Network Science **12** (1): 88–106, 2024
doi:10.1017/nws.2023.22

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Automated detection of edge clusters via an overfitted mixture prior

Hanh T. D. Pham and Daniel K. Sewell

University of Iowa, Iowa City, IA, USA
Corresponding author: Hanh T. D. Pham; Email: hanh-pham@uiowa.edu

Abstract

Most community detection methods focus on clustering actors with common features in a network. However, clustering edges offers a more intuitive way to understand the network structure in many real-life applications. Among the existing methods for network edge clustering, the majority are algorithmic, with the exception of the latent space edge clustering (LSEC) model proposed by Sewell (*Journal of Computational and Graphical Statistics, 30*(2), 390–405, 2021). LSEC was shown to have good performance in simulation and real-life data analysis, but fitting this model requires prior knowledge of the number of clusters and latent dimensions, which are often unknown to researchers. Within a Bayesian framework, we propose an extension to the LSEC model using a sparse finite mixture prior that supports automated selection of the number of clusters. We refer to our proposed approach as the automated LSEC or aLSEC. We develop a variational Bayes generalized expectation-maximization approach and a Hamiltonian Monte Carlo-within Gibbs algorithm for estimation. Our simulation study showed that aLSEC reduced run time by 10 to over 100 times compared to LSEC. Like LSEC, aLSEC maintains a computational cost that grows linearly with the number of actors in a network, making it scalable to large sparse networks. We developed the R package aLSEC which implements the proposed methodology.

Keywords: Clustering; community detection; network analysis; overfitted mixture model; variational Bayes

1. Introduction

A network consists of actors whose relationships are represented as edges connecting them. By analyzing the connections between actors and the overall structure of the network, researchers can gain valuable insights into the phenomenon from which the network arises. Examples of network applications exist in epidemiology (e.g., studying the spread of infectious disease, Salathé et al., 2010), social science (e.g., exploring the relationship between social media use and opinion polarization, Lee et al., 2014, biology e.g., identifying functionally coordinated interactions between proteins, Pereira-Leal et al., 2004), and many other fields. A widely studied topic in network analysis is community detection, which generally involves partitioning actors into groups such that actors are densely connected within their groups but sparsely connected with others (Harenberg et al., 2014). Among the vast amount of methods dedicated to clustering network actors in literature, many employed a model-based approach. Compared to algorithmic approaches, model-based community detection offers researchers advantages such as the capacity to quantify the uncertainty of estimates, infer missing data, and make predictions. Some notable model-based approaches for static networks are the latent position cluster model (Handcock et al., 2007), stochastic block model (SBM) (Nowicki & Snijders, 2001), degree-corrected SBM (Karrer & Newman, 2011), and mixed-membership SBM (Airoldi et al., 2008). Relevant methods for dynamic networks include the stochastic block transition model (Xu, 2015), and the latent

space approach to community detection built upon distance and projection models (Sewell & Chen, 2017).

While most community detection methods focus on clustering actors based on their connections, there are situations where clustering the edges provides a more intuitive way to characterize the network structure. For example, actors create friendships within a specific context in social networks, such as a classroom or workplace. An edge-centric approach to community detection can shed light on the context and facilitate knowledge about systems of flows in a network (Sewell, 2021). Despite the advantages of intuition and interpretation, few methods have been developed to cluster a network's edges. Most have been algorithmic in nature, using reasonable yet ad hoc approaches (Pereira-Leal et al., 2004; Evans & Lambiotte, 2009; Wu et al., 2010; Ahn et al., 2010). An exception to this, and the approach we focus on in this paper, is the latent space edge clustering (LSEC) method proposed by Sewell (2021). This model-based approach aims at finding the unobserved contexts in which edges form. In the LSEC model, the edges represent units of observation and are determined stochastically based on latent actor-level features and the latent environment of the edge contexts. This is in contrast to actor-centric methods where dyads are treated as the units of observations. That is, actor-centric approaches model the probability that actors i and j are connected (the edge is the random binary variable), whereas edge-centric approaches model the probability that an edge connects actors i and j (the incident actors are the random polychotomous variables) (Sewell, 2021).

Despite the promising results in simulation studies and real-world data analyses, the method has limitations. A challenge in fitting LSEC models is the double model selection problem, where the number of latent dimensions p and the number of clusters K must be specified beforehand. Sewell (2021) proposed fitting various models assuming different combinations of possible values of p and K before using the Akaike Information Criterion (AIC) and the Integrated Completed Likelihood (ICL) criterion, respectively, to select the optimal model. However, this approach ignores the uncertainty associated with estimating p and K and can be computationally costly to implement when considering multiple possible values of p and K. Under a Bayesian framework, we propose an adjustment for the priors in the LSEC model to simultaneously obtain estimates for both the number of clusters K and cluster-specific parameters. Even though we assume a known value of p, our simulation study showed that the penalty for incorrectly specifying p in our algorithm was minimal in most cases and eliminated the need to pick the correct value of p.

Several methods have been proposed in actor-centric community detection literature to tackle the challenge of selecting the number of clusters. Within a Bayesian framework, it is possible to select a prior for the number of components in the finite mixture model and then estimate it using a reversible jump Markov chain Monte Carlo (RJMCMC) sampler (Green, 1995). RJMCMC suffers from low acceptance rates, making it challenging to implement efficiently in practice (Stephens, 2000; Green & Hastie, 2009). An alternative is to use a Bayesian non-parametric model such as the Dirichlet Process Mixture Model (DPMM). However, Miller & Harrison (2013) offered some evidence showing that the posterior of the number of clusters in DPMM was not consistent. Other approaches to estimating the number of clusters include modularity maximization (Newman, 2006), efficient inference algorithms for mixture of finite mixture models with a prior on the number of components (Miller & Harrison, 2018; Geng et al., 2019), maximization of the posterior probability of the number of clusters given the network (Newman & Reinert, 2016), evaluation of eigenvalues of matrices associated with the network (Le & Levina, 2015), and hypothesis testing (Lo et al., 2001).

Unlike clustering the actors of a network, to the authors' knowledge there have been no prior attempts to automatically determine the number of clusters when clustering the edges of a network. This paper applies a sparse finite mixture approach to the LSEC model to simultaneously obtain estimates of the number of clusters and other model variables of interest. We derive a computationally efficient variational Bayes generalized EM algorithm and a gradient-based Monte

Carlo algorithm for estimation. We show in a simulation study that good estimation performance is obtained so long as the latent space's dimension is overspecified. By avoiding the need for a large number of repeated model fitting for various values of p and K as is necessary using the current state-of-the-art method, we reduced the computation time by 10–100 times in our analyses. Our generalized EM algorithm is efficient with a computational cost that grows linearly with the number of actors and edges in a network. The R package and code for analyzing the UK faculty network example in Section 4 can be found online.[1]

The remainder of this paper is as follows. Section 2 describes our method and estimation algorithm. Section 3 shows the results of our simulation study. Section 4 provides two examples of applying our method to real network data sets. Finally, Section 5 wraps up the paper with a discussion on the strengths and limitations of our approach.

2. Methods

2.1 Latent space edge clustering model

The latent space edge clustering model is formulated based on the idea that edges connect actors who share some characteristics with the same environment where the edges are formed (Sewell, 2021). Sewell explained two scenarios where this phenomenon emerges: relationship forming and opportunities for diffusion. In the former, the edges are formed between actors who engage in a joint latent group or environment, thus representing the various latent contexts of network connections. In the latter, the edges create systems of flows, where each system consists of edges that tend to act cohesively in a diffusion process.

Given a network with n actors and M edges, denote each edge $\boldsymbol{e}_m = (e_{m1}, e_{m2})$. We assume that the network is unweighted and directed, but the model can also be adapted to undirected networks. Let K be the number of edge classes and p be the number of latent dimensions. Let $\boldsymbol{Z}_{M\times K}$ be a latent class assignment matrix such that Z_{mk} equals one if the mth edge belongs to class k and zero otherwise. Let $\boldsymbol{W}_{K\times p}$ be a matrix such that each row $\boldsymbol{W}_k$ corresponds to the latent features of edge class k. Let $\boldsymbol{U}_{n\times p}$, $\boldsymbol{V}_{n\times p}$ be matrices where their respective rows $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ denote the latent sending and receiving features of actor i. Lastly, let $\boldsymbol{S} = (S_1, \ldots, S_n)$ and $\boldsymbol{R} = (R_1, \ldots, R_n)$ represent actors' specific overall propensities to send and receive edges respectively. The LSEC model proposed by Sewell (2021) is given as

$$
\begin{aligned}
f(\mathcal{E}|\boldsymbol{Z}) &= \prod_{m=1}^{M}\prod_{k=1}^{K}\left[f(\boldsymbol{e}_m|Z_{mk}=1)\right]^{Z_{mk}} \\
&= \prod_{m=1}^{M}\prod_{k=1}^{K}\left[f(e_{m1}|Z_{mk}=1)f(e_{m2}|e_{m1},Z_{mk}=1)\right]^{Z_{mk}},
\end{aligned}
$$

where

$$
f(e_{m1}=i|Z_{mk}=1) = \frac{e^{S_i+\boldsymbol{U}_i\boldsymbol{W}_k'}}{g_{uk}},
$$

$$
f(e_{m2}=j|e_{m1}=i,Z_{mk}=1) = \begin{cases} \frac{e^{R_j+\boldsymbol{V}_j\boldsymbol{W}_k'}}{g_{vk}-e^{R_i+\boldsymbol{V}_i\boldsymbol{W}_k'}} & \text{if } i\neq j \\ 0 \text{ otherwise,} \end{cases}
$$

$$g_{uk} = \sum_{i=1}^{n} e^{S_i + \boldsymbol{U}_i \boldsymbol{W}_k'},$$

$$g_{vk} = \sum_{j=1}^{n} e^{R_j + \boldsymbol{V}_j \boldsymbol{W}_k'}. \tag{1}$$

The model is always conditioning on ($\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$), which are omitted here for lack of space. Conditioning on the latent features of the actors and latent edge classes, the edges are independent. The conditional distribution of the probability of an edge m connecting actors i and j is a product of two multinomial distributions, where g_{uk} and g_{vk} are the normalizing constants. The first multinomial distribution corresponds to the probability of choosing the origin of a directed edge among the n actors. Conditioning on the choice of the origin, the second distribution corresponds to the probability of selecting the destination of the directed edge among the remaining $n-1$ actors. The formulation of the LSEC model implies that an edge in a specific environment will be more likely to be incident on certain actors. Those actors not only have high general propensity to send/receive edges, which is captured via ($\boldsymbol{S}, \boldsymbol{R}$), but also share some features with the edge's environment. The dot product between the actors' latent features ($\boldsymbol{U}, \boldsymbol{V}$) and the edges' latent features ($\boldsymbol{W}$) reflects the latter phenomenon.

We adopt a Bayesian approach to estimation, using the same priors as suggested in Sewell (2021) for $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{S}, \boldsymbol{R}, \boldsymbol{W}, \tau_U, \tau_V, \tau_S, \tau_R\}$, which are given by

$$\begin{aligned}
\boldsymbol{U}_i &\overset{\text{iid}}{\sim} N(\boldsymbol{0}, \tau_U^{-1}\mathcal{I}_p), & \tau_U &\sim \Gamma(a_U/2, b_U/2),\\
\boldsymbol{V}_i &\overset{\text{iid}}{\sim} N(\boldsymbol{0}, \tau_V^{-1}\mathcal{I}_p), & \tau_V &\sim \Gamma(a_V/2, b_V/2),\\
S_i &\overset{\text{iid}}{\sim} N(0, \tau_S^{-1}), & \tau_S &\sim \Gamma(a_S/2, b_S/2),\\
R_i &\overset{\text{iid}}{\sim} N(0, \tau_R^{-1}), & \tau_R &\sim \Gamma(a_R/2, b_R/2),\\
\boldsymbol{W}_k &\overset{\text{iid}}{\sim} N(\boldsymbol{0}, \mathcal{I}_p), &&
\end{aligned} \tag{2}$$

where $N(\boldsymbol{a}, \boldsymbol{B})$ is a multivariate normal distribution with mean vector $\boldsymbol{a}$ and covariance matrix $\boldsymbol{B}$, $\Gamma(a, b)$ is a gamma distribution with shape a and rate b, and $\mathcal{I}_p$ is a p-dimensional identity matrix. The identity matrix is used as the covariance of $\boldsymbol{W}_k$'s to avoid non-identifiability due to the dot products with $\boldsymbol{U}_i$'s and $\boldsymbol{V}_i$'s in the model likelihood. The relationship between parameters in the LSEC model is illustrated through a directed acyclic graph in Figure 1, along with the prior on $\boldsymbol{Z}$ for our proposed approach which will be described in the following section.

Since the number of components K and number of latent dimensions p are unknown, Sewell recommended fitting a model for each combination of plausible $\{K, p\}$ and choosing the best model according to some information criteria. However, obtaining estimation for each candidate model is challenging and time-consuming, given that the parameter space is high dimensional ($2n + 2np + Kp + 4$; in addition to the dimension of $\boldsymbol{Z}$ and its prior's hyperparameters). In what follows, we propose a different strategy that would fit the model only once in order to drastically save time and resources.

2.2 Sparse finite mixture models

Finite mixture models provide a flexible and computationally convenient framework to describe many complex distributions and perform clustering (McLachlan et al., 2019). To overcome the problem of choosing the appropriate number of components of a mixture distribution, researchers can use an overfitted model where the number of components exceeds what is supported by the

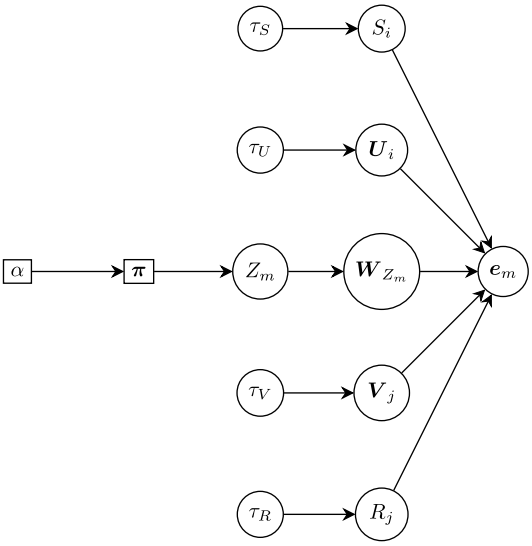

Figure 1. Relationship between parameters in the LSEC model (circles) and parameters in the proposed extension (rectangles).

data (Rousseau & Mengersen, 2011). Based on the theoretical results of overfitted mixture models derived by Rousseau & Mengersen (2011), Malsiner-Walli et al. (2016) formulated the concept of the sparse finite mixture model (SFMM) under the Bayesian framework of model-based clustering. SFMMs share many similar properties with other Bayesian non-parametric mixture models that assume an infinite number of components, including automatic detection of the number of clusters (Malsiner-Walli et al., 2017). However, SFMMs operate within the framework of finite mixtures and, therefore, come with a straightforward estimation strategy (Malsiner-Walli et al., 2016). While SFMMs were initially developed assuming Gaussian mixtures, Frühwirth-Schnatter & Malsiner-Walli (2019) further demonstrated how the method could be adapted to other applications of non-Gaussian mixtures, such as Poisson mixtures, latent class analysis, and mixtures of skew-normal distributions.

Assuming the data is generated from a finite mixture model, observations corresponding to the same component form a cluster. The key idea behind SFMM is to deliberately overfit the mixture models so that there are components that do not associate with any observation (Malsiner-Walli et al., 2016). As a result, the total number of mixture components K is greater than the true number of clusters in the data K_{true}. While K is fixed, K_{true} is a random variable and can take on values smaller than K with a high probability granted an appropriate choice of prior (Malsiner-Walli et al., 2016, 2017; Frühwirth-Schnatter & Malsiner-Walli, 2019). Given a prior on α that puts a large probability mass on small values near zero, the Dirichlet prior of π will show a strong preference for zero weights, resulting in empty clusters among the K components of the mixture models a priori (Frühwirth-Schnatter & Malsiner-Walli, 2019).

Inspired by the sparse finite mixture model approach, we assume the priors for $\boldsymbol{Z}$ to evaluate cluster assignment of the edges (Malsiner-Walli et al., 2016, 2017; Frühwirth-Schnatter & Malsiner-Walli, 2019) to be

$$\begin{aligned} \boldsymbol{Z}_m|\boldsymbol{\pi} &\overset{\text{iid}}{\sim} \text{Multinom}(1, \boldsymbol{\pi}) \\ \boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha \mathbf{1}_K) \\ \alpha &\sim \Gamma(a_a, b_a), \end{aligned} \tag{3}$$

where Multinorm$(1, \boldsymbol{a})$ is a multinomial distribution with 1 trial and event probabilities $\boldsymbol{a}$, Dirichlet$(\boldsymbol{a})$ is a Dirichlet distribution with concentration $\boldsymbol{a}$, $\mathbf{1}_K$ is a K-dimensional vector of 1's, and K is set intentionally large as an upper bound on the number of true clusters. In our model, the multinomial prior on the cluster label $\boldsymbol{Z}$ implies that the marginal likelihood of the edges follows a mixture distribution with weight $\boldsymbol{\pi}$.

$$f(\mathcal{E}_i) = \sum_{k=1}^{K} \pi_k \frac{\exp\{S_{\boldsymbol{e}_{m1}} + R_{\boldsymbol{e}_{m2}} + (\boldsymbol{U}_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m2}})\boldsymbol{W}_k'\}}{f_{uk}(f_{vk} - \exp\{R_{\boldsymbol{e}_{m1}} + \boldsymbol{V}_{\boldsymbol{e}_{m1}}\boldsymbol{W}_k'\})} \tag{4}$$

The SFMM is closely related to the DPMM. When the hyperparameter α is set to α_0/K and K goes to infinity, the SFMM converges to a DPMM with mixing distribution $DP(\alpha_0, H)$, where H is the base measure from which $\boldsymbol{W}_k$ are drawn *iid* (Malsiner-Walli et al., 2016). While the a priori expected number of clusters K_{true} is solely determined by the concentration parameter in the DPMM, it is also impacted by the number of mixture components K in the SFMM (Malsiner-Walli et al., 2017). One may seek to draw a connection between SFMM and other two-parameter Bayesian non-parametric mixture models, such as the Pitman-Yor Process Mixture Model (PYPMM) (Pitman & Yor, 1997). Malsiner-Walli et al. (2017) provided a detailed comparison of SFMM against both the DPMM and PYPMM. In SFMM, the prior probability of creating new clusters decreases with the increasing number of non-empty clusters, whereas it remains constant in the DPMM and increases in PYPMM (Malsiner-Walli et al., 2017). Additionally, as the sample size increases, the a priori expected number of clusters K_{true} increases in Baysian non-parametric mixture models including the DPMM and PYPMM, but this value remains constant in SFMM (Malsiner-Walli et al., 2017).

Researchers have studied the impact of the hyperparameter α values on the number of clusters in overfitted mixtures models. Using a fixed value of α instead of the Gamma hyperprior, Rousseau & Mengersen (2011) presented some theoretical justification for the upper bounds of α that is necessary to empty superfluous clusters. In particular, if $\alpha < d/2$, where d is the dimension of the component-specific parameter space, then the posterior expectation of the weights of superfluous clusters converges to zero as the sample size goes to infinity. Otherwise, there will exist at least two identical components with non-negligible weights, leading to overestimating the number of clusters due to these duplicated components (Rousseau & Mengersen, 2011). In practice, when we have a finite number of observations, it is necessary to select much smaller values for α than the proposed upper bound $d/2$ (Malsiner-Walli et al., 2016). We opted for using a Gamma hyperprior on α instead of fixing α at a small value based on evidence from Frühwirth-Schnatter & Malsiner-Walli (2019). In our analyses, we used their recommended hyperprior $\Gamma(1, 200)$ and truncation level $K = 10$ because Frühwirth-Schnatter & Malsiner-Walli showed that these specifications had good quality in emptying superfluous clusters in both the simulation study and real data analysis.

2.3 Estimation via variational Bayes generalized EM algorithm

We adopt a generalized expectation-maximization (GEM) approach to estimation, where the E step involves evaluating the conditional expectation of $(\boldsymbol{Z}, \boldsymbol{\pi})|\mathcal{E}$ and the M step entails performing conjugate gradient updates to maximize Q with respect to $\boldsymbol{\theta} := \{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \tau_S, \tau_R, \tau_U, \tau_V, \alpha\}$. In most GEM applications in the finite mixture models, $\boldsymbol{Z}$ is assumed to be the latent variable in the E step. Other variables, including the mixture weights, are chosen to maximize the expected log posterior in the M step. With this setup, the E step is tractable. However, in the case of the overfitted or sparse mixture models, the posterior modes of the mixture weights do not exist due to the anticipated empty components. An alternative option is to integrate out the weights, yet this renders the E step intractable. The expectation of $\boldsymbol{Z}$, as a result, requires approximation via either MCMC or variational Bayes (VB) method. VB is generally fast, but it can still impose a hefty computational cost in the absence of analytical solutions, which is, unfortunately, the case

here. Instead, we consider both $\boldsymbol{Z}$ and $\boldsymbol{\pi}$ as the latent variables in the E step and use the following variational approximation of the conditional posterior:

$$f(\boldsymbol{Z}, \boldsymbol{\pi} | \mathcal{E}, \tilde{\boldsymbol{\theta}}) \approx q(\boldsymbol{Z}) q(\boldsymbol{\pi}) = \left(\prod_{m=1}^{M} q(\boldsymbol{Z}_m) \right) q(\boldsymbol{\pi}). \tag{5}$$

The variational distributions are selected by minimizing the Kullback-Leibler divergence between $q(\boldsymbol{Z})q(\boldsymbol{\pi})$ and $f(\boldsymbol{Z}, \boldsymbol{\pi} | \mathcal{E}, \tilde{\boldsymbol{\theta}})$, which is the same as maximizing the evidence lower bound (ELBO) of the log marginal likelihood defined as

$$\text{ELBO} := E_{q(\boldsymbol{Z},\boldsymbol{\pi})} \left(\log f(\boldsymbol{\theta}, \boldsymbol{Z}, \boldsymbol{\pi} | \mathcal{E}) \right) - E_{q(\boldsymbol{Z},\boldsymbol{\pi})} \left(\log q(\boldsymbol{Z}, \boldsymbol{\pi}) \right). \tag{6}$$

Performing coordinate ascent to maximize the ELBO yields the following solutions, where $\boldsymbol{Z}_{-m}$ is the set of $\boldsymbol{Z}$'s that does not include $\boldsymbol{Z}_m$ (Wang & Blei, 2013; Beal, 2003).

$$q(\boldsymbol{Z}_m) \propto e^{E_{\boldsymbol{\pi}}(\log \pi(\boldsymbol{Z}_m, \mathcal{E} | \boldsymbol{\pi}, \boldsymbol{Z}_{-m} \tilde{\boldsymbol{\theta}}))}$$

$$q(\boldsymbol{\pi}) \propto e^{E_{\boldsymbol{Z}}(\log \pi(\boldsymbol{\pi}, \mathcal{E} | \boldsymbol{Z}, \tilde{\boldsymbol{\theta}})} \tag{7}$$

We then iterate between conditioning on either $\boldsymbol{Z}$ or $\boldsymbol{\pi}$ and updating the other until convergence. Given the multinomial-Dirichlet priors, the updates have analytical forms. In particular, the variational distribution $q(\boldsymbol{Z}_m)$ is a multinomial distribution with weights $\tilde{p}_{mk}$ and $q(\boldsymbol{\pi})$ is a Dirichlet distribution with concentration parameters $\tilde{\alpha}_k$, where $\tilde{p}_{mk}, \tilde{\alpha}_k$ are given by

$$\tilde{p}_{mk} \propto \left(\frac{e^{\boldsymbol{S}_{e_{m1}} + \boldsymbol{U}_{e_{m1}} \boldsymbol{W}_k'}}{g_{uk}} \right) \left(\frac{e^{\boldsymbol{R}_{e_{m2}} + \boldsymbol{V}_{e_{m1}} \boldsymbol{W}_k'}}{g_{vk} - e^{\boldsymbol{R}_{e_{m1}} + \boldsymbol{V}_{e_{m1}} \boldsymbol{W}_k'}} \right) e^{\psi(\tilde{\alpha}_k) - \psi(\sum_k \tilde{\alpha}_k)}$$

$$\tilde{\alpha}_k = \alpha + \sum_{m=1}^{M} \tilde{p}_{mk}, \tag{8}$$

where ψ is the digamma function.

We take a coordinate ascent approach to incrementally increase the Q function given each E step's output in the M step. The value of α to maximize Q is obtained using a combination of golden section search and successive parabolic interpolation (Brent, 2005). The updates of $\{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \tau_S, \tau_R, \tau_U, \tau_V\}$ are performed in the same manner as in Sewell (2021), with analytical solutions for the value of $\{\tau_S, \tau_R, \tau_U, \tau_V\}$ and conjugate gradient updates for $\{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$. Additionally, we use the same trick for computing the gradient in the LSEC paper to achieve a computational cost that grows linearly with the number of actors in a network. For details, see Appendix A.

To initialize the VB-GEM algorithm, we randomly generated $\boldsymbol{S}$ and $\boldsymbol{R}$ from a multivariate normal distribution with mean $\boldsymbol{0}$ and variance $\mathcal{I}_n$. We randomly sampled each row of W from a multivariate normal distribution with mean $\boldsymbol{0}$ and variance $\mathcal{I}_p$. We initialized U and V using results from performing spectral embedding on the graph adjacency matrix. We initialized α at the prior mean a_α / b_α ($= 1/200$ in our analyses). Convergence in the E step was evaluated by monitoring the changes in the ELBO. Convergence of the whole algorithm was determined based on the number of edges switching clusters between iterations. If 99.9% of the edges keep the same cluster assignment after the previous iteration, the algorithm will stop at the current iteration. To mitigate the fact that the performance of the general EM algorithm depends on the quality of the initialization, in the simulation study and real data analyses that follow, we ran the algorithm using 15 random starting points and chose the best model based on ICL (Biernacki et al., 2000). Additionally, we set the hyperparameters such that $a_U = b_U = a_S = b_S = a_V = b_V = a_R = b_R = 1$. While researchers can set these values depending on the problem at hand, we found that varying them had minimal impact on the result (see Appendix C).

2.4 Gradient-based Monte Carlo

While the GEM approach quickly produces point estimates, it lacks uncertainty estimation. In this section, we introduce a Hamiltonian Monte Carlo-within-Gibbs algorithm to obtain estimates and quantify uncertainty. The full log posterior is given by

$$
\begin{aligned}
l \propto & \sum_{m=1}^{M}\sum_{k=1}^{K} Z_{mk}\left[S_{e_{m1}} + R_{e_{m2}} + (\boldsymbol{U}_{e_{m1}} + \boldsymbol{V}_{e_{m2}})\boldsymbol{W}_k' - \log(f_{uk}) - \log(f_{vk} - e^{R_{e_{m1}} + \boldsymbol{V}_{e_{m1}}\boldsymbol{W}_k'})\right] \\
& + \log\Gamma(K\alpha) - K\log\Gamma(\alpha) + \sum_{k=1}^{K}(\alpha + \sum_{m=1}^{M} Z_{mk} - 1)\log\pi_k + (a_a - 1)\log\alpha - b_a\alpha \\
& - \frac{\tau_S}{2}||\boldsymbol{S}||^2 - \frac{\tau_R}{2}||\boldsymbol{R}||^2 - \frac{\tau_U}{2}||\boldsymbol{U}||_F^2 - \frac{\tau_V}{2}||\boldsymbol{V}||_F^2 - \frac{1}{2}||\boldsymbol{W}||_F^2 \\
& + \left(\frac{a_S + n}{2} - 1\right)\log\tau_S - \frac{b_S}{2}\tau_S + \left(\frac{a_R + n}{2} - 1\right)\log\tau_R - \frac{b_R}{2}\tau_S \\
& + \left(\frac{a_U + nD}{2} - 1\right)\log\tau_U - \frac{b_U}{2}\tau_U + \left(\frac{a_V + nD}{2} - 1\right)\log\tau_V - \frac{b_V}{2}\tau_V. \qquad (9)
\end{aligned}
$$

Due to semi-conjugacy, we can derive the full conditionals for $\boldsymbol{Z}_m$, which is a multinomial distribution with probabilities proportional to

$$
\pi_k \frac{e^{S_{e_{m1}} + R_{e_{m2}} + (\boldsymbol{U}_{e_{m1}} + \boldsymbol{V}_{e_{m2}})\boldsymbol{W}_k'}}{f_{uk}(f_{vk} - e^{R_{e_{m1}} + \boldsymbol{V}_{e_{m1}}\boldsymbol{W}_k'})}. \qquad (10)
$$

Similarly, we derive the full conditionals of $\boldsymbol{\pi}$ and the precision parameters as follows.

$$
\begin{aligned}
\boldsymbol{\pi}|\cdot &\sim \text{Dirichlet}\left(\alpha + \sum_{m=1}^{M}\boldsymbol{Z}_{m1}, \ldots, \alpha + \sum_{m=1}^{M}\boldsymbol{Z}_{mK}\right) \\
\tau_S|\cdot &\sim \Gamma\left(\frac{a_S + n}{2}, \frac{b_S + ||\boldsymbol{S}||^2}{2}\right) \\
\tau_R|\cdot &\sim \Gamma\left(\frac{a_R + n}{2}, \frac{b_R + ||\boldsymbol{R}||^2}{2}\right) \\
\tau_U|\cdot &\sim \Gamma\left(\frac{a_U + np}{2}, \frac{b_U + ||\boldsymbol{U}||_F^2}{2}\right) \\
\tau_V|\cdot &\sim \Gamma\left(\frac{a_V + np}{2}, \frac{b_V + ||\boldsymbol{V}||_F^2}{2}\right) \qquad (11)
\end{aligned}
$$

For the remaining parameters $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{R}, \boldsymbol{S}, \boldsymbol{W}, \alpha)$, we use a Hamiltonian Monte Carlo (HMC) algorithm to draw from the posterior. Since the posterior samples of $\boldsymbol{\pi}$ might contain zero values, which result in an undefined log posterior, we take a collapsed Gibbs sampling approach and integrate out π from the full conditionals of $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{S}, \boldsymbol{R}, \boldsymbol{W}, \alpha)$ (Van Dyk & Park, 2008). The gradients with respect to $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{R}, \boldsymbol{S}, \boldsymbol{W})$ can be computed in the same efficient manner as in the LSEC paper. Further details on these gradients can be found in Appendix B. Since α is strictly positive, we apply a log transformation to α before carrying out the HMC algorithm. Let $\lambda := \log\alpha$, then the derivative of the reduced log posterior, which is the log posterior after integrating out $\boldsymbol{\pi}$, with

respect to λ is given by

$$Ke^{\lambda}\psi(Ke^{\lambda}) - Ke^{\lambda}\psi(e^{\lambda}) + e^{\lambda}\sum_{k=1}^{K}\psi(e^{\lambda} + \sum_{m=1}^{M}\boldsymbol{Z}_{mk}) - Ke^{\lambda}\psi(Ke^{\lambda} + M) + a_a - b_a e^{\lambda}. \quad (12)$$

The MCMC output suffers from several non-identifiablity issues. The likelihood does not change if one rescales $\boldsymbol{W}$ by a constant c and then rescales $\boldsymbol{U}$ and $\boldsymbol{V}$ by $1/c$. Additionally, the likelihood is invariant to translations of $\boldsymbol{S}$, $\boldsymbol{R}$, and the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$. Like other latent space approaches, the latent actor positions $\boldsymbol{U}$, $\boldsymbol{V}$ and edge positions $\boldsymbol{W}$ are invariant to rotations, reflections, and translations. Lastly, the sparse finite mixture prior introduces the problem of aliasing of cluster labels and components.

To circumvent the non-identifiability issues, we post-process the MCMC output in six steps. Let the $^{(l)}$ superscript denote the lth MCMC sample. First, we rescale $\boldsymbol{W}^{(l)}$ so that the elements of $\boldsymbol{W}^{(l)}$ have unit variance and then rescale $\boldsymbol{U}^{(l)}$, $\boldsymbol{V}^{(l)}$, $\tau_U^{(l)}$, and $\tau_V^{(l)}$ accordingly. Second, we keep draws where the number of clusters equals the MAP estimate and discard the rest (Malsiner-Walli et al., 2016). Third, we remove the rows in $\boldsymbol{W}$ and elements in $\boldsymbol{\pi}$ corresponding to empty components in each sample (Malsiner-Walli et al., 2016). Fourth, we run the equivalence classes representatives algorithm to match cluster assignment in each sample to the MAP estimate and then reorder rows of $\boldsymbol{W}$ and elements in π according to permuted cluster label (Papastamoulis & Iliopoulos, 2010). Fifth, we apply a Procrustes transformation (Borg & Groenen, 2005) to rotate $\boldsymbol{U}^{(l)}$, $\boldsymbol{V}^{(l)}$ and $\boldsymbol{W}^{(l)}$ using the MAP as the reference matrix. Finally, we recenter $\boldsymbol{S}^{(l)}$ and $\boldsymbol{R}^{(l)}$ and the columns of $\boldsymbol{U}^{(l)}$ and $\boldsymbol{V}^{(l)}$ in each sample.

3. Simulation

We performed a simulation study to compare the clustering performance of our proposed algorithm to the existing approaches described below. There were twenty different scenarios assuming a different combination of the number of actors $n \in \{200, 400\}$, the true K (ranging from 2 to 6), and p (ranging from 2 to 3). For each scenario, we generated 200 network data sets. Each network had a density of 0.05, which translated to 1990 edges when $n = 200$ and 7980 edges when $n = 400$.

Given that indegree and outdegree are often highly correlated, each pair (S_i, R_i) was sampled from a bivariate normal distribution with mean $\mathbf{0}$, variances 2 and correlation 0.75. The matrix representing the edge cluster latent positions $\boldsymbol{W}$ was generated such that its row vectors $\boldsymbol{W}_k$'s were equally spaced on a circle when the true $p = 2$ and on a sphere when the true $p = 3$. The directions of the row vectors $\boldsymbol{U}_i$'s and $\boldsymbol{V}_i$'s were drawn from mixtures of von Mises-Fisher distributions with the mean directions coinciding with the direction of a $\boldsymbol{W}_k$ chosen with equal probability. Their magnitudes followed a Gamma distribution with shape 20 and rate 4, which we found from trial and error to encourage a reasonable level of clustering in the resulting networks. Across the twenty configurations, we set the concentrations of the mixtures of von Mises-Fisher distributions to achieve a similar degree of clustering in the network structure. This was measured by obtaining similar average values of edge assortativity (that is, the assortativity of the line graph using the edge cluster assignments as attributes), ranging between 0.6 and 0.8 across the ten different scenarios. Each edge cluster assignment $\boldsymbol{Z}_m$ was drawn from a multinomial distribution with equal event probabilities of $1/K$ where K was the correct number of clusters. An example of a simulated network when $(n = 400, K = 6, p = 2)$ is shown in Figure 2, where different edge colors correspond to different clusters.

Clustering results of the proposed approach were compared with other methods, including the original LSEC (Sewell, 2021), the mixed-membership blockmodeling method proposed by Ball et al. (2011) (BKN), and spectral clustering on a line graph (LG). Our approach always assumed that $p = 3$. We describe later a sensitivity analysis which shows that this appears to be a sound practice.

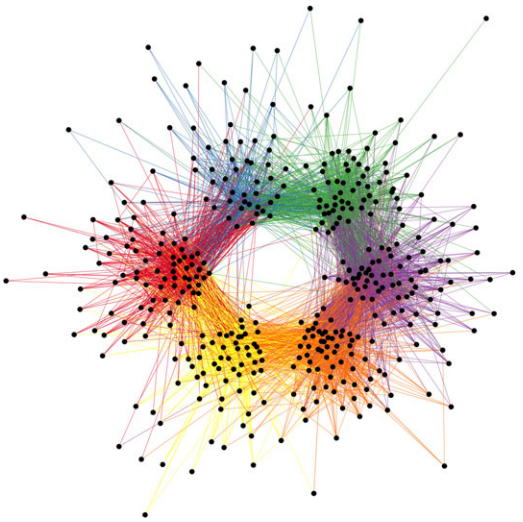

Figure 2. Example of simulated network where $n = 400, K = 6, p = 2$. Edges are colored according to their cluster assignment. The network layout is set to show the six different edge clusters clearly.

The results from LSEC were based on running the double model selection approach in Sewell (2021), performing estimation over a range of values for K (ranging from 2 to 10) and p (ranging from 2 to 5) and using both AIC and ICL to select a final model. There were no obvious ways to choose K and p for BKN and LG, and hence, we fit the oracle model assuming the true K and p were known. It is important to note that the results from BKN and LG are overestimates due to both of these methods being provided the true K and p. We refer to our proposed algorithm as the *automated* LSEC or aLSEC.

We evaluated clustering performance via normalized mutual information (NMI) (Danon et al., 2005) and adjusted Rand index (ARI) (Hubert & Arabie, 1985). The results are summarized in Table 1. Our proposed aLSEC method outperformed BKN and LG in all scenarios except when $K = 2$, despite both of these methods being provided knowledge of the true K and p. Compared to LSEC, aLSEC had similar or improved performance. The primary benefit of aLSEC over LSEC is the reduced computational cost. We investigated the ratio of running time of LSEC over aLSEC. We found that implementing aLSEC can reduce the run time by 10 to over 100 times compared to fitting the original LSEC model (Figure 3). In each simulation scenario, the majority of this ratio fell between 20 and 70 when $n = 200$, and between 20 and 40 when $n = 400$ (Figure 3).

We carried out a sensitivity analysis and found that there was little harm in overestimating the dimension of the latent space, especially when the actual number of clusters was large (Table 2). This finding is not overly surprising since setting p to be larger than the actual value has the potential to perfectly capture the latent space information embedded in a lower-dimensional space. We fixed the latent dimension to be $p = 3$ when fitting aLSEC in the simulation study. In practice, we would wish to visualize the latent space and save resources testing different values of $p's$ while still obtaining reasonably good clustering results.

4. Real network data analyses

4.1 Patient transfer network

We explored disease transmission in a patient transfer network between hospitals in California, which had 372 nodes and 12,853 directed edges (Justice et al., 2022). Each node in the network

Table 1. Clustering results of comparing aLSEC (assuming $p = 3$) with the existing methods. For both NMI and ARI, the higher the values, the better

True			NMI				ARI			
n	*K*	*p*	aLSEC	LSEC	BKN	LG	aLSEC	LSEC	BKN	LG
200	2	2	0.87	**0.99**	0.90	0.85	0.88	**0.99**	0.95	0.91
	3	2	**0.97**	**0.97**	0.91	0.88	**0.98**	**0.98**	0.94	0.92
	4	2	**0.90**	0.87	0.82	0.81	**0.93**	0.90	0.85	0.84
	5	2	**0.81**	0.80	0.74	0.80	**0.81**	**0.81**	0.74	0.80
	6	2	**0.70**	**0.70**	0.65	0.62	0.64	**0.65**	0.58	0.53
	2	3	0.84	**0.98**	0.87	0.84	0.84	**0.98**	0.93	0.91
	3	3	**0.88**	**0.88**	0.82	0.78	**0.92**	**0.92**	0.87	0.82
	4	3	0.81	**0.82**	0.77	0.78	0.80	**0.82**	0.75	0.77
	5	3	**0.78**	0.77	0.74	0.73	0.68	**0.69**	0.68	0.63
	6	3	**0.77**	0.75	0.73	0.74	**0.64**	0.63	**0.64**	0.61
400	2	2	0.95	**0.99**	0.95	0.90	0.95	**0.99**	0.98	0.95
	3	2	0.96	**0.97**	0.90	0.84	0.97	**0.98**	0.94	0.90
	4	2	**0.91**	0.89	0.83	0.78	**0.94**	0.92	0.86	0.80
	5	2	**0.81**	**0.81**	0.73	0.71	**0.82**	**0.82**	0.72	0.67
	6	2	0.71	**0.72**	0.66	0.61	**0.66**	**0.66**	0.60	0.52
	2	3	0.91	**0.98**	0.94	0.90	0.91	**0.99**	0.97	0.95
	3	3	0.88	**0.89**	0.82	0.76	0.92	**0.93**	0.87	0.78
	4	3	**0.84**	0.82	0.76	0.72	**0.87**	0.82	0.74	0.66
	5	3	**0.81**	0.79	0.77	0.75	**0.78**	0.76	0.72	0.64
	6	3	**0.78**	0.76	0.74	0.75	**0.68**	0.67	0.67	0.62

The highest values of NMI and ARI in each scenario are bolded.

represented a hospital. There was an edge from node i to node j if more than two patients were transferred from hospital i to hospital j during the seven years of the study. The threshold (e.g., two patients) was chosen based on a visual examination of the relationship between different cutoffs and the density of the resulting network graph to remove superfluous patient transfers from the network. The patient transfer data are available from the Healthcare Cost and Utilization Project (HCUP) State Inpatient Databases (SID).[2] HCUP data are available for a fee to all researchers following a standard application process and signing of a data use agreement.

Transfer patients have been shown to have significantly contributed to infectious disease outbreaks (Donker et al., 2012). Therefore, understanding the system of flows in this network will be helpful for public health officials and healthcare professionals to identify which transfer patients to screen or isolate in such events. For example, if a transfer patient, which corresponds to an edge in this network, tests positive for an infectious disease, it would be of great importance for hospitals within the region to know which other transfer patients to screen and isolate to prevent further spread.

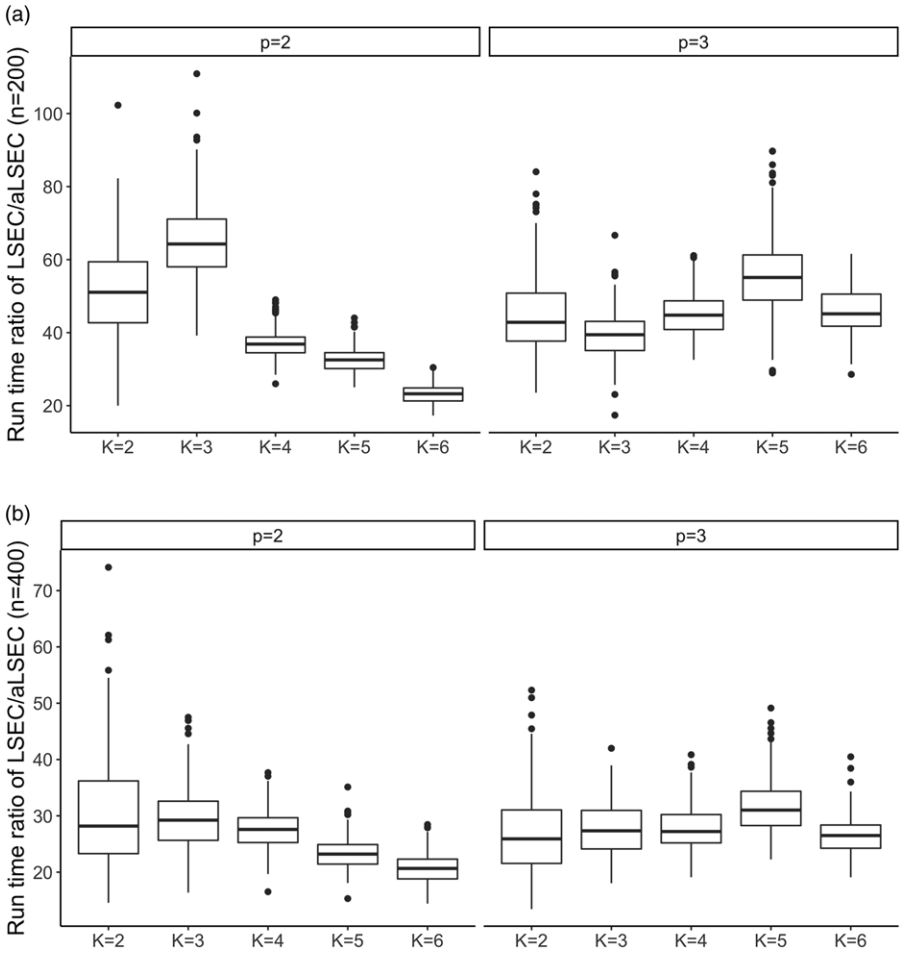

Figure 3. The ratios of the run time of the LSEC method over our aLSEC method.

Running the GEM algorithm with $p = 3$ revealed six clusters, shown in Figure 4. The MAP estimates of the latent position $\boldsymbol{U}$'s were shown along with the edge cluster assignments. Since there was no ground truth in this data set, we carried out epidemic simulations similar to those done by Sewell (2021) and examined the results. Each simulation started with infecting one hospital and allowed the transmission to spread via patient transfers between hospitals in the network. The period an infected hospital remains exposed was generated from a gamma distribution with a mean of 7 days and a standard deviation of 1 day. After exposure, the infected hospital became infectious and could transmit disease to other hospitals. The probability of a transfer patient carrying disease was assumed to be 0.1.

Figure 5(a) shows one archetypal example of the epidemic simulation results, noting that all other simulations were qualitatively similar. At each time point t, we calculated the cumulative number of edges in each cluster that had contributed to spreading infection. Once the disease entered an edge cluster, there was a rapid increase in the number of edges carrying transmission, followed by a plateau indicating that most of the edges in the group had already acted to transmit the disease. In contrast, Figure 5(b) shows the same epidemic simulation but with the cluster label randomly permuted. The increase in the number of edges carrying disease within each edge cluster was more gradual, implying that the permuted cluster labels failed to capture the systems of flows in the network.

Table 2. Sensitivity analysis assuming different values of p for aLSEC (represented by the different columns). The NMI values are averaged over 200 simulated networks. The higher the NMI values, the better

True		$n = 200$				$n = 400$			
K	p	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
2	2	1.00	0.87	0.87	0.91	1.00	0.95	0.92	0.93
3	2	0.95	0.97	0.93	0.93	0.93	0.96	0.93	0.92
4	2	0.79	0.90	0.90	0.89	0.80	0.91	0.90	0.89
5	2	0.70	0.81	0.82	0.83	0.70	0.81	0.82	0.82
6	2	0.57	0.70	0.71	0.71	0.57	0.71	0.72	0.72
2	3	0.99	0.84	0.82	0.86	0.99	0.91	0.86	0.85
3	3	0.89	0.88	0.85	0.85	0.90	0.88	0.85	0.83
4	3	0.67	0.81	0.85	0.84	0.78	0.84	0.84	0.82
5	3	0.74	0.78	0.79	0.80	0.77	0.81	0.82	0.82
6	3	0.69	0.77	0.78	0.78	0.75	0.78	0.79	0.79

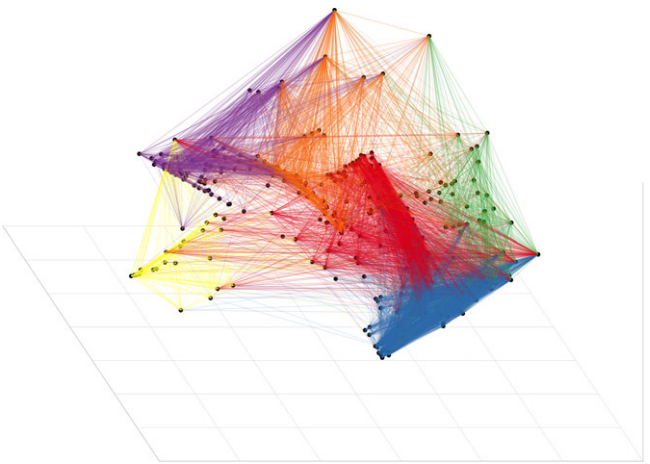

Figure 4. Results of applying aLSEC to the patient transfer network of California. Each point represents the MAP estimate latent position $\boldsymbol{U}$ of each actor. Edges are colored according to their cluster assignment.

4.2 UK faculty data

We applied the proposed method to a friendship network of UK university faculty, which consists of 81 nodes and 817 directed edges (Nepusz et al., 2008). Faculty members came from three different schools within the university. The school affiliation was known for all but two individuals. The UK Faculty data are publicly available as part of the `igraphdata` package in R (Csardi, 2015). The code to perform data analysis can be found online.[3]

We ran the HMC-within Gibbs algorithm on the UK faculty network, keeping every fifth iteration and obtaining 30, 000 samples. We then discarded the first 10, 000 samples, which we considered as a burn-in period. Figure 6 shows the posterior distribution of the number of clusters.

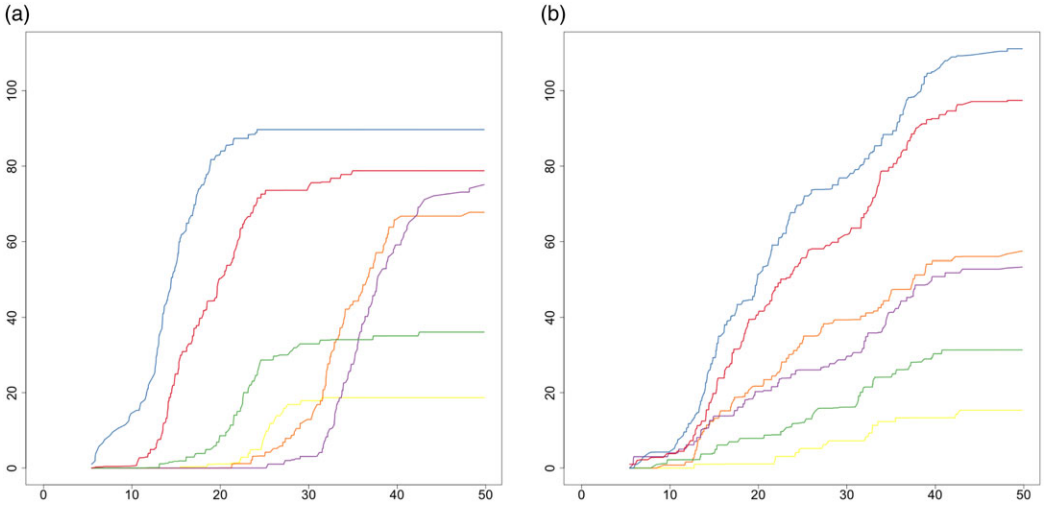

Figure 5. Results of running epidemic simulation on the network with edges labeled according to aLSEC results (a) and randomly (b). Each curve showed the number of edges carrying the transmission of a cluster over time.

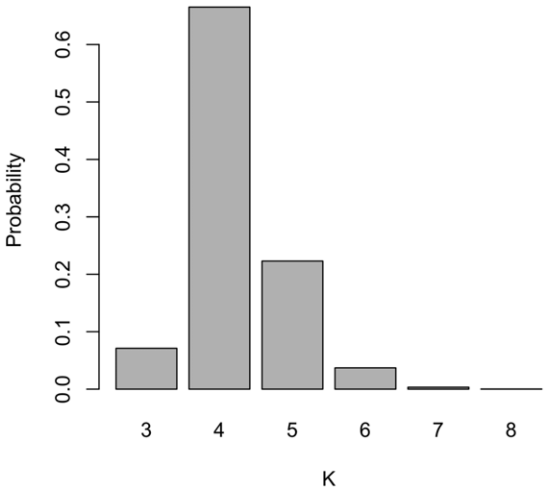

Figure 6. Posterior distribution of the number of clusters K after applying HMC-within-Gibbs algorithm to UK Faculty network.

Unlike the original LSEC approach, our approach allows for estimation of uncertainty associated with the estimated number of clusters K. There is an overwhelming posterior probability supporting $K = 4$.

After applying the post-processing steps, we examined the clustering results where the number of clusters $K = 4$. Figure 7 shows the MAP estimate of the latent position $\boldsymbol{U}^4$ of each actor, while edges are colored according to the MAP edge partition. Each shape of the node corresponds to each school to which each faculty belongs. The first three edge clusters in Figure 7(a), (b), and (c) mainly captured the within-school connections, while the last cluster Figure 7(d) represented the interaction across different schools. The results indicate that aLSEC is uncovering the hidden contexts in which the friendships are formed.

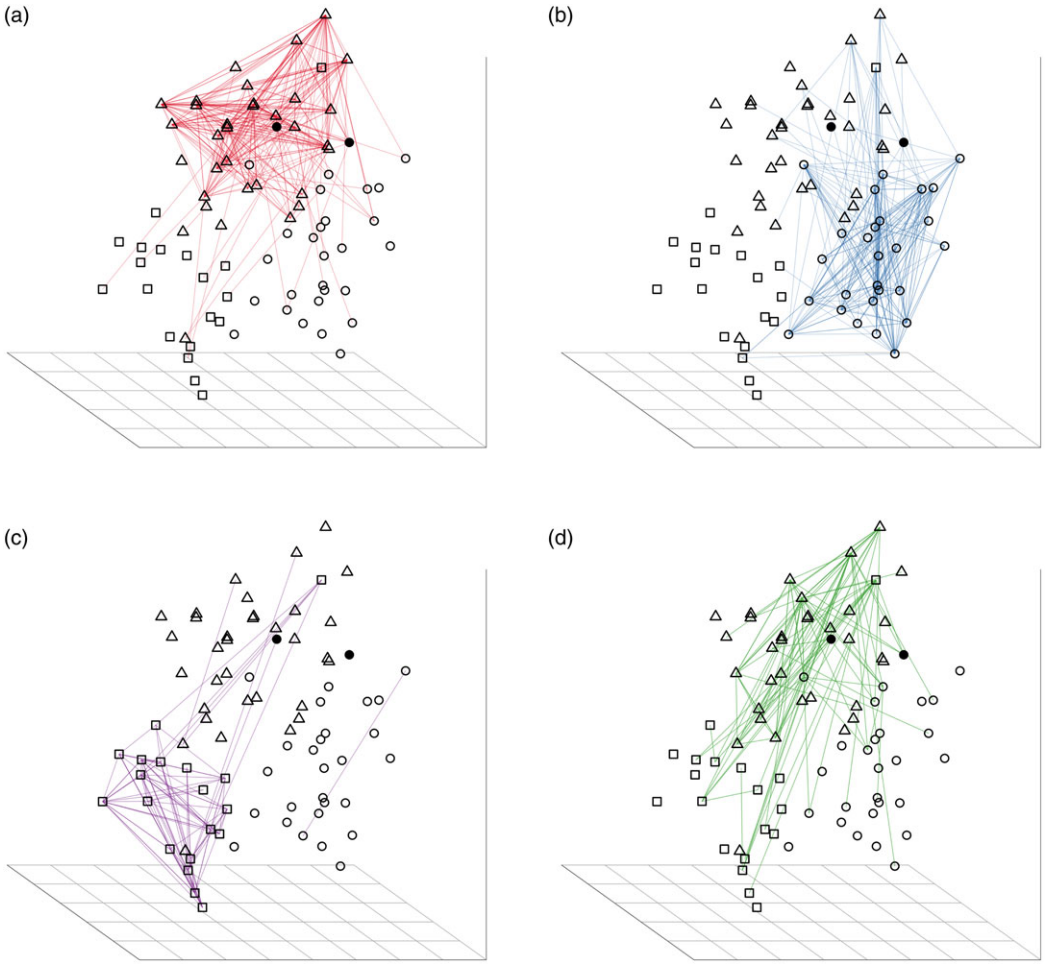

Figure 7. The MAP estimate of the latent positions $\boldsymbol{U}$. Edge colors correspond to the MAP edge partition. The three hollow shapes represent the three schools. The two solid circles represent the two individuals who did not mention their schools.

Our MCMC algorithm allows us to quantify the uncertainty in any model parameter, including the cluster assignment. Let $P_{M\times M}$ be a matrix where each entry P_{ij} represents the posterior probabilities of any two edges i and j sharing the same cluster. Let L denote the total number of post-processed MCMC samples, we calculated P_{ij} as follows.

$$P_{ij} = \frac{1}{L}\sum_{l=1}^{L} \mathbf{1}_{[\boldsymbol{Z}_i^{(l)}=\boldsymbol{Z}_j^{(l)}]}$$

Since we did not know the school affiliations of two individuals, we momentarily excluded these two actors and their corresponding connections. We then reordered the rows and columns of P into four blocks, where each of the first three blocks consisted of edges corresponding to within-school connections and the fourth block consisted of the remaining edges representing inter-school connections. The sizes of the four blocks were 317, 250, 96, and 152, respectively. Figure 8 is a heat map showing the posterior probabilities of edges sharing clusters after reordering, where darker colors indicate higher probabilities. A block structure is clearly shown on the diagonal, implying that the clustering assignment is similar across different samples.

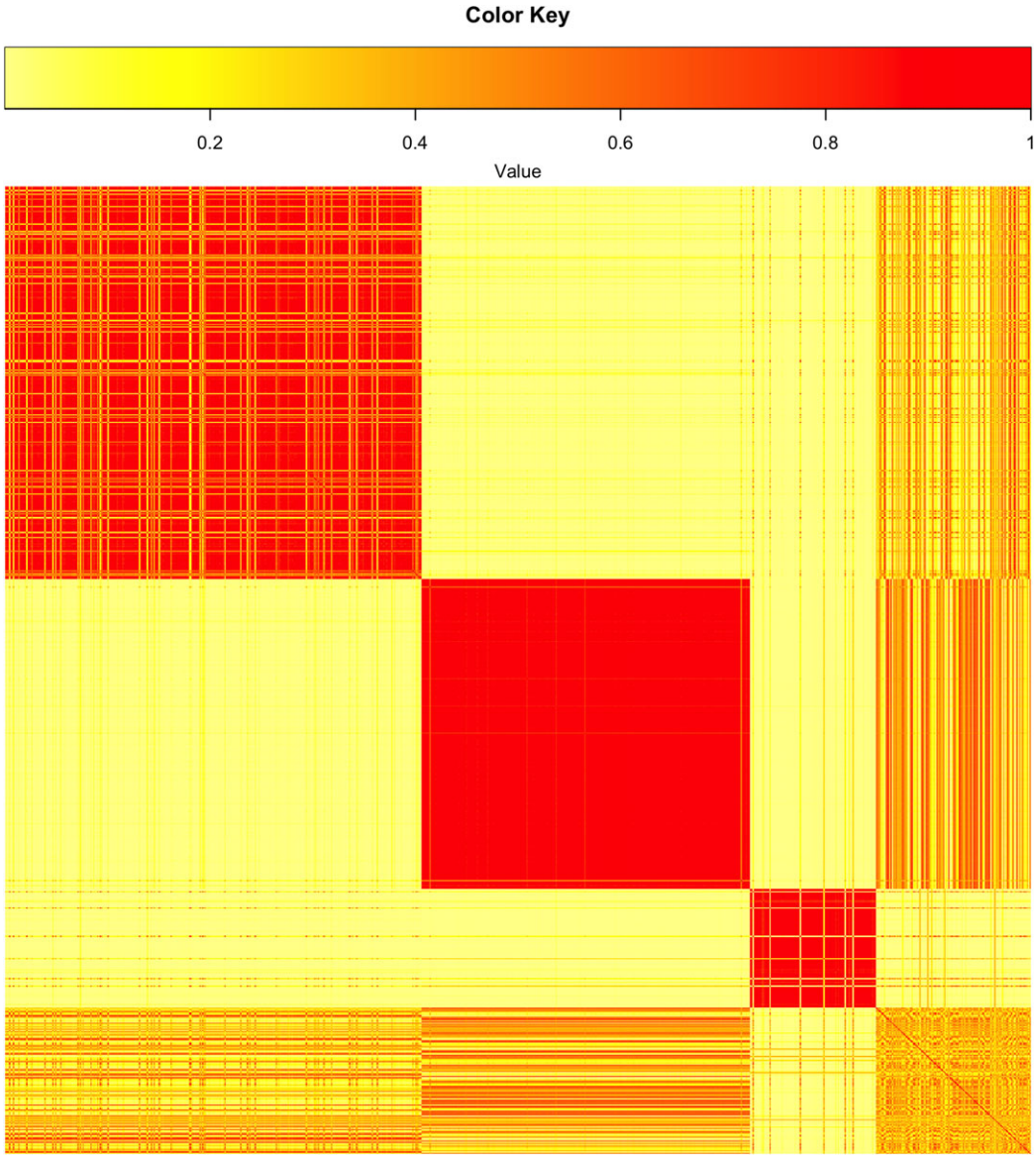

Figure 8. Heat map of the $P_{M\times M}$ matrix showing the posterior probabilities of any two edges sharing the same cluster. Darker colors imply higher probabilities. From left to right, edges are ordered into blocks, such as the first three blocks (size 317, 250, 96) are of edges representing within-school connections and the last block (size 152) is of edges representing inter-school connections.

5. Discussion

Despite receiving little attention in the community detection literature, edge clustering has the potential to provide an understanding of the network structure that is intuitive and easy to interpret. As shown in the applications of aLSEC to real data, identifying clusters of edges can reveal systems of flows in a network and contexts in which edge form. Sewell proposed a latent space edge clustering approach (LSEC) and showed favorable results when applying LSEC to both artificial and real networks. Nevertheless, implementing LSEC requires fitting the model multiple times to

obtain the best result. This paper presents an automatic LSEC approach using an overfitted mixture model prior to simultaneously estimate the number of clusters and cluster-specific variables. Fitting aLSEC models using a GEM approach drastically reduces the computational cost, averaging between 20 and 70 times as shown in the simulation study, and sometimes the computational gains can be up to a hundredfold.

To overcome the challenge of implementing aLSEC efficiently, we derive a variational Bayes GEM algorithm and an HMC-within Gibbs algorithm for estimation, both of which are efficient and have a computational cost that grows linearly with the number of actors in sparse networks. Our simulation study showed that aLSEC performed similar to or better than the original LSEC method but took 10–100 times less time to implement. We examined the sensitivity of the clustering results to different values of the number of dimensions p and found that there was little to no penalty in overspecifying this value. We further applied our method to analyzing patient transfer network and UK faculty network data. In the first application, our method successfully captured the systems of flows and informed us about how disease spreads in the network. In the second application, our approach revealed information about the context from which edges are formed.

Despite the promising performance, there are some limitations to our proposed method. We have yet to directly tackle the issue of specifying the number of latent dimensions p. While the sensitivity results show good performance when overspecifying p, more research is needed to understand the impact of using incorrect values of p on clustering performance. Another limitation is the selection of the hyperparameters for the Gamma priors. Here, we demonstrated some success by using the same prior as Frühwirth-Schnatter & Malsiner-Walli (2019), but the same setup was not guaranteed to work for every situation. Further knowledge of shrinkage priors in the sparse finite mixture model will help guide appropriate prior choices for future applications. Lastly, an important area for future work is to develop a goodness-of-fit measure supported by rigorous statistical methods, such as Bayesian p-values, in order to evaluate the appropriateness of LSEC or aLSEC to the data.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/nws.2023.22.

Competing interests. None.

Data availability statement. The patient transfer data are available from the Healthcare Cost and Utilization Project (HCUP) State Inpatient Databases (SID) (https://www.hcup-us.ahrq.gov/sidoverview.jsp). HCUP data are available for a fee to all researchers following a standard application process and signing of a data use agreement. The UK Faculty data are publicly available as part of the {igraphdata} package in R. The code to perform data analysis can be found online at https://github.com/hanhtdpham/aLSEC.

Funding statement. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Notes

1 https://github.com/hanhtdpham/aLSEC
2 https://www.hcup-us.ahrq.gov/sidoverview.jsp
3 https://github.com/hanhtdpham/aLSEC
4 $\boldsymbol{U}$ and $\boldsymbol{V}$ are highly correlated so plotting either one will produce almost the same plot.

References

Ahn, Y.-Y., Bagrow, J. P., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, *466*(7307), 761–764.

Airoldi, E. M., Blei, D., Fienberg, S., & Xing, E. (2008). Mixed membership stochastic blockmodels. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. *21*). Red Hook, NY: Curran Associates, Inc. https://proceedings.neurips.cc/paper/2008/file/8613985ec49eb8f757ae6439e879bb2a-Paper.pdf

Ball, B., Karrer, B., & Newman, M. (2011, September). Efficient and principled method for detecting communities in networks. *Physical Review E*, *84*, 036103. https://link.aps.org/doi/10.1103/PhysRevE.84.036103

Beal, M. J. (2003). Variational algorithms for approximate bayesian inference. *PQDT - Global*, *282*. http://login.proxy.lib.uiowa.edu/login?url=https://www.proquest.com/dissertations-theses/variational-algorithms-approximate-bayesian/docview/1775215626/se-2?accountid=14663 (Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-09-29).

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, *22*(7), 719–725. https://doi.org/10.1109/34.865189

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. New York: Springer Science & Business Media.

Brent, R. P. (2013). *Algorithms for minimization without derivatives*. North Chelmsford, MA: Courier Corporation.

Csardi, G. (2015). *igraphdata: A collection of network data sets for the 'igraph' package [Computer software manual]*. https://CRAN.R-project.org/package=igraphdata (R package version 1.0.1).

Danon, L., Díaz-Guilera, A., Duch, J., & Arenas, A. (2005, September). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, *2005*(09), P09008. https://doi.org/10.1088/1742-5468/2005/09/p09008

Donker, T., Wallinga, J., Slack, R., & Grundmann, H. (2012, April). Hospital networks and the dispersal of hospital-acquired pathogens by patient transfer. *PLoS ONE*, *7*(4), 1–8. https://doi.org/10.1371/journal.pone.0035002

Evans, T. S., & Lambiotte, R. (2009, July). Line graphs, link partitions, and overlapping communities. *Physical Review E*, *80*(1), 016105. https://doi.org/10.1103/PhysRevE.80.016105

Frühwirth-Schnatter, S., & Malsiner-Walli, G. (2019). From here to infinity: Sparse finite versus dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification*, *13*(1), 33–64. https://doi.org/10.1007/s11634-018-0329-y

Geng, J., Bhattacharya, A., & Pati, D. (2019). Probabilistic community detection with unknown number of communities. *Journal of the American Statistical Association*, *114*(526), 893–905. https://doi.org/10.1080/01621459.2018.1458618

Green, P. J. (1995, December). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732. https://doi.org/10.1093/biomet/82.4.711

Green, P. J., & Hastie, D. I. (2009). Reversible jump MCMC. *Genetics*, *155*(3), 1391–1403.

Handcock, M. S., Raftery, A. E., & Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *170*(2), 301–354.

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., . . .N., Samatova (2014). Community detection in large-scale networks: A survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, *6*(6), 426–439. https://doi.org/10.1002/wics.1319

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. https://doi.org/10.1007/BF01908075

Justice, S. A., Sewell, D. K., Miller, A. C., Simmering, J. E., Polgreen, P. M., & The CDC MInD-Healthcare Program (2022). Inferring patient transfer networks between healthcare facilities. *Health Services and Outcomes Research Methodology*, *22*(1), 1–15. https://doi.org/10.1007/s10742-021-00249-5

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

Le, C. M., & Levina, E. (2015). Estimating the number of communities in networks by spectral methods. *arXiv*. https://doi.org/10.48550/arXiv.1507.00827

Lee, J. K., Choi, J., Kim, C., & Kim, Y. (2014, January). Social media, network heterogeneity, and opinion polarization. *Journal of Communication*, *64*(4), 702–722. https://doi.org/10.1111/jcom.12077

Lo, Y., Mendell, N. R., & Rubin, D. B. (2001, October). Testing the number of components in a normal mixture. *Biometrika*, *88*(3), 767–778. https://doi.org/10.1093/biomet/88.3.767

Malsiner-Walli, G., Frühwirth-Schnatter, S., & Grün, B. (2016). Model-based clustering based on sparse finite gaussian mixtures. *Statistics and Computing*, *26*(1), 303–324.

Malsiner-Walli, G., Frühwirth-Schnatter, S., & Grün, B. (2017). Identifying mixtures of mixtures using bayesian estimation. *Journal of Computational and Graphical Statistics*, *26*(2), 285–295. https://doi.org/10.1080/10618600.2016.1200472

McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. *Annual Review of Statistics and Its Application*, *6*(1), 355–378. https://doi.org/10.1146/annurev-statistics-031017-100325

Miller, J. W., & Harrison, M. T. (2013). A simple example of dirichlet process mixture inconsistency for the number of components. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. *26*). Red Hook, NY: Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/f7e6c85504ce6e82442c770f7c8606f0-Paper.pdf

Miller, J. W., & Harrison, M. T. (2018). Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, *113*(521), 340–356. https://doi.org/10.1080/01621459.2016.1255636

Nepusz, T., Petróczi, A., Négyessy, L., & Bazsó, F. (2008, January). Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, *77*(1), 016107. https://doi.org/10.1103/PhysRevE.77.016107

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*(23), 8577–8582. https://doi.org/10.1073/pnas.0601602103

Newman, M. E. J., & Reinert, G. (2016, August). Estimating the number of communities in a network. *Physical Review Letters*, *117*(7), 078301. https://doi.org/10.1103/PhysRevLett.117.078301

Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, *96*(455), 1077–1087.

Papastamoulis, P., & 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*(2), 313–331.

Pereira-Leal, J. B., Enright, A. J., & Ouzounis, C. A. (2004). Detection of functional modules from protein interaction networks. *Proteins: Structure, Function, and Bioinformatics*, *54*(1), 49–57. https://doi.org/10.1002/prot.10505

Pitman, J., & Yor, M. (1997). The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, *25*(2), 855–900.

Rousseau, J., & Mengersen, K. (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(5), 689–710. https://doi.org/10.1111/j.1467-9868.2011.00781.x

Salathé, M., Kazandjieva, M., Lee, J. W., Levis, P., Feldman, M. W., & Jones, J. H. (2010). A high-resolution human contact network for infectious disease transmission. *Proceedings of the National Academy of Sciences*, *107*(51), 22020–22025. https://doi.org/10.1073/pnas.1009094108

Sewell, D. K. (2021). Model-based edge clustering. *Journal of Computational and Graphical Statistics*, *30*(2), 390–405. https://doi.org/10.1080/10618600.2020.1811104

Sewell, D. K., & Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, *12*(2), 351–377.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *The Annals of Statistics*, *28*(1), 40–74. Retrieved 03 May, 2022, from http://www.jstor.org/stable/2673981

Van Dyk, D. A., & Park, T. (2008). Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, *103*(482), 790–796.

Wang, C., & Blei, D. M. (2013). Variational inference in nonconjugate models. *Journal of Machine Learning Research*, *14*(1), 1005–1031.

Wu, Z., Lin, Y., Wan, H., & 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). https://doi.org/10.1109/ISKE.2010.5680851

Xu, K. (2015, May 09–12). Stochastic block transition models for dynamic networks. In G. Lebanon, & A. Vishwanathan (Eds.), *Proceedings of the eighteenth international conference on artificial intelligence and statistics* (Vol. *38*, pp. 1079–1087). San Diego, CA: PMLR. https://proceedings.mlr.press/v38/xu15.html

Cite this article: Pham H. T. D. and Sewell D. K. (2024). Automated detection of edge clusters via an overfitted mixture prior. *Network Science* **12**, 88–106. https://doi.org/10.1017/nws.2023.22