Applied Network Science

RESEARCH Open Access

Hierarchical Bayesian adaptive lasso methods on exponential random graph models

Dan Han[1*], Vicki Modisette[1], Melinda Forthofer[2] and Rajib Paul[2]

*Correspondence:
dan.han@louisville.edu

[1] Department of Mathematics, University of Louisville, 2301 S 3rd St, Louisville, KY 40292, USA
[2] Department of Public Health Sciences, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223, USA

Abstract

The analysis of network data has become an increasingly prominent and demanding field across multiple research fields including data science, health, and social sciences, requiring the development of robust models and efficient computational methods. One well-established and widely employed modeling approach for network data is the Exponential Random Graph Model (ERGM). Despite its popularity, there is a recognized necessity for further advancements to enhance its flexibility and variable selection capabilities. To address this need, we propose a novel hierarchical Bayesian adaptive lasso model (BALERGM), which builds upon the foundations of the ERGM. The BALERGM leverages the strengths of the ERGM and incorporates the flexible adaptive lasso technique, thereby facilitating effective variable selection and tackling the inherent challenges posed by high-dimensional network data. The model improvements have been assessed through the analysis of simulated data, as well as two authentic datasets. These datasets encompassed friendship networks and a respondent-driven sampling dataset on active and healthy lifestyle awareness programs.

Keywords: ERGM, Bayesian Analysis, Network, Penalized Model, Adaptive Lasso

Introduction

Multiple disciplines such as sociology, political science, and biology have extensively employed network analysis and random graph studies to comprehend and represent relationships among entities, ranging from friendships and global trading partners to proteins and genes. Early models generating random graphs assumed equal probability among graphs of the same size or independence among edges, but these models had evident limitations (Erdös and Rényi 1959). Holland and Leinhardt presented the next advancement by introducing a model for directed graphs that solely employed independent dyads (Holland and Leinhardt 1981). Subsequent work overcame the limitations of independence assumptions and introduced Markov random graph models, establishing the foundation for ERGMs that have been endured for decades (Frank and Strauss 1986), however, traditional statistical methods have limitations in effectively capturing the complexities of relational data. The ERGM has emerged as a valuable tool for quantifying such data, elucidating how local interactions shape the overall structure of a network. ERGMs acknowledge and capture the inherent interdependence embedded within network structures. The probability of an edge's existence

is influenced not only by the presence of other edges but also by various network configurations, such as triangles, and the characteristics of nodes throughout the entire network. This assumption of dependence aligns closely with our intuitive understanding of how networks are formed and operate. It is noteworthy that the development of ERGMs by Frank and Strauss (1986) was primarily motivated by the recognition of tie-dependence in networks.

Fundamentally, ERGMs are analogous to logistic regression when the dyads are independent, offering regression-like analysis on random networks. ERGMs estimate the probability of tie existence between pairs of nodes in a network. Since ERGMs share commonalities with logistic regression, let us recall the traditional lasso method in classical linear regression and discuss its development and relation to Bayesian theory, providing hints about the potential problems developing lasso estimates on the exponential random network. The lasso of Tibshirani is a method for simultaneous shrinkage and model selection in regression problems. Tibshirani (1996) In the context of linear regression, the lasso is a regularization technique for simultaneous estimation and variable selection where if $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ where $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^\top$ is the response vector, $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_p)$ is an $n \times p$ predictor matrix, $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p)$ is a corresponding vector of regression coefficients, $\boldsymbol{\epsilon} = (\epsilon_1, \cdots, \epsilon_n)$ are independent normal distributed errors, then the lasso estimates are defined as

$$\hat{\beta}(lasso) = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{x_j}\beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1}$$

where the second term in (1) is the so-called "l^1 penalty". The tuning parameter λ controls the amount of penalty. Fan and Li (2001) studied a class of penalized models including the lasso. They proved that the lasso can perform automatic variable section because of the singularity of l^1 penalty at the origin. If certain conditions are not satisfied, the lasso estimates could be inconsistent. To overcome the above issues, Zou in 2006 and Wang et al. proposed to use an adaptive lasso that enjoys the consistency and the oracle properties: namely, it performs as well as if the true underlying model were given in advance. Zou (2006), Wang and Leng (2008) Tibshirani suggested that lasso estimates can be interpreted as posterior mode estimates when the regression parameters have independent and identical Laplace (i.e., double-exponential) priors. Tibshirani (1996) Targeting at finding this mode, several other authors studied subsequently different Bayesian contexts. Yuan and Lin (2006), Park and Casella (2008), Leng et al. (2014), Alhamzawi and Ali (2018) However, all these studies are for linear regressions and they are not built on random networks.

In the context of ERGMs, estimation encounters computational challenges when there is dependence among dyads. These challenges are primarily attributed to the intractability of the normalizing constant and the issue of degeneracy. Chatterjee and Diaconis (2013) Intractability refers to the computational difficulties associated with calculating the normalizing constant, which ensures that the probability mass function sums to one. On the other hand, degeneracy refers to the phenomenon where the models assign a significant proportion of their probability mass to a small subset of graphs. This leads to a cascading effect throughout the graph, resulting in the

model assigning most of its probability mass to very sparse or very dense graphs. Bayesian computational methods have proven instrumental in circumventing these challenges. Caimo and Friel were the first to develop complete Bayesian frameworks for network models, enabling the incorporation of Bayesian analysis into real-world networks, which often exhibit large-scale, high-dimensional, and complex structures with numerous attribute variables associated with nodes (Caimo and Friel 2011). Subsequently, Caimo et al. integrated a transdimensional reversible jump Markov Chain Monte Carlo (RJMCMC) approach, initially introduced by Green (1995), with the exchange algorithm (Caimo and Friel 2013, 2014). This algorithm incorporates an independence sampler, utilizing a distribution that fits a parametric density approximation to the within-model posterior. This method is appealing in model selection since it relies exclusively on probabilistic considerations but is challenging computationally since it needs to estimate the posterior probability for each competing model. In scenarios with a high number of variables, the presence of numerous potential models becomes more pronounced. The increased dimensionality leads to a larger set of competing models, making the task of model selection more challenging and critical. This motivates the development of the penalized exponential random graph model developed in this paper.

While penalized estimation methods have been discussed in the context of graphical models by various researchers, these studies either lack a specific focus on ERGMs or fail to fully account for the inherent dependencies present in network data, often transforming the problem into generalized penalized linear regression. Meinshausen and Bühlmann (2006), Shojaie et al. (2012), Shojaie and Michailidis (2010), Shojaie (2013), Fan et al. (2009) Motivated by the need to explore network model uncertainty and achieve parsimony in exponential random graphs, we propose a more flexible and adaptive lasso-type penalized model within the framework of the ERGM. This model aims to improve parameter estimations and prediction accuracy, enabling effective variable selection within high-dimensional network data. Through comprehensive evaluations and comparisons with existing methods, our model demonstrates its superiority in terms of efficiency and effectiveness in selecting significant variables. It promises substantial improvements in the field by addressing the critical challenge of model selection in the analysis of high-dimensional network data.

In summary, the utilization of Bayesian adaptive lasso model offers two prominent advantages: (1) Enhanced convergence speed and improved parameter mixing: adaptive lasso addresses a notable limitation of the conventional lasso regularization technique, which often exhibits sluggish convergence and difficulties in selecting significant variables within high-dimensional datasets. Consequently, it facilitates faster convergence and more effective mixing of parameters. This characteristic proves particularly advantageous in scenarios involving extensive datasets or a substantial number of predictors. (2) Effective variable selection: Bayesian adaptive lasso exponential random graph model demonstrates exceptional proficiency in this task by automatically identifying pertinent variables while concurrently shrinking or eliminating less relevant ones. The process is facilitated through the utilization of multiple chains generated by a parallel direction sampling algorithm, which enhances the efficiency and accuracy of variable selection. These benefits are the primary focus of the discussed article.

This article is structured as follows. Section 2 provides a basic introduction to exponential random graph models, offering a foundation for the subsequent discussions. In Sect. 3, we introduce a Bayesian Exponential Adaptive Lasso Model for the exponential random graph, which enhances the Monte Carlo maximum likelihood method proposed by Geyer and the Bayesian ERGM (BERGM) presented by Caimo and Friel (Geyer 1991; Caimo et al. 2022). Section 4 presents a derivation of the Gibbs sampling theory underlying the model, shedding light on the underlying theoretical framework. In Sect. 5, we introduce the adaptive parallel direction sampling algorithm, which is incorporated into the Gibbs sampling theory to improve the mixing of the Monte Carlo chains, thereby enhancing the overall performance of the model. Section 6 outlines the algorithm procedure and provides a comparative analysis with the BERGM method proposed by Caimo et al., highlighting the strengths and advantages of our proposed approach. Caimo and Friel (2013), Caimo and Friel (2014), Caimo et al. (2022) In Sect. 7, we describe the network dataset called Faux Dixon High, which is used to test the model and present simulation results. Additionally, this section includes the results of applying the proposed model to data collected in a study conducted with the Prevention Research Center at USC and Sumter County Active Lifestyles (SCAL). In Sect. 8, we discuss the goodness of fit for the proposed Bayesian adaptive lasso method, providing an evaluation of its performance and suitability. Finally, in Sect. 9, we summarize the key findings and contributions of the paper and identify open problems and avenues for future research

Exponential random graph models

Examples and context

Exponential Random Graph Models (ERGMs) are widely applicable to research questions in the social and health sciences. In psychology, researchers studied Romanian school children's friendship networks to find that sex and mental health showed patterns of homophily, concluding that ERGM are a "promising avenue for further research." Baggio et al. (2017) Also in the social and health sciences, Becker et al. considered the friendship network of members of a sorority and the influence of disordering eating habits on friendship finding that women tended to have disordered eating habits, unlike their friends (Becker et al. 2018). This unexpected result has implications for understanding the complex social dynamics that go into a serious health concern. Solo et al. note the utility and suitability of ERGM for modeling connections within the brain compared to more traditional methods, though they also note the computational difficulty of ERGM (Solo et al. 2018). On a much larger scale, ERGM have been used to understand the influences of information sharing on tourism. The model helped answer questions about the existence of patterns in the network including whether or not the network exhibited the characteristic of homophily and how organizations should understand their role in the network (Williams and Hristov 2018). In the biological world, Stivala et al. show that ERGM can address some of the limitations that previous research had found in modeling biological processes (Stivala and Lomi 2021). These examples show the incredible flexibility and significance of exponential random graph models.

Model structure

For any network, it can be expressed with an adjacency matrix. The connectivity of the network's graph is described by an $n \times n$ adjacency matrix $\boldsymbol{Y}$. Its i-j entry $Y_{i,j} = 1$ if node i will give referral to node j and $Y_{i,j} = 0$ otherwise. Let $\mathcal{Y}$ be the set of all possible graphs on n nodes and let $\boldsymbol{y}$ be a realization of $\boldsymbol{Y}$. A given network $\boldsymbol{y}$ consists of n nodes and m edges that define a relationship between pairs of nodes called dyads. The adjacency matrix of the network graph $\boldsymbol{Y}$ allows for the analysis of the structural relationship in the observed network.

For general exponential random graph models, the network has the following exponential family type density: (Lusher et al. 2013)

$$\pi(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{z(\boldsymbol{\theta})} e^{\boldsymbol{\theta}^T s(\boldsymbol{y})} \tag{2}$$

where $\boldsymbol{y}$ is the observed network, $\boldsymbol{\theta}$ is a vector of parameters, and $s(\boldsymbol{y})$ is a vector of network statistics. Each i-th network statistic $s_i(\cdot)$ has a corresponding parameter θ_i. A positive value of θ_i indicates that the edges involved in the formation of network statistics s_i are more likely to be connected with each other. The normalizing constant $z(\boldsymbol{\theta})$ is the summation $\sum_{\boldsymbol{y} \in \mathcal{Y}} e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}$ where $\mathcal{Y}$ is the set of all possible graphs with the same number of nodes as $\boldsymbol{y}$. The number of possible graphs with n nodes is $2^{n(n-1)/2}$ which becomes very large for all but the smallest graphs. Lusher et al. (2013) Hence, the calculation of $z(\boldsymbol{\theta})$ is feasible only for small networks in computer computation. It becomes challenging to find this normalization constant for large networks or even moderate-sized networks.

Let $\boldsymbol{\delta} = s(\boldsymbol{y}_{ij}^+) - s(\boldsymbol{y}_{ij}^-)$ be the vector of changes in the statistics in $\boldsymbol{s}$ when the edge y_{ij} between node i and j in the graph $\boldsymbol{y}$ changes from 1 to 0 along with the complement part $\boldsymbol{y}_{ij}^c$ same. Conditioned on the state of the rest of the graph represented $\boldsymbol{Y}_{-ij}$, the log odds of the probability of a tie existing between node i and j is:

$$\log \frac{P(Y_{ij} = 1 | \boldsymbol{Y}_{-ij} = \boldsymbol{y}_{-ij}, \boldsymbol{\theta})}{P(Y_{ij} = 0 | \boldsymbol{Y}_{-ij} = \boldsymbol{y}_{-ij}, \boldsymbol{\theta})} = \boldsymbol{\theta}^T \boldsymbol{\delta} \tag{3}$$

These network statistics can be overlapping subgraph configurations such as the number of edges, mutual edges, triangles, and uniform homophily etc. The representation above gives the intuitive explanation of the model parameter $\boldsymbol{\theta}$ about their effect on the probability of an edge between node i and j.

Classical inference for ERGMs

Estimation methods

The inferential statistical goal is to find an appropriate estimate of $\boldsymbol{\theta}$ such that the corresponding generated network has the probability distribution centered on the observed network on average. That is, we want to solve the moment equation:

$$\mathbb{E}_{\boldsymbol{\theta}}(s(\boldsymbol{y})) = s(\boldsymbol{y}_{\text{obs}}) \tag{4}$$

where $\boldsymbol{y}_{obs}$ is the observed network and $s(\boldsymbol{y})$ is a vector of network statistics in the proposed graph and $s(\boldsymbol{y}_{\text{obs}})$ is a vector of the network statistics in the observed graph.

However, in most cases, the moment equation cannot be solved analytically. This challenge leads to two mainstream simulations: Maximum Pseudolikelihood estimation and Monte Carlo Maximum Likelihood estimation.

Maximum pseudolikelihood estimation

The direct Maximum likelihood estimation of ERGMs is complicated since the likelihood function is difficult to compute for models and networks of moderate or large size. Strauss et al. proposed a standard approximation with maximum pseudolikelihood estimation (MPLE). Strauss and Ikeda (1990) Instead of conditioning each tie on the state of the entire graph, the assumption is that the dependence of each dyad is weak. In particular, the MPLE estimates can be obtained by assuming the independence among values of Y_{ij}:

$$P(Y_{ij} = 1|\boldsymbol{\delta}_{-ij} = \boldsymbol{y}_{-ij}) = P(Y_{ij} = 1)$$

This allows for the pseudolikelihood function that has the strength of quick estimation but has been shown to not provide reliable estimates. van Duijn et al. (2009), Friel et al. (2009)

$$\pi(\boldsymbol{y}|\boldsymbol{\theta}) \approx \pi_{pseudo}(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i \neq j} \pi(y_{ij}|\boldsymbol{y}_{-ij}, \boldsymbol{\theta}) \tag{5}$$

$$= \prod_{i \neq j} \frac{\pi(y_{ij} = 1|\boldsymbol{y}_{-ij}, \boldsymbol{\theta})^{y_{ij}}}{[1 - \pi(y_{ij} = 0|\boldsymbol{y}_{-ij}, \boldsymbol{\theta})]^{y_{ij}-1}} \tag{6}$$

This will only provide the true estimate for ERGM with dyadic independence or when the change statistics can be found only considering one tie without knowing the rest of the graph. Research by van Duijn et al. compares the maximum pseudo-likelihood and maximum likelihood estimates, and their study shows the pseudo-likelihood estimation is biased and MPLE can only approximate the transitivity pattern in the network well. van Duijn et al. (2009)

Monte Carlo maximum likelihood estimation

Similar to methods in linear regression, ERGMs are log-linear, and a typical method for finding the maximum likelihood requires finding the roots of the derivative of the log of the function. This results in the $s(\boldsymbol{y})^T - \mathbb{E}_{\boldsymbol{\theta}}(s(\boldsymbol{y})) = 0$ found earlier. The Monte Carlo maximum likelihood estimation in ERGM case needs to find the following important ratio: (van Duijn et al. 2009)

$$\frac{z(\boldsymbol{\theta})}{z(\boldsymbol{\theta}_0)} = \mathbb{E}_{\boldsymbol{y}|\boldsymbol{\theta}_0}\left[\frac{e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}}{e^{\boldsymbol{\theta}_0^T s(\boldsymbol{y}_{\text{obs}})}}\right]. \tag{7}$$

The log-likelihood equation, however, is not directly solvable without computing the normalizing constant. As previously mentioned, this is computationally intensive for all but the smallest graphs. With this approximation, though, the normalizing constant can be estimated by generating m graphs from the density $\pi(\boldsymbol{\pi}|\boldsymbol{\theta_0})$ and finding $e^{(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T s(\boldsymbol{y}_i)}$

for each graph and use importance sampling technique. The estimates of $\boldsymbol{\theta}$ can be obtained by maximizing the log-likelihood ratio approximated as the following:

$$\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}_0) \approx (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T - \ln\left[\frac{1}{m}\sum_{i=1}^{m} e^{(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T s(\boldsymbol{y}_i)}\right] \tag{8}$$

However, in this method, the choice of the initial $\boldsymbol{\theta}_0$ is tricky and should be near the maximum likelihood estimate of $\boldsymbol{\theta}_0$. Poor choice of $\boldsymbol{\theta}_0$ can lead to the failure of the maximization log-likelihood function and degeneracy problem. van Duijn et al. (2009), Handcock (2003)

Bayesian adaptive lasso exponential random graph model

This work is motivated by the need to explore model uncertainty and flexibility. With these objectives, we consider the following exponential random graph model, this model is a particular class of discrete exponential random exponential families that represent the probability distribution of the adjacency matrix $\boldsymbol{Y} \in \mathcal{Y}$ where $\mathcal{Y}$ is the set of all possible graphs on n nodes. Let $\boldsymbol{y}$ a realization of $\boldsymbol{Y}$. The likelihood function of an ERGM stands for the probability density of a random network and can be expressed as:

$$\pi(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{q(\boldsymbol{y}|\boldsymbol{\theta})}{z(\boldsymbol{\theta})} = \frac{e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}}{z(\boldsymbol{\theta})} \tag{9}$$

where $q(\boldsymbol{y}|\boldsymbol{\theta}) = e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}$ is the unnormalized likelihood.

We consider the following adaptive lasso estimator on the exponential random network:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} l(\boldsymbol{\theta}|\boldsymbol{y}) - P(\boldsymbol{\theta}), \tag{10}$$

$$P(\boldsymbol{\theta}) = \sum_{j=1}^{p} \lambda_j |\theta_j| \tag{11}$$

where $l(\boldsymbol{\theta}|\boldsymbol{y}) = \ln(\pi(\boldsymbol{y}|\boldsymbol{\theta}))$ is the log-likelihood function of $\boldsymbol{\theta}$ and each λ_j is a different penalty parameter used for the coefficients. In dyadic independence ERGMs, maximizing the log-likelihood function (10) is equivalent to maximizing the following log pseudo-likelihood function:

$$l(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{\boldsymbol{y}} y_{ij} \ln(\pi_{ij}) + \sum_{\boldsymbol{y}} (1 - y_{ij}) \ln(1 - \pi_{ij}) - \sum_{j=1}^{p} \lambda_j |\theta_j| \tag{12}$$

where $\pi_{ij} = P(Y_{ij} = 1|\boldsymbol{y}_{ij}^c) = P(Y_{ij} = 1)$. In this case, the network estimation problems are transformed into the classical adaptive lasso logistic linear regression model. For example, the coordinate descent algorithm developed in glmnet package for R (Tay et al. 2023; Friedman et al. 2010) can get estimations of the parameters $\theta_j,\ j = 1, 2, 3, \cdots, p$ with penalties include the lasso, ridge and the elastic net. However, different from the generalized linear regression models, the challenge of estimation on the dyadic dependent ERGMs relies on the intractable normalizing constant appearing in the log-likelihood

function. With the review of ERGMs likelihood-based methods in Sect. 2, the solution to the equation (10) has similar obstacles. To get around those obstacles, we will study this problem with an adaptively Bayesian estimate obtained from the lasso penalized method on the random networks.

Assume that a prior distribution $\pi(\boldsymbol{\theta})$ is placed on $\boldsymbol{\theta}$, and we are interested in the posterior distribution

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \tag{13}$$

We consider a conditional Laplace prior specification of the form similar to the classical Bayesian lasso linear regression developed in Park and Casella (2008) but with different penalty terms so that we have λ_j for $j = 1, 2, 3, \cdots, p$:

$$\pi(\boldsymbol{\theta}|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda_j}{2\sqrt{\sigma^2}} e^{-\lambda_j|\theta_j|/\sqrt{\sigma^2}} \tag{14}$$

We can now formulate a hierarchical model on the exponential random graph, which we can use to implement this version of the Bayesian lasso with a Gibbs sampler, using the Laplace distribution as a scale mixture of Gaussians. When the mixing distribution is exponential, the resulting distribution is Laplace. Andrews and Mallows (1974)

$$\frac{a}{2} e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}} e^{-\frac{z^2}{2s}} \frac{a^2}{2} e^{-\frac{a^2 s}{2}} ds, \quad a > 0 \tag{15}$$

Now we use a latent parameter τ^2 to make the prior (14) as a scale mixture of normal distributions (15). We can consider τ_js as additional parameters that assign different variances to the prior of $\boldsymbol{\theta}$. When $\tau_j \to 0$, the coefficient of $s_j(\boldsymbol{y})$ is shrunk to zero.

Assume $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_p)$ follows normal distributions centered at zero with variance defined below.

$$\boldsymbol{\theta}|\sigma^2, \tau_1^2, \tau_2^2, ..., \tau_p^2 \sim \mathcal{N}(0_p, \sigma^2 \boldsymbol{D}_\tau) \tag{16}$$

where $\sigma^2 > 0$ and $\boldsymbol{D}_\tau = diag(\tau_1^2, \tau_2^2, \cdots, \tau_p^2)$ is a matrix that allows each parameter to come from a normal distribution with a different variance.

Different than the basic Bayesian lasso model proposed by Park and Casella (2008) in which $\boldsymbol{\tau}$ follows

$$\pi(\boldsymbol{\tau}^2) = \frac{\boldsymbol{\lambda}^2}{2} e^{-\frac{\lambda^2 \tau^2}{2}}, \tag{17}$$

our Bergm adaptive lasso model sets up different shrinkage parameters for different coefficients. This motivates us to define a more adaptive penalty in the hierarchical structure:

$$\pi(\sigma^2, \tau_1, \tau_2, \cdots, \tau_p) \propto \pi(\sigma^2) \prod_{j=1}^{p} \frac{\lambda_j^2}{2} e^{-\frac{\lambda_j^2 \tau_j^2}{2}} \tag{18}$$

and an independent non-informative scale-invariant marginal prior $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$ on σ^2 suggested by Park and Casella. Park and Casella (2008) The conditional distribution on

σ^2 guarantees a unimodal full posterior distribution for the estimate $\boldsymbol{\theta}$ on the network. (See Appendix A). The unimodal posterior distribution ensures the quick convergence of the Gibbs sampling algorithm and ensures the meaningful single point estimate of $\boldsymbol{\theta}$.

Finally, the simplest prior for the penalty term λ_j, for $j = 1, 2, 3, \cdots, p$ would be a uniform distribution, but this proved to be problematic with complex networks, particularly when a model has many parameters. Thus, following the notation of Park and Casella (2008) we propose a prior such that λ_j^2 follows Gamma distribution with shape parameter r and rate parameter δ_j:

$$\pi(\lambda_j^2) = \frac{\delta_j^r}{\Gamma(r)}\left(\lambda_j^2\right)^{r-1} e^{-\delta_j \lambda_j^2} \qquad \text{for } \lambda_j, r, \delta_j > 0. \tag{19}$$

This prior mixes well with the other choices for the Gibbs sampling and as Park and Casella (2008) note, this prior can approach 0 as $\boldsymbol{\lambda} \to \infty$ and can concentrate probability near the MLE.

In summary, the hierarchical formulation of the Bayesian adaptive lasso Model on the exponential random graph is as follows:

$$\pi(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{z(\boldsymbol{\theta})} e^{\boldsymbol{\theta}^T s(\boldsymbol{y})} \tag{20}$$

$$\boldsymbol{\theta}|\sigma^2, \tau_1^2, \tau_2^2, ..., \tau_p^2 \sim \mathcal{N}(0_p, \sigma^2 \boldsymbol{D}_\tau) \tag{21}$$

$$\boldsymbol{D}_\tau = diag(\tau_1^2, \cdots, \tau_p^2) \tag{22}$$

$$\pi(\sigma^2, \tau_1, \tau_2, \cdots, \tau_p|\lambda_j) \propto \pi(\sigma^2) \prod_{j=1}^{p} \frac{\lambda_j^2}{2} e^{-\frac{\lambda_j^2 \tau_j^2}{2}} \tag{23}$$

$$\pi(\lambda_j^2) = \frac{\delta_j^r}{\Gamma(r)}\left(\lambda_j^2\right)^{r-1} e^{-\delta_j \lambda_j^2} \tag{24}$$

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2} \tag{25}$$

for $\sigma^2, r, \delta_j, j = 1, 2, 3, \cdots, p$ and $\tau_1^2, \tau_2^2, \cdots, \tau_p^2 > 0$.

The major differences of this formulation compared with the Bayesian lasso in Park and Casella (2008) are first, the Bayesian lasso method in Park and Casella (2008) is applied to linear regression model $\boldsymbol{y} = \mu \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ without any network structure. In other words, $\boldsymbol{y}$ in Park and Casella (2008) follows the normal distribution $\mathcal{N}(\mu \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n)$, where $\boldsymbol{y}$ is a $n \times 1$ vector of responses which doesn't involve random graph. Second, our model allows different penalty variables λ_j, one for each different parameter. In this case, each τ_j^2 can have its own distribution and thus the variance of each normal distribution can be different. With the flexibility of the penalties, the lasso estimate of the parameter for less important random variables on the exponential random graph will have a larger penalty. And smaller penalty will be

applied to those important random variables. And compared with the existing Bayesian Adaptive Lasso model (Leng et al. 2014),Alhamzawi and Ali (2018), our model is built on the random network. And compared with the Bayesian Exponential Random Graph Model (BERGM) by Caimo and Friel (2011), our model Bayesian Adaptive Lasso Exponential Random Graph Model(BALERGM) has more accurate estimations, and the structure is more flexible and adaptive to the network statistics level by adopting distinct shrinkage and penalties for different network statistics. The estimates $\hat{\theta}_j$ of θ_j for $j = 1, 2, 3, \cdots, p$ will be small and close to 0 if it does not provide much improvement on predicting the random network $\boldsymbol{Y}$. So it naturally leads to an estimator with an automatic variable selection property. The value of λ_j will affect the estimates θ_j. The larger $\hat{\lambda}_j$ exists in the model, the sparser $\boldsymbol{\theta}$ will be. (namely, more coefficients are small and near 0) whereas smaller $\hat{\theta}_j$ leads to a less sparse $\boldsymbol{\theta}$. Sparsity is a common belief in high-dimensional statistics because we anticipate only a few covariates are actually related to the response and most covariates are useless. BALERGM is very powerful in this scenario because it leads to a sparse estimator on the network (many coefficients are near 0). Note that high-dimensional problems in network science are very common. For example, in genetics, there are many genes per individual but often we have few patients in our study, or in neuroscience, the fMRI machine produces many voxels per person at a given time.

Gibbs sampler implementation

Now we will implement the model with a Gibbs sampler. The Gibbs sampling method is a Markov Chain Monte Carlo (MCMC) algorithm. In our case, the joint distribution is difficult to sample from directly, but the conditional distribution of each variable is known and is easier to sample from. The Gibbs sampling algorithm generates an instance from the distribution of each variable in turn, conditioned on the current values of the other variables. The construction of the hierarchical model (20) makes the derivation of the full conditional distributions for each component of the estimates feasible.

Thus we can write the joint density as the product of the conditional density of $\boldsymbol{y}|\boldsymbol{\theta}$ and the density of $\boldsymbol{\theta}$. Using the pieces of the hierarchical formulation of the model from (20) we can substitute in each piece that we have already chosen to find the joint distribution.

$$\begin{aligned}\pi(\boldsymbol{y}, \boldsymbol{\theta}, \sigma, \boldsymbol{\lambda}, \boldsymbol{\tau}) &= \pi(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\ &= \pi(\boldsymbol{y}|\boldsymbol{\theta}) \prod_{j=1}^{p} \pi(\theta_j|\tau_j^2, \sigma^2)\pi(\tau_j^2|\lambda_j)\pi(\lambda_j)\pi(\sigma^2) \\ &= \frac{1}{z(\boldsymbol{\theta})} e^{\boldsymbol{\theta}^T s(\boldsymbol{y})} \prod_{j=1}^{p} \frac{1}{(2\sigma^2\tau_j^2)^{1/2}} e^{-\frac{1}{2\sigma^2\tau_j^2}\theta_j^2} \frac{\lambda_j^2}{2} e^{-\frac{\lambda_j^2\tau_j^2}{2}} \frac{\delta_j^r}{\Gamma(r)} \left(\lambda_j^2\right)^{r-1} e^{-\delta_j\lambda_j^2} \frac{1}{\sigma^2}\end{aligned} \tag{26}$$

To implement the Gibbs sampling, we require the distribution of each parameter $\tau_j, \lambda_j, \sigma^2$ to update in turn. From the joint distribution (26), we consider all parts of that joint distribution that depend on each variable. As summarized in Table 1, we consider the full conditional distributions for $\tau_j, \lambda_j,$ and σ^2 respectively.

Table 1 Sampling distributions from joint distribution for each variable

Variable	Proportional Distribution
$\frac{1}{\tau_j}$	Inverse Gaussian $\left(\sqrt{\frac{\lambda_j^2\sigma^2}{\theta_j^2}}, \lambda_j^2\right)$
λ_j^2	Gamma $\left(2, \frac{\tau_j^2}{2}\right)$
σ^2	Inverse Gamma $\left(\frac{p}{2}, \frac{1}{2}\boldsymbol{\theta}^T D_{\boldsymbol{\tau}}^{-1}\boldsymbol{\theta}\right)$

Sample τ_j

For each τ_j we have the following distribution.

$$\pi(\tau_j|\boldsymbol{y}, \boldsymbol{\theta}, \sigma, \boldsymbol{\lambda}) \propto (\tau_j^2)^{\frac{-1}{2}} e^{-\frac{1}{2}\left(\frac{\theta_j^2/\sigma^2}{\tau_j^2} + \lambda_j^2\tau_j^2\right)} \tag{27}$$

To find what distribution each τ_j follows, we begin by considering the following transformation. Chhikara and Folks (1988) If a random variable $x \sim$ Inverse Gaussian(μ, λ'), that is

$$f(x, \mu, \lambda') = \left(\frac{\lambda'}{2\pi x^3}\right)^{\frac{1}{2}} e^{-\frac{\lambda'(x-\mu)^2}{2\mu^2 x}}, \tag{28}$$

then with the change of variable, we can find the density f' of $w = x^{-1}$ as

$$f(w, \mu, \lambda') = \left(\frac{\lambda'}{2\pi w^3}\right)^{\frac{1}{2}} e^{-\frac{\lambda'(1-\mu w)^2}{2\mu^2 w}}. \tag{29}$$

Hence

$$f'(w, \mu, \lambda') = \mu w f(w, \mu^{-1}, \lambda'\mu^{-2}). \tag{30}$$

Then we can rewrite equation 27 into the reciprocal of the Inverse Gaussian distribution

$$\left(\frac{1}{\tau_j^2}\right)^{-\frac{3}{2}} \exp\left\{-\frac{1}{2}\left(\frac{\theta_j^2}{\tau_j^2} + \frac{\lambda_j^2}{1/\tau_j^2}\right)\right\} \propto \left(\frac{1}{\tau_j^2}\right)^{-\frac{3}{2}} exp\left\{-\frac{\theta_j^2\left(\frac{1}{\tau_j^2} - \sqrt{\frac{\lambda_j^2\sigma^2}{\theta_j^2}}\right)^2}{2\sigma^2\frac{1}{\tau_j^2}}\right\} \tag{31}$$

thus $\frac{1}{\tau_j^2}$ follows inverse Gaussian distribution with parameters $\sqrt{\frac{\lambda_j^2\sigma^2}{\theta_j^2}}$ and λ_j^2:

$$\frac{1}{\tau_j^2} \sim \text{Inverse Gaussian}\left(\sqrt{\frac{\lambda_j^2\sigma^2}{\theta_j^2}}, \lambda_j^2\right) \tag{32}$$

Sample σ^2

Similar to the other parameters, we now look at σ^2 with the following conditional distribution:

$$\pi(\sigma^2|\boldsymbol{y},\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\tau}) \propto (\sigma^2)^{-1-\frac{p}{2}} e^{-\frac{1}{2\sigma^2}\boldsymbol{\theta}^T \mathrm{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\theta}}. \tag{33}$$

If $x \sim$ Inverse Gamma (α, β) with the shape parameter α and scale parameter β, then it has the following density function:

$$f(x,\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\frac{\beta}{x}}. \tag{34}$$

We can compare the conditional density (33) with (34) to find:

$$\pi(\sigma^2|\boldsymbol{y},\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\tau}) \propto \text{Inverse Gamma}\left(\frac{p}{2}, \frac{1}{2}\boldsymbol{\theta}^T D_{\boldsymbol{\tau}}^{-1}\boldsymbol{\theta}\right). \tag{35}$$

Sample λ_j^2

To sample the penalty term $\boldsymbol{\lambda}$, we have developed three methods providing flexibility depending on the network model requirements (Table 2).

Method A: The simplest prior for the penalty term λ_j, for $j = 1, 2, 3, \cdots, p$ would be a uniform distribution, but this proved to be problematic with complex networks, particularly when a model has many parameters. Thus, following the notation of Park and Casella (2008), we propose an adaptive prior such that $\lambda_j^2 \sim$ Gamma (r, δ_j). That is,

$$\pi(\lambda_j^2) = \frac{\delta_j^r}{\Gamma(r)} \left(\lambda_j^2\right)^{r-1} e^{-\delta_j \lambda_j^2} \text{ for } \lambda_j, r, \delta_j > 0 \tag{36}$$

For the full Bayes estimation of λ_j^2, we have to find the following distribution.

$$\pi(\lambda_j^2|\boldsymbol{y},\boldsymbol{\theta},\sigma,\boldsymbol{\tau}) \propto \frac{\lambda_j^2}{2} e^{-\frac{\lambda_j^2 \tau_j^2}{2}} \left(\lambda_j^2\right)^{r-1} e^{-\delta\lambda_j^2} \tag{37}$$

$$= \frac{\left(\lambda_j^2\right)^r}{2} \exp\left\{-\lambda_j^2\left(\frac{\tau_j^2}{2} + \delta_j\right)\right\} \tag{38}$$

This shows us that λ_j^2 is proportional to a gamma distribution with $\alpha = r + 1$ and $\beta = \frac{\tau_j^2}{2} + \delta_j$, since a standard gamma probability density function is $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$.

Therefore we can conclude:

$$\pi(\lambda_j^2|\boldsymbol{y},\boldsymbol{\theta},\sigma,\boldsymbol{\tau}) \propto \text{Gamma}\left(r+1, \frac{\tau_j^2}{2} + \delta_j\right) \tag{39}$$

Table 2 Methods for Sampling $\boldsymbol{\lambda}$

	Methods
Method A:	Full Bayes with a $\lambda_j \sim$ Gamma $(r, \delta_j), j = 1, 2, \cdots, p$
Method B:	Partial empirical Bayes with an empirical update of $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_p)$
Method C:	Full empirical Bayes with an empirical update of $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_p)$

where r and $\boldsymbol{\delta}$ are chosen constants/vectors of constants.

Method B: In contrast to the previous Method A, where the parameters δ_j, for $j = 1, 2, \cdots, p$, were treated as fixed constants, the proposed method incorporates an empirical update of the hyperparameter vector $\boldsymbol{\delta}$ using the Monte Carlo Expectation-Maximization (E-M) algorithm (Levine and Casella 2001). The empirical update of the parameters δ_j is performed using the following formula:

$$\delta_j = \frac{r}{\mathbf{E}_{\delta_j^{(k-1)}}\left[\lambda_j^2 | \delta_j^{(k-1)}, \boldsymbol{y}^{(k-1)}\right]}. \tag{40}$$

The full derivation of this method is presented in Appendix B.

The empirical update of the parameters δ_j using the E-M algorithm brings several advantages to the estimation process. Firstly, it eliminates the need for manually specifying appropriate hyperparameter values, as the parameter values are estimated directly from the observed data. This data-driven approach enables the selection of hyperparameters based on the characteristics of the data, enhancing the flexibility and adaptability of the model. Additionally, the empirical update of the parameters δ_j allows the model to capture intricate nuances and complexities that may not be adequately accounted for by Method 1, which relies on a fixed hyperparameter. By updating the parameters based on the observed data, the model can better capture the intricacies and variability present in the data, leading to improved estimation accuracy and model performance.

Method C: This method uses a full empirical Bayes that directly estimates $\boldsymbol{\lambda}$ from observed data without assuming any specific distribution or model. The full derivation is in the Appendix B, but we can update λ_j^2

$$\lambda_j^2 = \frac{r}{\mathbf{E}_{\boldsymbol{\lambda}_j^{(k-1)}}\left[\frac{\tau_j^2}{2} | \lambda_j^{(k-1)}, y^{(k-1)}\right] + \delta_j} \tag{41}$$

While this Method C offers several advantages, such as adapting to the data and improving exploration of the parameter space, they also have certain disadvantages that should be considered.

One of the primary disadvantages of full empirical MCMC is its computational cost. Empirical MCMC methods typically require additional iterations and computations compared to traditional MCMC algorithms. The empirical updates of parameters or proposal distributions can be computationally intensive, particularly when dealing with large datasets or complex models. This can result in longer execution times, limiting the scalability of the method.

Another disadvantage is the potential for bias or inefficiency in the estimation process. Empirical updates rely on the observed network data to estimate the parameters and the proposal distribution of the network. If the nodal sufficient statistics are not fully representative or the observations of nodal random variables contain outliers, the empirical estimates may introduce biases or inefficiencies in the MCMC sampling. Additionally, the convergence of this method 3 needs careful tuning of the other hyperparameters to achieve optimal performance. The optimization of hyperparameters can be nontrivial and needs expert knowledge or extensive experimentation.

Adaptive parallel direction sampling algorithm

There have been considerable developments in the approaches dealing with the problem of sampling from a distribution with a doubly intractable normalizing constant. For example, the easy-to-implement and more direct single variable exchange algorithm proposed by Murray et al. (2012). However, if there is strong temporal dependence in the state process and a strong correlation between model parameters, the exchange algorithm performs slow mixing. Caimo and Friel (2011) and Caimo and Mira (2015) apply the ideas in Murray et al. (2012) to increase MCMC sampling efficiency by combining delayed rejection and adaptive Monte Carlo techniques. First, a collection of H parallel Markov chains are generated. Then the next element of a current chain h is found using estimates from chains h_1 and h_2 as below.

Algorithm: Parallel Adaptive Sampling Algorithm

while $i = 1, ..., N$ **do**

Define a scalar ADS move factor γ, for each chain $\boldsymbol{h} \in \{1, 2, 3, \cdots, H\}$:

1. Sample two current states h_1, h_2 and $h_1 \neq h_2 \neq h$.
2. Sample the error term from a symmetric normal distribution. $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\sigma}_{\epsilon}^2)$.
3. The sampling of $\boldsymbol{\theta}_h$ performs a simple random walk: $\boldsymbol{\theta}_h^{'} = \boldsymbol{\theta}_h + \gamma(\boldsymbol{\theta}_{h_1} - \boldsymbol{\theta}_{h_2}) + \boldsymbol{\epsilon}$.
4. Sample $\boldsymbol{y}'$ from $\pi(\cdot|\boldsymbol{\theta}_h^{'})$.
5. Accept $\boldsymbol{\theta}_h^{'}$ with probability $\min(1, \frac{q(\boldsymbol{y}|\boldsymbol{\theta}_h^{'})\pi(\boldsymbol{\theta}_h^{'})q(\boldsymbol{y}'|\boldsymbol{\theta}_h)}{q(\boldsymbol{y}|\boldsymbol{\theta}_h)\pi(\boldsymbol{\theta}_h)q(\boldsymbol{y}'|\boldsymbol{\theta}_h^{'})})$ (42)

where $q(\boldsymbol{y}|\boldsymbol{\theta}) = e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}$ is the unnormalized likelihood.

end while

The move of $\boldsymbol{\theta}$ is illustrated in Figure 1. Here, two other chains h_1 and h_2 are chosen at random. The difference between the corresponding estimates in the other two chains $\boldsymbol{\theta}_{h_1}$ and $\boldsymbol{\theta}_{h_2}$ are used to find the distance to move away from $\boldsymbol{\theta}_h$. A normal distribution with a very small variance is used to slightly adjust the estimate for the new $\boldsymbol{\theta}$.

Bayesian adaptive lasso algorithm

In this section, we will list the algorithm of the Bayesian Exponential Random Graph Model (BERGM) by Caimo et al. (2017) and the algorithm of our Bayesian Adaptive Lasso Exponential Random Graph Model (BALERGM)) for easy comparison. Caimo et al. (2017) set up the exchange algorithm with a Gibbs update of $\boldsymbol{\theta}'$ and then $\boldsymbol{y}'$ using Markov Chain Monte Carlo iteration without penalized terms. The algorithm can be written in the following concise way:

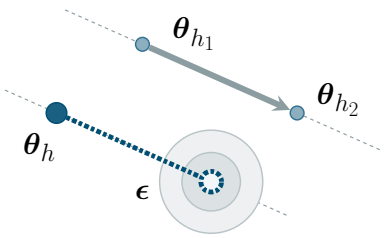

Fig. 1 The parallel ADS move of $\boldsymbol{\theta}_h$ is generated based on the difference of the states $\boldsymbol{\theta}_{h_1}$ and $\boldsymbol{\theta}_{h_2}$ in other Markov chains and $\boldsymbol{\epsilon}$ is a random error term

Algorithm: Bayesian Exponential Random Graph Model

while $i = 1, ..., N$ **do**

while $h = 1, ..., H$ **do**

1. generate h_1 and h_2 such that $h_1 \neq h_2 \neq h$

2. generate $\boldsymbol{\theta}'_h$ from $\gamma(\boldsymbol{\theta}_{h_1} - \boldsymbol{\theta}_{h_2}) + \epsilon(\dot{\cdot}\cdot|\boldsymbol{\theta}_h)$

3. simulate y' from $\pi(\dot{\cdot}\cdot|\boldsymbol{\theta}'_h)$

4. update $\boldsymbol{\theta}_h \rightarrow \boldsymbol{\theta}'_h$ with the log of the probability

$$\min\left(0, [\boldsymbol{\theta}_h - \boldsymbol{\theta}'_h]^T[s(\boldsymbol{y}') - s(\boldsymbol{y})] + \log\left[\frac{\pi(\boldsymbol{\theta}'_h)}{\pi(\boldsymbol{\theta}_h)}\right]\right)$$

end while

end while

Where $s(\boldsymbol{y})$ and $s(\boldsymbol{y}')$ are functions of the observed and simulated vector of network statistics respectively.

For the new Bayesian Adaptive Lasso model, we use the parallel adaptive direction sampler method suggested by BERGM and combine with Gibbs sampling to generate samples to find estimates for $\boldsymbol{\theta}$.

Algorithm: Bayesian Adaptive Lasso Exponential Random Graph Model Algorithm

Require: Set the initial value for $\boldsymbol{\lambda}, \sigma^2, \gamma$, Use ERGM to find MPLE (Maximizer to the Psuedolikelihood Function) to find initial values for $\boldsymbol{\theta}$. Denote samples of $\boldsymbol{\theta}$ in the hth chain, as $\boldsymbol{\theta}_h$.

while $i = 1, ..., N$ **do**

while $h = 1, ..., H$ **do**

1. sample $\boldsymbol{\theta}_h$ with Parallel Adaptive Direction Sampler:

a. generate h_1 and h_2 such that $h_1 \neq h_2 \neq h$

b. update $\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}$

c. generate $\boldsymbol{\theta}'_h$ from $\gamma(\boldsymbol{\theta}_{h_1} - \boldsymbol{\theta}_{h_2}) + \epsilon(\dot{\cdot}\cdot|\boldsymbol{\theta}_h)$

d. simulate $\boldsymbol{y}'$ from $\pi(\dot{\cdot}\cdot|\boldsymbol{\theta}'_h)$

e. update $\boldsymbol{\theta}_h \rightarrow \boldsymbol{\theta}'_h$ with the log of the probability

$$\min\left(0, [\boldsymbol{\theta}_h - \boldsymbol{\theta}'_h]^T[s(\boldsymbol{y}') - s(\boldsymbol{y})] + \log\left[\frac{\pi(\boldsymbol{\theta}'_h)}{\pi(\boldsymbol{\theta}_h)}\right]\right)$$

where $\pi(\boldsymbol{\theta}) \sim \mathcal{N}(0_p, \sigma^2\boldsymbol{D}_{\boldsymbol{\tau}})$

2. sample σ^2 from Inverse Gaussian$(\frac{p}{2}, -\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\theta})$

3. sample τ_j^2 for $j = 1, 2, 3, ..., p$ from Inverse Gaussian $\left(\sqrt{\frac{\lambda_j^2\sigma^2}{\theta_j^2}}, \lambda_j^2\right)$

4a. full Bayes update of $\boldsymbol{\lambda}$

1. sample λ_j^2 for $j = 1, 2, 3, ..., p$ from Gamma $(r + 1, \frac{\tau_j^2}{2} + \delta_j)$

OR

4b. empirical update of $\boldsymbol{\delta}$ and update of $\boldsymbol{\lambda}$

1. update δ_j for $j = 1, 2, 3, ..., p$ with the mean of the last five $\boldsymbol{\lambda}$ samples estimating the expected value.

$$\delta_j = \frac{r}{\mathbf{E}_{\delta_j^{(k-1)}}\left[\lambda_j^2|\delta_j^{(k-1)}, y^{(k-1)}\right]}$$

2. sample λ_j^2 for $j = 1, 2, 3, ..., p$ from Gamma $(r + 1, \frac{\tau_j^2}{2} + \delta_j)$

OR

4c. full empirical update of $\boldsymbol{\lambda}$

1. update $\boldsymbol{\lambda}$ with the mean of the last five $\boldsymbol{\tau}$ estimating the expected value

Algorithm: Bayesian Adaptive Lasso Exponential Random Graph Model Algorithm

$$\lambda_j^2 = \frac{r}{\mathbf{E}_{\lambda_j^{(k-1)}}\left[\frac{\tau_j^2}{2}|\lambda_j^{(k-1)}, y^{(k-1)}\right]+\delta_j}$$

end while

end while

This code was built with R version 4.1.1 (2021-08-10). R Core Team (2021) The following package versions were also used: coda 0.19-4, mcmc 0.9-7, Bergm 5.0.3, ergm.count 4.0.2, ergm 4.1.2, mvtnorm 1.1-3.

This BALERGM package is shared on Github:xxxx. (The link will be provided upon the acceptance of this paper).

Simulation and application

In this section, we will show the strength of BALERGM in three key ways. The first way uses the Faux Dixon High School data set to simulate 100 graphs to compare BERGM and BALERGM. The results of trials shows BALERGM is a stable model with accurate estimation, in addition to providing improvements to BERGM with a higher acceptance rate and effective sampling size, and lower MSE. The next two ways showcase the parameter selection abilities of BALERGM. The first on a simulated parameter and the second with network data collected in a study by Prevention Research Center at USC and Sumter County Active Lifestyles (SCAL).

Data

The network object Faux Dixon High represents a friendship network among junior high and high school students based on data gathered by a National Longitudinal Study of Adolescent Health, see details in Resnick et al. (1997). This study, first conducted in 1994–1995, considered more than 90,0000 American students. Students were asked to list friends, and a tie is formed between them in the network if both students claimed friendship (Goodreau et al. 2008).

The final network has 248 nodes with 1,197 directed edges. Each node has three characteristics: grade, sex, and race. The grades include 7th-12th and race is first delineated by Hispanic and non-Hispanic which was further split into Asian, Black, Native American, Other, and White. Figure 2 shows the network plotted with nodes colored for each grade showing the homophily.

Executing the BALERGM algorithm requires choosing network statistics with both nodal and edge attributes and structural features such as triangles and triads (Morris et al. 2008). The count of these network statistics is found with the adjacency matrix realization $\boldsymbol{y}$ of $\boldsymbol{Y}$ with i-j entry in the matrix defined as y_{ij}. For a directed network, the following summations demonstrate the counting procedure.

$$\text{Edges: } \sum_{i\neq j} y_{ij} \qquad \text{Mutual Edges: } \sum_{i\neq j} y_{ij}y_{ji} \qquad \text{Cyclic Triads: } \sum_{i\neq j\neq k} y_{jk}y_{i,k}y_{ij}$$

A natural network statistic for this data is the instances of homophily between students in the same grade, since as seen in Fig. 2, nodes with the same attribute (in this case

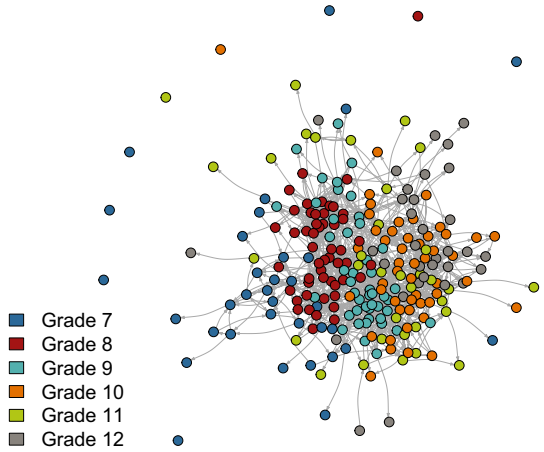

Fig. 2 Generated in R, this plot shows the clustering of student friendship with students that have the same grade

Table 3 Summary table for the connections among different grades. The $i - j$ position in this table shows the number of connections from Grade i to Grade j, $i = 7, 8, 9, 10, j = 7, 8, 9, 10$

	Grade 7	Grade 8	Grade 9	Grade 10	Grade 11	Grade 12	Sum
Grade 7	42	5	8	3	3	1	62
Grade 8	9	263	48	10	7	4	341
Grade 9	13	53	184	35	32	15	332
Grade 10	3	14	46	183	14	13	273
Grade 11	0	2	13	12	42	16	85
Grade 12	0	4	11	10	8	71	104
Sum	67	341	310	253	106	120	1197

grade) appear visually to have more connections. As seen in Table 3, with the diagonal entries of the mixing matrix from Grade i to Grade i for $i \in \{7, 8, 9, 10, 11, 12\}$, most of the connections are between students in the same grade. This feature can be included in network models with the R code nodematch.

Simulation

To demonstrate the overall effectiveness of BALERGM, we conducted a comparative analysis between BALERGM and BERGM (Caimo and Friel 2014). Our evaluation involved the generation of 100 independent exponential random graphs using the Faux Dixon High dataset, with known and fixed parameters $\boldsymbol{\theta}$. Specifically, we focused on two selected network statistics: the count of edges in the network (θ_1 : edges) and the count of occurrences of homophily, where students of the same grade have a friendship connection (θ_2 : nodematch.Grade). Without loss of generality, we fixed the parameter values $\boldsymbol{\theta} = (-4.8, 2.3)$ and generated 100 independent exponential random graphs based on the Faux Dixon High dataset, considering them as new instances with associated node attributes. This approach allowed us to create 100 distinct opportunities to estimate the parameter vector $\boldsymbol{\theta}$ using both the BALERGM and BERGM algorithms, enabling a comprehensive performance comparison against the true parameter values $\boldsymbol{\theta} = (-4.8, 2.3)$.

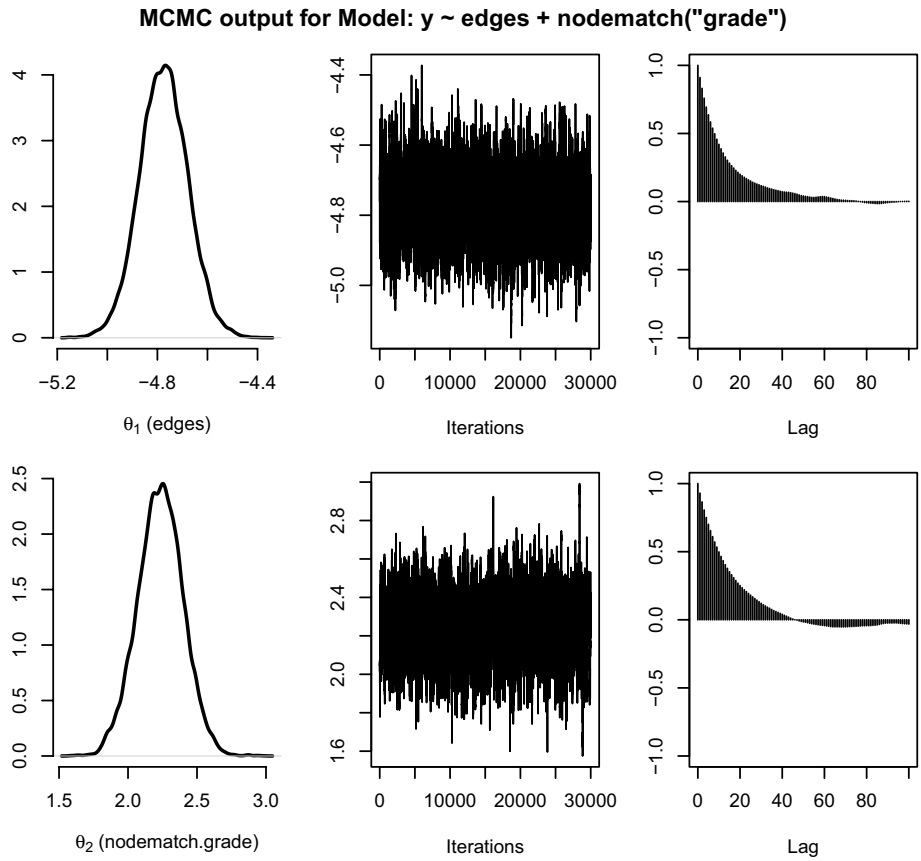

Fig. 3 MCMC output: Distribution of samples on the left, the trace of samples in the center, autocorrelation plot on the right

In each run of BERGM and BALERGM, the main chain for either model consists of 2000 iterations and the burn-in number is 50 iterations. In 100 simulations, each model generates a sequence of values estimating each $\boldsymbol{\theta}$ in each simulation. To confirm the stability of the model, the following representation of the MCMC results (Fig. 3) shows the strength and stability of the BALERGM algorithm after relatively few iterations. The unimodal distribution of estimates is on the left, and the center column shows the trace of the estimates indicating a stable estimating process. The final column shows the autocorrelation plot with the lag decreasing quickly; by 50 iterations, the process has stabilized to minimal lag.

Using both the mean and median of these estimates we can compare several metrics. Table 4 compares the acceptance rate of generated estimates for each run, the mean effective sample size, and both the mean and median square error (*MSE*) of the estimates compared to the chosen true values, where $MSE = \frac{1}{n}\boldsymbol{e}^\top \boldsymbol{e}$, $\boldsymbol{e}$ is the error vector, that is $\boldsymbol{e} = \hat{\boldsymbol{\theta}} - (-4.8, 2.3)$.

Table 4 shows that using either the mean or median of the generated estimates in MCMC for $\boldsymbol{\theta}$ (1) BALERGM has a better overall acceptance rate and effective sample size on average than BERGM. The acceptance rate or the percentage of generated

Table 4 Results of both BERGM and BALERGM using formula y ~ edges + nodematch("Grade")

Results					
		Mean AR	**Mean ESS**	**Mean squared error**	**Median squared error**
BERGM	θ_1	0.5286	222.16	4.641585	2.845891
	θ_2		220.69		
BALERGM	θ_1	0.5607	246.72	0.1960137	0.06896186
	θ_2		221.82		

samples that are accepted in the MCMC process is increased. This implies BALERGM adjusts to the true parameter for each single variable faster than BERGM. (2) BALERGM offers an improvement over BERGM with a lower mean squared error (MSE). The mean squared error is dramatically lower with the BALERGM process no matter whether the mean or median in MCMC is used as the estimate for $\boldsymbol{\theta}$. This can be seen in the quantiles for each estimate of $\boldsymbol{\theta}$ since the true values are $\boldsymbol{\theta} = (-4.8, 2.3)$, the BALERGM estimates are much closer to these true values.

In Table 5, the true known value of each $\boldsymbol{\theta}$ is estimated by either the mean or median of the generated samples. The quantiles for estimates of $\boldsymbol{\theta}$ show the spread of each estimate.

Once results are generated, the estimates produced can be used to calculate the probability of a tie, using the $\boldsymbol{\theta}$. Using the previous example, if no new tie is created, so the change statistic for θ_1 is zero, then the probability that a tie is between students of the same grade can be calculated as follows.

$$P(Y_{ij} = 1|\theta_2 = 2.8550) = \frac{e^{2.8550}}{1 + e^{2.8550}} = 0.945577.$$

Table 5 Results of simulating 100 graphs and comparing results for BERGM and BALERGM using means as the estimates of $\boldsymbol{\theta}$

Mean of the MCMC output as the estimate for θ								
		True Value [a]	**Estimate** [b]	**Quantiles**[c]				
				2.5%	***25%***	***50%***	***75%***	***97.5%***
BERGM	θ_1	−4.800	−5.3470	−5.607	−5.456	−5.349	−5.253	−5.031
	θ_2	2.300	5.2656	4.455	5.026	5.251	5.543	6.185
BALERGM	θ_1	−4.800	−4.9444	−5.172	−5.018	−4.938	−4.871	−4.739
	θ_2	2.300	2.8550	2.496	2.680	2.824	2.996	3.304
Median of the MCMC output as the estimate for $\boldsymbol{\theta}$								
		True Value	Estimate [d]	Quantiles				
				2.5%	*25%*	*50%*	*75%*	*97.5%*
BERGM	θ_1	−4.800	−5.1367	−5.386	−5.235	−5.137	−5.045	−4.874
	θ_2	2.300	4.6296	3.903	4.397	4.632	4.822	5.355
BALERGM	θ_1	−4.800	−4.8778	−5.077	−4.949	−4.873	−4.806	−4.682
	θ_2	2.300	2.6131	2.398	2.501	2.595	2.713	2.918

[a] Chosen true value for parameter for each simulated graph

[b] Mean of MCMC outputs

[c] Quantiles from MCMC output

[d] Median of MCMC outputs

That means the conditional probability of observing an edge (not involved in the creation of other network statistics included in the model) is about 94.56%.

Variable selection

BALERGM not only improves sampling efficiency compared to previous models but also demonstrates strong performance in variable selection through its adaptive lasso component. This indicates the ability of the model to identify and highlight parameters that are either more or less significant to the network structure. The example using the following simulated dataset showcases the effectiveness of BALERGM in terms of variable selection.

For this example, we still use Faux Dixon high school dataset. The chosen network statistics are the count of the edges in the network (edges), the counts of the occurrences of homophily where students of the same grade have a friendship connection (nodematch. Grade), and the third artificial created term: the counts of the occurrences of homophily from a generated nodal attribute for the wealth of a parent (nodemath.Wealth). This additional nodal attribute "Wealth" is generated from the uniform distribution on 20 and 75. Given that this nodal variable is generated uniformly at random, it is intentionally designed to have no impact on the network structure. Our objective is to test whether BALERGM can effectively identify and exclude this artificially created insignificant nodal variable. Running the BALERGM algorithm produces the following results, demonstrating that the model accurately estimates the value of θ_3 to be close to zero. This outcome aligns with our expectations, as the variable has no meaningful influence on the network structure (Table 6, Fig. 4).

Sumter county active lifestyles (SCAL) network analysis

With reports by The US Burden of Disease Collaborators (2018) of worsening metrics of American health, communities are working on addressing and understanding the factors that might improve health outcomes. To this end, the University of South Carolina Prevention Research Center and Sumter County Active Lifestyles (SCAL) based in Sumter County, South Carolina conducted a respondent-driven sampling study in 2014 to better understand the dynamics of social networks and health outcomes.

In this study, community ambassadors chosen for their history of community involvement were given a set compensation for their participation. Each ambassador was instructed to share the survey with those in their social network. Each of these respondents was also compensated for both completion of the survey and sharing the survey with others that completed the survey. Using referral codes, a network can be

Table 6 Results of BALERGM using formula y ~ edges + nodematch("Grade") + nodematch("Wealth")

BALERGM result for parameter selection				
	Estimate Mean	**SD**	**Naive SE**	**Time-series SE**
θ_1 (edges)	−4.8005296753	0.07659603	0.0001398446	0.001769280
θ_2 (nodematch("Grade"))	2.2939927192	0.09614381	0.0001755338	0.002292080
θ_3 (nodematch("Wealth"))	0.0001696885	0.01967663	0.0000359245	0.000151961

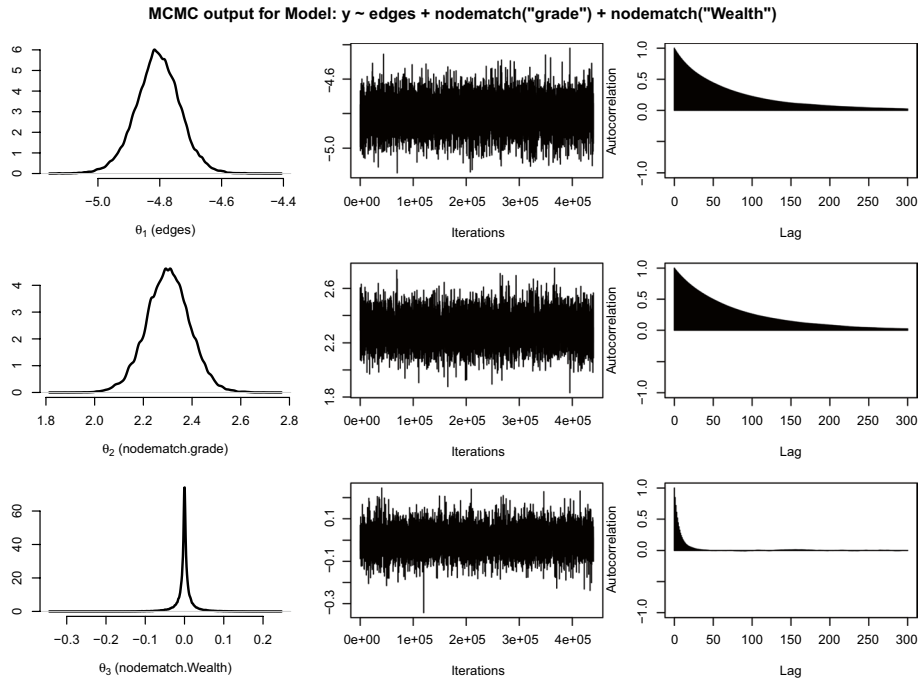

Fig. 4 MCMC output: Distribution of samples on the left, the trace of samples in the center, autocorrelation plot on the right

created with nodes representing survey respondents, edges formed by survey sharing, and nodal characteristics from the results of the survey. The final network has 80 nodes with the data for 30 questions for each respondent. Figure 5 is the network plot labeled with one of the 30 questions: "Have you heard of a group called Sumter County Active Lifestyles (SCAL)?"

The survey was intended to be a brief but broad look at self-reported health benchmarks. Questions cover demographic characteristics revealing that the respondents

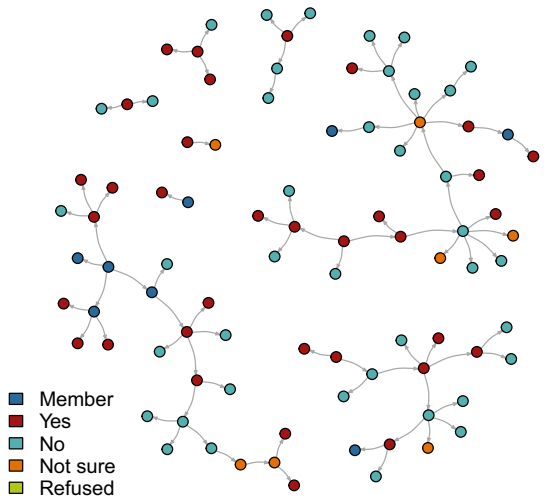

Fig. 5 Generated in R, this plot shows results of asking "Have you heard of a group called Sumter County Active Lifestyles (SCAL)?"

are primarily white (87%), female (78%), likely to be older than 50 (44%), and more educated with 46% being college graduates. Other questions focused on self-reported health outcomes and activities including exercise habits, eating habits, and social support dynamics. The question forms included qualitative questions about physical activities and opportunities for physical activities in the community. For the purposes of this network, network attributes were assigned using the answers to only multiple-choice questions.

The resulting network contains many nodal attributes where ERGM and BERGM cannot be applied effectively. This motivates a model like BALERGM which enables understanding which of these network statistics contribute less to the network structure.

Results

Using the SCAL data set from the previous section, we use the network statistics in Table 7 to analyze this model, where we use the ergm terms "nodematch", "nodefactor" and "nodecov". These three terms all provide measures of homophily. "nodematch" counts the instances of nodes with the same attribute for a given attribute. "nodecov" performs a similar function but for continuous variables. "nodefactor" creates network statistics for each discrete level of a nodal attribute and counts the occurrences of connected nodes with the same attribute level. For more details, see Morris et al. (2008).

Table 7 shows the BALERGM output on the SCAL social network. Here the sparsity of the network can be seen in the large negative values for the network statistics for edges and the out-degree of the nodes. While the standard deviations vary with each estimate, the MCMC outputs show stable estimating with symmetric distributions as the quantile values indicate. It 7 provides valuable insights into the relationships between different variables in the network analysis. One interesting observation is that individuals who maintain a healthy diet (θ_6-θ_{13} are all positive) tend to have positive connections with each other. This suggests a clustering effect among individuals with similar dietary habits, indicating a potential influence of shared health-conscious behaviors on network connections.

Furthermore, the result highlights that participation in a walking program (variables θ_{18} to θ_{19}) is positively associated with network connections. This implies that individuals who engage in walking programs are more likely to know each other within the network. This finding suggests a potential social bonding effect among individuals who actively participate in health-promoting activities, leading to the formation of connections and social ties.

The adaptive lasso penalty in BALERGM is useful for shrinking $\boldsymbol{\theta}$ values for network statistics that are less significant to the network structures. Depending on the model and network conditions, the parameter estimate might not reach exactly zero. For example, the estimate for both $\theta_{26} = -.001$ and $\theta_{20} = -.076$ are small, but this mean of the generated samples as the single factor utilized doesn't allow for a nuanced ranking of how significant each parameter is. Using the distribution of $\boldsymbol{\theta}$ found in the Gibbs sampling

Table 7 Results from BALERGM with variable selection on SCAL data

Result for parameter selection							
Network statistic	**Mean**	**SD**	**Quantiles**				
			2.5%	**25%**	**50%**	**75%**	**97.5%**
θ_1 (edges)	−5.794	0.995	−7.770	−6.460	−5.782	−5.121	−3.879
θ_2 (out degree 0)	1.212	0.64	−0.051	0.795	1.216	1.631	2.477
θ_3 (out degree 1)	−1.025	0.526	−2.066	−1.372	−1.025	−0.679	0.016
θ_4 (out degree 2)	−0.629	0.407	−1.428	−0.901	−0.632	−0.363	0.189
θ_5 (out degree 3)	−0.207	0.319	−0.853	−0.410	−0.209	−0.002	0.431
θ_6 (1 serving fruit/day) [a]	0.654	0.308	0.046	0.453	0.655	0.859	1.265
θ_7 (2 servings fruit/day) [a]	0.536	0.304	−0.063	0.334	0.534	0.738	1.134
θ_8 (3-4 servings fruit/day) [a]	0.608	0.329	−0.054	0.391	0.612	0.829	1.245
θ_9 (5+ servings fruit/day) [a]	0.877	0.461	−0.066	0.577	0.887	1.186	1.772
θ_{10} (1 serving vegetables/day) [a]	0.451	0.310	−0.164	0.248	0.454	0.655	1.062
θ_{11} (2 servings vegetables/day) [a]	0.477	0.295	−0.106	0.283	0.478	0.675	1.056
θ_{12} (3-4 servings vegetables/day) [a]	0.587	0.292	0.018	0.392	0.587	0.782	1.165
θ_{13} (5+ servings vegetables/day) [a]	0.096	0.359	−0.644	−0.128	0.109	0.335	0.770
θ_{14} (vigorous phys. activities/week)[b]	−0.011	0.206	−0.426	−0.145	−0.008	0.125	0.389
θ_{15} (moderate phys. activities/week)[c]	0.008	0.042	−0.075	−0.019	0.009	0.037	0.090
θ_{16} (days walking 10min/week)[c]	−0.018	0.035	−0.088	−0.042	−0.019	0.006	0.052
θ_{17} (days using parks/month) [c]	−0.030	0.028	−0.090	−0.049	−0.030	−0.011	0.022
θ_{18} (heard of walking program) [b]	0.333	0.214	−0.087	0.190	0.334	0.478	0.752
θ_{19} (participate in walking program)[b]	0.186	0.244	−0.305	0.027	0.188	0.346	0.668
θ_{20} (heard of SCAL)[b]	0.076	0.191	−0.310	−0.049	0.081	0.206	0.448
θ_{21} (general health is very good)[a]	−0.203	0.206	−0.616	−0.339	−0.201	−0.065	0.194
θ_{22} (general health is good) [a]	−0.247	0.204	−0.650	−0.381	−0.248	−0.113	0.154
θ_{23} (general health is fair) [a]	−0.072	0.207	−0.470	−0.211	−0.077	0.063	0.349
θ_{24} (general health is poor) [a]	−0.479	0.477	−1.505	−0.773	−0.456	−0.160	0.402
θ_{25} (gender) [a]	−0.064	0.15	−0.362	−0.163	−0.064	0.035	0.226
θ_{26} (age) [c]	−0.001	0.005	−0.012	−0.005	−0.001	0.002	0.009
θ_{27} (highest year of school completed)[b]	−0.165	0.205	−0.571	−0.302	−0.166	−0.029	0.240

[a] nodefactor

[b] nodematch

[c] nodecov

process, we can find the probability that half of the distribution is less than zero at some significance level α.

$$|P(\boldsymbol{\theta} < 0) - 0.5| = \alpha$$

A larger value of α indicates a higher importance of the variable in the context of the model. This creates the ability to rank variables. The following Table 8 shows the parameters less significant to the construct of the network at various significance levels. For example, the network statistic of the age of the participant (θ_{26}) is less significant than for the network statistic of having heard of the SCAL program (θ_{20}). While both are not the primary factors, BALERGM gives researchers insights into the social dynamics of Sumter County allowing for targeted programs to improve health outcomes.

Table 8 Variable Selection with Different Tolerance Levels

Tolerance Level	Variable Index Number
0.05	26
0.10	16 20 25 26
0.15	4 16 20 25 26
0.20	4 14 15 16 20 22 25 26

This example highlights the powerful functionalities of BALERGM, particularly in the context of variable selection and importance ranking in network analysis. In network studies, the presence of numerous network variables is common. The identification of the most relevant variables is crucial as it enables researchers to concentrate their analysis and interpretation on the factors that significantly influence the network's structure and behavior. By focusing on these key variables, we can gain a deeper understanding of the underlying mechanisms that drive network dynamics.

Goodness of fit

To assess the performance and goodness of fit of Exponential Random Graph Models (ERGMs), various diagnostics can be employed. These diagnostics involve comparing key statistical measures calculated from observed networks with those obtained from simulated networks generated based on the estimated network parameters. In the Bayesian framework, evaluating the goodness of fit of the model involves conducting posterior predictive assessments. This entails comparing the observed network to a collection of networks simulated from the posterior distribution of the model's parameter estimates, as determined by Caimo and Friel (2011).

The set of statistics used for the comparison contains degree distributions, the minimum geodesic distance, and the number of edge-wise shared partners. Since the SCAL network graph is a directed graph, the degree distributions for both in and out degrees are included. Since the graph includes isolated nodes and clusters such that there is no path between some nodes, the minimum geodesic distance or the minimum number of edges needed to connect any two nodes is infinite leading to the spike in the plot for minimum geodesic distance in 6. Finally, the edge-wise shared partners are concentrated in the lower values since the number of nodes in common for any number of edges is small.

The Bayesian goodness of fit diagnostic in Fig. 6 evaluates the implemented model in section 8. The observed network is compared with 300 randomly simulated network samples drawn from the estimated posterior distribution using 50 auxiliary iterations for the network simulation step. Figure 6 illustrates the summary results of these 300 generated graphs in black and gray, alongside the original network represented in red. The comparison reveals a strong alignment in the high-level characteristics that are not explicitly modeled. This indicates that the posterior mean obtained through BALERGM accurately generates networks with corresponding structures.

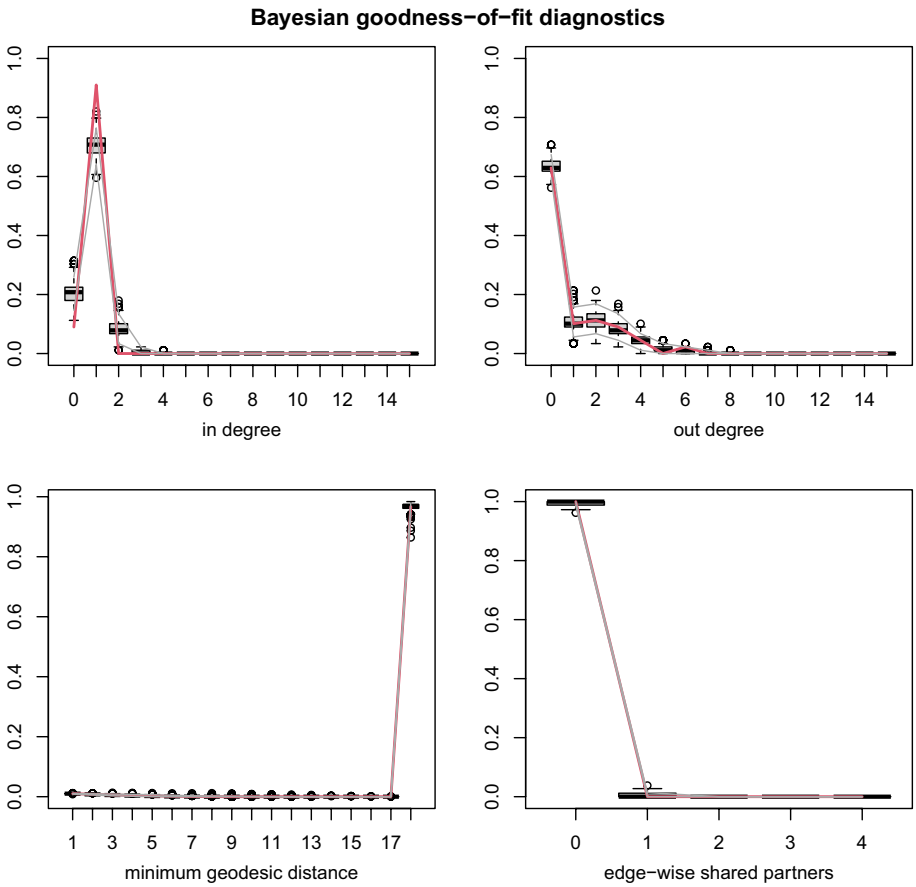

Fig. 6 Bayesian goodness of fit diagnostic for the estimated parameter posterior distribution for BALERGM model on SCAL dataset

Discussion

Bayesian adaptive lasso exponential random graph model (BALERGM) offers several notable advantages in the field of network analysis. One key advantage is its ability to perform automatic variable selection, facilitated by the integration of the Lasso regularization technique. By employing the Lasso penalty, BALERGM effectively identifies and emphasizes the most relevant network parameters while diminishing the influence of less significant ones towards zero. This feature streamlines the modeling process and extracts valuable insights from intricate network data, enhancing the interpretability of the results. Moreover, the Lasso penalty promotes sparsity in parameter estimates, resulting in a more parsimonious model that aids in discerning the influential factors governing network behavior.

Another compelling advantage of BALERGM is its superior adaptive estimation performance. Through the adaptive adjustment of penalties for each parameter, the model swiftly adapts to the data, allowing it to focus on the most relevant network parameters and capture underlying patterns and relationships more effectively. Researchers can readily select key network factors based on their significance levels, providing valuable insights and actionable knowledge.

We have shown the effectiveness of the the proposed algorithm in simulation and compared its performance against the currently popular BERGM method. One promising direction for future work involves the development of more generalized penalized forms within the context of network analysis. While the Lasso penalty has demonstrated its efficacy in variable selection and sparsity promotion, the incorporation of ridge penalty distributions could offer additional benefits. Combining the strengths of both Lasso and ridge penalties would strike a balance between model complexity and over/underfitting issues, leading to more robust parameter estimation.

Appendix A: proof of unimodal posterior

The chosen prior needs to result in a unimodal posterior for faster Gibbs sample convergence and confidence that the estimates found are actually best.

Theorem:

The joint posterior distribution is unimodal for typical choices of $\pi(\sigma^2)$ and any choice of $\lambda \geq 0$.

Proof:

We begin by representing the joint distribution of θ and $\sigma^2 > 0$ using distributions already defined.

$$\pi(\boldsymbol{\theta}, \sigma^2) \propto \pi(\boldsymbol{\theta}|\sigma^2)\pi(\sigma^2) \tag{42}$$

$$= \prod_{j=1}^{p} \frac{\lambda_j}{2\sqrt{\sigma^2}} e^{-\lambda_j|\theta_j|/\sqrt{\sigma^2}} \frac{1}{\sigma^2} \tag{43}$$

We have choose the prior such that $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$ according to the recommendation of the literature. Park and Casella (2008)

We wish to show that the posterior is unimodal in the sense that every upper-level set of $\{(\boldsymbol{\theta}, \sigma^2)|\pi(\boldsymbol{\theta}, \sigma^2) > x, \sigma^2 > 0\}$, for $x > 0$ is connected. We will show this is true under a continuous transform with continuous inverse since the continuous image of a connected set is connected. Munkres and Davis (2018)

The posterior is shown here

$$\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{y}) \propto \pi(y|(\boldsymbol{\theta}, \sigma^2))\pi(\boldsymbol{\theta}, \sigma^2) \tag{44}$$

$$= \pi(\boldsymbol{y}|(\boldsymbol{\theta}, \sigma^2))\pi(\boldsymbol{\theta}|\sigma^2)\pi(\sigma^2) \tag{45}$$

$$= \frac{1}{z(\boldsymbol{\theta})} e^{\boldsymbol{\theta}^T s(\boldsymbol{y})} \prod_{j=1}^{p} \frac{\lambda_j}{2\sqrt{\sigma^2}} e^{-\lambda_j|\theta_j|/\sqrt{\sigma^2}} \frac{1}{\sigma^2} \tag{46}$$

$$= \frac{e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}}{z(\boldsymbol{\theta})} \frac{1}{\sigma^2} \frac{1}{2^p \sqrt{\sigma^2}^p} \prod_{j=1}^{p} \lambda_j e^{-\lambda_j|\theta_j|/\sqrt{\sigma^2}} \tag{47}$$

We now take the natural log of the equation above.

$$\ln \pi(\boldsymbol{\theta}, \sigma^2|y) = -\ln(\sigma^2) + \boldsymbol{\theta}^T s(\boldsymbol{y}) - \sum_{j=1}^{p} \lambda_j |\theta_j| \frac{1}{\sqrt{\sigma^2}} + \sum_{j=1}^{p} \ln(\lambda_j) - p\ln(2) - \frac{p}{2}\ln(\sigma^2) \tag{48}$$

The following transform allows for easier calculation.

$$\phi_j \leftrightarrow \frac{\theta_j}{\sqrt{\sigma^2}} \qquad \rho \leftrightarrow \frac{1}{\sqrt{\sigma^2}} \qquad j = 1, 2, 3, ...p$$

This is continuous with a continuous inverse when $0 < \sigma^2 < \infty$, so the upper-level sets for the original parameters correspond under the transformation to upper-level sets for the original parameters.

Let $\boldsymbol{\phi} = (\phi_1, \phi_2, ...\phi_p)^T$ be the column vector for ease of notation. This transform is one-to-one and continuous for $0 < \sigma^2 < \infty$, therefore the unimodality of the transformed equation is equivalent to the unimodality of the original equation.

Using the transform and algebra we get the following expression

$$\begin{aligned} h(\boldsymbol{\phi}, \rho) &= \ln \rho^2 + (\sqrt{\sigma^2}\boldsymbol{\phi})^T s(\boldsymbol{y}) - \sum_{j=1}^{p} \lambda_j |\phi_j| + \frac{p}{2}\ln(\rho^2) \\ &= (p+2)\ln(\rho) + \frac{\boldsymbol{\phi}^T s(\boldsymbol{y})}{\rho} - \sum_{j=1}^{p} \lambda_j |\phi_j| \end{aligned} \tag{49}$$

We can show that (A8) is unimodal by showing it is a concave function in $(\boldsymbol{\phi}, \rho)$. We will do that by considering each term of the equation in turn.

$$h_1 = \ln(\rho) \qquad h_2 = \frac{\boldsymbol{\phi}^T s(\boldsymbol{y})}{\rho} \qquad h_3 = -\sum_{j=1}^{p} \lambda_j |\phi_j|$$

We will determine the concavity of the first two functions by checking the spectral property of the corresponding Hessian matrix.

$$H_{h_i} = \begin{bmatrix} \frac{\partial^2 h_i}{\partial \phi^2} & \frac{\partial^2 h_i}{\partial \boldsymbol{\phi} \partial \rho} \\ \frac{\partial^2 h_i}{\partial \rho \partial \boldsymbol{\phi}} & \frac{\partial^2 h_i}{\partial \rho^2} \end{bmatrix}, i = 1, 2. \tag{50}$$

For the first term and the second term $h_1 = \ln(\rho)$ and $h_2 = \frac{\boldsymbol{\phi}^T s(\boldsymbol{y})}{\rho}$, the corresponding Hessian matrix H_{h_1} and H_{h_2} are both negative semi-definite and thus h_1 and h_2 are both concave in $(\boldsymbol{\phi}, \rho)$.

For the third term $h_3 = -\sum_{j=1}^{p} \lambda_j |\phi_j|$, we see this is a sum of the negative of a constant times an absolute value function. This is a concave function in $\boldsymbol{\phi}, \rho$, since the j the term in h_3 is $h_3(j) = -\lambda_j |\phi_j|$ which is a concave function of ϕ_j and the sum of concave functions is a concave function.

Using the same reasoning that the sum of concave functions is concave gives that (A8) is concave, and hence the posterior distribution is concave.

Therefore, we can conclude that our posterior distribution is unimodal.

Appendix B: empirical bayes

The Monte Carlo Expectation-maximization algorithm for empirical Bayes estimation of hyperparameters proposed by Levine and Casella (2001) essentially treats the parameters as missing data and then uses the E-M algorithm to iteratively approximate the hyperparameters substituting Monte Carlo estimates for any expected values that cannot be computed explicitly. For BALERGM, the Gibbs sampler is used to estimate the expected values.

Method B: Estimating δ_j

To begin this process, we consider the part of the joint distribution that depends on $\boldsymbol{\delta}$, since when taking the derivative all other terms will become zero.

$$\pi(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\delta}) = \frac{\delta_j^r}{\Gamma(r)} \left(\lambda_j^2\right)^{(r-1)} e^{-\delta_j \lambda_j^2} + \text{ terms not involving } \delta_j^2. \tag{51}$$

We then take the natural log of the resulting equation.

$$\ln(\delta_j | \boldsymbol{y}, \boldsymbol{\theta}) \propto r \ln(\delta_j) - \delta_j \lambda_j^2. \tag{52}$$

1. Expectation step

$$Q(\delta_j | \delta_j^{(k-1)}, y^{(k-1)}) = \mathbb{E}_{\delta^{(k-1)}} \left[\ln(\delta_j | \boldsymbol{y}, \boldsymbol{\theta}) | \delta_j^{(k-1)}, y^{(k-1)} \right] \tag{53}$$

$$= r \ln(\delta_j) - \delta_j \mathbb{E}\left[\lambda_j^2 | \delta_j^{(k-1)}, y^{(k-1)} \right] + \text{ other terms not involving } \delta_j \tag{54}$$

2. Maximization step

$$\delta_j^{(k)} = \arg\max_{\delta_j} Q(\delta_j | \delta_j^{(k-1)}, \boldsymbol{y}^{(k-1)}). \tag{55}$$

Solve

$$\frac{\partial Q}{\partial \delta_j} = \frac{r}{\delta_j} - \mathbb{E}\left[\lambda_j^2 | \delta_j^{(k-1)}, y^{(k-1)} \right] = 0, \tag{56}$$

we get

$$\delta_j = \frac{r}{\mathbb{E}\left[\lambda_j^2 | \delta_j^{(k-1)}, y^{(k-1)} \right]}. \tag{57}$$

Method C: Estimating λ_j

The empirical process of estimating λ_j begins with the joint distribution terms that depend on λ_j.

$$\pi(\boldsymbol{\theta}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{\tau} | \boldsymbol{y}, s(\boldsymbol{y})) \propto \pi(\boldsymbol{y} | \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \sigma^2, \boldsymbol{\tau}) \prod_{j=1}^{p} \pi(\boldsymbol{\tau} | \lambda_j^2) \pi(\lambda_j^2) \pi(\sigma^2) \tag{58}$$

$$= \frac{e^{\boldsymbol{\theta}^T s(\boldsymbol{y})}}{z(\boldsymbol{\theta})} \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi j^2}} exp\left\{ -\frac{1}{2\tau_j^2}\theta_j^2 \right\} \frac{\lambda_j^2}{2} \exp\left\{ -\frac{\lambda_j^2 \tau_j^2}{2} \right\} \frac{\delta_j^r}{\Gamma(r)} \left(\lambda_j^2\right)^{r-1} e^{-\delta_j \lambda_j^2} \frac{1}{\sigma^2} \quad (59)$$

Next, we take the natural log

$$\ln \pi(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{\tau} | \boldsymbol{y}, s(\boldsymbol{y})) = \sum_{j=1}^{p} \left[r \ln(\lambda_j^2) - \lambda_j^2 \left(\frac{\tau_j^2}{2} + \delta_j \right) \right] + \text{ terms not involving } \boldsymbol{\lambda}. \quad (60)$$

1. Expectation step

$$Q(\lambda | \lambda^{(k-1)}, \boldsymbol{y}^{(k-1)}) = \mathbb{E}_{\lambda^{(k-1)}} \left[\ln \pi(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{\tau} | \boldsymbol{y}, s(\boldsymbol{y})) | \lambda^{(k-e)}, \boldsymbol{y}^{(k-e)} \right]$$

$$= \sum_{j=1}^{p} r \ln(\lambda_j^2) - \sum_{j=1}^{p} \lambda_j^2 \left(\mathbb{E}_{\lambda^{(k-1)}} \left[\tau_j^2 | \boldsymbol{y}^{(k-1)}, \lambda^{(k-1)} \right] + \delta_j \right) + \text{ terms not involving } \boldsymbol{\lambda}. \quad (61)$$

2. Maximization step

$$\boldsymbol{\lambda}^{(k)} = \arg\max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda} | \boldsymbol{\lambda}^{(k-1)}, \boldsymbol{y}^{(k-1)}). \quad (62)$$

Solve

$$\frac{\partial Q}{\partial \lambda_j} = \frac{2r}{\lambda_j} - 2\lambda_j \left(\mathbb{E}_{\boldsymbol{\lambda}^{(k-1)}} \left[\tau_j^2 | \boldsymbol{y}^{(k-1)}, \lambda^{(k-1)} \right] + \delta_j \right) = 0, \quad (63)$$

we get

$$\lambda_j^2 = \frac{r}{\mathbb{E}_{\boldsymbol{\lambda}^{(k-1)}} \left[\tau_j^2 | \boldsymbol{y}^{(k-1)}, \lambda^{(k-1)} \right] + \delta_j}. \quad (64)$$

Thus these conditional expectations are just the posterior expectations under the hyper-parameter $\lambda^{(k-1)}$ thus they can be estimated using the sample averages from a run of the Gibbs sampler described in the section.

Author contributions
RP and DH conceived the research. MF collected the data. DH developed the mathematical model and designed the MCMC algorithm. DH and VM generated the code and analyzed the data. DH and VM drafted the first version of the manuscript with input from all authors. All authors contributed to the critical revision of the manuscript for important intellectual content. All authors have seen and approved the final version and agreed to its publication. DH and VM had full access to all the data in the study and take responsibility for the accuracy of the mathematical analysis.

Funding
This work was not supported by any funding.

Availability of data and materials
The Faux Dixon High dataset is avlaible in ERGM R package (Hunter et al. 2008). The SCAL dataset can be requested from the University of South Carolina Prevention Research Center: http://prevention.sph.sc.edu/Resources/SCOTM.html by filling out an interest form. The R codes used during the current study are available in the GitHub repository https://github.com/dan-han-mathematics/Bergm-lasso.git.

Declarations

Ethics approval and consent to participate
Not Applicable.

Competing interests
The authors declare no competing fnancial or non-fnancial interests.

Received: 26 November 2023 Accepted: 6 April 2024
Published online: 23 April 2024

References

Alhamzawi R, Ali HTM (2018) The Bayesian adaptive lasso regression. Math Biosci 303:75–82. https://doi.org/10.1016/j.mbs.2018.06.004
Andrews DF, Mallows CL (1974) Scale mixtures of normal distributions. J Roy Stat Soc: Ser B (Methodol) 36(1):99–102
Baggio S, Luisier V, Vladescu C (2017) Relationships between social networks and mental health. Swiss J Psychol 76(1):5–11. https://doi.org/10.1024/1421-0185/a000186
Becker KR, Stojek MM, Clifton A, Miller JD (2018) Disordered eating in college sorority women: a social network analysis of a subset of members from a single sorority chapter. Appetite 128:180–187. https://doi.org/10.1016/j.appet.2018.06.013
Caimo A, Bouranis L, Krause R, Friel N (2022) Statistical network analysis with bergm. J Stat Softw 104(1):1–23. https://doi.org/10.18637/jss.v104.i01
Caimo A, Friel N (2011) Bayesian inference for exponential random graph models. Soc Netw 33(1):41–55. https://doi.org/10.1016/j.socnet.2010.09.004
Caimo A, Friel N (2013) Bayesian model selection for exponential random graph models. Soc Netw 35(1):11–24. https://doi.org/10.1016/j.socnet.2012.10.003
Caimo A, Friel N (2014) Bergm: Bayesian exponential random graphs in R. J Stat Softw 61:1–25. https://doi.org/10.18637/jss.v061.i02
Caimo A, Mira A (2015) Efficient computational strategies for doubly intractable problems with applications to Bayesian social networks. Stat Comput 25(1):113–125
Caimo A, Pallotti F, Lomi A (2017) Bayesian exponential random graph modelling of interhospital patient referral networks. Stat Med 36(18):2902–2920. https://doi.org/10.1002/sim.7301
Chatterjee S, Diaconis P (2013) Estimating and understanding exponential random graph models. Ann Stat 41(5):2428–2461. https://doi.org/10.1214/13-AOS1155
Chhikara RS, Folks L (1988) The Inverse Gaussian Distribution: Theory, Methodology, and Applications. CRC Press
Erdös P, Rényi A (1959) On random graphs I. Publicationes Mathematicae Debrecen 6:290
Fan J, Feng Y, Wu Y (2009) Network exploration via the adaptive LASSO and SCAD penalties. Ann Appl Stat 3(2):521–541. https://doi.org/10.1214/08-AOAS215
Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96(456):1348–1360
Frank O, Strauss D (1986) Markov Graphs. J Am Stat Assoc 81(395):832–842
Friedman J, Tibshirani R, Hastie T (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22. https://doi.org/10.18637/jss.v033.i01
Friel N, Pettitt A, Reeves R, Wit E (2009) 06. Bayesian Inference in Hidden Markov Random Fields for Binary Data Defined on Large Lattices. J Comput Graph Stat 18:243–261. https://doi.org/10.1198/jcgs.2009.06148
Geyer CJ (1991) Markov chain monte carlo maximum likelihood
Goodreau SM, Handcock MS, Hunter DR, Butts CT, Morris M (2008) A statnet Tutorial. J Stat Softw 24(9):1–26. https://doi.org/10.18637/jss.v024.i09
Handcock MS (2003) Assessing degeneracy in statistical models of social networks. Technical report, Working paper
Holland PW, Leinhardt S (1981) An exponential family of probability distributions for directed graphs. J Am Stat Assoc 76(373):33–50
Hunter D, Handcock M, Butts C, Goodreau S, Morris M (2008) Ergm: a package to fit, simulate and diagnose exponential-family models for networks. J Stat Softw 24(3):1–29. https://doi.org/10.18637/jss.v024.i03
Leng C, Tran MN, Nott D (2014) Bayesian adaptive lasso. Ann Inst Stat Math 66(2):221–244
Levine RA, Casella G (2001) Implementations of the Monte Carlo em algorithm. J Comput Graph Stat 10(3):422–439
Lusher D, Koskinen J, Robins G (2013) Exponential random graph models for social networks: theory, methods, and applications. Cambridge University Press
Meinshausen N, Bühlmann P (2006) High-dimensional graphs and variable selection with the lasso. Ann Stat 34(3):1436–1462
Morris M, Handcock MS, Hunter DR (2008) Specification of exponential-family random graph models: terms and computational aspects. J Stat Softw 24(4):1–24. https://doi.org/10.18637/jss.v024.i04
Munkres JR, Davis L (2018) Topology. Pearson Prentice Hall
Murray I, Ghahramani Z, MacKay D (2012) MCMC for doubly-intractable distributions
Park T, Casella G (2008) The Bayesian Lasso. J Am Stat Assoc 103(482):681–686. https://doi.org/10.1198/016214508000000337
R Core Team (2021) R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing
Resnick M, Bearman P, Blum R, Bauman K, Harris K, Jones J, Tabor J, Beuhring T, Sieving R, Shew M, Ireland M, Bearinger L, Udry J (1997) Protecting adolescents from harm Findings from the National Longitudinal Study on Adolescent Health. JAMA 278(10):823–32
Shojaie A (2013) Link prediction in biological networks using multi-mode exponential random graph models. In: 11th Workshop on Mining and Learning with Graphs, pp 987–991. Citeseer

Shojaie A, Basu S, Michailidis G (2012) Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. Stat Biosci 4(1):66–83
Shojaie A, Michailidis G (2010) Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. Biometrika 97(3):519–538
Solo V, Poline JB, Lindquist MA, Simpson SL, Bowman FD, Chung MK, Cassidy B (2018) Connectivity in fMRI: Blind spots and Breakthroughs. IEEE Trans Med Imaging 37(7):1537–1550. https://doi.org/10.1109/tmi.2018.2831261
Stivala A, Lomi A (2021) Testing biological network motif significance with exponential random graph models. Appl Netw Sci. https://doi.org/10.1007/s41109-021-00434-y
Strauss D, Ikeda M (1990) Pseudolikelihood estimation for social networks. J Am Stat Assoc 85(409):204–212
Tay JK, Narasimhan B, Hastie T (2023) Elastic net regularization paths for all generalized linear models. J Stat Softw 106:1
The US Burden of Disease Collaborators (2018) The State of US Health, 1990–2016: burden of diseases, injuries, and risk factors among US states. JAMA 319(14):1444–1472. https://doi.org/10.1001/jama.2018.0158
Tibshirani R (1996) Regression shrinkage and selection via the Lasso. J Roy Stat Soc: Ser B (Methodol) 58(1):267–288
van Duijn MA, Gile KJ, Handcock MS (2009) A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. Social Networks 31(1):52–62. https://doi.org/10.1016/j.socnet.2008.10.003
Wang H, Leng C (2008) A note on adaptive group lasso. Comput Stat Data Anal 52(12):5277–5286
Williams NL, Hristov D (2018) An examination of DMO network identity using Exponential Random Graph Models. Tour Manage 68:177–186. https://doi.org/10.1016/j.tourman.2018.03.014
Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R Stat Soc Series B (Statistical Methodology) 68(1):49–67
Zou H (2006) The Adaptive Lasso and its Oracle Properties. J Am Stat Assoc 101(476):1418–1429

Publisher's Note