=1}^{M}\prod_{k=1}^{K}\left[\pi(\boldsymbol{e}_m|\boldsymbol{Z}_{mk}=1)\right]^{\boldsymbol{Z}_{mk}}, \\
&= \prod_{m=1}^{M}\prod_{k=1}^{K}\left[\pi(\boldsymbol{e}_{m1}|\boldsymbol{Z}_{mk}=1)\pi(\boldsymbol{e}_{m2}|\boldsymbol{e}_{m1},\boldsymbol{Z}_{mk}=1)\right]^{\boldsymbol{Z}_{mk}}, \\
\pi(\boldsymbol{e}_{m1}=i|\boldsymbol{Z}_{mk}=1) &= \frac{e^{S_i+\boldsymbol{U}_i\boldsymbol{W}_k'}}{f_{uk}}, \\
\pi(\boldsymbol{e}_{m2}=j|\boldsymbol{e}_{m1}=i,\boldsymbol{Z}_{mk}=1) &= \begin{cases} \frac{e^{R_j+\boldsymbol{V}_j\boldsymbol{W}_k'}}{f_{vk}-e^{R_i+\boldsymbol{V}_i\boldsymbol{W}_k'}} & \text{if } j \neq i \\ 0 & \text{otherwise,} \end{cases} \\
f_{uk} &= \sum_{i=1}^{n} e^{S_i+\boldsymbol{U}_i\boldsymbol{W}_k'}, \\
f_{vk} &= \sum_{i=1}^{n} e^{R_i+\boldsymbol{V}_i\boldsymbol{W}_k'}.
\end{aligned} \tag{1}
$$

Implicitly we are also always conditioning on $(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$, but for the sake of space we have omitted this dependence.

Note that in the same spirit as actor centric LSMs, the edges are conditionally independent given the latent features of the actors and edge classes. Given these latent features, the conditional probability of an edge m connecting two actors i and j follows a product of two multinomial distributions, where the source (from which the edge originates) of the

directed edge is chosen first from all n actors, and then given the source actor, the target actor (towards which the edge is directed) is chosen from the remaining $n-1$ actors.

We choose to work within a Bayesian framework, and hence we must place priors on the unknown quantities $(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{Z})$. Of course these priors will vary from researcher to researcher and from context to context, but the general form we assume in our estimation algorithm in Section 2.3 is

$$
\begin{aligned}
\boldsymbol{U}_i &\overset{iid}{\sim} N(\mathbf{0}, \tau_u I_p), & \tau_s &\sim \Gamma(a_s/2, b_s/2), \\
\boldsymbol{V}_i &\overset{iid}{\sim} N(\mathbf{0}, \tau_v I_p), & \tau_r &\sim \Gamma(a_r/2, b_r/2), \\
\boldsymbol{W}_k &\overset{iid}{\sim} N(\mathbf{0}, I_p), & \tau_u &\sim \Gamma(a_u/2, b_u/2), \\
S_i &\overset{iid}{\sim} N(0, \tau_s), & \tau_v &\sim \Gamma(a_v/2, b_v/2), \\
R_i &\overset{iid}{\sim} N(0, \tau_r), & \boldsymbol{\alpha} &\sim Dir(\alpha_0 \mathbb{1}_K), \\
\boldsymbol{Z}_m &\overset{iid}{\sim} Multinom(1, \boldsymbol{\alpha}), & &
\end{aligned} \tag{2}
$$

where $N(\mathbf{a}, \mathbf{B})$ denotes the multivariate normal distribution with mean $\mathbf{a}$ and precision matrix $\mathbf{B}$, $\Gamma(a, b)$ denotes the gamma distribution with shape a and rate b, $Multinom(n, \mathbf{a})$ denotes the multinomial distribution drawing n times from probability vector $\mathbf{a}$, $Dir(\mathbf{a})$ denotes the dirichlet distribution with shape parameters $\mathbf{a}$, and $\mathbb{1}_K$ denotes the K-dimensional vector of ones. Figure 2 gives a schematic for the LSEC model parameters.

The multinomial prior on the latent edge classes implies the marginal likelihood of the edges is the mixture distribution given by

$$
\pi(\mathcal{E}) = \prod_{m=1}^{M} \sum_{k=1}^{K} \alpha_k \frac{\exp\{S_{\boldsymbol{e}_{m1}} + R_{\boldsymbol{e}_{m2}} + (\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}})\boldsymbol{W}'_k\}}{f_{uk}\Big(f_{vk} - \exp\{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}}\boldsymbol{W}'_k\}\Big)}. \tag{3}
$$

The proposed LSEC model is an edge exchangeable network distribution restricted to binary edges (that is, not hypergraphs). This class of models formally introduced by Crane and Dempsey (2018) (see also, e.g., Broderick and Cai, 2015; Janson, 2018) treats the edges as the fundamental units of observation. A network $\mathcal{E} = \{\boldsymbol{e}_m\}_{m=1}^M$ based on a mapping $\phi_0 : \{1, 2, \ldots, M\} \mapsto \mathcal{A} \times \mathcal{A}$ (so $\boldsymbol{e}_m = \phi_0(m)$) is an edge exchangeable network if for any permutation $\sigma : \{1, \ldots, M\} \mapsto \{1, \ldots, M\}$, $\mathcal{E}_\sigma \overset{\mathcal{D}}{=} \mathcal{E}$, where the network $\mathcal{E}_\sigma = \{\boldsymbol{e}_{\sigma(m)}\}_{m=1}^M$

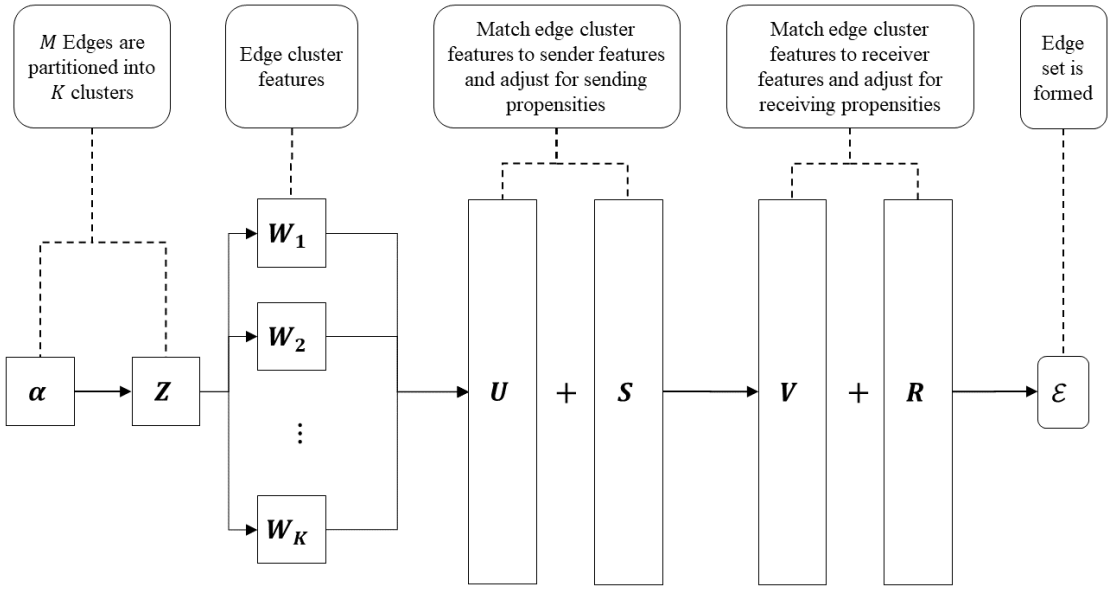


Figure 2: Schematic of the LSEC model.

is that which is induced by the mapping $\phi_1(m) := \phi_0(\sigma(m))$. Probability distributions are built for edge exchangeable networks by (for the binary edge context) placing a probability distribution over $\mathcal{A} \times \mathcal{A}$ and having each edge be an *iid* draw from this distribution.

2.2 Motivation by latent group participation

As can be seen in (1), the more similar an actor's features are to that of an edge class, the higher the likelihood that an edge of that class will involve that actor. For the case where these edge classes correspond to latent groups, we may go a step further and consider the binary participation of each of the actors. Suppose in a social network we knew which workplaces, social clubs, religious organizations, etc. to which each actor belonged, and we wished to model the probability that a friendship formed through the k^{th} such group connected actors i and j. It seems reasonable to model this probability as a function of i's overall tendency to send edges, call it $\widetilde{S}_i$, j's overall tendency to receive edges, call it $\widetilde{R}_j$, and whether or not actor i (j) participates in group k in a capacity which facilitates sending (receiving) edges, call it $\boldsymbol{Z}^u_{ik}$ ($\boldsymbol{Z}^v_{jk}$) taking values of 0 or 1. Then using a linear

model we would have

$$\mathbb{P}(\boldsymbol{e}_m = (i,j) | \boldsymbol{Z}^e_{mk} = 1) \propto e^{\eta_{k,ij}}, \tag{4}$$
$$\eta_{k,ij} := \widetilde{S}_i + \widetilde{R}_j + \beta_u \boldsymbol{Z}^u_{ik} + \beta_v \boldsymbol{Z}^v_{jk},$$

where $\boldsymbol{Z}^e_m$ is the 0-1 vector corresponding to the m^{th} edge class (what in the preceding sections was denoted simply as $\boldsymbol{Z}_m$), and β_u and β_v are regression coefficients. Now in practice we will not generally have this participation information. However, we can view participation in these latent groups as an unobserved bipartite network of actors and groups and employ a random dot product graph model (Young and Scheinerman, 2007). That is, the probability that actor i participates in group k as a sender is $\mathbb{P}(\boldsymbol{Z}^u_{ik} = 1) = \widetilde{\boldsymbol{U}}_i \widetilde{\boldsymbol{W}}'_k$ and as a receiver is $\mathbb{P}(\boldsymbol{Z}^v_{ik} = 1) = \widetilde{\boldsymbol{V}}_i \widetilde{\boldsymbol{W}}'_k$, which again follows from the same logic as before, that an actor which shares features similar to a latent group will be more likely to participate in that group.

Since we do not know $\boldsymbol{Z}^u_{ik}$ or $\boldsymbol{Z}^v_{jk}$, it seems reasonable to replace $\eta_{k,ij}$ with its expected value. In so doing we have

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{e}_m = (i,j) | \boldsymbol{Z}^e_{mk} = 1) \\
&\propto \exp\{\widetilde{S}_i + \widetilde{R}_j + \beta_u \widetilde{\boldsymbol{U}}_i \widetilde{\boldsymbol{W}}'_k + \beta_v \widetilde{\boldsymbol{V}}_j \widetilde{\boldsymbol{W}}'_k\} \\
&\propto \exp\Big\{ \Big(\widetilde{S}_i + \beta_u \widetilde{\boldsymbol{U}}_i \overline{\widetilde{\boldsymbol{W}}}'\Big) + \Big(\widetilde{R}_j + \beta_v \widetilde{\boldsymbol{V}}_j \overline{\widetilde{\boldsymbol{W}}}'\Big) \\
&\qquad + \beta_u \big(\widetilde{\boldsymbol{U}}_i - \overline{\widetilde{\boldsymbol{U}}}\big)\big(\widetilde{\boldsymbol{W}}_k - \overline{\widetilde{\boldsymbol{W}}}\big)' + \beta_v \big(\widetilde{\boldsymbol{V}}_j - \overline{\widetilde{\boldsymbol{V}}}\big)\big(\widetilde{\boldsymbol{W}}_k - \overline{\widetilde{\boldsymbol{W}}}\big)' \Big\} \\
&= \exp\big\{S_i + R_j + \boldsymbol{U}_i \boldsymbol{W}'_k + \boldsymbol{V}_j \boldsymbol{W}'_k\big\}
\end{aligned} \tag{5}$$

where

$$S_i := \widetilde{S}_i + \beta_u \widetilde{\boldsymbol{U}}_i \overline{\widetilde{\boldsymbol{W}}}', \qquad \boldsymbol{U}_i := \sqrt{\frac{\beta_u}{\beta_v}}\Big(\widetilde{\boldsymbol{U}}_i - \overline{\widetilde{\boldsymbol{U}}}\Big),$$

$$R_j := \widetilde{R}_j + \beta_v \widetilde{\boldsymbol{V}}_i \overline{\widetilde{\boldsymbol{W}}}', \qquad \boldsymbol{V}_j := \sqrt{\frac{\beta_v}{\beta_u}}\Big(\widetilde{\boldsymbol{V}}_j - \overline{\widetilde{\boldsymbol{V}}}\Big),$$

$$\text{and } \boldsymbol{W}_k := \sqrt{\beta_u \beta_v}\Big(\widetilde{\boldsymbol{W}}_k - \overline{\widetilde{\boldsymbol{W}}}\Big).$$

This then matches the LSEC model of (1).

2.3 Estimation

2.3.1 Generalized EM

We propose using a generalized EM algorithm (Dempster et al., 1977) to obtain the maximum a posteriori (MAP) estimates of the unknown quantities, where the E step is the expectation of $\boldsymbol{Z}|\mathcal{E}$, and the M step is the maximization with respect to $\boldsymbol{\theta} := \{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \tau_s, \tau_r, \tau_u, \tau_v, \boldsymbol{\alpha}\}$. The E step can be computed exactly. However, there are no closed form solutions for the M step. Instead we take a coordinate ascent approach, increasing the output of the E step (the expectation of the log posterior) incrementally rather than attempting to converge to the maximum at each iteration.

First, it should be noted that $\boldsymbol{Z}_m|\mathcal{E}, \boldsymbol{\theta}$ follows a multinomial distribution with probabilities proportional to

$$\frac{\alpha_k \exp\{S_{\boldsymbol{e}_{m1}} + R_{\boldsymbol{e}_{m2}} + (\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}})\boldsymbol{W}'_k\}}{f_{uk}\left(f_{vk} - \exp\{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}}\boldsymbol{W}'_k\}\right)}. \tag{6}$$

The complete likelihood is given as

$$\pi(\mathcal{E}, \boldsymbol{Z}|\boldsymbol{\theta}) = \prod_{m=1}^{M}\prod_{k=1}^{K}\left[\alpha_k \frac{\exp\{S_{\boldsymbol{e}_{m1}} + R_{\boldsymbol{e}_{m2}} + (\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}})\boldsymbol{W}'_k\}}{f_{uk}\Big(f_{vk} - \exp\{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}}\boldsymbol{W}'_k\}\Big)}\right]^{\boldsymbol{Z}_{mk}}, \tag{7}$$

and hence, letting $\widetilde{\boldsymbol{\theta}}$ denote the current estimate of $\boldsymbol{\theta}$, the quantity we wish to maximize is

$$\begin{aligned}
Q(\boldsymbol{\theta}|\widetilde{\boldsymbol{\theta}}) := \sum_{k=1}^{K}\sum_{m=1}^{M} p_{mk}\Big\{ & S_{\boldsymbol{e}_{m1}} + R_{\boldsymbol{e}_{m2}} + (\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}})\boldsymbol{W}'_k \\
& - \log(f_{uk}) - \log\left(f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}}\boldsymbol{W}'_k}\right)\Big\} \\
& - \frac{\tau_s}{2}\|\boldsymbol{S}\|^2 - \frac{\tau_r}{2}\|\boldsymbol{R}\|^2 - \frac{\tau_u}{2}\|\boldsymbol{U}\|_F^2 - \frac{\tau_v}{2}\|\boldsymbol{V}\|_F^2 - \frac{1}{2}\|\boldsymbol{W}\|_F^2 \\
& + \left(\frac{a_s + n}{2} - 1\right)\log\tau_s - \frac{b_s\tau_s}{2} + \left(\frac{a_r + n}{2} - 1\right)\log\tau_r - \frac{b_r\tau_r}{2} \\
& + \left(\frac{a_u + np}{2} - 1\right)\log\tau_u - \frac{b_u\tau_u}{2} + \left(\frac{a_v + np}{2} - 1\right)\log\tau_v - \frac{b_v\tau_v}{2} \\
& + \sum_{k=1}^{K}(\alpha_0 + p_{\cdot k} - 1)\log\alpha_k,
\end{aligned} \tag{8}$$

where p_{mk} is the (normalized) posterior probability from (6) (which is implicitly a function of $\widetilde{\boldsymbol{\theta}}$, not $\boldsymbol{\theta}$), and $p_{\cdot k} = \sum_m p_{mk}$.

Given the current estimates of $\boldsymbol{S}$, $\boldsymbol{R}$, $\boldsymbol{U}$, and $\boldsymbol{V}$, the maximizing values of τ_s, τ_r, τ_u, τ_v, and $\boldsymbol{\alpha}$ can be found explicitly as

$$\begin{aligned} \hat{\tau}_s &= \frac{a_s + n - 2}{b_s + \|\boldsymbol{S}\|^2}, & \hat{\tau}_u &= \frac{a_u + np - 2}{b_u + \|\boldsymbol{U}\|_F^2}, \\ \hat{\tau}_r &= \frac{a_r + n - 2}{b_r + \|\boldsymbol{R}\|^2}, & \hat{\tau}_v &= \frac{a_v + np - 2}{b_v + \|\boldsymbol{V}\|_F^2}, \\ \text{and } \hat{\alpha}_k &= \frac{\alpha_0 + p_{\cdot k} - 1}{K\alpha_0 + M - K}. \end{aligned} \tag{9}$$

Updating $\{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ requires more care, as there is no analytical solution to this optimization problem. Further, because the size of this set is $\mathcal{O}(n)$, we cannot rely on any optimization algorithm which utilizes second derivatives such as Newton-Raphson or quasi-Newton approaches without incurring unacceptable computational cost and memory requirements. To this end we employ a conjugate-gradient approach to maximize $\{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ given the current estimate of $\{\tau_s, \tau_r, \tau_u, \tau_v, \boldsymbol{\alpha}\}$. Before giving these gradients, we first introduce a little more notation. First, let $\mathcal{M}_{i1} := \{m : \boldsymbol{e}_{m1} = i\}$ and similarly $\mathcal{M}_{i2} := \{m : \boldsymbol{e}_{m2} = i\}$ denote the sets of edges from and to i respectively. Let $p_{(i1)k} := \sum_{m \in \mathcal{M}_{i1}} p_{mk}$ and similarly $p_{(i2)k} := \sum_{m \in \mathcal{M}_{i2}} p_{mk}$ denote the total expected number of edges of class k from and to i respectively. Then the gradients can be found to be

$$\frac{\partial Q}{\partial (S_i, \boldsymbol{U}_i)} = \sum_{k=1}^{K} \left(p_{(i1)k} - p_{\cdot k} \frac{e^{S_i + \boldsymbol{U}_i \boldsymbol{W}_k'}}{f_{uk}} \right) \begin{pmatrix} 1 & \boldsymbol{W}_k \end{pmatrix} - (S_i, \boldsymbol{U}_i) \begin{pmatrix} \tau_s & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_u I_p \end{pmatrix}, \tag{10}$$

$$\frac{\partial Q}{\partial (R_i, \boldsymbol{V}_i)} = \sum_{k=1}^{K} \left(p_{(i2)k} - \sum_{m \notin \mathcal{M}_{i1}} \frac{p_{mk} e^{R_i + \boldsymbol{V}_i \boldsymbol{W}_k'}}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}_k'}} \right) \begin{pmatrix} 1 & \boldsymbol{W}_k \end{pmatrix} - (R_i, \boldsymbol{V}_i) \begin{pmatrix} \tau_r & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_v I_p \end{pmatrix}, \tag{11}$$

$$\begin{aligned} \frac{\partial Q}{\partial \boldsymbol{W}_k} = \sum_{m=1}^{M} p_{mk} \Bigg[& \boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}} - \frac{1}{f_{uk}} \sum_{i=1}^{n} e^{S_i + \boldsymbol{U}_i \boldsymbol{W}_k'} \boldsymbol{U}_i \\ & - \frac{1}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}_k'}} \sum_{i \neq \boldsymbol{e}_{m1}} e^{R_i + \boldsymbol{V}_i \boldsymbol{W}_k'} \boldsymbol{V}_i \Bigg] - \boldsymbol{W}_k. \end{aligned} \tag{12}$$

At first blush, a derivate based estimation algorithm looks to erase any computational gains of having a likelihood of $\mathcal{O}(n)$. Indeed, evaluating (11) for all i would cost $\mathcal{O}(nM)$ as would evaluating (12) for each k. However, by precomputing certain quantities, these derivatives can be computed in $\mathcal{O}(M + n)$ time.

The key quantities necessary to make these gradients scalable are as follows. Let

$$\boldsymbol{s}_{uk} := \sum_{i=1}^{n} e^{S_i + \boldsymbol{U}_i \boldsymbol{W}_k'} \boldsymbol{U}_i, \qquad \boldsymbol{s}_{vk} := \sum_{i=1}^{n} e^{R_i + \boldsymbol{V}_i \boldsymbol{W}_k'} \boldsymbol{V}_i, \qquad \text{and } H_k := \sum_{m=1}^{M} \frac{p_{mk}}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}_k'}}. \tag{13}$$

Then we can equivalently write the gradients in (11) and (12) as

$$\frac{\partial Q}{\partial (R_i, \boldsymbol{V}_i)} = \sum_{k=1}^{K} \left(p_{(i2)k} - e^{R_i + \boldsymbol{V}_i \boldsymbol{W}_k'} \left(H_k - \frac{p_{(i1)k}}{f_{vk} - e^{R_i + \boldsymbol{V}_i \boldsymbol{W}_k'}} \right) \right) \begin{pmatrix} 1 & \boldsymbol{W}_k \end{pmatrix} - (R_i, \boldsymbol{V}_i) \begin{pmatrix} \tau_r & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_v I_p \end{pmatrix}, \tag{14}$$

$$\frac{\partial Q}{\partial \boldsymbol{W}_k} = \sum_{m=1}^{M} p_{mk} \left[\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}} - \frac{\boldsymbol{s}_{uk}}{f_{uk}} - \frac{\boldsymbol{s}_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}_k'} \boldsymbol{V}_{\boldsymbol{e}_{m1}}}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}_k'}} \right] - \boldsymbol{W}_k. \tag{15}$$

The cost of precomputing the $\boldsymbol{s}_{uk}$'s, $\boldsymbol{s}_{vk}$'s is $\mathcal{O}(n)$, and for the H_k's, $p_{(i1)k}$'s, and the $p_{(i2)k}$'s is $\mathcal{O}(M)$. Once computed, the cost of computing (14) for all i is $\mathcal{O}(n)$ and the cost of computing (15) is $\mathcal{O}(M)$, and thus the total cost of computing the gradient of Q is $\mathcal{O}(M + n)$.

For the purposes of verifying the scalability of this GEM algorithm, we took six real datasets of varying size and for each one ran 25 iterations of the algorithm setting $K = 3$ and $p = 2$. Computations were done using `R` in conjunction with `c++` code via `Rcpp` (Eddelbuettel and Balamuta, 2017) on an Intel i7-9850H 2.6GHz processor. The first dataset is the ubiquitous Sampson monk network (Sampson, 1968); the second is a patient transfer network described in detail in Section 4; the third dataset is the famous 2004 political blogs dataset (Adamic and Glance, 2005); the last three datasets can be found on the Stanford Large Network Dataset Collection[3]. An important note here is that with edge clustering, it only makes sense to cluster on the giant component. Any two edges clearly should not be clustered if they belong to different components of the graph. Table 1 gives the number of actors, number of edges, and computational time, and Figure 3 shows the near perfect linear relationship with respect to the number of actors and edges. The least squares plane has an R^2 equal to 0.998.

[3] `https://snap.stanford.edu/data/`

One last comment is that the initialization of this algorithm as implemented in the simulation studies and real data analyses was as follows. We set the row vectors of $\boldsymbol{W}$ to be equally spaced through $\Re^p$, and then randomly generated $\boldsymbol{S}$, $\boldsymbol{R}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ from a multivariate normal with mean $\mathbf{0}$ and covariance I_p. The results did vary somewhat based on the random starting position, but we found in practice that trying $\sim$10 or 15 starting positions outperformed more considered (and computationally expensive) initialization schemes.

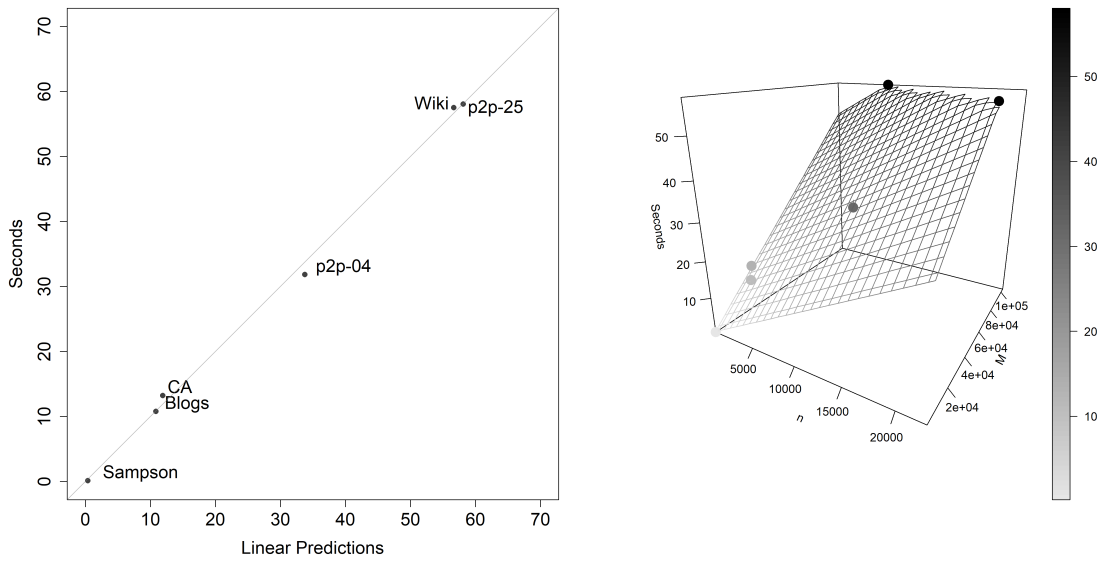
	Time (sec)	n	M
Sampson	0.15	18	88
CA	13.25	385	23,821
Political blogs	10.8	1,222	19,089
Wiki-vote	58.05	7,066	103,663
p2p-Gnutella04	31.84	10,876	39,994
p2p-Gnutella25	57.53	22,663	54,693

Table 1: Computing time in seconds for six real datasets with varying number of actors and edges. These times reflect 25 iterations setting $K = 3$ and $p = 2$.

2.3.2 Gradient-based Monte Carlo

Due to the high dimension of the parameter space $((2n + K)(p + 1) + 3)$ (in addition to the dimension of $\boldsymbol{Z}$), we suggest using a gradient-based Metropolis-Hastings step within a Gibbs algorithm. As mentioned above in (2.3.1), the full conditionals for $\{\boldsymbol{Z}_m\}_{m=1}^M$ are independent multinomial draws with probabilities given by (6). The priors for the precision parameters and $\boldsymbol{\alpha}$ are semi-conjugate, where the full conditionals are given by

$$\begin{aligned}
\tau_s|\cdot &\sim \Gamma\left(\frac{a_s+n}{2}, \frac{b_s+\|\boldsymbol{S}\|^2}{2}\right), & \tau_r|\cdot &\sim \Gamma\left(\frac{a_r+n}{2}, \frac{b_r+\|\boldsymbol{R}\|^2}{2}\right), \\
\tau_u|\cdot &\sim \Gamma\left(\frac{a_u+np}{2}, \frac{b_u+\|\boldsymbol{U}\|_F^2}{2}\right), & \tau_v|\cdot &\sim \Gamma\left(\frac{a_v+np}{2}, \frac{b_v+\|\boldsymbol{V}\|_F^2}{2}\right), \\
\boldsymbol{\alpha}|\cdot &\sim Dir\left(\alpha_0+M_1, \ldots, \alpha_0+M_K\right), & & \qquad (16)
\end{aligned}$$



(a) (b)

Figure 3: Time to run the GEM algorithm for six real datasets with varying number of actors and edges. These times reflect 25 iterations setting $K = 3$ and $p = 2$. The R^2 of the least squares plane equals 0.998.

where $M_k := \sum_m \boldsymbol{Z}_{mk}$ is the number of edges belonging to the k^{th} cluster. Given $(\{\boldsymbol{Z}_m\}_{m=1}^M, \tau_s, \tau_r, \tau_u, \tau_v, \boldsymbol{\alpha})$, one can then draw $(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W})$ using Metropolis-adjusted Langevin algorithm, Hamiltonian Monte Carlo, etc. While a simpler approach such as a random-walk Metropolis algorithm may be employed, due to the high dimensionality it is to be expected that a gradient based approach would lead to quicker convergence (Roberts and Rosenthal, 1998). The success of this is, of course, predicated on being able to compute the gradient in an efficient manner. Fortunately, we can use similar tricks as those in Section 2.3.1 to achieve this computational efficiency.

Letting $z_m \in \{1, \ldots, K\}$ denote the cluster assignment of the m^{th} edge, the partial

gradient of the full conditional with respect to $(S_i, \boldsymbol{U}_i)$ is given as

$$\begin{aligned}\frac{\partial \log \pi(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}|\cdot)}{\partial (S_i, \boldsymbol{U}_i)} &= \sum_{m \in \mathcal{M}_{i1}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - \sum_{m=1}^{M} \frac{e^{S_i + \boldsymbol{U}_i \boldsymbol{W}'_{z_m}}}{f_{u z_m}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - (S_i, \boldsymbol{U}_i) \begin{pmatrix} \tau_s & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_u I_p \end{pmatrix} \\ &= \sum_{m \in \mathcal{M}_{i1}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - \sum_{k} M_k \frac{e^{S_i + \boldsymbol{U}_i \boldsymbol{W}'_k}}{f_{uk}} \begin{pmatrix} 1 & \boldsymbol{W}_k \end{pmatrix} - (S_i, \boldsymbol{U}_i) \begin{pmatrix} \tau_s & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_u I_p \end{pmatrix}. \end{aligned} \tag{17}$$

If we precompute

$$\widetilde{H}_k := \sum_{m=1}^{M} \frac{1_{\{z_m = k\}}}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}'_k}},$$

then the gradient of $\log \pi(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}|\cdot)$ with respect to $(R_i, \boldsymbol{V}_i)$ is

$$\begin{aligned}&\frac{\partial \log \pi(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}|\cdot)}{\partial (R_i, \boldsymbol{V}_i)} \\ &= \sum_{m \in \mathcal{M}_{i2}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - \sum_{m \notin \mathcal{M}_{i1}} \frac{e^{R_i + \boldsymbol{V}_i \boldsymbol{W}'_{z_m}}}{f_{v z_m} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}'_{z_m}}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - (R_i, \boldsymbol{V}_i) \begin{pmatrix} \tau_r & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_v I_p \end{pmatrix} \\ &= \sum_{m \in \mathcal{M}_{i2}} \begin{pmatrix} 1 & \boldsymbol{W}_{z_m} \end{pmatrix} - \sum_{k} e^{R_i + \boldsymbol{V}_i \boldsymbol{W}'_k} \left(\widetilde{H}_k - \frac{|\mathcal{M}_{i1} \cap \{m : z_m = k\}|}{f_{vk} - e^{R_i + \boldsymbol{V}_i \boldsymbol{W}'_k}} \right) \begin{pmatrix} 1 & \boldsymbol{W}_k \end{pmatrix} - (R_i, \boldsymbol{V}_i) \begin{pmatrix} \tau_r & \boldsymbol{0} \\ \boldsymbol{0}' & \tau_v I_p \end{pmatrix} \end{aligned} \tag{18}$$

and with respect to $\boldsymbol{W}_k$ is

$$\sum_{m : z_m = k} \left(\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}} - \frac{\boldsymbol{s}_{uk}}{f_{uk}} - \frac{\boldsymbol{s}_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}'_k} \boldsymbol{V}_{\boldsymbol{e}_{m1}}}{f_{vk} - e^{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}} \boldsymbol{W}'_k}} \right) - \boldsymbol{W}_k, \tag{19}$$

where $\boldsymbol{s}_{uk}$ and $\boldsymbol{s}_{vk}$ are as given above in (13).

There are several sources of non-identifiability in the likelihood that must be addressed when considering the MCMC output. The usual LSM issues arise, namely the invariance to rotations, reflections and translations of the latent actor positions $\boldsymbol{U}$ and $\boldsymbol{V}$, and, in the LSEC, the latent cluster positions $\boldsymbol{W}$. In addition, the likelihood is invariant to rescaling $\boldsymbol{W}$ by some constant c and subsequently rescaling both $\boldsymbol{U}$ and $\boldsymbol{V}$ by $1/c$, as well as invariant to translations of $\boldsymbol{S}$, $\boldsymbol{R}$, and the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$. Finally, the usual mixture model issues arise with aliasing of the cluster labels.

We can address these issues through post-processing the MCMC output in four steps. First, we rescale $\boldsymbol{W}^{(\ell)}$ such that the elements have unit variance, where the $^{(\ell)}$ superscript

here and elsewhere will correspond to the ℓ^{th} MCMC draw. We then subsequently rescale $\boldsymbol{U}^{(\ell)}$, $\boldsymbol{V}^{(\ell)}$, $\tau_u^{(\ell)}$, and $\tau_v^{(\ell)}$ appropriately. Second, we perform a Procrustes transformation (see, e.g., Borg and Groenen, 2005) to rotate the matrix formed by stacking $\boldsymbol{U}^{(\ell)}$ and $\boldsymbol{V}^{(\ell)}$ using the MAP as a reference matrix, and then subsequently rotate $\boldsymbol{W}^{(\ell)}$. Third, for each iteration we recenter $\boldsymbol{S}^{(\ell)}$, $\boldsymbol{R}^{(\ell)}$, and the columns of both $\boldsymbol{U}^{(\ell)}$ and $\boldsymbol{V}^{(\ell)}$. Finally, we perform Equivalence Classes Representatives algorithm (Papastamoulis and Iliopoulos, 2010) to permute the labels of $\boldsymbol{Z}^{(\ell)}$; the rows and elements of $\boldsymbol{W}^{(\ell)}$ and $\boldsymbol{\alpha}^{(\ell)}$ respectively are then also permuted accordingly.

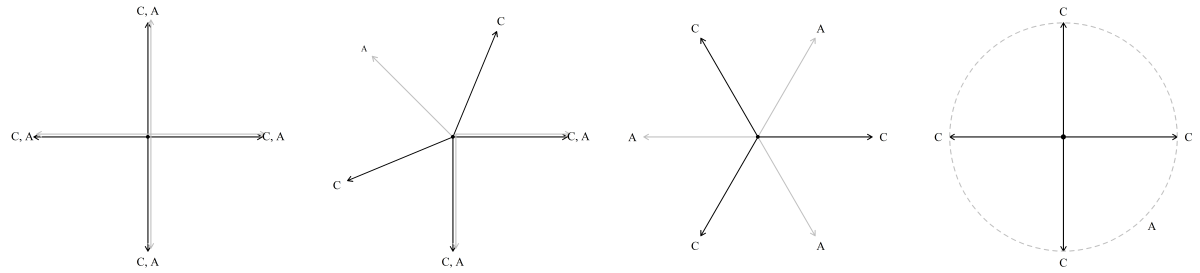
3 Simulation Studies

A huge challenge in both clustering and latent space network models is model selection-choosing the number of clusters in the former, and choosing the dimension of the latent space in the latter. Our context involves a double model selection, making it necessary to have both a computationally efficient estimation method as well as an effective selection criterion. Our GEM algorithm is sufficiently fast to fit multiple K's and multiple p's (the number of clusters and latent space dimension respectively). What remains is finding an effective model selection criterion. We performed a simulation study involving six scenarios (that is, six methods of generating the data) and for each data set fit 36 models corresponding to $K \in \{2, 3, \ldots, 10\}$ and $p \in \{2, 3, 4, 5\}$.

In each simulation scenario, there were 400 actors, and overall there was an average of 9608 edges. Each scenario varied how the actors and edge classes shared features. In the first four scenarios, the direction of the $\boldsymbol{U}_i$'s and $\boldsymbol{V}_i$'s were drawn from a mixture of von Mises-Fisher distributions with concentration parameter 50, 50, 50, and 0 respectively, and their magnitudes were drawn from a gamma distribution with (shape,rate) parameters $(20, 4)$; the (S_i, R_i) pairs were drawn from a bivariate normal distribution with mean $\mathbf{0}$, variances equal to 2, and correlation equal to 0.75; and $\boldsymbol{\alpha}$ was set equal to $(1/K, \ldots, 1/K)$. Figure 4 shows the mean directions of the von Mises-Fisher distributions from which the actors' latent features were drawn as well as the directions of the edge clusters $\boldsymbol{W}$. From this one can see that the number of clusters in the four scenarios are 4, 4, 3, and 4 respectively.

Note that for the fourth scenario the direction of each $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ is uniform. In the first three scenarios, there are sets of actors that share similar features. The first scenario is most similar to traditional community detection in which there are groups of actors which are densely intraconnected and sparsely interconnected. The second and third scenarios allows the latent edge class features to diverge from those of the actors. The fourth scenario removes any semblance of groups of actors. The final two scenarios build off of the first and fourth scenario described above, and have both been modified in the same way. We have increased p to 3 and added two additional clusters in opposite direction to each other and orthogonal to the plane displayed in Figure 4 (i.e., the cluster directions are $(I_3, -I_3)'$).

To perform double model selection, we tried several strategies. The first three used a single criterion, BIC, AIC, or ICL (Biernacki et al., 2000), to select both K and p (we computed all information criteria based on the MAP estimators; see, e.g., Fraley and Raftery, 2007). These simple strategies uniformly performed poorly. We also combined either BIC or AIC with either ICL or NEV (Celeux and Soromenho, 1996). Specifically, we used for each possible p either ICL or NEV to choose K; we subsequently chose the p (and its corresponding K as chosen by either ICL or NEV) that had the lowest BIC or AIC. Using two criteria performed better overall, although NEV was clearly inferior in this simulation study to using ICL. BIC performed poorly in selecting p, as it tended to overpenalize, always selecting $p = 2$. The best strategy was to use the combination of ICL and AIC. For scenario 2, no method performed well in selecting more than two clusters. This is perhaps not too surprising given the close proximity in angle between the vectors of $\boldsymbol{W}$, but unfortunately seems to indicate that clusters that are not well separated will not be detected as distinct clusters. For the other scenarios, this model selection approach performed quite well, especially given the difficulty of this double model selection problem. We present these results in Table 2. To make sure that the clustering results were accurate when the model selection chose the correct K and p, we computed the variation of information (VI) (Meila, 2003), normalized mutual information (NMI) (Danon et al., 2005), and the adjusted Rand Index (ARI) (Hubert and Arabie, 1985). As a point of comparison, we also fit the method of Ball et al. (2011) to the data and



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Figure 4: Four scenarios of the simulation studies. These figures provide the directions of the edge clusters ($\boldsymbol{W}_k$'s) in black and denoted by a "C", and the mean directions of the von Mises-Fisher distributions from which the actor features ($\boldsymbol{U}_i$'s and the $\boldsymbol{V}_i$'s) were drawn in gray and denoted by a "A". Note that in scenario 4, the direction of the actor features were drawn from a uniform distribution.

performed spectral clustering of the line graph. These comparisons are given in Table 3. For Scenarios 1 and 5 where the actors' latent positions are well separated, the line graph spectral clustering performs slightly better than the LSEC; however, for all other contexts, LSEC dramatically outperforms the other two approaches. This demonstrates that the true clustering assignments are being accurately recovered.

In addition, we wished to determine if the latent positions were being accurately recovered. To this end, for each of the six simulation scenarios we plotted the estimated $\boldsymbol{W}$'s along with the true values of $\boldsymbol{W}$. For each of the estimated latent position matrix, we found the row permutation and rotation which allowed us to most closely align it with the true $\boldsymbol{W}$. Figure 5 shows these plots, which demonstrate accurate recovery of the latent positions.

Scenario	p	K 2	3	4	5	6	7	8	9	10	Marginal Probability
1	2	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	
2	2	**0.90**	0.09	0.01	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.90	0.09	0.01	0.00	0.00	0.00	0.00	0.00	0.00	
3	2	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
4	2	0.00	0.00	**0.99**	0.004	0.004	0.00	0.00	0.00	0.00	1.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.00	0.00	0.99	0.004	0.004	0.00	0.00	0.00	0.00	
5	2	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.04	0.02	0.14
	3	0.00	0.00	0.00	0.00	**0.61**	0.12	0.08	0.02	0.02	0.86
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.00	0.00	0.00	0.00	0.61	0.16	0.12	0.06	0.04	
6	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	**0.75**	0.11	0.10	0.03	0.01	1.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Marginal Probability		0.00	0.00	0.00	0.00	0.75	0.11	0.10	0.03	0.01	

Table 2: Simulation study results for double model selection problem using ICL to choose K and AIC to choose p. The most commonly selected configurations of K and p are given in bold, and the true configurations are boxed.

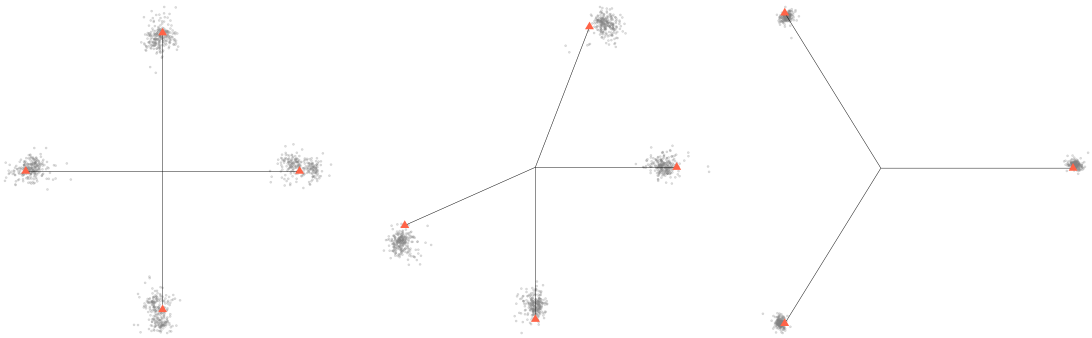
	VI			NMI			ARI		
Scenario	LSEC	BKN	LG	LSEC	BKN	LG	LSEC	BKN	LG
1	0.17	0.27	**0.14**	0.94	0.90	**0.95**	0.96	0.92	**0.97**
2	**0.98**	1.34	1.22	**0.64**	0.49	0.46	**0.62**	0.37	0.39
3	**0.96**	1.33	1.29	**0.56**	0.39	0.06	**0.64**	0.42	0.02
4	**0.32**	1.37	1.64	**0.88**	0.50	0.16	**0.92**	0.55	0.10
5	0.42	0.61	**0.35**	0.88	0.83	**0.90**	0.91	0.84	**0.92**
6	**0.79**	2.22	2.22	**0.78**	0.37	0.12	**0.81**	0.38	0.04

Table 3: Simulation study results for evaluating the accuracy in recovering the true cluster assignments. The VI (lower is better), NMI (higher is better), and ARI (higher is better) are given for the proposed method (LSEC), the method of Ball et al. (2011) (BKN), and spectral clustering of the line graph.

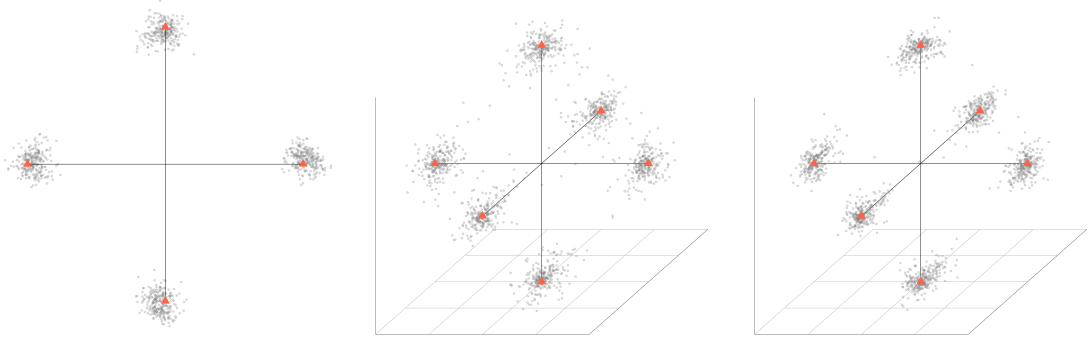
4 Illustrative Analyses

4.1 Patient transfer network

We considered a patient sharing network occurring from 2005 to 2011 in the state of California. The actors in this network are hospitals, and two hospitals i and j are connected by a directed edge if i transfers more than one patient to j over the study period. This network consisted of 385 actors and 13,724 edges. Patient sharing networks are important to understand due to their role in disease transmission; patients act as a vector for disease transmission between hospitals and have been shown to cause epidemic outbreaks and to significantly increase regional hospital acquired infections (Svoboda et al., 2004; Sewell et al., 2019). As an outbreak emerges, due to the risk of interfacility spread by transfer patients, public health officials and healthcare professionals need to know which patients to isolate for disease control or screen for surveillance. By identifying systems of flows in the patient transfer network, the knowledge of which transfer patients have been previously tested positive for the infectious disease allows us to better inform isolation or screening practice. Note that unlike actor-based clustering which would target all incoming transfers



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3



(d) Scenario 4 (e) Scenario 5 (f) Scenario 6

Figure 5: Simulation study results showing the estimated $\boldsymbol{W}$ (gray circles) and the true $\boldsymbol{W}$ (orange triangles connected to the origin by segments) providing evidence that the latent positions are being accurately recovered. (Color online.)

in a set of clustered facilities, edge clustering gives more precision by determining exactly which patient transfers to screen or isolate.

We fit the LSEC model to the data letting the number of clusters K range from 2 to 15 and letting the dimension of the latent space p range from 2 to 5. We set the hyperparameters $a_s = b_s = a_r = b_r = a_u = b_u = a_v = b_v = \alpha_0 = 1$. The AIC/ICL strategy indicated $p = 3$ and $K = 8$. Figure 6 shows the estimated latent features $\boldsymbol{U}$ of the hospitals along with the edges, where the edges have been colored according to their cluster. ($\boldsymbol{U}$ had a 0.935 correlation with $\boldsymbol{V}$, so plotting either would lead to similar figures. Note also that this may be interpreted to mean that hospitals send and receive patients based on similar features.) The degree to which edge clusters are well-separated can be viewed through $\boldsymbol{W}$.

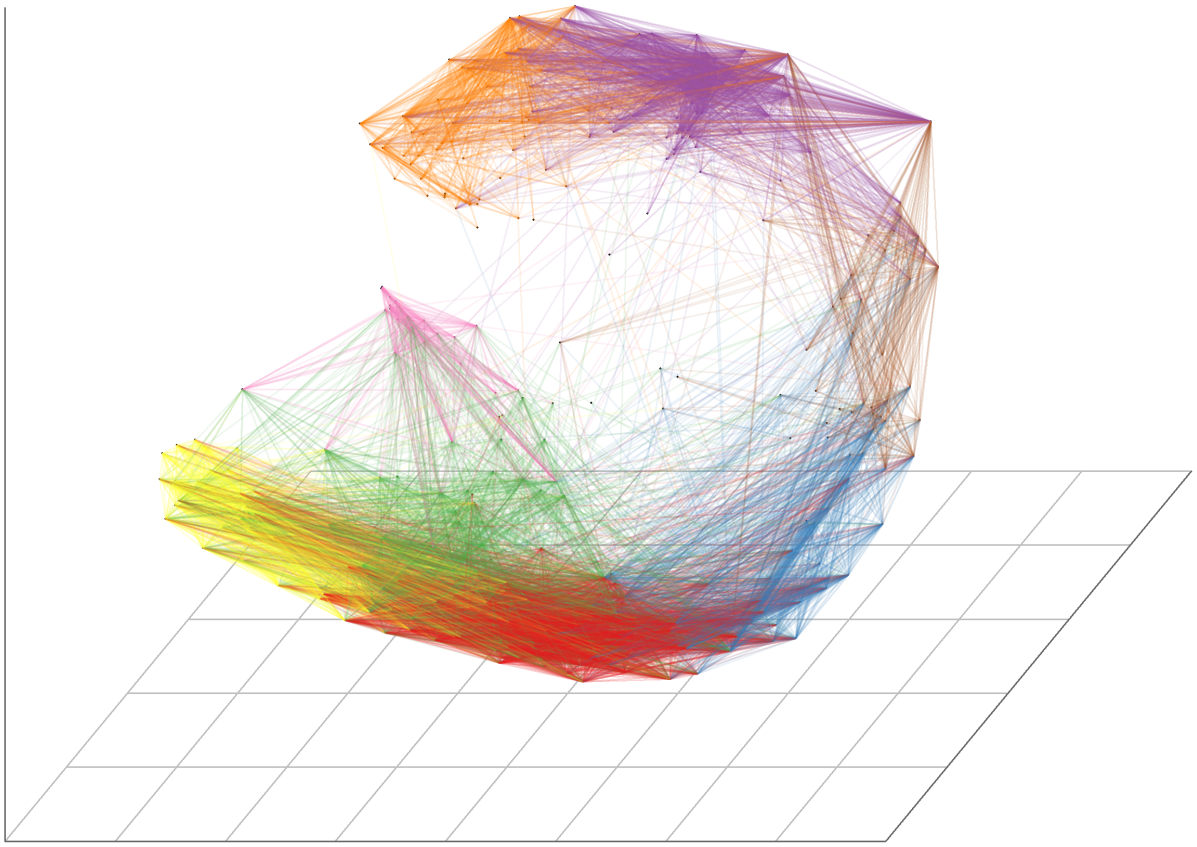


Figure 6: Patient sharing network. The estimated latent positions of the hospitals $\boldsymbol{U}$ are plotted along with the edges in the network. The edges have been colored according to the 8 clusters identified from fitting the LSEC model to the data. (Color online.)

If the angles between the $\boldsymbol{W}_k$'s are small, then this would indicate that the clusters are not overly distinguishable from one another, whereas large angles indicate that the clusters are in fact well-separated. Table 4 gives the angles between the estimated $\boldsymbol{W}_k$'s, which are all large, indicating well-separated clusters of connections between the hospitals.

There is no ground truth in this data set. However, to investigate whether these estimated systems of flows have any bearing on epidemic outbreaks we simulated epidemics on the network using a susceptible-exposed-infectious (SEI) epidemic model. These simulations were based loosely on those done by Karkada et al. (2011). We initially infected one hospital in the network and allowed the infection to travel from hospital to hospital

k	2	3	4	5	6	7	8
1	111	115	98	50	41	158	57
2		126	126	61	96	88	106
3			74	155	100	56	123
4				123	133	61	55
5					55	146	69
6						152	97
7							108

Table 4: Patient sharing network. Angles (in degrees) between the estimated cluster vectors $\boldsymbol{W}_k$. Larger angles indicate well-separated clusters.

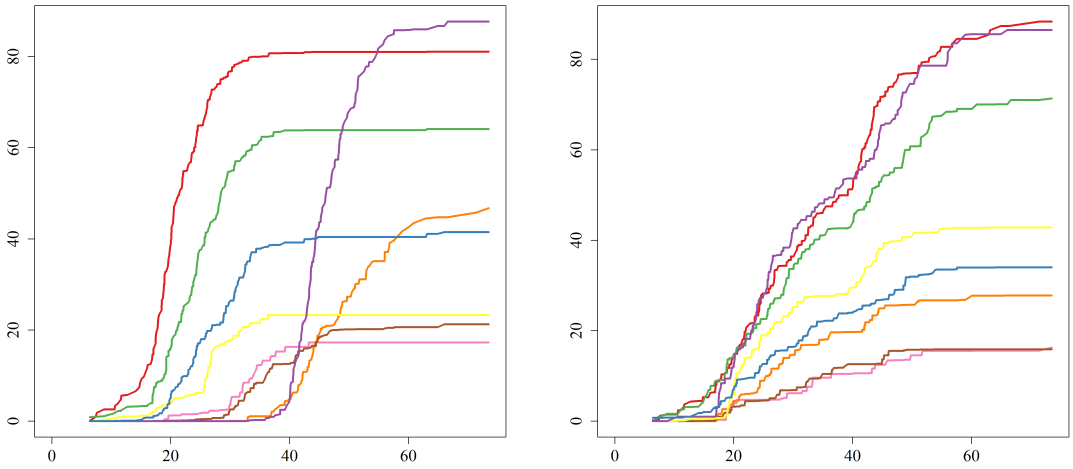
over the edges (i.e., by patients being transferred across hospitals). Once infected, a hospital remained in the exposed state for a certain time which was randomly drawn from a gamma distribution with mean 7 days and standard deviation of 1 day. After this exposure time, the hospital became infectious and could transmit the disease to other hospitals. This transmission occurred according to an exponential distribution governed by the overall rate of patient transfers in the data and a transmission probability of 0.1. Combining these led to a transmission rate of 0.046/day across edges. The infectious state was considered an absorbing state, and the simulations were run until all reachable[4] hospitals were infected.

All the simulations we looked at gave similar qualitative results, so we only show here one example. Figure 7a shows for each of the 8 edge clusters the expected number of edges in that cluster that have led to a new infection over time. That is, for each time point t and each $k \in \{1, \ldots, 8\}$ we compute

$$\sum_{m=1}^{M} p_{mk} 1_{\{\boldsymbol{e}_{m1} \text{ infected } \boldsymbol{e}_{m2} \text{ at or before } t\}}$$

Due to the varying number of edges belonging to each group, the terminal/plateau points are different for each edge cluster. What is clear from these simulations is that as soon as the disease enters a particular system of flows, there is a rapid increase in the number

[4]One hospital only transferred out patients while never receiving patients from elsewhere.



(a) (b)

Figure 7: Epidemic simulation on the patient sharing network. Each curve shows the expected number of edges for a specific edge cluster which have transmitted disease from one hospital to another (vertical axis) at or before a particular time (horizontal axis, in days). The left plot corresponds to the LSEC edge clusters, while the right plot is a random permutation of those edge clusters. (Color online.)

of edges of that same class that act to transmit disease. We can compare these results to Figure 7b which shows the same epidemic but with the edge cluster labels permuted. From the figure based on the permuted edge labels we see much flatter slopes and less sudden jumps in transmission within a particular system of flows. This would seem to imply that once an infection is detected in a transfer patient we may leverage the edge clusters to identify which future transfer patients to expend efforts in screening or isolation procedures.

4.2 UK faculty friendships

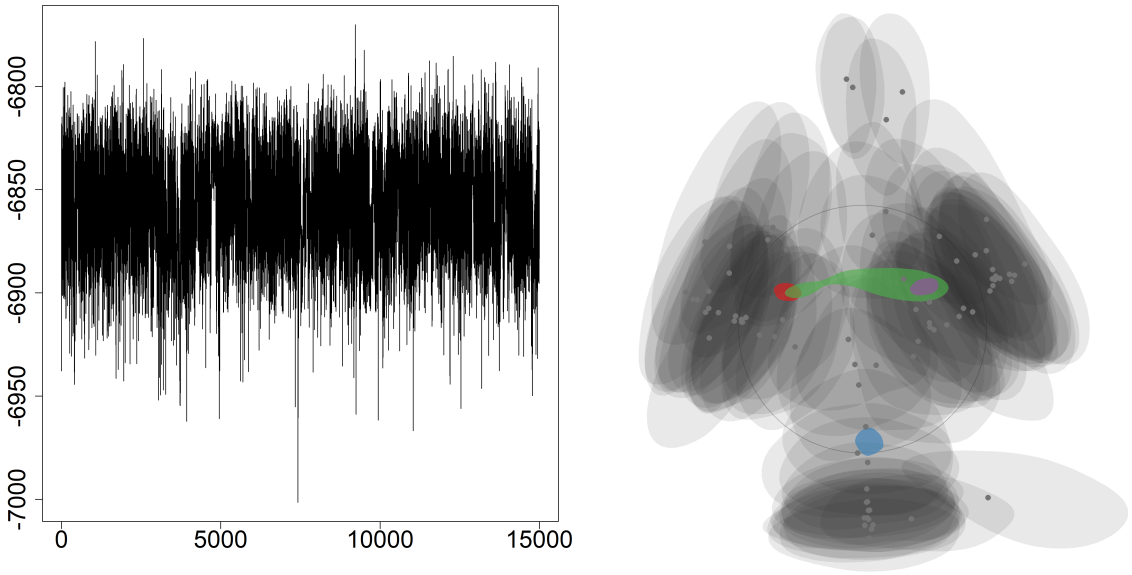
We considered a friendship network amongst faculty members at three schools within a university located in the United Kingdom (Nepusz et al., 2008). Friendship data was collected by questionnaire from 81 faculty members, although two failed to note their

school. We analyzed this data using the LSEC model, setting the hyperparameters to be $a_s = b_s = a_r = b_r = a_u = b_u = a_v = b_v = 1$ and $\alpha_0 = 2$. The AIC/ICL strategy indicated $p = 2$ and $K = 3$. However, the second highest ICL indicated $K = 4$, and this scenario led to an interesting result which we will describe shortly. After determining K and p via the GEM algorithm, we then ran a Hamiltonian Monte Carlo-within-Gibbs sampling algorithm using 15,000 samples as a burn-in period, and keeping every 25^{th} iteration afterwards until another 15,000 samples were obtained. Figure 8a provides the trace plot of the log posterior, and 8b shows the 50% posterior credible regions of $\boldsymbol{U}$ (as with the patient transfer network, there was a strong correlation between $\boldsymbol{U}$ and $\boldsymbol{V}$ leading to similar graphics) and $\boldsymbol{W}$. From this figure it appears that there is considerable overlap between the green cluster and the purple and red clusters, which may well explain why ICL preferred 3, rather than 4, clusters. On average, however, the posterior estimate of the angle between the rows of $\boldsymbol{W}$ corresponding to the green cluster and the purple and red clusters (53° and 74° respectively) indicated a reasonable level of cluster separation. The posterior means of the angles between the rows of $\boldsymbol{W}$ (i.e., cluster vectors) are given in Table 5.

Figure 9 shows the estimates of $\boldsymbol{U}$ with the edges colored by their MAP estimated cluster assignation. It is apparent that three of the clusters primarily capture within-school edges. The fourth, however, primarily captures edges connecting faculty of different schools. That is, while we might expect an actor-centric community detection algorithm to detect the three clusters corresponding to the three schools, by focusing on the edges we can determine which edges form within the environment of a specific school, and which edges take place in a broader setting at the university level.

k	2 (blue)	3 (purple)	4 (green)
1 (red)	119	120	75
2 (blue)		121	136
3 (purple)			53

Table 5: UK faculty friendship network. Angles (in degrees) between the estimated cluster vectors $\boldsymbol{W}_k$. Larger angles indicate well-separated clusters.



(a) (b)

Figure 8: UK faculty friendship network. (a) Trace of the log posterior from applying HMC-within-Gibbs. (b) 50% credible regions for the $\boldsymbol{U}_i$'s (gray) and the $\boldsymbol{W}_k$'s (red, blue, purple, green). (Color online.)

Although the intention of the proposed methodology is for clustering the edges of a network, edge clustering can be used to induce a K-dimensional compositional vector for each of the actors, much akin to a mixed membership vector which may be interpreted to signify the probability that an actor will select a particular role with which to form edges. Once the edges are clustered, one may look at each actor i, and for each edge cluster k compute the number of edges incident on i which belong to the k^{th} cluster. This may be interpreted as describing where each of the actors have formed edges. Figure 10a provides the posterior mean of the $\boldsymbol{U}_i$'s connected by the edges of the network where, as in Figure 9, the edges have been colored according to their cluster; the actors have been marked by a pie graph showing the edge cluster distribution for the edges incident on each actor. Specifically, these actor-specific allocations correspond to the posterior expected number of edges in each cluster, given by

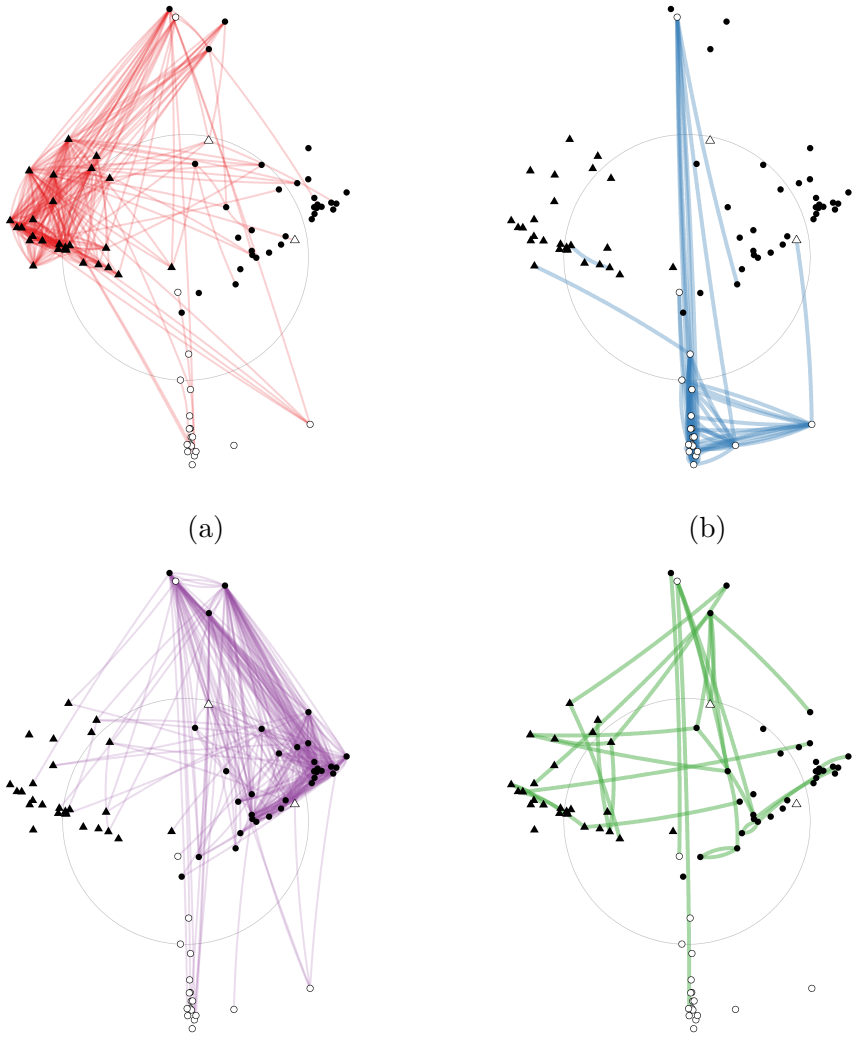
$$\sum_{m \in \mathcal{M}_{i1} \cup \mathcal{M}_{i2}} \Big(\mathbb{P}(z_m = 1|\mathcal{E}), \quad \mathbb{P}(z_m = 2|\mathcal{E}), \quad \ldots \quad \mathbb{P}(z_m = K|\mathcal{E}) \Big)$$

As a point of comparison, we also fit a mixed membership stochastic blockmodel (MMSB) (Airoldi et al., 2008) with four clusters and compared this visually with the mixed membership induced by the edge clustering (Figure 10b). As a general point, we argue that while the interpretations of the actors' compositional vectors resulting from the LSEC and from those via the MMSB are similar, it is much more interpretable to discuss the actor-specific edge cluster distributions than the output of a MMSB. For example, in a friendship network, edge clustering might result in declaring that an individual might have 50% of his/her frienships formed from a workplace, 30% through a community organization, etc., whereas MMSB would result in declaring that an individual has 50% belonging to his/her workplace, 30% belonging to a community organization, etc. Clearly the individual fully belongs to each of these groups, but a different proportion of friendships may form within each context. In the specific case of the UK faculty friendship network, the MMSB fails to capture the various contexts in which edges are formed. Rather than capturing the fourth environment in which friendships formed (university), this method split one of the schools into two, leading to results which are not easily interpreted.

5 Discussion

Although community detection is one of the hottest research topics in network analysis, most work is focused on clustering the actors of the network. We argue that it is just as, or perhaps more, natural to think of the *edges* as belonging to one of many clusters rather than *actors*. Edge clustering has the potential to be useful in contexts where it is of interest to study flow through the network as demonstrated by our simple epidemic simulation studies. Edge clustering also has the potential to be useful in contexts such as social networks where edges are assumed to have formed within a particular context or environment.

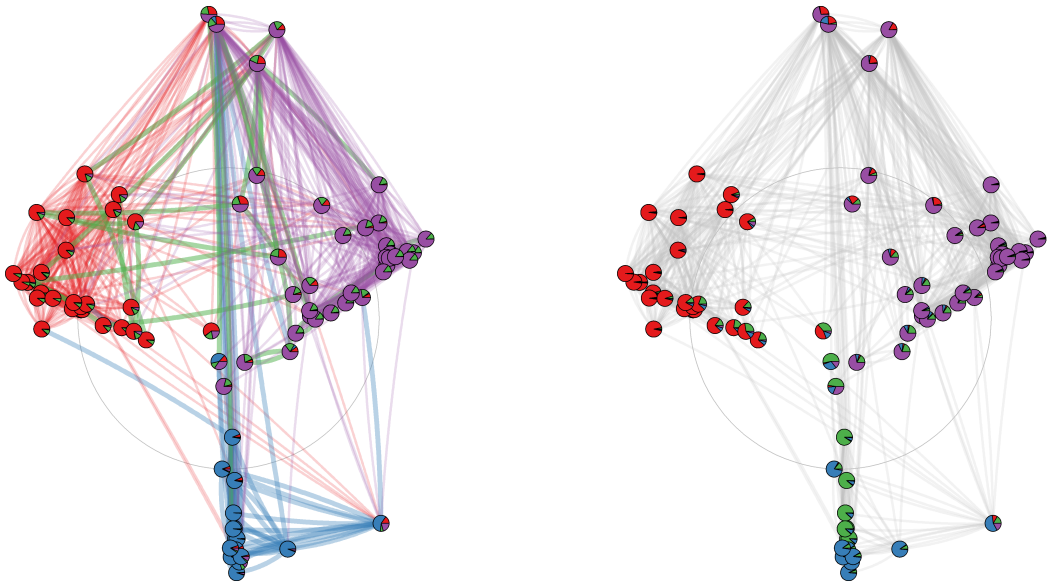
We have proposed a novel approach for edge clustering that is based on a probabilistic generative model over edge exchangeable networks. We have based this on intuitive ideas of actor and edge features building on the widely used class of LSMs. We derived a generalized EM algorithm for estimation and provided guidance on gradient-based Monte



(a) (b)

(c) (d)

Figure 9: The estimated latent features $\boldsymbol{U}$ from fitting the LSEC model to the UK faculty friendship network and the edges as colored by the MAP edge partition. The three schools are represented by different shapes. The two respondents who did not indicate a school are given as hollow triangles. (Color online.)



(a) LSEC (b) MMSB

Figure 10: The estimated latent features $\boldsymbol{U}$ from fitting the LSEC model to the UK faculty friendship network. (a) The edges are colored according to the cluster assignments from the LSEC model, and actors are shown as pie graphs depicting the edge cluster distribution of those edges incident on each actor. (b) The actors are shown as pie graphs depicting the mixed membership vectors as estimated from the MMSB model. (Color online.)

Carlo algorithms which have a computational cost that is linear in the number of actors and edges. This is in contrast with existing LSMs which have a cost that grows quadratically with the number of actors. This approach uses a fundamentally different vantage point, namely at looking at the edges as the units of observations rather than the dyads.

One limitation to the proposed model is that there is no constraint disallowing multiple edges, although in practice most networks do not allow multiple edges between a single pair of actors. This is an issue inherent in edge exchangeable network models. However, because our simulation studies show positive results and our real data analyses give intuitive answers, we conclude that ignoring this constraint is negligibly detrimental in gaining important insights into the latent group structure of network data. Additionally, as is often the case with network clustering approaches such as blockmodels or BKN, the proposed model does not explicitly account for transitivity or other higher order dependencies. Addressing these issues would be an important contribution to what has been presented here. Finally, while we have described gradient-based Monte Carlo methods for going from point estimation to inference, Markov chain Monte Carlo methods, regardless of the computationally efficient gradients presented, will in general be too slow and memory intensive for large networks. Ergo it would be useful to develop, e.g., a variational Bayes method for the LSEC model in future work.

Acknowledgments

This work was supported by the US Centers for Disease Control and Prevention (5 U01 CK000531-02).

References

Adamic, L. A. and Glance, N. (2005), "The Political Blogosphere and the 2004 U.S. Election: Divided They Blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, New York, NY, USA: ACM, LinkKDD '05, pp. 36–43.

Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. (2010), "Link Communities Reveal Multiscale Complexity in Networks," *Nature*, 466, 761.

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

Ball, B., Karrer, B., and Newman, M. E. J. (2011), "Efficient and principled method for detecting communities in networks," *Phys. Rev. E*, 84, 036103.

Bello-Orgaz, G., Salcedo-Sanz, S., and Camacho, D. (2018), "A Multi-Objective Genetic Algorithm for overlapping community detection based on edge encoding," *Information Sciences*, 462, 290 – 314.

Biernacki, C., Celeux, G., and Govaert, G. (2000), "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 719–725.

Borg, I. and Groenen, P. J. (2005), *Modern Multidimensional Scaling: Theory and Applications*, Springer.

Broderick, T. and Cai, D. (2015), "Edge-exchangeable graphs, sparsity, and power laws," in *NIPS 2015 Workshop on Bayesian Nonparametrics: The Next Generation*.

Cai, Q., Ma, L., Gong, M., and Tian, D. (2016), "A survey on network community detection based on evolutionary computation," *International Journal of Bio-Inspired Computation*, 8, 84–98.

Celeux, G. and Soromenho, G. (1996), "An entropy criterion for assessing the number of clusters in a mixture model," *Journal of Classification*, 13, 195–212.

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

Danon, L., Díaz-Guilera, A., Duch, J., and Arenas, A. (2005), "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, 2005, P09008–P09008.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, 1–38.

Deng, X., Li, G., Dong, M., and Ota, K. (2017), "Finding overlapping communities based on Markov chain and link clustering," *Peer-to-Peer Networking and Applications*, 10, 411–420.

Dhumal, A. and Kamde, P. (2015), "Survey on community detection in online social networks," *International Journal of Computer Applications*, 9, 35 – 41.

Eddelbuettel, D. and Balamuta, J. J. (2017), "Extending extitR with extitC++: A Brief Introduction to extitRcpp," *PeerJ Preprints*, 5, e3188v1.

Enugala, R., Rajamani, L., Ali, K., and Kurapati, S. (2015), "Community detection in dynamic social networks: A survey," *International Journal of Research and Applications*, 2, 278 – 285.

Evans, T. S. and Lambiotte, R. (2009), "Line graphs, link partitions, and overlapping communities," *Phys. Rev. E*, 80, 016105.

— (2010), "Line graphs of weighted networks for overlapping communities," *The European Physical Journal B*, 77, 265–272.

Fortunato, S. (2010), "Community detection in graphs," *Physics Reports*, 486, 75 – 174.

Fraley, C. and Raftery, A. E. (2007), "Bayesian regularization for normal mixture estimation and model-based clustering," *Journal of Classification*, 24, 155–181.

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

Hubert, L. and Arabie, P. (1985), "Comparing partitions," *Journal of Classification*, 2, 193–218.

Janson, S. (2018), "On edge exchangeable random graphs," *Journal of Statistical Physics*, 173, 448–484.

Karkada, U., Adamic, L. A., Kahn, J., and Iwashyna, T. (2011), "Limiting the spread of highly resistant hospital-acquired microorganisms via critical care transfers: a simulation study," *Intensive Care Medicine*, 37, 1633–1640.

Kim, P. and Kim, S. (2015), "Detecting overlapping and hierarchical communities in complex network using interaction-based edge clustering," *Physica A: Statistical Mechanics and its Applications*, 417, 46 – 56.

Kim, Y. and Jeong, H. (2011), "Map equation for link communities," *Phys. Rev. E*, 84, 026110.

Malliaros, F. D. and Vazirgiannis, M. (2013), "Clustering and community detection in directed networks: A survey," *Physics Reports*, 533, 95 – 142, clustering and Community Detection in Directed Networks: A Survey.

Meila, M. (2003), "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines*, Springer, pp. 173–187.

Nepusz, T., Petróczi, A., Négyessy, L., and Bazsó, F. (2008), "Fuzzy communities and the concept of bridgeness in complex networks," *Phys. Rev. E*, 77, 016107.

O'Connor, L., Médard, M., and Feizi, S. (2015), "Maximum Likelihood Latent Space Embedding of Logistic Random Dot Product Graphs," *arXiv preprint arXiv:1510.00850*.

Papastamoulis, P. and Iliopoulos, G. (2010), "An Artificial Allocations Based Solution to the Label Switching Problem in Bayesian Analysis of Mixtures of Distributions," *Journal of Computational and Graphical Statistics*, 19, 313–331.

Pereira-Leal, J. B., Enright, A. J., and Ouzounis, C. A. (2004), "Detection of functional modules from protein interaction networks," *Proteins: Structure, Function, and Bioinformatics*, 54, 49–57.

Pizzuti, C. (2018), "Evolutionary Computation for Community Detection in Networks: A Review," *IEEE Transactions on Evolutionary Computation*, 22, 464–483.

Raftery, A. E., Niu, X., Hoff, P. D., and Yeung, K. Y. (2012), "Fast inference for the latent space network model using a case-control approximate likelihood," *Journal of Computational and Graphical Statistics*, 21, 901–919.

Rastelli, R., Maire, F., and Friel, N. (2018), "Computationally efficient inference for latent position network models," *arXiv preprint arXiv:1804.02274*.

Roberts, G. O. and Rosenthal, J. S. (1998), "Optimal scaling of discrete approximations to Langevin diffusions," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 255–268.

Salter-Townshend, M. and Murphy, T. B. (2013), "Variational Bayesian inference for the latent position cluster model for network data," *Computational Statistics & Data Analysis*, 57, 661–671.

Sampson, S. F. (1968), "A novitiate in a period of change: An experimental and case study of relationships," *Unpublished Ph. D. dissertation, Department of Sociology, Cornell University*.

Sewell, D. K., Simmering, J. E., Justice, S., Pemmaraju, S. V., Segre, A. M., and Polgreen, P. M. (2019), "Estimating the Attributable Disease Burden and Effects of Inter-Hospital Patient Sharing on Clostridium difficile Infections," *Infection Control and Hospital Epidemiology*, 40, 656–661.

Shi, C., Cai, Y., Fu, D., Dong, Y., and Wu, B. (2013), "A link clustering based overlapping community detection algorithm," *Data & Knowledge Engineering*, 87, 394 – 404.

Svoboda, T., Henry, B., Shulman, L., Kennedy, E., Rea, E., Ng, W., Wallington, T., Yaffe, B., Gournis, E., Vicencio, E., Basrur, S., and Glazier, R. H. (2004), "Public health measures to control the spread of the Severe Acute Respiratory Syndrome during the outbreak in Toronto," *New England Journal of Medicine*, 350, 2352–2361, pMID: 15175437.

Wu, Z., Lin, Y., Wan, H., and Tian, S. (2010), "A fast and reasonable method for community detection with adjustable extent of overlapping," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, pp. 376–379.

Xie, J., Kelley, S., and Szymanski, B. K. (2013), "Overlapping Community Detection in Networks: The State-of-the-art and Comparative Study," *ACM Comput. Surv.*, 45, 43:1–43:35.

Young, S. J. and Scheinerman, E. R. (2007), "Random dot product graph models for social networks," in *Algorithms and Models for the Web-Graph*, Springer, pp. 138–149.

Zhang, Y., Zhang, Y., Chen, Q., Ai, Z., and Gong, Z. (2018), "True-link clustering through signaling process and subcommunity merge in overlapping community detection," *Neural Computing and Applications*, 30, 3613–3621.