

GLMLE: graph-limit enabled fast computation for fitting exponential random graph models to large social networks

Ran He · Tian Zheng

Received: 29 April 2014/Revised: 7 February 2015/Accepted: 17 February 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Large network, as a form of big data, has received increasing amount of attention in data science, especially for large social network, which is reaching the size of hundreds of millions, with daily interactions on the scale of billions. Thus analyzing and modeling these data to understand the connectivities and dynamics of large networks is important in a wide range of scientific fields. Among popular models, exponential random graph models (ERGMs) have been developed to study these complex networks by directly modeling network structures and features. ERGMs, however, are hard to scale to large networks because maximum likelihood estimation of parameters in these models can be very difficult, due to the unknown normalizing constant. Alternative strategies based on Markov chain Monte Carlo (MCMC) draw samples to approximate the likelihood, which is then maximized to obtain the maximum likelihood estimators (MLE). These strategies have poor convergence due to model degeneracy issues and cannot be used on large networks. Chatterjee et al. (Ann Stat 41:2428–2461, 2013) propose a new theoretical framework for estimating the parameters of ERGMs by approximating the normalizing constant using the emerging tools in graph theory—graph limits. In this paper, we construct a complete computational procedure built upon their results with practical innovations which is fast and is able to scale to large networks. More specifically, we evaluate the likelihood via simple function approximation of the corresponding

ERGM's graph limit and iteratively maximize the likelihood to obtain the MLE. We also discuss the methods of conducting likelihood ratio test for ERGMs as well as related issues. Through simulation studies and real data analysis of two large social networks, we show that our new method outperforms the MCMC-based method, especially when the network size is large (more than 100 nodes). One limitation of our approach, inherited from the limitation of the result of Chatterjee et al. (Ann Stat 41:2428–2461, 2013), is that it works only for sequences of graphs with a positive limiting density, i.e., dense graphs.

Keywords Exponential random graph models · Graph limits · Maximum likelihood estimator · Likelihood ratio test

1 Introduction

There has been growing interest in applying exponential random graph models (also known as p^* models) to social network analysis (see Frank and Strauss 1986; Robins et al. 2007; Handcock and Gile 2010). The class of exponential random graph models, to which we refer as ERGMs for short, includes many popular random graph models such as the dyadic independence models and the Markov random graphs, which makes ERGMs one of the most widely used and flexible models for complex networks.

Despite its popularity, parameter estimation of ERGMs for large networks remains a challenging problem. This is due to the fact that the normalizing constant in the likelihood function depends on the parameters of interest and is a summation over all possible graphs of n nodes. When n is large, the normalizing constant contains an astronomical

R. He · T. Zheng (✉)
Department of Statistics, Columbia University,
New York, NY, USA
e-mail: tzheng@stat.columbia.edu

R. He
e-mail: ran@stat.columbia.edu

number of terms, making evaluating the likelihood—let alone maximizing it—computationally infeasible.

An early solution proposed by Strauss and Ikeda (1990) is a pseudolikelihood method that estimates parameters by maximizing a pseudo-likelihood function, which ignores the dependency among edges. It is computationally expedient, but has been shown by van Duijn et al. (2009) to produce unreliable estimates with sizable bias when dependence in network is strong. Other approaches have been focused on using Monte Carlo schemes to obtain maximum likelihood estimator (Geyer and Thompson 1992). In particular, Markov chain Monte Carlo (MCMC) can be used to draw samples of random graphs from a distribution at each iteration to approximate the likelihood function so that the likelihood can be subsequently maximized. However, this method is computationally expensive since it usually requires a very large number of draws and iterations, especially when the initial values of parameters are far from the unknown true maximum likelihood estimates (Handcock 2003). Thus it is not able to scale to large networks, because sampling a single network of size hundreds of millions is infeasible, let alone drawing a large number of samples in each iteration. One variation of Markov chain Monte Carlo maximum likelihood estimator (MCMCMLE) is to use the Robbins-Monro stochastic approximation algorithm proposed by Snijders (2002), which utilizes iterated simple direct updates of parameter estimates that improve the likelihood function locally. Again, all the above issues with MCMC-based method remain, especially large network issue described above and poor convergence issue due to degeneracy issue of ERGMs.

Recent developments of graph limits due to Lovász and Szegedy (2006) and Borgs et al. (2008) and their coauthors add new depth to our understanding of random graphs, especially very large graphs. In particular, graph limits theory shows that a sequence of random graphs $\{G_n\}$ converges to a limit if the homomorphism density (informally, the proportion) of edges, triangles and other small subgraphs in $\{G_n\}$ converges. Furthermore, the limit object is a measurable function $w : [0, 1]^2 \rightarrow [0, 1]$, which can be viewed as an infinite weighted graph on the points of the unit interval. We provide an overview in Sect. 2.3. Chatterjee et al. (2013) have built upon this emerging tool a new theoretical framework to estimate the parameters in exponential random graph models. First, they prove in the language of graph limits that almost all ERGM graphs converge to a graph limit realization where a certain function is maximized. Then they give an approximation for the normalizing constant based on the maximum of this function. Their work is crucial in our development of a computational algorithm

to fit ERGMs to large networks and is reviewed in Sect. 2.3.

In this paper, we propose a new practical approach of estimating exponential random graph models via graph limits tools, which is an extension based on the work of Chatterjee et al. (2013). Section 2 reviews existing methods for estimating ERGMs, graph limits and theoretical results of Chatterjee et al. (2013). Section 3 introduces our algorithm augmented with simple function approximation of graph limits to improve the generalization of their work. Section 4 demonstrates the advantages of our algorithm with comparison to MCMC-based method via both simulations and real data examples. Section 4 also gives a discussion about the degeneracy issue of ERGMs. Further, we investigate likelihood ratio tests for ERGMs under different hypotheses. We conclude with a discussion in Sect. 5.

2 Background

A social network can be represented by a graph, in which nodes typically represent individuals and ties (or edges) represent a specified relationship of interest between individuals, such as friendship. A graph, denoted by $G = (V, E)$, comprises a set of nodes, V , together with a set of edges, E . Let \mathcal{G}_n be the space of all simple graphs G on n nodes, where simple graphs are undirected graphs with no loops or multiple edges. Let U_1, U_2, \dots, U_k denote real-valued functions on \mathcal{G}_n , i.e., each $U_i(G)$ is a *feature* (graph statistic) of a graph G in the space \mathcal{G}_n . Typical features are geometrically natural quantities such as the count of edges or the count of triangles in the graph.

Given a set of k features $\mathbf{U}(G) = (U_1(G), \dots, U_k(G))$ and a vector of real-valued parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$, the exponential random graph model (ERGM) assumes that G follows a probability distribution in the following exponential form:

$$p_{\boldsymbol{\theta}}(G) \stackrel{\text{def}}{=} \exp \left\{ \sum_{i=1}^k \theta_i U_i(G) - \psi(\boldsymbol{\theta}) \right\} \tag{1}$$

$$= \exp \{ \boldsymbol{\theta}' \mathbf{U}(G) - \psi(\boldsymbol{\theta}) \},$$

where $\psi(\boldsymbol{\theta})$ is a normalizing constant such that the total mass of $p_{\boldsymbol{\theta}}(G)$ is 1. Explicitly,

$$\psi(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \log \sum_{G \in \mathcal{G}_n} \exp(\boldsymbol{\theta}' \mathbf{U}(G)).$$

Given a simple graph G as data, our interest is to find the MLE of $\boldsymbol{\theta}$, which maximizes $p_{\boldsymbol{\theta}}(G)$. However, since analytic form of $\psi(\boldsymbol{\theta})$ is unknown due to the combinatorial complexity of summing over all possible $2^{\binom{n}{2}}$ graphs in \mathcal{G}_n , the MLE cannot generally be found analytically. Therefore,

evaluation of the normalizing constant $\psi(\theta)$ remains a major obstacle in ERGM estimation and we seek a reliable and easy-to-implement solution. Many different approaches have been proposed. We introduce, in the following, the most widely used and representative methods.

2.1 Pseudolikelihood approach

The maximum pseudo-likelihood estimator (MPLE) of Strauss and Ikeda (1990), motivated by methods from spatial statistics (Besag 1975), is a fast and convenient method for parameter estimation.

Consider the conditional formulation of the model (1):

$$\text{logit} \left[p_{\theta}(G_{ij} = g_{ij} | G_{ij}^c) \right] = \theta' \delta(G_{ij}^c),$$

where $G_{ij} = I_{(i,j) \in E(G)}$ (the (i,j) th entry of adjacency table of G) and $\delta(G_{ij}^c) = U(G) - U(G_{ij}^c)$, the change in $U(G)$ when the (i,j) edge toggled in G while the rest of the network remains G_{ij}^c . The pseudolikelihood for the model (1) is just the product of conditional probability of all pairs of (i,j) by ignoring the dependency among edges and thus gives a likelihood with an easy analytical form:

$$\prod_{i,j} p_{\theta}(G_{ij} = g_{ij} | G_{ij}^c) = \prod_{i,j} \text{logit}^{-1} \left(\theta' \delta(G_{ij}^c) \right). \tag{2}$$

The form of pseudolikelihood (2) is identical to the likelihood of a logistic regression model, where the true edge state, g_{ij} , is treated as an independent observation with the

et al. (2009). Though its properties are poorly understood for analyzing social networks, the MPLE has been commonly used as a rough approximate of the MLE, especially providing initial values for other iterative methods such as MCMC-based approaches and stochastic approximation methods. For example, MPLE is the default method to obtain the initial values for MCMC-based algorithm implemented in R function `ergm` from the `ergm` package (Hunter et al. 2008).

2.2 Monte Carlo-based approach

Geyer and Thompson Geyer and Thompson (1992) have proposed a Monte Carlo scheme to approximate the likelihood, using m samples $\{G_i^t\} \sim p_{\theta^{(t)}}$ for a known $\theta^{(t)}$ to approximate the normalizing constant $\psi(\theta)$. More specifically,

$$\hat{\psi}(\theta; \{G_i^t\}) = C_n + \log \left\{ \frac{1}{m} \sum_{i=1}^m \exp(\theta' U(G_i^t)) \right\},$$

where C_n is a constant depending only on the number of nodes n . Plugging in the above approximation $\hat{\psi}$ for ψ in (1) yields the approximated likelihood function,

$$\hat{p}_{\theta}(G; \{G_i^t\}) = \exp \left\{ \theta' U(G) - \hat{\psi}(\theta; \{G_i^t\}) \right\}.$$

The approximated likelihood \hat{p}_{θ} can then be iteratively maximized to obtain the Monte Carlo maximum likelihood estimator (MCMLE) of θ .

Algorithm MCMLE or MCMCMLE

1. Give a starting point $\theta^{(0)}$, usually taken to be the MPLE.
 2. For each t ,
 - (a) sample m graphs $\{G_i^t\}$ from $p_{\theta^{(t)}}$;
 - (b) set $\theta^{(t+1)} = \text{argmax}_{\theta} \hat{p}_{\theta}(G; \{G_i^t\})$.
 3. Stop once $\|\theta^{(t+1)} - \theta^{(t)}\| < \epsilon$ for some fixed ϵ . And the corresponding $\theta^{(t+1)}$ is the MCMLE.
-

corresponding row of the design matrix given by $\delta(G_{ij}^c)$. With standard logistic regression algorithms employed, the MLE for this logistic regression model is exactly the same as the MPLE for the corresponding ERGM, which is easy for implementation.

Despite their easy implementation, algorithms to compute the MLE for logistic regression models can become unreliable and lead to non-convergence if the models are nearly degenerate. In addition, the MPLE approach ignores the dependence between edges, which can be strong in many ERGMs. Consequently, MPLEs usually suffer from substantial bias. The standard error estimates derived from the MPLE method are problematic, which is shown in a simulation study by van Duijn

A host of techniques for sampling graphs $\{G_i\}$ from ERGM with parameters θ have been proposed. Liu (2008) uses importance sampling method. Handcock et al. (2008) recommend an MCMC-based approach that uses a local Markov chain through adding or deleting edges via the Metropolis algorithm, which has been most commonly used. The corresponding estimators are usually referred to as MCMCMLE.

Another MCMC-related approach is Snijders' suggestion (2002) on using the Robbins-Monroe stochastic approximation algorithm for computing moment estimates, which solves

$$E\{Z_{\theta}\} = 0, \tag{3}$$

where $Z_\theta = U(G) - U(G^{\text{obs}})$ and G^{obs} is the observed graph. The iteration step in the Robbins–Monroe procedure for solving (3) with step-size a_t , is

$$\hat{\theta}_{t+1} = \hat{\theta}_t + a_t Z_t,$$

where Z_t is a random variable from the distribution of Z_θ specified by $\theta = \hat{\theta}_t$ that noisily estimates U_θ . The step sizes a_t are a sequence of positive numbers converging to 0, with classical choice of $a_t = 1/t$. For more details, please see Snijders (2002) and also Robbins and Monro (1951). Note that in exponential families, moment estimates are also maximum likelihood estimates. Thus, this procedure provides a promising tool to approximate MLE for ERGMs, while also requiring drawing samples from p_θ for a given θ based on MCMC.

These Monte Carlo-based procedures are, in theory, guaranteed to converge to the MLE if it exists. They have been popular among practitioners. Handcock et al. (2008) have implemented these methods in statnet suite of packages in the R statistical language. The particular package for ERGMs is called ergm (Hunter et al. 2008). Despite their popular use, one common difficulty that these schemes share is the choice of initial values. If the starting point is close to the true MLE, these algorithms may perform well at finding the MLE. This is certainly not the case in practice because we usually lack the knowledge of the approximate location of the MLE. For the case where the starting point is far from the MLE, the convergence of these approaches is rather poor. Bhamidi et al. (2008) give a theoretical explanation: if the parameters are non-negative, then for large n , either the p_θ model is essentially the same as an Erdos-Renyi model or the Markov chain takes exponential time to mix. This limits the application of MCMC-based approach to large networks. In fact, since sampling-based methods require a large number of samples in each iteration, they are very time and memory consuming and become incomputable when applied to large networks. For example, ergm fails to run for a network of size 10,000, which is relatively small compared to real-world network data.

2.3 Graph limits based approach

One of recent exciting developments in graph theory is the theory of graph limits, due to Lovász, Szegedy, Borgs and their coauthors. Below, we first introduce their definition of the limit of a sequence of dense graphs.

Definition 1 (Lovász and Szegedy 2006) For two simple graphs H and G , let $\text{hom}(H, G)$ denote the number of homomorphisms (adjacency-preserving maps) from $V(H)$ to $V(G)$, where $V(H)$ and $V(G)$ are vertex sets. This number is normalized to get the homomorphism density

$$t(H, G) \stackrel{\text{def}}{=} \frac{\text{hom}(H, G)}{|V(G)|^{|V(H)|}}. \tag{4}$$

Thus $t(H, G)$ is the probability that a random map of $V(H) \rightarrow V(G)$ is a homomorphism. It is defined that a sequence of simple graphs $\{G_n\}$ is *convergent*, if the sequence $t(H, G_n)$ of (4) has a limit for every simple graph H , in the sense that G_n become more and more similar as n goes to infinity.

The main result of Lovász and Szegedy (2006) is that convergent graph sequences have a limit object, which can be represented as a measurable function. Let \mathcal{W} denote the space of all measurable functions $w : [0, 1]^2 \rightarrow [0, 1]$ that satisfy $w(x, y) = w(y, x)$ for all $x, y \in [0, 1]$. For every simple subgraph H and $w \in \mathcal{W}$, we define

$$d(H, w) \stackrel{\text{def}}{=} \int_{[0,1]^{|V(H)|}} \prod_{ij \in E(H)} w(x_i, x_j) dx \tag{5}$$

as the homomorphism density of H in w . The intuition behind this definition is that the interval $[0, 1]$ represents a “continuum” of vertices, serving as locations, and $w(x, y)$ denotes the probability of having an edge between “vertices” x and y . A sequence of graphs $\{G_n\}$ is said to converge to a limit object w if for every finite simple graph H ,

$$\lim_{n \rightarrow \infty} t(H, G_n) = d(H, w).$$

The above result works for a sequence of graphs $\{G_n\}$ whose corresponding graph limit object w exists when the number of nodes n goes to infinity. In other words, w is positive and fixed in the limit, rather than going to zero. This results in a sequence of dense graphs, which is the main limitation of the framework of Lovász and Szegedy (2006). But parallel theories for sparse graphs are beginning to emerge in Bollobás and Riordan (2011). On the other hand, every finite simple graph G can also be represented as a graph limit w^G in a natural way. Split the interval $[0, 1]$ into n equal intervals I_1, \dots, I_n , where $n = |V(G)|$. For $x \in I_i, y \in I_j$, define

$$w^G(x, y) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } ij \in E(G), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The papers on graph limit theory define not only the limit of a sequence of graphs, but also and more importantly, the space of limit objects, \mathcal{W} . And (6) maps any given simple graph into this space. This makes sense because the constant sequence of any finite simple graph $G, \{G, G, \dots\}$ converges to the graph limit w^G . Therefore, motivated by this representation, Chatterjee et al. (2013) propose to map any ERGM graph into a quotient space with an induced probability measure.

Let $\widetilde{\mathcal{W}}$ be a quotient space in which every simple graph G has an equivalence class \widetilde{G} under measure-preserving bijections. Specifically, denote Σ as the space of measure preserving bijections $\sigma : [0, 1] \rightarrow [0, 1]$. Say that $w_1, w_2 \in \mathcal{W}$ are equivalent if $w_1(x, y) = w_2(\sigma x, \sigma y)$ for some $\sigma \in \Sigma$. To any finite graph G , we associate its graph limit representation w^G as in (5) and its equivalent class \widetilde{w}^G . A (unique) representation of this class, \widetilde{G}^{rep} , can be obtained by relabeling nodes of G according to (strictly) ascending orders of degrees (Bickel and Chen 2009). For notational simplicity, we drop the superscript $(\cdot)^{rep}$ and denote \widetilde{G} as a (unique) representation of \widetilde{w}^G . Define a distance δ_{\square} such that $(\widetilde{\mathcal{W}}, \delta_{\square})$ is a metric space (see Chatterjee et al. 2013; Lovász and Szegedy 2006 for more details on definitions). Intuitively, graph limit theory can be regarded as projecting any graph on a two-dimensional symmetric function space according to the representation (5) and introducing distance on this space in order to define limit, continuous, etc. And exponential random graph model can then be defined on this metric space using “statistics of graphs” on this space.

Let $T : \widetilde{\mathcal{W}} \rightarrow \mathbb{R}$ be a bounded continuous function on the metric space $(\widetilde{\mathcal{W}}, \delta_{\square})$. Then T induces an exponential random graph model of (1) on \mathcal{G}_n and the probability mass function p_n is defined as

$$p_n(G) \stackrel{\text{def}}{=} \exp \left\{ n^2 (T(\widetilde{G}) - \psi_n) \right\}, \tag{7}$$

where \widetilde{G} is the image of G in the quotient space $\widetilde{\mathcal{W}}$ and ψ_n is the normalizing constant. Note that a linear combination of continuous function is still continuous, $T(\widetilde{G})$ can also be written as $T(\widetilde{G}) = \sum_{i=1}^k \theta_i T_i(\widetilde{G})$, where k features $(T_1(\widetilde{G}), \dots, T_k(\widetilde{G}))$ are of interest and the parameters are $(\theta_1, \dots, \theta_k)$.

Typical choice of $T(\cdot)$ is the homomorphism density $d(H, \cdot)$ as in (5), which is continuous with respect to δ_{\square} distance on $\widetilde{\mathcal{W}}$, where H can be any finite simple graph motif. This choice coincides with the commonly used ERGM terms such as number of edges or triangles and the corresponding definition of (7) is equivalent to (1). For example, consider an ERGM with number of edges, two-stars and triangles as features; then

$$\begin{aligned} T(\widetilde{G}) &= \sum_{i=1}^3 \theta_i T_i(\widetilde{G}) = \sum_{i=1}^3 \theta_i d(H_i, \widetilde{G}) \\ &= \frac{2\theta_1(\# \text{ edges in } G)}{n^2} + \frac{6\theta_2(\# \text{ two-stars in } G)}{n^3} \\ &\quad + \frac{6(\theta_3 - 2\theta_2)(\# \text{ triangles in } G)}{n^3}, \end{aligned} \tag{8}$$

where number of two-stars is defined as the number of connected triples of vertices. On the other hand, choice of $T(\cdot)$ is not limited to homomorphism densities. In fact, the main results of Chatterjee et al. (2013), the theoretical basis of our algorithm, work for many other “continuous function” on graph space, such as the degree sequence or the eigenvalues of the adjacency matrix. Therefore, results of Chatterjee et al. (2013), as well as those of this paper, can be applied to more general cases of ERGMs.

Based on the Erdos-Renyi measures defined in Chatterjee and Varadhan (2011), where they prove that these probability measures obey a large deviation principle in the space $\widetilde{\mathcal{W}}$, Chatterjee et al. (2013) give an asymptotic formula for ψ_n of (7):

$$\lim_{n \rightarrow \infty} \psi_n = \sup_{\widetilde{w} \in \widetilde{\mathcal{W}}} (T(\widetilde{w}) - I(\widetilde{w})), \tag{9}$$

where

$$I(\widetilde{w}) = \iint_{[0,1]^2} I(w(x, y)) dx dy$$

is the rate function of the large deviation principle mentioned above with

$$I(u) = \frac{1}{2} u \log u + \frac{1}{2} (1 - u) \log(1 - u).$$

A more important finding is that when n is large, almost all random graphs G_n drawn from ERGMs (7) are close to graphs F with high probability when $T(\widetilde{F}) - I(\widetilde{F})$ reaches maximum. The approximation error between \widetilde{G}_n and \widetilde{F} is given in theorem 3.2 in their paper. For any $\eta > 0$ there exist $C, \gamma > 0$ such that for any n ,

$$P(\delta_{\square}(\widetilde{G}_n, \widetilde{F}) > \eta) \leq C e^{-n^{\gamma}}.$$

In other words, \widetilde{F} is the graph limit of ERGMs (7) and can be obtained by maximizing $T(\widetilde{w}) - I(\widetilde{w})$.

Based on these findings, Chatterjee et al. (2013) introduce a method to approximate MLE of (7), by evaluating $\psi(\theta)$ on a fine grid in θ space and then carrying out the maximization by classical methods such as a grid search. The corresponding θ is an estimate of MLE. However, this method only works for some specific ERGM when its graph limit is known but not for arbitrary ERGM when the graph limit is unknown. In addition, evaluating $\psi(\theta)$ on a fine grid may be impossible since parameter space of θ is infinite. Thus, it is hard to determine the range of grid unless we know approximately where the true MLE is located, which is not the case in practice. These issues limit the application of their work to more general cases.

3 Methods

In this section, we describe a new computational procedure of finding MLE of ERGMs for large networks via graph limits tools. More specifically, motivated by Chatterjee et al. (2013), we propose an improved algorithm to approximate the normalizing constant $\psi(\theta)$ via (9). In addition, providing that \tilde{W} is the space of all symmetric function $w : [0, 1]^2 \rightarrow [0, 1]$, we propose to use two-dimensional simple functions to approximate the elements in \tilde{W} .

3.1 Approximating the graph limit of an ERGM using two-dimensional simple functions

By definition, a two-dimensional simple function is a finite linear combination of indicator functions of measurable regions. In our case, for any m , split $[0, 1]^2$ into m^2 lattices with equal area,

$$A_{ij} \stackrel{\text{def}}{=} \left\{ (x, y) : x \in \left[\frac{i-1}{m}, \frac{i}{m} \right) \text{ and } y \in \left[\frac{j-1}{m}, \frac{j}{m} \right) \right\}, \tag{10}$$

where $i, j = 1, \dots, m$. And let $\{c_{ij}\}$ be a sequence of real numbers between 0 and 1. Define a two-dimensional simple function $f_m(x, y) : [0, 1]^2 \rightarrow [0, 1]$ of the form

$$f_m(x, y) \stackrel{\text{def}}{=} \sum_{i,j=1}^m c_{ij} \mathbf{1}_{A_{ij}}(x, y), \tag{11}$$

where $c_{ij} = c_{ji}$ for any pair of ij . The above simple functions have the following properties:

1. $f_m(x, y) = f_m(y, x)$.
2. The sum, difference and product of two simple functions are again simple functions.
3. The integral of a simple function is very easy to compute, i.e.,

$$\int_{[0,1]^2} f_m(x, y) dx dy = \frac{1}{m^2} \sum_{i,j=1}^m c_{ij}. \tag{12}$$

4. For any element $f \in \tilde{W}$, there is a sequence of simple functions f_m such that

$$f(x, y) = \lim_{m \rightarrow \infty} f_m(x, y), \quad \forall (x, y) \in [0, 1]^2.$$

Therefore, we can use a simple function defined in (11), with an appropriate choice of m , to approximate any function in \tilde{W} , i.e., the graph limit object.

Let G_n be a random graph on n nodes drawn from the ERGM distribution (7). Recall that, as proved in Chatterjee et al. (2013), the graph limit of G_n can be obtained via maximizing $T(\tilde{w}) - I(\tilde{w})$, while the graph constructed from this graph limit captures the properties of G_n . When the graph limit is a constant (the corresponding ERGM is a simple Erdos-Renyi random graph model), it is trivial to solve this optimization problem. And Chatterjee et al. (2013) provide some cases of ERGMs with constant graph limits. When the graph limit is more complicated and unknown, however, it is hard to solve this optimization problem since \tilde{W} contains all symmetric two-dimensional functions $w : [0, 1]^2 \rightarrow [0, 1]$. We address this problem using simple functions approximation described above.

Algorithm Simple function approximation

1. Give a bounded continuous function

$$T(\tilde{G}) = \sum_{i=1}^k \theta_i T_i(\tilde{G})$$

that induces a probability mass function of ERGM, which has form (7).

2. Define a two-dimensional simple function

$$w_m = \sum_{i,j=1}^m c_{ij} \mathbf{1}_{A_{ij}}(x, y),$$

where A_{ij} is defined as above.

3. Solve the optimization problem

$$\sup_{c_{ij}} (T(w_m) - I(w_m))$$

and the corresponding arguments are \hat{c}_{ij} .

4. An approximation of the graph limit of the ERGM induced by T is

$$\hat{w}_m = \sum_{i,j=1}^m \hat{c}_{ij} \mathbf{1}_{A_{ij}}(x, y).$$

The advantage of this approximation is that it simplifies the search for $\sup_{\tilde{w}}(T(\tilde{w}) - I(\tilde{w}))$, making it computationally tractable. For example, consider the ERGM with homomorphism densities of edges, two-stars and triangles as T_i , which is (8). In the third step of the above algorithm, $T(w_m) - I(w_m)$ can be written as

$$T(w_m) - I(w_m) = \frac{\theta_1}{m^2} \sum_{ij} c_{ij} + \frac{\theta_2}{m^3} \sum_{ijk} c_{ij}c_{jk} + \frac{\theta_3}{m^3} \sum_{ijk} c_{ij}c_{jk}c_{ik} - \frac{1}{2m^2} \sum_{ij} [c_{ij} \log c_{ij} + (1 - c_{ij}) \log(1 - c_{ij})]. \tag{13}$$

Then commonly used optimization methods, such as *conjugate gradient* or *simulated annealing*, can be employed to solve this optimization problem with $\frac{1}{2}m(m + 1)$ parameters.

3.2 Estimating parameters of ERGMs via graph limits

Based on the above simple function approximation algorithm to maximize $T(\tilde{w}) - I(\tilde{w})$, we obtain the corresponding graph limit as well as an approximation of the normalizing constant $\psi_n(\theta)$ in (7) via the optimization procedure (9). Suppose the estimated simple function based on a known T and $\theta^{(t)}$ is $\hat{w}_m^{(t)}$, then the approximated normalizing constant is

$$\hat{\psi}_n(\theta; \hat{w}_m^{(t)}) \stackrel{\text{def}}{=} T_\theta(\hat{w}_m^{(t)}) - I(\hat{w}_m^{(t)}).$$

Plugging in the approximated $\hat{\psi}_n$ leads to the approximated log-likelihood function of θ :

$$\log \hat{p}_n(\theta; G, \hat{w}_m^{(t)}) \stackrel{\text{def}}{=} n^2 [T_\theta(\tilde{G}) - \hat{\psi}_n(\theta; \hat{w}_m^{(t)})]. \tag{14}$$

Maximizing $\log \hat{p}_n$ provides the *graph limit maximum likelihood estimator* (GLMLE) of θ . It should be noted that the bias of GLMLE greatly depends on the accuracy of the approximation of the log-likelihood, $\log \hat{p}_n$, which is based on the approximation of normalizing constant using simple functions. Thus, we propose an iterative procedure as follows:

Algorithm GLMLE

1. Give an initial value of $\theta, \theta^{(0)}$.
 2. For each t ,
 - (a) Use simple function approximation to estimate $\tilde{w}^{(t)}$ by maximizing $T_{\theta^{(t)}}(\tilde{w}) - I(\tilde{w})$. The corresponding simple function is

$$\hat{w}_m^{(t)} = \sum_{i,j=1}^m \hat{c}_{ij} \mathbf{1}_{A_{ij}}(x, y);$$
 - (b) set $\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \log \hat{p}_n(\theta; G, \hat{w}_m^{(t)})$.
 3. Stop once $\|\theta^{(t+1)} - \theta^{(t)}\|$ converges. And the corresponding $\theta^{(t+1)}$ is the GLMLE.
-

This framework of our algorithm can be interpreted as an iterative refinement approach. The motivation is as follows: if we know the current value of the parameters θ , we can find the best value of the “latent variables” w_m , as in step 2(a); conversely, if we know the value of the “latent variables” w_m , we can find an update of the parameters θ , as in step 2(b). These two steps make our algorithm a maximization–maximization procedure, similar to the motivation of the generalized expectation–maximization algorithm, though the latter requires an expectation step. Note that the above iterative algorithm provides $\theta^{(t+1)}$ in each step such that

$$\log \hat{p}_n(\theta^{(t+1)}; G, \hat{w}_m^{(t+1)}) > \log \hat{p}_n(\theta^{(t)}; G, \hat{w}_m^{(t)}),$$

when n and m are large enough. This guarantees that our algorithm converges to the MLE in theory if it exists, as the likelihood function of ERGM follows an exponential family and is globally concave. The maximum found is also unique.

3.3 Practical remarks

3.3.1 Initial values

To compute an initial value reasonably close to the MLE quickly, we estimate $\theta^{(0)}$ by constraining the graph limit to be a constant function, i.e.,

$$w_0(x, y) = c, \quad \forall (x, y) \in [0, 1]^2,$$

in which case the corresponding graph is an Erdos-Renyi graph. \hat{c} is obtained by solving (9). As the optimization problem (9) is reduced to a one-dimensional problem, we are able to compute this rough estimate as an initial value very fast.

3.3.2 Updating w_m

Note that $T(w_m) - I(w_m)$, the function to be maximized to obtain \hat{w}_m is a nonlinear function and has a simple expression as in, for example, (13). Many nonlinear optimization techniques, ranging from slower but more accurate strategies such as simulated annealing to faster greedy strategies such as nonlinear conjugate gradient method, can be used.

Since this optimization is carried out in each iteration, a faster method may be preferred for better computational efficiency in certain applications. The initial value of w_m is the simple function representation of w^G , w_m^G , by averaging the values in each $\lfloor \frac{n}{m} \rfloor \times \lfloor \frac{n}{m} \rfloor$ block of w^G .

3.3.3 Updating θ

ERGMs, whose distribution is in exponential families, have log-likelihood

$$\begin{aligned} \log p_n(\theta; G) &= n^2 \left[\sum_{i=1}^k \theta_i T_i(\tilde{G}) - \psi(\theta) \right] \\ &= n^2 \left[\theta' \mathbf{T}(\tilde{G}) - \psi(\theta) \right]. \end{aligned}$$

And a very useful property of exponential family is that, for any θ ,

$$\nabla \psi(\theta) = E_\theta[\mathbf{T}(\tilde{G})].$$

This property implies that we can calculate the first derivative of the log-likelihood function (gradient) using $E_\theta[\mathbf{T}(\tilde{G})]$, rather than the annoying $\nabla \psi(\theta)$, which is intractable. Specifically, the gradient for an ERGM graph G is

$$\nabla \log p_n(\theta; G) = n^2 \left\{ \mathbf{T}(\tilde{G}) - E_\theta[\mathbf{T}(\tilde{G})] \right\}.$$

Thus the problem is converted to determine the expected value of ERGM statistics, $E_\theta[\mathbf{T}(\tilde{G})]$, which is a function of graph limit w . We illustrate this with some commonly used ERGM terms $t(H_i, G)$, homomorphism density of edges, two-stars and triangles, as defined in (4). But (8) indicates that they are equivalent to $U_i(G)$, number of edges, two-stars or triangles in G , under a linear transformation. And

$$\begin{aligned} U_1(G) &= \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^n I\{(i,j) \in E(G)\}, \\ U_2(G) &= \frac{1}{2} \sum_{\substack{i,j,k=1 \\ i \neq j \neq k}}^n I\{(i,j) \in E(G)\} \times I\{(i,k) \in E(G)\}, \\ U_3(G) &= \frac{1}{6} \sum_{\substack{i,j,k=1 \\ i \neq j \neq k}}^n I\{(i,j) \in E(G)\} \times I\{(i,k) \in E(G)\} \\ &\quad \times I\{(j,k) \in E(G)\}. \end{aligned}$$

Note that the graph limit $w(x, y)$ is the probability of having an edge between node i and j given their position $X_i = x$, $X_j = y$. Then

$$\begin{aligned} E[U_1(G)] &= E[E(U_1|x_1, \dots, x_n)] \\ &= \sum_{i=1}^{n-1} \sum_{j>i} E[E(I\{(i,j) \in E(G)\}|x_1, \dots, x_n)] \\ &= \sum_{i=1}^{n-1} \sum_{j>i} \int_{[0,1]^n} w(x_i, x_j) dx_1, \dots, dx_n \\ &= \binom{n}{2} \iint w(x, y) dx dy. \end{aligned}$$

Similarly, we have

$$\begin{aligned} E[U_2(G)] &= 3 \binom{n}{3} \iiint w(x, y) w(x, z) dx dy dz, \\ E[U_3(G)] &= \binom{n}{3} \iiint w(x, y) w(x, z) w(y, z) dx dy dz. \end{aligned}$$

Therefore, using the simple function approximation $\hat{w}_m^{(t)}$ in t th step, $E_\theta[\mathbf{T}(\tilde{G})]$ can be easily approximated, since integrals reduce to summations as shown in (12). This gives us an approximation of the first derivative of the log-likelihood, $\nabla \log \hat{p}_n(\theta; G)$. Then a host of gradient methods can be employed to update θ . These techniques include, for example, gradient descent, coordinate descent, conjugate gradient, etc. One algorithm that worths mentioning is *long-range search* algorithm for exponential families (Okabayashi and Geyer 2012). This algorithm uses only the first derivative of log-likelihood and is theoretically guaranteed to converge to the MLE if it exists and the convergence rate is fast in terms of number of iterations. Therefore, we can employ this method to update θ in our algorithm.

However, when $E_\theta[\mathbf{T}(\tilde{G})]$ is hard to calculate, gradient-based methods will no longer work. One simple example is triangle percent. Suppose T_2, T_3 is the homomorphism densities for two-stars and triangles, respectively. Though $E[T_2]$ and $E[T_3]$ are easy to calculate as shown above, $E[\frac{T_3}{T_2}]$ is hard to compute. In these cases, we can exploit non-gradient-based nonlinear optimization methods, such as Nelder–Mead method 1965, which only uses function values. It is robust but relatively slow, which impacts the computational complexity of step 2b of our algorithm though that of step 2a is not affected and remains $O(m^3)$.

3.3.4 Stopping criteria

In practice, our algorithm stops if it is unable to reduce the objective function \hat{p} by a factor of $\delta(|\hat{p}| + \delta)$, where $\delta = 10^{-8}$. This popular stopping criteria is, in fact, the default stopping rule of R function `optim`.

3.3.5 Computational complexity

Our algorithm does not depend on the network size n except for obtaining w_m^G in the initial step, whose time complexity is $O(n^2)$. This guarantees our algorithm scales well to large networks.

On the other hand, the complexity of our algorithm highly depends on the most complex ERGM term in the model because of the nonlinear function to be maximized in step 2a. Though the complexity is still unknown for some ERGM terms, it is known for many commonly used ERGMs. For example, if we consider an ERGM with number of edges and two-stars as statistics, the time complexity of initial step 1 is $O(1)$. And in each iteration, the computational complexity of step 2a is $O(m^3)$.

4 Results

Here we illustrate our method through simulation studies and real data analyses. In both cases, we compare our algorithm with MCMC-based algorithm, which is the most commonly used method.

4.1 Simulation study

For our simulation study, we consider an ERGM using homomorphism densities $t(H_i, \cdot)$ as sufficient statistics, where H_1 is edge, H_2 is two-star and H_3 is triangle. This model is actually identical to the ERGM using number of edges, two-stars and triangles as statistics under a reparameterization of θ , which is shown in (8). We specify the true value of the parameters θ to be $\theta = (-2, -1, 1)$, which is obtained by rounding parameter estimates of this ERGM fitted to a small Facebook social network data. Using the R function `simulate.ergm` from the `ergm` package (Hunter et al. 2008), we generate ERGM graphs of different sizes

($n = 100, 200, 500, 1000, 2000, 4000$) for this model. In each case, we simulate 100 graphs and apply our algorithm as well as MCMC algorithm (R function `ergm`) to model these data. For simple function approximation, we set $m = 10$.

We measure the performances of these two approaches in terms of bias and standard errors of fitted value $\hat{\theta}$. Our method outperforms MCMC method in almost all cases for all parameters (see Table 1), especially when the size of graph n is large. However, we notice that the bias and standard error of GLMLE increase as n increases, which is in line with those of MCMCMLE except for the $\hat{\theta}_3$, the parameter for triangle term. This may suggest that the R function `simulate.ergm`, which draws samples using MCMC, may fail to generate large ERGM random graph of given parameters, due to the convergence issue of Markov chains. To illustrate this, we use the W -random graph approach in Lovász and Szegedy (2006) to simulate graphs from the graph limit of the same ERGM and repeat the simulation comparison.

W -random graph is a method to generate random graph using a given graph limit w . Given a two-dimensional function (graph limit) $w \in \tilde{W}$ and an integer $n > 0$, we can generate a random graph $G(n, w)$ with n nodes as follows: first generate n independent numbers X_1, \dots, X_n from the uniform distribution $U(0, 1)$; then connect nodes i and j by an edge with probability $w(X_i, X_j)$, independently for every pair. Lovász and Szegedy (2006) prove that the graph sequence $G(n, w)$ is convergent with probability 1 in δ_{\square} measure and its limit is the function w . This means $G(n, w)$ captures the property of any graph G_n with limit w when n is large. Therefore, instead of using `simulate.ergm` to generate problematic large random graphs, we can first obtain the graph limit w_{θ} of the above ERGM with the true value of θ and then simulate the corresponding W -random graphs $G(n, w_{\theta})$ as random draws from the ERGM. All other settings are exactly the same as above. The results are summarized in Table 2.

Table 1 Bias and standard errors of parameter estimates by GLMLE and MCMCMLE for random graphs of various sizes generated by the R function `simulate.ergm`

Size n	GLMLE			MCMCMLE		
	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)	Bias($\hat{\theta}_3$) se($\hat{\theta}_3$)	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)	Bias($\hat{\theta}_3$) se($\hat{\theta}_3$)
100	-0.017 (0.206)	-0.429 (5.055)	0.929 (7.161)	0.042 (0.163)	-0.496 (1.738)	9.800 (7.638)
200	-0.022 (0.100)	0.137 (1.369)	0.075 (1.667)	0.033 (0.188)	-1.757 (3.968)	23.780 (18.074)
500	-0.490 (0.019)	0.285 (0.491)	0.079 (2.433)	-0.481 (0.069)	0.598 (1.725)	-9.748 (43.559)
1000	-0.922 (0.013)	0.045 (0.381)	0.154 (0.330)	-0.917 (0.048)	0.483 (2.660)	-27.233 (102.808)
2000	-1.347 (0.009)	-0.209 (0.347)	0.355 (0.255)	-1.346 (0.029)	0.458 (3.787)	-20.266 (188.530)
4000	-1.741 (0.007)	-0.417 (0.307)	0.547 (0.127)	-1.742 (0.023)	0.588 (6.431)	18.510 (379.371)

Table 2 Bias and standard errors of parameter estimates by GLMLE and MCMCMLE for random graphs of various sizes generated by the W -random graph method

Size n	GLMLE			MCMCMLE		
	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)	Bias($\hat{\theta}_3$) se($\hat{\theta}_3$)	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)	Bias($\hat{\theta}_3$) se($\hat{\theta}_3$)
100	0.110 (0.694)	-2.412 (16.639)	0.182 (10.243)	0.004 (0.150)	0.487 (1.546)	7.164 (8.593)
200	-0.018 (0.045)	0.357 (0.661)	-0.098 (2.275)	-0.015 (0.114)	0.803 (1.125)	-6.063 (17.025)
500	-0.009 (0.012)	0.223 (0.064)	-0.103 (0.127)	-0.031 (0.068)	0.979 (0.661)	-1.681 (8.269)
1000	-0.009 (0.006)	0.225 (0.021)	-0.125 (0.040)	-0.031 (0.051)	0.962 (0.520)	-0.557 (5.283)
2000	-0.007 (0.003)	0.219 (0.021)	-0.110 (0.045)	-0.031 (0.030)	0.982 (0.307)	-1.263 (4.180)
4000	-0.007 (0.002)	0.212 (0.017)	-0.094 (0.029)	-0.035 (0.024)	1.029 (0.240)	-1.452 (2.960)

Table 3 Bias and standard errors of parameter estimates by GLMLE and MCMCMLE for model 2

Size n	GLMLE		MCMCMLE	
	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)	Bias($\hat{\theta}_1$) se($\hat{\theta}_1$)	Bias($\hat{\theta}_2$) se($\hat{\theta}_2$)
50	0.109 (0.246)	0.103 (0.331)	-0.007 (0.089)	0.074 (0.087)
100	0.091 (0.141)	0.071 (0.246)	0.000 (0.046)	0.075 (0.056)
200	0.061 (0.092)	0.037 (0.140)	-0.046 (0.028)	0.112 (0.021)
300	0.026 (0.087)	-0.045 (0.090)	-0.091 (0.017)	0.177 (0.029)
400	-0.028 (0.088)	-0.025 (0.091)	-0.126 (0.013)	0.137 (0.017)
500	0.021 (0.074)	-0.001 (0.081)	-0.149 (0.011)	0.209 (0.010)

Comparing the bias and standard errors of estimates in Table 2 and those in Table 1, we find that both methods generate more sensible estimates from the W -random graphs when n becomes large. This indicates that W -random graph simulating method is more likely to generate random graph from the desired ERGM when n is large, comparing with MCMC-based approach (R function simulate.ergm). However, this simulation procedure utilizes w_θ , which is an approximation of true w corresponding to true θ , and may deliver biased samples if m is too small.

Based on the more reliable results in Table 2, it can be seen that graph limit approach outperforms the MCMC-based approach under almost all settings, especially when n is large. For small graphs such as $n = 100$, the performance of MCMCMLE is comparable to that of GLMLE or better. This is reasonable as GLMLE is built upon limiting behavior of large graphs. Especially, our algorithm is based on (9), an asymptotic formula for the normalizing constant ψ_n , which may work less effectively when n is small.

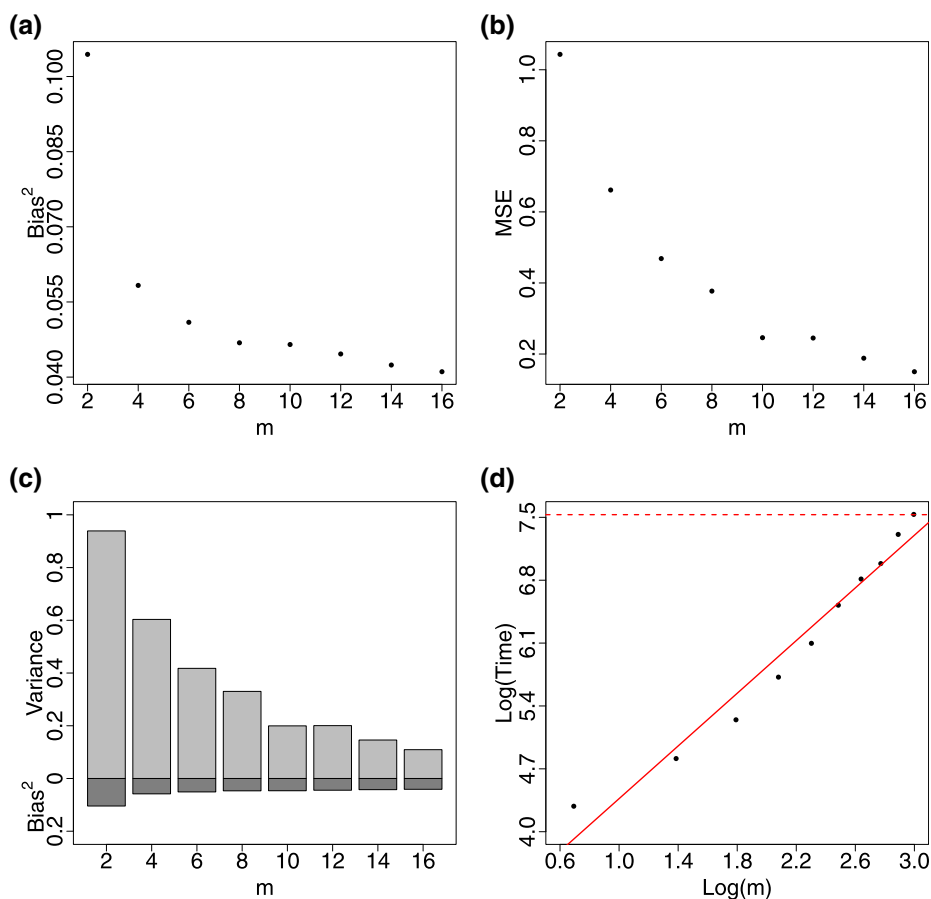
The function $T : \widetilde{\mathcal{W}} \rightarrow \mathbb{R}$ that induces an exponential random graph model of form (7) is not just limited to be the homomorphism densities of edges, two-stars or triangles. Actually, it can be any bounded continuous function, which

greatly generalizes ERGMs. In order to illustrate that our graph-limit-based algorithm works on more general cases of ERGMs, we consider another model that uses homomorphism density of edges and triangle percent as terms. Similar to the previous model, the $T(\widetilde{G})$ in this ERGM can also be expressed as a function of number of edges, two-stars and triangles, that is,

$$\begin{aligned}
 T(\widetilde{G}) &= \theta_1(\text{edges density}) + \theta_2(\text{triangle percent}) \\
 &= \frac{2\theta_1(\#\text{edges in G})}{n^2} \\
 &\quad + \frac{\theta_2(\#\text{triangles in G})}{(\#\text{two-stars in G}) - 2 \times (\#\text{triangles in G})}.
 \end{aligned}
 \tag{15}$$

We specify the true value of the parameters θ to be $\theta = (-1.8, -0.2)$. Using W -random graphs generating algorithm, we simulate ERGM graphs of different sizes ($n = 50, 100, 200, 300, 400, 500$). And all other settings are the same as those for the previous simulation. The results in Table 3 indicate that GLMLE also works on models other than (8) and performs better than MCMC-based algorithm when network size n is large.

Fig. 1 **a** Plot of bias^2 vs. m , where $\text{bias}^2 = \|\hat{\theta} - \theta\|^2$, and the bias^2 for MCMCMLE is 2.3333; **b** plot of MSE vs. m , where $\text{MSE} = \|\hat{\theta} - \theta\|^2$, and the MSE for MCMCMLE is 13.926; **c** bar plot of variance and bias^2 vs. m where $\text{variance} = \text{MSE} - \text{bias}^2$; and **d** log-log plot of running times vs. m and the fitted line with a slope of 1.466, where the red dashed line is for MCMC-based algorithm.



Besides the above investigation into how network size n impacts the performance of our algorithm, it is also very important to examine the choice of m , the parameter used in simple function approximation. This is a crucial parameter for two reasons. First, a too small m may cause corresponding simple function fail to correctly approximate the true graph limit. Second, a too large m may lead to computational infeasibility since the theoretical complexity in each iteration is about $O(m^3)$. Thus, we conduct a simulation study using different m to investigate into the impact of m on GLMLE, under criteria of bias, MSE and running time. We set $n = 4000$ and specify the true value of the parameters $\theta = (-2, -1, 1)$ again. We generate 100 random graphs from this ERGM model and apply our algorithm using different choices of $m = (2, 4, 6, 8, 10, 12, 14, 16)$.

The results are shown in Fig. 1. Clearly, bias, MSE and variance of GLMLE decrease as m increases, and all are much smaller than those of MCMCMLE. Plots (a–c) also reveal that MSE seems to have a faster decreasing rate than bias, indicating variance of GLMLE decreases faster than bias since $\text{MSE} = \text{bias}^2 + \text{variance}$. This phenomenon can be explained by the fact that larger m yields a more accurate

approximation of w and hence more stable, which in return, produces more stable estimator of θ compared with GLMLE using smaller m . Plot (d) illustrates the polynomial increase of computation time as m increases with the order of 1.466 approximately. It does not contradict the theoretical $O(m^3)$ rate, because $O(m^3)$ is the computational complexity in each iteration while $O(m^{1.466})$ is that for the entire algorithm. This also implies that our algorithm converges at a faster rate for larger m , which makes sense since large m provides a more accurate estimate of w and a larger value of likelihood function. The running time of our method increases significantly with choice of large value of m , for example, total computational cost reaches that of MCMC-based method when $m = 20$. On the other side, the improvement of GLMLE in terms of bias or MSE is not that significant for m greater than 6. Thus, the choice of $m = 10$ in the above simulation studies as well as the following real data analyses seems reasonable.

4.2 Real data analysis

We apply our method to two real large social networks from Slashdot (Leskovec et al. 2009). Slashdot is a technology-

related news website that has a large specific user community. In 2002, it introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. The two networks used below are this ‘‘Slashdot Zoo social networks’’ where links represent friend/foe between users of Slashdot. The first social network *Slashdot0811* was obtained in November 2008, while *Slashdot0902* was obtained in February 2009. The links are directional in the original data but we converted the data to undirected graphs for our examples. Statistics of these two networks are as follows:

	Nodes	Edges	Two-stars	Triangles	Transitivity ratio
<i>Slashdot0811</i>	77,360	469,180	68,516,301	551,724	0.02416
<i>Slashdot0902</i>	82,168	504,230	74,983,589	602,592	0.02411

We first fit the ERGM in (8) to these two networks. Although MCMC-based approach works in theory for large networks, it fails in practice, primarily because these two networks are too large to be coerced to objects to which the ergm function can be applied. Our GLMLE algorithm works efficiently no matter how large the network is, as the algorithm takes sufficient statistics as inputs and employs simple function approximation with pre-fixed $m = 10$. The estimated GLMLE are

1. *Slashdot0811*: $(-4.5109, -1.5863, 1.6871)$,
2. *Slashdot0902*: $(-4.6502, -1.8122, 1.9430)$.

The running time for obtaining w^G of *Slashdot0811* on a 2.66 GHz processor is 392 s, while that for *Slashdot0902* is 436 s. And the running time for estimating the parameters of ERGM on *Slashdot0811* is 153 s, while that for *Slashdot0902* is 124 s.

To interpret the fitted ERGM parameters, consider adding one more edge to the graph such that a two-star is converted to a triangle. Then, the fitted values of θ indicate

that the log-likelihood is decreased by 9.0216 for *Slashdot0811*, while by 9.3004 for *Slashdot0902*. This implies that if we treated these two as independent networks, the people in the former network are more likely to connect to people who have same friends/foes compared with the latter one. This agrees with the observed transitivity ratios of these two network. However, they are not independent networks but two timestamps of the same graph, indicating the underlying generative models have changed since November 2008, which may reveal some interesting phenomenon on the evolution of social networks.

Note that our algorithm has a crucial tuning parameter m ; a robustness check is necessary to study the effect of the choice of m in real data analysis. Hence, we apply our graph limit-based algorithm to the network *Slashdot0902* using different values of $m = (2, 4, 6, 8, 10, 12, 14, 16)$. The performance is measured by bias^2 and values of normalized log-likelihood $\frac{1}{n^2} p_n(\hat{\theta}_m)$, where we treat $\hat{\theta}$ of $m = 16$ as the baseline to evaluate $\text{bias}^2(\hat{\theta}_m) = \|\hat{\theta}_m - \hat{\theta}_{16}\|^2$. The plots in Fig. 2 indicate that GLMLE is robust to the choice of m since ‘‘bias’’ and likelihood values remain steady after significant improvement for small m . This also suggests using m ranging from 8 to 16 may be appropriate in practice, considering the expensive computational cost for larger m .

In order to compare our method with MCMC-based approach, we obtain a random subnetwork G_{sub} from the *Slashdot0902* network via link-tracing-based sampling method. Starting with a randomly selected node, we trace all the nodes whose distances to the seed node are less or equal to k , where k , the hop of the link-tracing subsample, is the smallest number when the subnetwork size exceeds 300. In our case, the actual value of k associated with the resulting subnetwork is $k = 2$. This subnetwork G_{sub} contains 376 nodes, 1,609 edges, 48,915 two-stars and 1,661 triangles, which is a much smaller graph such that both algorithms can be applied on, for an illustration.

Fig. 2 **a** Plot of bias^2 vs. m for *Slashdot0902*, where $\text{bias}^2 = \|\hat{\theta}_m - \hat{\theta}_{16}\|^2$; **b** plot of $\frac{1}{n^2} p_n(\hat{\theta}_m)$ vs. m .

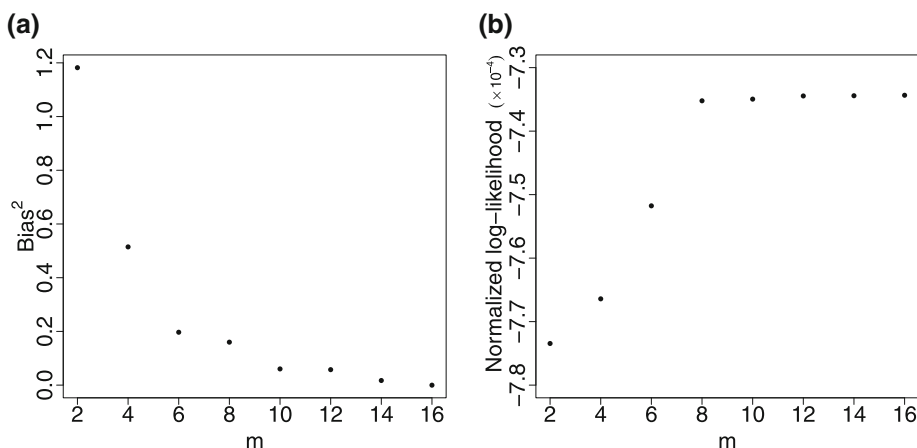


Fig. 3 Heat map of graph limits w_1, w_2, w_3, w_4 and the graph limit representation of G_{sub}, w^G , as in Table 4. The different shades of gray represent the values of $w(x, y) \in [0, 1]$, with black being 1 and white 0.

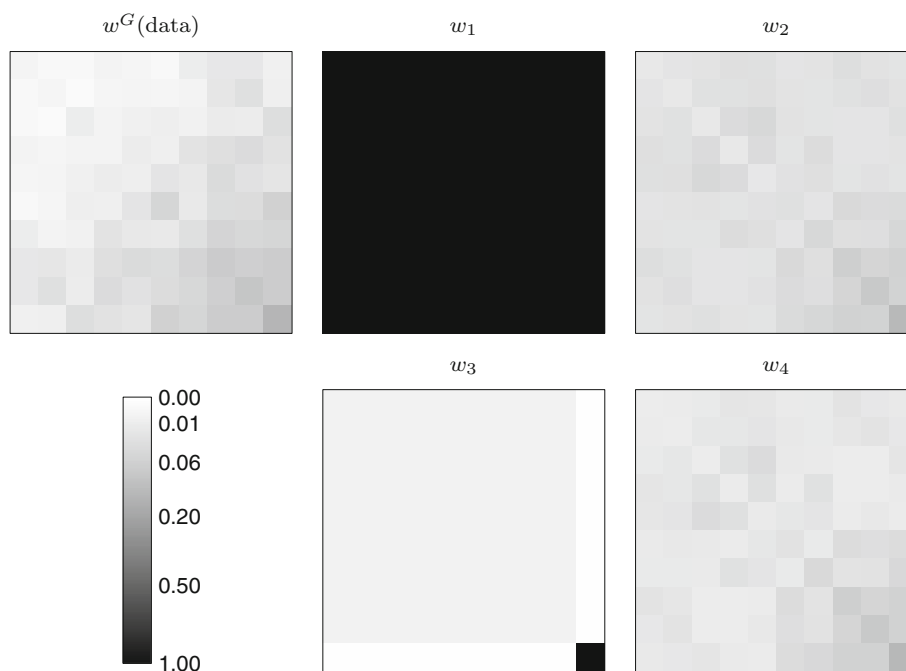


Table 4 Estimates by MCMCMLE and GLMLE for two ERGMs applied to a sub-network of *Slashdot0902*

Method	$\hat{\theta}$	Corresponding w	$\frac{1}{n^2} \log(p_n)$
Model 1			
MCMCMLE	(-2.5161, 3.3917, 43.2382)	w_1	-44.1442
GLMLE	(-1.8415, -0.7689, 0.7705)	w_2	-0.0558
GLMLE	(-1.8415, -0.7689, 0.7705)	w^G	-0.0523
Model 2			
MCMCMLE	(-1.6072, 0.1206)	w_3	-0.1408
GLMLE	(-2.1921, 0.0714)	w_4	-0.0518
GLMLE	(-2.1921, 0.0714)	w^G	-0.0497

Again consider two different ERGMs: one is (8) (referred to as model 1) while the other one is (15) (referred to as model 2). For either model, we apply both MCMCMLE and GLMLE to the subnetwork G_{sub} . Then we obtain the graph limit corresponding to the fitted value $\hat{\theta}$ through (9) to approximate the log-likelihood via (14). In addition, we calculate the approximated log-likelihood using the GLMLE and the w^G , the graph limit representation of G_{sub} as described in (6). All graph limit objects used, $w : [0, 1]^2 \rightarrow [0, 1]$, are visualized in Fig. 3. It demonstrates that the graph limits corresponding to GLMLE are closer to the graph limit representation w^G (observed data) under both models, indicating that GLMLE estimates of ERGM are closer to the true underlying unknown parameters. The numerical results are listed in Table 4.

For model 1, MCMC algorithm fails to converge in 50 iterations (the default in *ergm* is 20), yielding a *degenerate* ERGM. In fact, the corresponding w_1 of this estimate is

$w_1(x, y) = 0.9999, \forall x, y$ (see Fig. 3), indicating that it represents a complete graph and the estimates fall into the degeneracy region. This may be the reason why this algorithm fails to converge. On the other hand, our method works well and the estimates have a much larger value of the approximated log-likelihood than that of MCMCMLE. However, comparing values of log-likelihood by plugging in the graph limit corresponding to the GLMLE and the “data” w^G , we find that the latter is better. The difference, not surprisingly, is due to bias coming from simple function approximation as discussed in Sect. 4.1.

For model 2, MCMC algorithm again fails to converge after 50 updates. However, the corresponding graph limit and the value of log-likelihood indicate that it performs much better on this model than the performance for model 1. Moreover, the estimates of these two methods are more similar and so are their corresponding graph limits, while GLMLE still outperforms MCMCMLE in terms of the log-

likelihood. It is interesting to note that the MCMCMLE of the parameter for the triangle percent term is much larger than that of the GLMLE. This means that the graphs drawn from the ERGM fitted by MCMCMLE are more clustered than that of GLMLE, which is captured by the graph limit (Fig. 3).

4.3 Near-degeneracy issue of ERGMs

Many previous attempts to develop MCMC-based estimation for ERGMs have found that the algorithms nearly always converge to degenerate graphs—graphs that are either empty or complete—or that the algorithms do not converge consistently. This is because the ERGMs have near-degeneracy issue, which is defined by the distribution of some ERGMs placing disproportionate probability on a small set of outcomes. More specifically, the distribution is concentrated partly on very high-density (complete) and partly on very low-density (empty) graphs. Handcock et al. (2003) showed that this issue is a function of the form of the model and algorithm used.

As Snijders et al. (2006) pointed out, some parts of the parameter space of ERGMs correspond to nearly degenerate distributions, which may lead to convergence problems of estimation algorithms such as MCMC-based algorithm. The reason is that, in each step, MCMC-based algorithm needs to draw samples to update the parameters. But once the unreliable samples drive the values of parameters update into a near-degenerate region of parameter spaces, it is hardly to get out of that region, leading to nonconvergent MLE which may correspond to either complete or empty graph. We observed this phenomenon and showed it in Fig. 3. On the other side, graph limit-based algorithm is a deterministic method that does not need to draw samples and thus does not have this convergence issue due to degenerate region of parameter space. This is why our method is superior than MCMC-based algorithm in handling the degeneracy issue of ERGMs.

From model perspective, Snijders et al. (2006) proposed new specifications for ERGMs that represent structural properties to solve near degeneracy problem. One proposed class of models is called alternating k -triangle model. Take the ERGM using counts of edges, two-stars and triangles as sufficient statistics as an illustration, which is a special case of k -triangle ERGM model. As explained in Snijders et al. (2006), if all three parameters are positive, the model will tend to complete graph, while strongly negative value of edge parameter will force the model toward the empty graph. But if the two forces are balanced, the combined effect is a mixture of (nearly) empty and (nearly) complete graphs, which is closer to

realistic observations. This is why we choose $(-2, -1, 1)$ as the parameter values in our simulation study, which follows the above idea of alternating k -triangle model. Explicitly, the negative value of two-star parameter will offset the effect of positive value of triangle parameter, indicating the parameter values we choose are not in the degenerate region of ERGM we use.

4.4 Likelihood ratio test on ERGMs

Likelihood ratio test (LRT), a widely-used inference tool, is desirable for examining important features of ERGM graphs, by testing whether the estimate for each parameter in an ERGM is statistically significant. However, there are very little literature on the examination of LRT on ERGMs, primarily because of two issues:

1. The normalizing constant in ERGMs is intractable, which makes it computationally infeasible to calculate the value of likelihood function;
2. The distribution of LRT test statistics is unknown.

With the help of our GLMLE method, the first issue can be easily solved because our approach returns an evaluation of approximated likelihood function. The second issue remains challenging because it is very difficult to determine the exact or even asymptotic distribution of the LRT test statistics. Traditional theoretical properties of LRT, such as test statistics following χ^2 distribution according to Wilks' theorem 1938, do not directly generalize to the case of ERGMs for two reasons. First, general ERGM graph is not an IID (independent and identically distributed) data. Second, the distribution of test statistics depends on the choice of model terms as well as network size. We here, using our proposed GLMLE, carry out a close scrutiny of these problems associated with LRT and introduce a new method based on empirical p values as an alternative way to conduct LRT for ERGMs.

4.4.1 Test setup

Likelihood ratio test compares two models with one model nested in another. Specifically, the hypotheses should be in the form of

$$\begin{aligned}\mathcal{H}_0 &: \theta_i = 0, \\ \mathcal{H}_a &: \theta_i \neq 0,\end{aligned}$$

where $i = 1, \dots, k$. The test is based on the likelihood ratio, which expresses how many times the data are more likely to be fitted under the full model than the nested one. And the test statistic is twice the difference in two log-likelihoods for the full model and the nested model, which is also referred as deviance, i.e.,

$$\begin{aligned}
 D &= -2 \log \left(\frac{\text{likelihood for null model}}{\text{likelihood for full model}} \right) \\
 &= -2 \log \hat{p}_n(\hat{\theta} | \mathcal{H}_0) + 2 \log \hat{p}_n(\hat{\theta} | \mathcal{H}_a).
 \end{aligned}
 \tag{16}$$

4.4.2 Distribution of the LRT test statistic on ERGMs

As mentioned above, it is very difficult to determine the exact distribution of the test statistic (deviance) because of the complex form of ERGM statistics. Thus investigating into the asymptotic distribution of the test statistic may be more feasible. Given that deviance can be expressed in the form of maximum likelihood estimator by Taylor expansion, this problem is equivalent to examining the asymptotic distribution of MLE. Kolaczyk and Krivitsky (2011) addressed this problem by working on a simple ERGM and proving that the asymptotic distribution of MLE is normal. According to this result, the deviance will be asymptotically χ^2 -distributed with degree of freedom equal to the difference in the dimensions of parameter spaces of two models, which coincides with Wilks' theorem. However, Wilks' theorem holds for IID samples where the sample sizes of the full and nested models are the same. In other words, adding an ERGM term will not change the effective sample size, which is true in most cases when LRT is applied but not in the situation of ERGMs, due to the special nature of graph data. For instance, assuming there is no dependence among edges, the effective sample size N equals $\binom{n}{2}$ when the ERGM only contains number of edges

as the model term. An insight into the effective sample size of a network for an ERGM can be obtained by studying the asymptotic behavior of the Fisher information \mathcal{I} . For the simple ERGM that considers only number of edges, \mathcal{I} is on the order of $O(\binom{n}{2})$. But N is about $3\binom{n}{3}(4n - 9)$ when we add the number of two-stars into the model terms. This indicates that the test statistic depends not only on the difference of dimensions of parameter spaces, but also on n , the size of networks.

In order to show this, we run a simulation with different network sizes n . The continuous function used in ERGM is $T(\tilde{G}) = \theta_1 t(H_1, G) + \theta_2 t(H_2, G)$,

where $t(H_i, G)$ is defined as the homomorphism density of simple graph H_i in G , the same as (8), where, again, H_1 represents for edge and H_2 stands for two-star. And the hypotheses are

$$\begin{aligned}
 \mathcal{H}_0 &: \theta_2 = 0, \\
 \mathcal{H}_a &: \theta_2 \neq 0,
 \end{aligned}
 \tag{17}$$

which is to test whether the corresponding parameter for homomorphism density of two-stars is significant or not. We generate 100 networks under the null model and calculate the approximated likelihood of two models to show the distribution of the test statistic under the null hypothesis H_0 .

The boxplots in Fig. 4 clearly show that the distribution of test statistics is not equally distributed as χ^2_1 over different sizes of network, contrasting to the Wilks' theorem. Instead, the distribution depends on the size n . In addition, the results in Table 5 indicate that the mean and variance of test statistics increase as the network size increases. In order to further examine how network size n impacts the distribution of LRT test statistics, we plot logarithm of means and variances vs. logarithm of network sizes using 21 values of n ranging from 100 to 4000 (Fig. 5). And the slopes of the lines fitted are 0.63 and 1.12, respectively, which quantitatively reveal how LRT test statistic's null distribution depends on network size n .

Therefore, when we carry out a likelihood ratio test for the hypotheses in (17), the test statistic is not χ^2_1 -distributed, but is expected to have a mixed χ^2 distribution, where the mixture parameter depends on n . This still needs

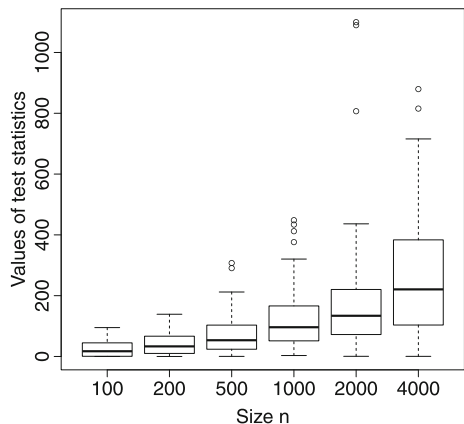


Fig. 4 Boxplots of test statistics for LRT under different settings of network size n

Table 5 Mean and variance of test statistics for (17) under different network sizes n

Network size	$n = 100$	$n = 200$	$n = 500$	$n = 1000$	$n = 2000$	$n = 4000$
Mean	24.688	40.681	70.219	120.221	177.113	266.937
Variance	647.074	1184.312	3577.469	8652.449	32,385.296	42,124.496

Fig. 5 Regression plots of logarithm of mean and variance of LRT test statistics on logarithm of network size n

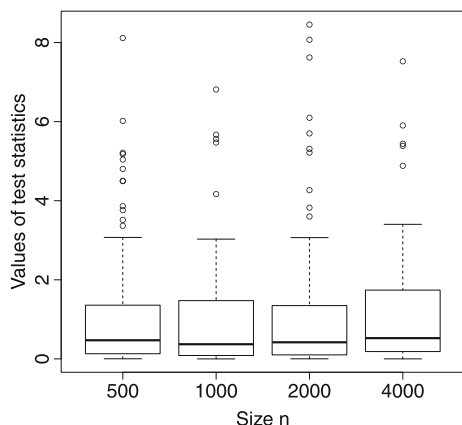
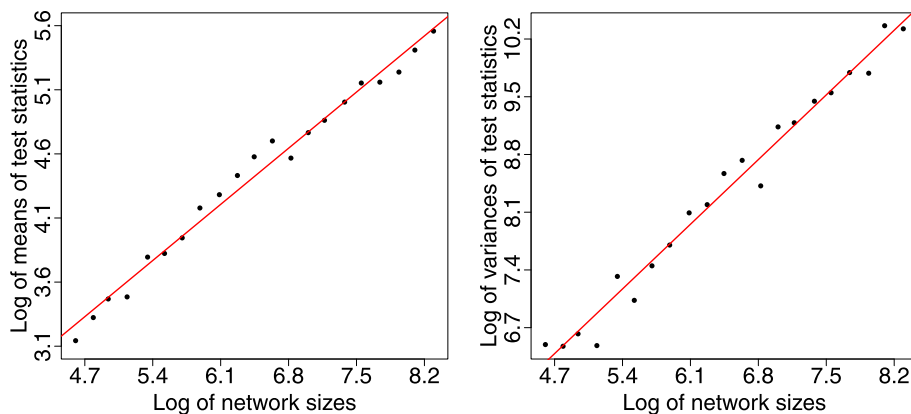


Fig. 6 Boxplots of LRT test statistics for (18) under different settings of network size n

Table 6 Mean and variance of test statistic for (18) and the corresponding KS test p value under different settings of network sizes n

Size	$n = 500$	$n = 1000$	$n = 2000$	$n = 4000$
Mean	1.1036	0.9478	1.1927	1.1288
Variance	2.5370	1.7808	3.2884	2.1232
KS test p value	0.1309	0.7507	0.1367	0.9241

a rigorous proof and the proof may be expected to be rather complicated.

Besides different model terms used in ERGMs, another thing has made it more difficult to determine the distribution of the test statistic of LRT, that is, the dependence among edges in ERGM graphs. More specifically, the effective sample size N is not very clear for a random graph from ERGMs due to the dependence in the graph. For example, $N = \binom{n}{2}$, the number of edges, for any dyadic independence model. But N is smaller than $\binom{n}{2}$ when dependence among edges exists, which is the case for ERGMs. However, except for extreme cases of ERGMs when dependence among edges are very strong, N is at

Table 7 Analysis of deviance table for two ERGMs fitted to a sub-network of *Slashdot0902*

Model	Log-likelihood	Deviance	Empirical p value
Model 1			
NULL	-48,997.19	-	-
T_1 only	-8085.31	81,823.76	<0.001
T_1 and T_2	-8019.34	131.94	0.001
Model 1	-7887.76	263.16	<0.001
Model 2			
NULL	-48,997.19	-	-
T_1 only	-8085.31	81,823.76	<0.001
Model 2	-7321.27	1528.08	0.007

least in the same order of network size, n , no matter it is for sparse graph or dense graph. This indicates the amount of information in graph data increases indefinitely as networks size n increases.

On the other hand, when the ERGM terms are the same for the full and nested models and edges are IID distributed as Bernoulli variables, assumptions of Wilks' theorem hold and we can still apply Wilks' theorem to claim the LRT test statistic follow a χ^2 distribution. This is the special case when we fit the same ERGM on two samples of networks with equal size and we use likelihood ratio test to test whether the parameters are the same, to test whether these two networks are from the same ERGM distribution. Specifically, if we consider the ERGM as in (8), the hypotheses are of the following form:

$$\begin{aligned} \mathcal{H}_0 : \theta_1 = \theta'_1, \theta_2 = \theta'_2, \theta_3 = \theta'_3, \\ \mathcal{H}_a : \theta_1 \neq \theta'_1, \theta_2 \neq \theta'_2, \theta_3 \neq \theta'_3, \end{aligned} \tag{18}$$

where $(\theta_1, \theta_2, \theta_3)$ are the parameters for the first network while $(\theta'_1, \theta'_2, \theta'_3)$ are those for the other one.

We carry out a simulation study to verify this. The boxplots in Fig. 6 indicate that distributions of test statistics

are similar over different settings of network size n . And this is validated by the numerical results in Table 6, the means and variances of test statistics. These results are in contrast to those in Fig. 4 and Table 5, when ERGM terms and the corresponding degrees of freedom are different for the full and nested models. Moreover, the result of Kolmogorov–Smirnov test shows that the test statistics are indeed from χ^2 distribution with degree of freedom equal to their mean. And p-value of KS test indicates that this result is statistically significant under any commonly used confidence level, such as 0.01, 0.05 or 0.1. This special case of hypothesis testing, from another perspective, confirms the above discussion about distribution of LRT test statistics. However, one thing that needs to be mentioned is that the degree of freedom of χ^2 distribution that test statistic follows is not 3, which is different from what Wilks' theorem states. This can possibly be explained by the fact that the number of edges, two-stars and triangles are not independent, which reduces the dimension of parameter space for this ERGM.

4.4.3 LRT based on empirical p values

Because the test statistic has an unknown exact distribution or asymptotic distribution, determining the p-value of the likelihood ratio test seems not to be tractable, where p-value is the probability of obtaining a LRT test statistic at least as extreme as the one that is actually observed under the null hypothesis. Explicitly,

$$p\text{-value} = \mathcal{P}\{D \geq D_{\text{obs}} | \Theta_0\},$$

where D is the test statistic (deviance) defined in (16) and D_{obs} is the observed value of it. However, we can still carry out a likelihood ratio test based on an empirical p-value that approximates the exact p-value without relying on asymptotic distributional theory or exhaustive enumeration. And Monte Carlo procedure can be used to obtain such empirical p-values (see Hanneke et al. 2010). That is, we sample a large number of networks from ERGMs in the null. For each network, we compute the MLE under the null hypothesis as well as the MLE under the alternative hypothesis and then calculate the LRT test statistic (deviance). This Monte Carlo procedure provides an empirical distribution of the deviance under the null hypothesis. Thus we can compare the observed test statistic with this empirical distribution to obtain the empirical p value, which is the percentage in the set of replicated samples that the value of deviance is at least the observed value. We apply this method to the above sub-network of *Slashdot0902* to test whether GLMLE for each parameter is statistically significant or not.

The results shown in Table 7 indicate that inclusion of homomorphism density of edges, T_1 , substantially improves the model fit, as does the inclusion of those of two-stars and triangles, where the last one can be seen as transitivity term. For model 2, the results are similar, where triangle percent term captures the transitivity of the graph. Moreover, based on AIC criteria, model 2 performs better than the first model. However, model 1 is to be preferred based on theoretical results by Snijders et al. (2006). They suggest a certain class of ERGMs that exhibit the desired transitivity and clumping properties of network and model 1 is a special case in this class.

5 Discussion

In this paper, we propose a new computationally efficient method for estimating the parameters of a popular model of networks—exponential random graph models (ERGMs). Motivated by the latest developments of graph limits theory, Chatterjee et al. (2013) propose a theoretical framework for estimating ERGMs based on a large-deviation approximation to the normalizing constant. We extend their ideas to more general cases of ERGMs, where the unknown corresponding graph limits are not constant, by exploiting simple function approximation and other practical tactics. Both simulation study and real data analysis are used to compare the performance of our algorithm and the most commonly used method—MCMC based-algorithm.

One limitation of our method is that it applies to a sequence of dense graph with a positive limiting density, which is inherited from the definition of graph limit; while most interest in empirical large graphs is in sparse graphs, where the graph limit tends to zero. However, this does not limit our method to be applied to empirical large graphs. This is because we can only observe one image of a large graph G , rather than a sequence of graphs. And the constant sequence $\{G, G, \dots\}$ does have a graph limit object w^G , which may be very small but still positive. On the other side, for example, though Erdos–Renyi graph with fixed p is designed for density graph, it can still be fitted to an empirical sparse graph (with a very small value of fitted p), while this is not common in practice since Erdos–Renyi model cannot capture complex structure of empirical network and thus we turn to ERGMs. In fact, the theoretical result of Chatterjee et al. (2013) shows that, in the limit, the normalizing constant of an ERGM can be approximated by solving an optimization problem, (9). In other words, when network size n is large enough, one set of parameter values correspond to a graph limit object under an equivalent class. And the idea of our algorithm is that we

want to use (9) to find a set of parameter values of ERGMs, whose corresponding graph limit object is the closest to the graph limit representation of the observed graph. Therefore, the limitation of dense graph sequences does not really hurt the application of our algorithm to the empirical large graphs.

Our method is primarily built upon two asymptotically consistent approximations. The first one is an asymptotic formula of the normalizing constant, as shown in (9). This requires the network size n , the number of nodes, to be large so that this approximation is close to the true normalizing constant. Simulations show that the asymptotic results are valid for $n > 100$. On the other hand, we use simple function approximation to estimate the corresponding graph limit w of any values of parameters θ . Theoretically, this approach works when m , as defined in (11), is large. However, we show that this approximation procedure works adequately well for m as small as 10. In order to have a more accurate approximation of graph limits (also the log-likelihood), we should employ larger m , which will result in higher computational cost. Choosing a good value of m for a particular network analysis and numerical stability of m are important problems that deserve further research.

The comparison with MCMCMLE using simulations and real data examples shows that our method, GLMLE, remarkably outperforms MCMCMLE, in terms of bias, standard errors of estimates and values of log-likelihood. Furthermore, the computation of MCMCMLE becomes impractically expensive for large graphs. The only situation where MCMCMLE performs better is when n is small, as we discussed earlier. Therefore, our proposed method, GLMLE, provides a computationally efficient alternative to MCMCMLE for large networks. We also discover that when n is large, the MCMC-based random ERGM network generating method fails. As an alternative, we incorporate the W -random graph generating procedure to simulate random graphs from ERGMs, which is shown to be a reliable method.

No proof is yet available for the consistency and asymptotic normality of the GLMLE, which are intuitively plausible based on the results of simulation studies. Table 2 indicates that GLMLE are consistent and Table 6 does support the expectation that the estimators are asymptotically normal; otherwise the test statistic will not be χ^2 -distributed. On the one hand, it is trivial to prove the asymptotic normality of the GLMLE under some certain assumptions, such as using number of edges as ERGM term and assuming no dependence exist among edges. On the other hand, nonstandard assumptions (lack of independence or using complicated ERGM terms) imply that a proof is quite difficult. This leads to determining the distribution of likelihood ratio test statistic also very hard,

because it is related to the asymptotic distribution of GLMLE. But likelihood ratio test is still doable since the empirical p value can be used as an approximation to p value via Monte Carlo procedures. Another problem caused by the unknown asymptotic distribution of GLMLE is the evaluation of the GLMLE's sampling uncertainty, i.e., standard error. For MLE with known asymptotic distributions, theoretical estimates are available such as the inverse of the Fisher Information matrix. For GLMLE, we may need to resort to resampling techniques such as bootstrap methods for the approximation of the variance of GLMLE, which requires further investigation.

Graph limits-based method for fitting ERGM is still in its early stage and needs further research. For example, we are considering more flexible functional classes for approximating a two-dimensional function (the graph limit), to improve accuracy. Moreover, in this paper, we only considered two exponential random graph models—one in (8) and another in (15). We will apply our algorithm to more general exponential random graph models, such as the new specification of ERGMs of Snijders et al. (2006) using alternating k -star, k -triangle and k -twopath as model terms. Theoretically, our algorithm can be generalized to any ERGM with continuous $T(\cdot)$, which is defined in (7).

Acknowledgments This research is, in parts, supported by NSF Grant SES-1023176.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Besag J (1975) Statistical analysis of non-lattice data. *The statistician*. pp 179–195
- Bhamidi S, Bresler G, Sly A (2008) Mixing time of exponential random graphs. In: *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on, IEEE*, pp 803–812
- Bickel PJ, Chen A (2009) A nonparametric view of network models and newman-girvan and other modularities. *Proc Natl Acad Sci* 106(50):21068–21073
- Bollobás B, Riordan O (2011) Sparse graphs: metrics and random models. *Random Struct Algorithms* 39(1):1–38
- Borgs C, Chayes JT, Lovász L, Sós VT, Vesztegombi K (2008) Convergent sequences of dense graphs i: subgraph frequencies, metric properties and testing. *Adv Math* 219(6):1801–1851
- Chatterjee S, Varadhan S (2011) The large deviation principle for the erdős-rényi random graph. *Eur J Comb* 32(7):1000–1017
- Chatterjee S, Diaconis P et al (2013) Estimating and understanding exponential random graph models. *Ann Stat* 41(5):2428–2461
- 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. *Soc Netw* 31(1):52–62

- Frank O, Strauss D (1986) Markov graphs. *J Am Stat Assoc* 81(395):832–842
- Geyer CJ, Thompson EA (1992) Constrained monte carlo maximum likelihood for dependent data. *J Royal Stat Soc Ser B (Methodological)*:657–699
- Handcock MS (2003) Statistical models for social networks: inference and degeneracy. *Dyn Soc Netw Model Anal* 126:229–252
- Handcock MS, Gile KJ (2010) Modeling social networks from sampled data. *Ann Appl Stat* 4(1):5–25
- Handcock MS, Robins G, Snijders TA, Moody J, Besag J (2003) Assessing degeneracy in statistical models of social networks. Tech rep, Working paper
- Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M (2008) statnet: software tools for the representation, visualization, analysis and simulation of network data. *J Stat Softw* 24(1):1548
- Hanneke S, Fu W, Xing EP (2010) Discrete temporal models of social networks. *Electron J Stat* 4:585–605
- Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M (2008) ergm: A package to fit, simulate and diagnose exponential-family models for networks. *J Stat Softw* 24(3):nihpa54860
- Kolaczyk ED, Krivitsky PN (2011) On the question of effective sample size in network modeling. arXiv preprint arXiv:11120840
- Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6(1):29–123
- Liu JS (2008) Monte Carlo strategies in scientific computing. Springer, New York
- Lovász L, Szegedy B (2006) Limits of dense graph sequences. *J Combin Theory Ser B* 96(6):933–957
- Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput J* 7(4):308–313
- Okabayashi S, Geyer CJ (2012) Long range search for maximum likelihood in exponential families. *Electron J Stat* 6:123–147
- Robbins H, Monro S (1951) A stochastic approximation method. *Ann Math Stat*:400–407
- Robins G, Snijders T, Wang P, Handcock M, Pattison P (2007) Recent developments in exponential random graph (p^*) models for social networks. *Soc Netw* 29(2):192–215
- Snijders TA (2002) Markov chain monte carlo estimation of exponential random graph models. *J Soc Struct* 3(2):1–40
- Snijders TA, Pattison PE, Robins GL, Handcock MS (2006) New specifications for exponential random graph models. *Sociol Methodol* 36(1):99–153
- Strauss D, Ikeda M (1990) Pseudolikelihood estimation for social networks. *J Am Stat Assoc* 85(409):204–212
- Wilks SS (1938) The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann Math Stat* 9(1):60–62