A Graphon-based Framework for Modeling Large Networks

Ran He

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2015

ABSTRACT

A Graphon-based Framework for Modeling Large Networks

Ran He

This thesis focuses on a new graphon-based approach for fitting models to large networks and establishes a general framework for incorporating nodal attributes to modeling. The scale of network data nowadays, renders classical network modeling and inference inappropriate. Novel modeling strategies are required as well as estimation methods.

Depending on whether the model structure is specified a priori or solely determined from data, existing models for networks can be classified as parametric and non-parametric. Compared to the former, a non-parametric model often allows for an easier and more straightforward estimation procedure of the network structure. On the other hand, the connectivities and dynamics of networks fitted by non-parametric models can be quite difficult to interpret, as compared to parametric models.

In this thesis, we first propose a computational estimation procedure for a class

of parametric models that are among the most widely used models for networks, built upon tools from non-parametric models with practical innovations that make it efficient and capable of scaling to large networks.

Extensions of this base method are then considered in two directions. Inspired by a popular network sampling method, we further propose an estimation algorithm using sampled data, in order to circumvent the practical obstacle that the entire network data is hard to obtain and analyze. The base algorithm is also generalized to consider the case of complex network structure where nodal attributes are involved. Two general frameworks of a non-parametric model are proposed in order to incorporate nodal impact, one with a hierarchical structure, and the other employs similarity measures.

Several simulation studies are carried out to illustrate the improved performance of our proposed methods over existing algorithms. The proposed methods are also applied to several real data sets, including Slashdot online social networks and in-school friendship networks from the National Longitudinal Study of Adolescent to Adult Health (AddHealth Study). An array of graphical visualizations and quantitative diagnostic tools, which are specifically designed for the evaluation of goodness of fit for network models, are developed and illustrated with these data sets. Some observations of using these tools via our algorithms are also examined and discussed.

Contents

List of Tables

List of Figures

Acknowledgments

I owe my deeply-felt thanks to my Ph.D advisor, Professor Tian Zheng, for her constant encouragement, support and guidance throughout my graduate study and my thesis-writing period. Professor Zheng has been not only a great academic advisor, but a great mentor and company in my life. She has taught me so much that I can still benefit greatly from for the years to come.

I would also like to thank Professors Peter Orbanz, Liam Paninski, Xiaodong Lin and Babak Heydari for agreeing to serve on my defense committee and for their helpful comments and suggestions that direct the revision of my thesis.

I am greatly indebted to all my friends at the Department of Statistics at Columbia University for their constant encouragement and support throughout the past five years, especially to Yuting Ma for her many valuable advices for this thesis. I have had the opportunity to learn from many professors in the department and got inspirations from my fellow Ph.D classmates.

Last but most importantly, I want to express my deepest thanks to my girlfriend, parents and grandparents, for their tolerance, understanding and love.

To my girlfriend, parents and those who educate me

Chapter 1

Introduction

1.1 Background

Networks are all around us. We have seen them in social networks (e.g., friendship networks and affiliation networks), communication networks (e.g., email networks and calling networks), information networks (e.g., citation networks and web graphs), biological networks (e.g., neural networks and protein-protein interaction networks), product networks (e.g., co-purchasing networks) and many more. With the rapid development of information technology, the sizes of networks are growing at an astonishing speed. Particularly, social networks have reached the sizes of hundreds of millions, with daily interactions on the scale of billions. The substantially increasing scale provides new opportunities as well as challenges for scientific research. Recently, large networks, as a form of big data, have received increased amount of attention in data science. Analyzing and modeling these data in order to understand the connectivities and dynamics of large networks are important in a wide range of scientific fields, including statistics, social science, computer science and biology.

Among popular models in the statistics literature, exponential random graph

models (also known as p^* models), to which we refer as ERGMs for short, have been developed to study these complex networks by directly modeling network structures and features (see Wasserman and Pattison 1996; Pattison and Wasserman 1999; Robins *et al.* 1999; 2007b; Handcock and Gile 2010). The broad class of ERGMs, include many popular random graph models such as the dyadic independence models and the Markov random graph models of Frank and Strauss (1986), making ERGMs one of the most widely-used and flexible parametric models for complex networks.

Despite its popularity, parameter estimation of ERGMs remains a challenging problem. Much research has been done on this challenging estimation problem. Strauss and Ikeda (1990) propose a method using the maximum pseudo-likelihood, which is inspired by Besag (1975), by assuming independence among edges. However, it is known that for many data sets pseudo-likelihood estimates are not accurate (van Duijn *et al.* 2009). Thus in the general statistical community, it has given way to Monte Carlo-based estimation methods, first proposed by Geyer and Thompson (1992). Although there are many variants of the Monte Carlo estimation techniques developed by a number of authors (see Snijders 2002; Wasserman and Robins 2005; Snijders *et al.* 2006; Handcock *et al.* 2008; Liu 2008), they are exclusively grounded on the same central idea: simulating the distribution of random graphs from a starting set of parameter values, and subsequently refining the parameter values by comparing the distribution of graphs against the observed graph, with this process repeated until the parameter estimates stabilize (Robins *et al.* 2007a). Though these approaches have good theoretical properties and have been used as standard methods for fitting ERGMs, they suffer from several limitations including the convergence issue. More importantly, Monte Carlo-based algorithms are not able to scale to large networks, 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 the same size. When the network size is large, the normalizing constant is composed of a formidably large number of terms, making the evaluation of the likelihood—let alone the maximization of it—computationally infeasible.

On the other hand, recent developments of graph limits theory due to Lovász and Szegedy (2006), Borgs *et al.* (2008) and their coauthors add new depth to the understanding of random graphs, especially of very large graphs. Graph limits theory shows that a sequence of undirected random graphs converges to a limit object under some conditions. And this graph limit object, also referred to as graphon, is defined on a two-dimensional symmetric measurable function space. This sheds light on the development of a non-parametric model for networks, usually called as W-random graph model, which is fully characterized by graphons. The non-parametric model, based on graphons, enjoys a considerably easier and more straightforward estimation procedure of the network structure, while it is difficult to interpret the fitted models, especially for the connectivities and dynamics of networks, as compared to a parametric model such as ERGM as described above.

Chatterjee *et al.* (2013) have built upon this emerging tool a new theoretical framework to understand exponential random graph models. They prove in the language of graph limits theory that any sequence of random graphs drawn from an ERGM converges to a graphon function, which can be obtained via solving an optimization problem. This exciting result reveals the connection between parametric models (ERGMs) and non-parametric models (W-random graph models) for networks and, further, the connection between ERGMs and observed network data via the bridge of graphons, making the parameter estimation of ERGMs for large networks possible. Their work is crucial in our development of a computational algorithm to fit ERGMs on large networks and is reviewed in Section 2.4.

1.2 Main results

This thesis mainly focuses on developing frameworks and inference for modeling large networks, built upon the tools from the graph limits theory. We first propose a computational estimation procedure for a popular parametric model in the network analysis, with practical innovations that make it easy-to-implement, efficient and scalable to large networks.

Extensions of this base method are then considered in two directions in order to broaden its applications. Inspired by a popular network sampling method, we further propose an estimation algorithm using sampled data, in order to overcome the practical obstacle that entire network data are usually hard to obtain and analyze. The base algorithm is also generalized to consider the case of complex network structure involving nodal attributes. Two general frameworks of incorporating nodal effect into modeling are proposed, one with a hierarchical structure and the other employs kernel functions.

Several simulation studies are carried out to illustrate some key properties of our proposed methods as well as their improved preformance over existing algorithms. The proposed methods are also applied to several real data sets, including Slashdot online social networks and in-school social networks from the National Longitudinal Study of Adolescent to Adult Health (AddHealth Study). An array of graphical visualizations and quantitative diagnostic tools, which are specifically designed for the evaluation of goodness of fit for network models, are developed and illustrated with these data sets. Some observations of using these tools via our algorithms are also studied and discussed.

The algorithm developed in this thesis is implemented in the `R` statistical language (2012) and the source code is publicly available at `http://dx.doi.org/10.7916/D8HH6HQR`.

1.3 Organization of the thesis

Chapter 2 reviews the background of exponential random graph models (ERGMs) and graph limits (or graphons) as well as the connection between these two. Chapter 3 develops a new estimation procedure for ERGMs. Simulation results and real data applications of the proposed algorithm are presented and its application on hypothesis testings is also discussed and investigated in this chapter. Chapter 4 considers the extension of our proposed method to sampled network data. Chapter 5 studies another extension to network data containing nodal attributes, resorting to the development of a general framework of modeling these data. Chapter 6 concludes and discusses research directions for the future.

Chapter 2

Background

In this chapter, we start with introducing exponential random graph models (ERGMs), one of the most widely used parametric models for random graphs, in Section 2.1. Section 2.2 reviews existing inference methods for ERGMs. In Section 2.3, we introduce graphon (or graph limits). At last, we introduce ERGMs from the perspective of graphon and presents some existing theoretical results in Section 2.4.

2.1 Exponential Random Graph Models

A network can be represented by a graph and different kinds of networks have different meanings in nodes and edges. For example, in a social network, nodes (or vertexes) in the corresponding graph typically represent individuals and edges (or ties) represent specified relationships of interest between individuals, such as friendship. In a citation network, nodes stand for scientific papers and edges represent citations. Graph provides a structural model that makes it possible to analyze and understand how these separate systems act together.

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 with n nodes, where simple

graphs are undirected graphs with no loops or multiple edges. Let $U_1, U_2, \cdots, U_k$ 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 $\boldsymbol{U}(G) = (U_1(G), \cdots, U_k(G))$ and a vector of real-valued parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_k)$, the exponential random graph model (ERGM) assumes an exponential form of the probability distribution of G:

$$\begin{aligned} p_{\boldsymbol{\theta}}(G) &\stackrel{\text{def}}{=} \exp\left\{ \sum_{i=1}^{k} \theta_i U_i(G) - \psi(\boldsymbol{\theta}) \right\} \\ &= \exp\left\{ \boldsymbol{\theta}' \boldsymbol{U}(G) - \psi(\boldsymbol{\theta}) \right\}, \end{aligned} \tag{2.1.1}$$

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

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

There have been a large number of major theoretical and empirical developments of exponential random graph models, making this class of models one of the most widely studied and used parametric models for networks (see Robins *et al.* (2007a) for a detailed summary).

Firstly, ERGMs attain their advantage in their generality. Note that there is no restriction on model statistics $\boldsymbol{U}(G)$, implying that the framework of ERGMs is so general that many popular random graph models are covered. For example, Bernoulli graphs assume independence among ties and its special case — Erdos-Renyi graphs (Erdős and Rényi 1960) — imposes a homogeneity assumption of edges and has a probability distribution of the above form with number of edges as statistics. Dyadic models, with a slightly more complicated assumption for directed networks that dyads are independent of each others, can be categorized as ERGM models as well, with statistics for single edges and reciprocated edges. Another important class of random graphs with distributions in the exponential

form (2.1.1) are Markov random graphs of Frank and Strauss (1986), which assume Markov dependence, i.e., a possible edge (i, j) is contingent on any other possible edges involving nodes i and j.

Besides the generality of the form of exponential random graph models, new specifications of ERGMs have made them more flexible and more realistic in many circumstances, both empirically and theoretically. They have enabled ERGMs to capture properties of real-world networks. In fact, the elaborations of exponential random graph models that go beyond the above Markov dependency assumption have been developed. For example, node-level effects can be introduced by adding model terms containing nodal attributes to ERGMs, such as the number of same-gender and cross-gender edges. Non-Markov dependence structures, presented by Pattison and Robins (2002), assume that edges without sharing an actor may be interdependent through third party links, inducing the new specifications of ERGMs that include higher order terms such as k-stars or k-triangles (Snijders *et al.* 2006).

Moreover, ERGMs are pervasive in network analysis as a class of parametric models. Consequently, the connectivities and formation of a network can be better understood through the interpretation of fitted parameters. Some formal inference methods for diagnosing model fitting, such as the goodness of fit testing procedure, have been proposed by Hunter *et al.* (2008a), making it possible to examine the effects of adding an ERGM model term.

Despite its popularity, exponential random graph models also suffer from criticisms in several aspects. The first one is the model formulation, related to the flexibility of ERGMs. This seemingly paradoxical fact arises because choosing an appropriate model term is hard sometimes, simply due to the reason that any statistic of a graph can be used as an ERGM configuration. Furthermore, if the model terms are contingent to each other, the interpretability of estimated coefficients may also be problematic. Another disadvantage of ERGMs is the near *degeneracy*

issue, i.e., the distribution of some ERGM places a disproportionate probability on a small set of outcomes. More precisely, the distribution is concentrated partly on graphs of either very high density (complete) or very low density (empty). Many researchers have studied these criticisms and responded with some solutions (see Handcock *et al.* 2003; Snijders *et al.* 2006; Hunter *et al.* 2008a).

2.2 Model inference

Inference of the aforementioned class of exponential random graph models is of interest in this section. Specifically, given a simple graph G as the data, fitting an ERGM on G requires finding the *maximum likelihood estimators* (MLE) of the parameters that maximize the likelihood function $p_{\boldsymbol{\theta}}(G)$. However, since the analytic form of the normalizing constant $\psi(\boldsymbol{\theta})$ is unknown due to the combinatorial complexity of summing over all possible $2^{\binom{n}{2}}$ graphs in $\mathcal{G}_n$, the MLE generally cannot be found analytically. Therefore, the evaluation of $\psi(\boldsymbol{\theta})$ remains a major obstacle in the estimation of ERGMs. Many different approaches have been proposed. We introduce, in the following, the most widely used and representative methods for fitting exponential random graph models on networks.

2.2.1 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 (2.1.1):

$$\operatorname{logit}\Big[p_{\boldsymbol{\theta}}\big(G_{ij} = g_{ij} \big| G_{ij}^c\big)\Big] = \boldsymbol{\theta}'\boldsymbol{\delta}(G_{ij}^c)$$

where $G_{ij} = \mathbf{1}\big\{(i,j) \in E(G)\big\}$ (the (i,j)th entry of adjacency table of G) and $\boldsymbol{\delta}(G_{ij}^c) = \boldsymbol{U}(G) - \boldsymbol{U}(G_{(i,j)\text{-toggled}})$, the change in $\boldsymbol{U}(G)$ when the (i,j) edge toggled

in G while the rest of the network remains G^c_{ij}. The pseudolikelihood for the model (2.1.1) 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:

$$\prod_{i,j} p_{\boldsymbol{\theta}}\left(G_{ij} = g_{ij} \middle| G^c_{ij}\right) = \prod_{i,j} \text{logit}^{-1}\left(\boldsymbol{\theta}'\boldsymbol{\delta}(G^c_{ij})\right). \tag{2.2.1}$$

The form of the pseudolikelihood (2.2.1) 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 corresponding row of the design matrix given by $\boldsymbol{\delta}(G^c_{ij})$. 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 simple implementation, the 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 among edges, while such dependence can be strong in many real-world networks. Consequently, MPLEs usually suffer from substantial bias. The standard error estimates derived from the MPLE method are also problematic, which is shown in a simulation study by van Duijn *et al.* (2009). Though its properties are poorly understood for analyzing network data, the MPLE has been commonly used as a rough approximate of the MLE, especially its providing initial values for other iterative methods such as Monte Carlo Markov Chain (MCMC) based approaches and stochastic approximation methods. For example, the 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.* 2008b).

2.2.2 Monte Carlo-based approach

Other approaches have been focused on using Monte Carlo schemes to obtain maximum likelihood estimator. Samples of random graphs are drawn from the distribution of ERGMs with given parameter values to approximate the likelihood function, so that the likelihood can be maximized by subsequently refining parameter values until stabilization.

In particular, Geyer and Thompson (1992) have proposed a Monte Carlo scheme to approximate the likelihood, using m samples $\{G_i^t\}_{i=1}^m \sim p_{\boldsymbol{\theta}^{(t)}}$ for a known $\boldsymbol{\theta}^{(t)}$ to approximate the normalizing constant $\psi(\boldsymbol{\theta})$. More specifically,

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

where C_n is a constant depending only on the number of nodes n. Plugging in the above approximation $\hat{\psi}$ in substitute of ψ in (2.1.1) yields the approximated likelihood function,

$$\hat{p}_{\boldsymbol{\theta}}\big(G; \{G_i^t\}_{i=1}^m\big) = \exp\Big\{\boldsymbol{\theta}'\boldsymbol{U}(G) - \hat{\psi}(\boldsymbol{\theta}; \{G_i^t\})\Big\}.$$

Hence, the approximated likelihood $\hat{p}_{\boldsymbol{\theta}}$ can be iteratively maximized to obtain the Monte Carlo maximum likelihood estimator (MCMLE) of $\boldsymbol{\theta}$.

Algorithm 2.1 MCMLE or MCMCMLE

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

A host of techniques for sampling graphs $\{G_i\}$ from ERGM with parameters $\boldsymbol{\theta}$ have been proposed. Liu (2008) uses the importance sampling method. Handcock *et al.* (2008) invent an MCMC-based approach that uses a local Markov chain by adding or deleting edges via the Metropolis algorithm, which has been most commonly used in the literature. The resulting estimator is usually referred to as the *Markov Chain Monte Carlo-based maximum likelihood estimator* (MCMCMLE).

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

$$E\{Z_{\boldsymbol{\theta}}\} = 0, \tag{2.2.2}$$

where $Z_{\boldsymbol{\theta}} = \boldsymbol{U}(G) - \boldsymbol{U}(G^{\text{obs}})$ and G^{obs} is the observed graph. The iteration step in the Robbins-Monroe procedure for solving (2.2.2) is

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

where a_t is the step-size and Z_t is a random variable from the distribution of $Z_{\boldsymbol{\theta}}$ specified by $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_t$ that roughly estimates $U_{\boldsymbol{\theta}}$. The step sizes a_t are a sequence of positive numbers converging to 0, with classical choice of $a_t = \frac{1}{t}$. See Snijders (2002) and Robbins and Monro (1951) for more details. Note that in the exponential families, moment estimates are also maximum likelihood estimates. Thus, this procedure provides a promising tool for approximating MLE for ERGMs, while still requiring drawing samples from $p_{\boldsymbol{\theta}}$ for a given $\boldsymbol{\theta}$ based on MCMC.

Estimators based on the Monte Carlo-based procedures are theoretically guaranteed to converge to the MLE if exists, rendering them very popular among practitioners. Handcock *et al.* (2008) implement these methods in `statnet` suite of packages in the `R` statistical language. The particular package for ERGMs is called `ergm` (Hunter *et al.* 2008b). Although their prevalent use, one common difficulty shared by these schemes is the choice of initial values. If the starting point is

close to the actual MLE, these algorithms may perform well at finding the MLE. It is certainly not the case in practice due to the lack of the knowledge of the approximate location of the MLE. In the case where the starting point is far from the true 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 $p_{\boldsymbol{\theta}}$ 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 the sampling-based methods requires a large number of samples in each iteration, they are very time- and memory-consuming and become computationally infeasible when applied to large networks. For example, `ergm` fails to run on a network of size $10,000$, which is relatively small compared to real-world networks.

2.3 Graphon

One of recent exciting developments in graph theory is the theory of graph limits, due to Lovász and Szegedy (2006), Borgs *et al.* (2008) and their coauthors. Since graph limit is also commonly referred to as graphon, we use graph limit and graphon interchangeably throughout the thesis. In the following, we first introduce the definition of the limit of a sequence of undirected dense graphs.

Definition 2.3.1. (Lovász and Szegedy 2006) For two simple graphs H and G, let $\hom(H, G)$ 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

$$t(H, G) \stackrel{\text{def}}{=} \frac{\hom(H, G)}{|V(G)|^{|V(H)|}}. \tag{2.3.1}$$

Thus $t(H, G)$ is the probability that a random map of $V(H) \to V(G)$ is a homomorphism. It is defined that a sequence of simple graphs $\{G_n\}$ is *convergent*, if the

sequence $t(H, G_n)$ of (2.3.1) 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 a convergent undirected graph sequence has a limit object, which can be represented as a measurable function. Let $\mathcal{W}$ denote the space of all measurable functions $w : [0,1]^2 \to [0,1]$ 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

$$d(H,w) \stackrel{\text{def}}{=} \int_{[0,1]^{|V(H)|}} \prod_{(i,j)\in E(H)} w(x_i, x_j) d\boldsymbol{x} \tag{2.3.2}$$

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 or indices, 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,

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

Intuitively, a sequence of random graphs $\{G_n\}$ converges to a limit if the proportion (informally) of edges, triangles and other small subgraphs in $\{G_n\}$ converges. And this limit object, graphon, is a two-dimensional symmetric measurable function $w : [0,1]^2 \to [0,1]$, which can be viewed as an infinite weighted graph on the points of the unit interval.

The above result works for a sequence of graphs $\{G_n\}$ whose corresponding graph limit object w exists when the number of nodes 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). Nevertheless, 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 graphon w^G in a natural way. Split the interval $[0,1]$ into n equal intervals $I_1, \cdots, I_n$, where $n = |V(G)|$. For $x \in I_i$, $y \in I_j$, define

$$w^G(x,y) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if} \quad (i,j) \in E(G), \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.3}$$

This representation makes sense because the constant sequence of any finite simple graph G, $\{G, G, \cdots\}$ converges to the graph limit w^G.

With the definition of a graph limit for a given sequence of graphs, a question arises naturally: how to find a sequence of graphs whose limit is a given graphon function? Lovász and Szegedy (2006) develop W-random graph models in order to answer this question, providing another perspective to the understanding of graph limits. It is a nonparametric generative model that can be used to generate random graphs $G(n,w)$ of size n from a given graphon function w. Explicitly, given a two-dimensional function (graphon) $w \in W$, W-random graph models assume that the probability of any two nodes i and j to be connected is $w(x_i, x_j)$, where x_i, x_j are the latent coordinates generated independently from $U(0,1)$. Suppose e_{ij} is the indicator function of whether nodes i and j are connected, W-random graph models can be summarized as follows.

Algorithm 2.2 Generating W-random graphs

$$\begin{aligned} X_i &\sim U(0,1) \qquad\qquad i = 1, \cdots, n \\ w(\cdot,\cdot) &\quad \text{is a graphon function} \\ P(e_{ij}=1) &= w(x_i, x_j). \end{aligned}$$

Furthermore, the corresponding graph sequence $\{G(n,w)\}$ is proved to be convergent with probability 1 to the graphon w. To this end, w fully characterizes W-random graphs $\{G(n,w)\}$, and on the other side, $G(n,w)$ captures the property

of any graph G_n whose limit is w when n is large. In other words, fitting W-random graph models on any observed network G is actually the same as estimating the underlying graphon function of G.

Much attention has been paid on the statistical inference of W-random graph models or equivalently, the estimation of graphon functions, based on observed networks. Some of the works are model-based. For example, Palla *et al.* (2010) consider a blockwise constant model and propose a MCMC-like algorithm. Airoldi *et al.* (2013) develop an approximation method for graphons based on stochastic block models (SBM) and provide related theoretical results. Similarly, Latouche and Robin (2013) also rely on the connection between SBM and W-random graph models but employ variational Bayes techniques. Others are not model-based but are dependent on nonparametric methods, such as the sorting-and-averaging algorithm proposed by Chan and Airoldi (2014) (see also Wolfe and Olhede 2013). Though different methods with various computational techniques are developed, they all share a common idea, that is, using the blockwise constant structure to estimate graphon. In fact, algorithms developed in this thesis are also based on the same motivation, where we use two-dimensional simple functions (blockwise constant functions) to approximate graphons. However, this thesis focuses more on fitting exponential random graph models via the tool of graphon rather than the estimation of graphons.

Compared to the parametric models for networks such as the exponential random graph models, the non-parametric W-random graph model offers several advantages, such as its considerably easier and more straightforward estimation procedures and its ability of capturing the structure of a network. However, interpreting the fitted models may be difficult, because the estimated graphon function can be regarded as a smoothed version of the adjacency matrix of the corresponding graph.

Therefore, if we are able to find a way to combine these two methods of ana-

lyzing networks or to find the connection between these two, advantages of both methods will be inherited. More importantly, the tool of graphon will help us better understand and estimate the exponential random graph models, especially for large networks. Since graphon estimation is built upon asymptotic approximations, it implies that this framework is favorable for large networks. Chatterjee *et al.* (2013) have initiated research in this direction and developed a theoretical framework that makes the connection between ERGMs and graphons clear. Their results will be reviewed in the following section.

2.4 ERGMs with graphon

The papers on graph limit theory define not only the limit of a sequence of graphs, but also, more importantly, the space of limit objects, $\mathcal{W}$. Any given simple graph can be mapped into this space via (2.3.3). Motivated by this representation, Chatterjee *et al.* (2013) propose to project 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 $\sigma : [0,1] \to [0,1]$. Define 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 graphon representation w^G, as in (2.3.3), 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 $\widetilde{w^G}$. We define a distance $\delta_\square$ 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 (2.3.3)

and introducing distance on this space in order to define the limit, the continuity and etc. Then the exponential random graph models can be defined on this metric space using "statistics of graphs" defined on this space.

Let $T : \widetilde{\mathcal{W}} \to \mathbb{R}$ be a bounded continuous function on the metric space $(\widetilde{\mathcal{W}}, \delta_\square)$. Then T induces an exponential random graph model of (2.1.1) on $\mathcal{G}_n$ and the probability mass function is defined as:

$$p_n(G) \stackrel{\text{def}}{=} \exp\Big\{n^2\big(T(\widetilde{G}) - \psi_n\big)\Big\}, \tag{2.4.1}$$

where $\widetilde{G}$ is the image of G in the quotient space $\widetilde{W}$ and ψ_n is the normalizing constant. Since a linear combination of continuous functions 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 $\big(T_1(\widetilde{G}), \cdots, T_k(\widetilde{G})\big)$ are of interest and the corresponding coefficients are $(\theta_1, \cdots, \theta_k)$.

A typical choice of $T(\cdot)$ is the homomorphism density $d(H, \cdot)$ as in (2.3.2), which is continuous with respect to $\delta_\square$ distance on $\widetilde{W}$, 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, resulting in the new definition of ERGMs on graphon space $\widetilde{\mathcal{W}}$ of (2.4.1) asymptotically equivalent to the traditional definition on graph space $\mathcal{G}_n$ of (2.1.1), under a reparameterization. For example, consider an ERGM with number of edges, two-stars and triangles as features, then for large n

$$\begin{aligned} T(\widetilde{G}) &= \sum_{i=1}^{3} \theta_i T_i(\widetilde{G}) = \sum_{i=1}^{3} \theta_i d(H_i, \widetilde{G}) \\ &\approx \frac{2\theta_1(\#\text{ edges in G})}{n^2} + \frac{6\theta_2(\#\text{ two-stars in G})}{n^3} \\ &\quad + \frac{6(\theta_3 - 2\theta_2)(\#\text{ triangles in G})}{n^3}, \end{aligned} \tag{2.4.2}$$

where number of two-stars is defined as the number of connected triples of vertices. On the other hand, the 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 functions" on graph space, such as the

distribution of degree sequence (indeed, homomorphism densities of k-stars with different k) or the eigenvalues of the adjacency matrix. Therefore, the results of Chatterjee *et al.* (2013), as well as those of this thesis, 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 $\widetilde{W}$, Chatterjee *et al.* (2013) give an asymptotic formula for the normalizing constant ψ_n:

$$\lim_{n\to\infty} \psi_n = \sup_{\widetilde{w}\in\widetilde{W}} \Big(T(\widetilde{w}) - I(\widetilde{w}) \Big), \tag{2.4.3}$$

where

$$I(\widetilde{w}) = \int_{[0,1]^2} I(w(x,y))dxdy$$

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 significant finding is that, when n is large, for a given continuous function $T(\cdot)$, almost all random graphs G_n drawn from ERGMs, defined by $T(\cdot)$ via (2.4.1), are close to W-random graphs F with high probability when $T(\widetilde{F}) - I(\widetilde{F})$ reaches the maximum. The approximation error between $\widetilde{G}_n$ and $\widetilde{F}$ is given in Theorem 3.2 in their paper. Precisely, for any $\eta > 0$ there exist $C, \gamma > 0$ such that for any n,

$$P\Big(\delta_\square(\widetilde{G}_n, \widetilde{F}) > \eta \Big) \le Ce^{-n^2\gamma}.$$

In other words, $\widetilde{F}$ is the underlying graphon function of ERGMs induced by $T(\cdot)$, 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 ERGMs, by evaluating $\psi(\boldsymbol{\theta})$ on a fine grid in the parameter space of $\boldsymbol{\theta}$ and then carrying out the maximization by classical methods such as a grid search. The corresponding $\hat{\boldsymbol{\theta}}$ is the estimated MLE. However, this method is more

theoretical than practical and has several limitations, which will be overcome by our proposed algorithm introduced in the next chapter.

Chapter 3

Graphon-based Estimation Method for ERGMs

In this chapter, we develop a new estimation algorithm for ERGMs, augmented with simple function approximations of graphons and an iterative procedure. We start with presenting the motivation of developing this algorithm in Section 3.1. Details of our proposed method as well as some practical remarks are included in Section 3.2. This section also discusses the identifiability issue of graphons. Section 3.3 demonstrates the advantages of our estimation procedure with comparisons to the existing method via both simulations and real data examples. A discussion about the near degeneracy issue of ERGMs is included in this section too. Further, Section 3.4 investigates likelihood ratio tests for ERGMs under different hypothesis testings, with the help of our proposed estimation algorithm. We conclude with a discussion in Section 3.5.

3.1 Motivation

As mentioned in the previous chapter, two pioneers Chatterjee *et al.* (2013), have examined exponential random graph models via the tool of graph limits (or graphons) and, more importantly, proposed a method for estimating parameters. However, their estimation method works only for some specific ERGM when its underlying graphon is known to be a constant but not for arbitrary ERGM with nonconstant graphon. In addition, evaluating the normalizing constant on a fine grid may be impossible since parameter space is infinite. Thus, it is hard to determine the range of grid unless we know approximately where the actual MLE is located, which is not the case in practice. These issues limit the application of their work to more general cases.

Motivated by their theoretical results, we propose a new graphon-based computational procedure of finding MLE of ERGMs for large networks, referred to as *graph limit-based maximum likelihood estimation* (GLMLE) algorithm. It is an improved algorithm to approximate the normalizing constant so that the likelihood function can be approximated and subsequently maximized. More specifically, providing that $\widetilde{W}$ is the space of all graphon functions $w : [0, 1]^2 \to [0, 1]$, we propose to use two-dimensional simple functions to approximate the elements in $\widetilde{W}$.

Contradict to the MCMC-based algorithm that directly estimates parameters of ERGMs from data, our GLMLE algorithm uses graphon functions as a bridge that connects models and observed network data. More precisely, any network data has an underlying graphon function via the projection (2.3.3) and any ERGM model also corresponds to an item in the graphon space according to the theoretical result developed by Chatterjee *et al.* (2013). Figure 3.1 summarizes our ideas. We describe the GLMLE algorithm in details in the following sections.

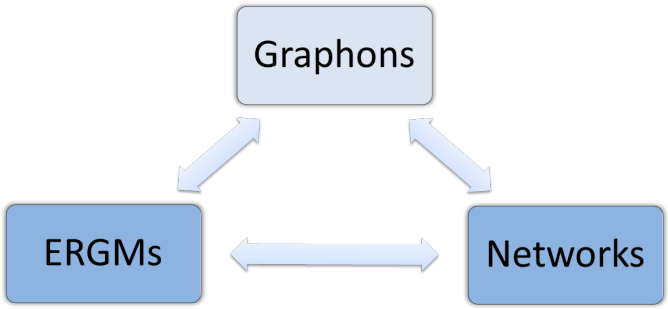

Figure 3.1: Connections among exponential random graph models, graphon functions and network data. The lighter color indicates latency.

3.2 GLMLE algorithm

3.2.1 Simple function approximation of a graphon

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,

$$A_{ij} \stackrel{\text{def}}{=} \left\{(x,y) : x \in \left[\frac{i-1}{m}, \frac{i}{m}\right) \text{ and } y \in \left[\frac{j-1}{m}, \frac{j}{m}\right)\right\}, \tag{3.2.1}$$

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

$$f_m(x,y) \stackrel{\text{def}}{=} \sum_{i,j=1}^{m} c_{ij} \mathbf{1}_{A_{ij}}(x,y), \tag{3.2.2}$$

where the indicator function $\mathbf{1}_{A_{ij}}(x,y) = \begin{cases} 1 & (x,y) \in A_{ij} \\ 0 & \text{otherwise} \end{cases}$ and $c_{ij} = c_{ji}$ 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.,

$$\int_{[0,1]^2} f_m(x,y)dxdy = \frac{1}{m^2}\sum_{i,j=1}^{m} c_{ij}. \tag{3.2.3}$$

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

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

Therefore, we can use a simple function defined in (3.2.2), with an appropriate choice of m, to approximate any function in $\widetilde{W}$, i.e., the graphon function.

Let G_n be a random graph on n nodes drawn from the ERGM distribution induced by $T(\cdot)$. Recall that, as proved in Chatterjee *et al.* (2013), the corresponding graphon of G_n can be obtained via maximizing $T(\widetilde{w}) - I(\widetilde{w})$, while on the other side, graphs constructed from this graphon function capture the properties of G_n. When the underlying graphon 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 several such examples of ERGMs whose graphons are known to be constant. When the latent graphon is unknown and more complicated, however, it is hard to solve this optimization problem since $\widetilde{W}$ contains all two-dimensional symmetric measurable functions. We address this problem by a reliable and easy-to-implement method, i.e., simple functions approximation, summarized below.

Algorithm 3.1 Simple function approximation

1. Give a bounded continuous function

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

that induces a probability mass function of ERGM, which has form (2.4.1).

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

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

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

4. An approximation of the graphon function 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 main advantage of this approximation is that it converts the optimization problem of (2.4.3) in the complicated form of integrals of functions to an optimization problem in a simple form of summations of constants, because of the property of simple functions in (3.2.3). Thus, Algorithm 3.1 simplifies the search for $\widetilde{w}$ in solving the maximization problem $\sup_{\widetilde{w}} \big(T(\widetilde{w}) - I(\widetilde{w}) \big)$, making it computationally tractable. For example, consider the ERGM of (2.4.2), which use homomorphism densities of edges, two-stars and triangles as model terms. In the third step of the above algorithm, $T(w_m) - I(w_m)$ can be written as

$$
\begin{aligned}
& T(w_m) - I(w_m) \\
= \; & \frac{\theta_1}{m^2} \sum_{ij} c_{ij} + \frac{\theta_2}{m^3} \sum_{ijk} c_{ij} c_{jk} + \frac{\theta_3}{m^3} \sum_{ijk} c_{ij} c_{jk} c_{ik} \\
& - \frac{1}{2m^2} \sum_{ij} \left[c_{ij} \log c_{ij} + (1 - c_{ij}) \log(1 - c_{ij}) \right]. \qquad (3.2.4)
\end{aligned}
$$

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

3.2.2 Parameter estimation of ERGMs via graphons

Based on the above simple function approximation algorithm, we obtain the corresponding graphon for any ERGM model as well as an approximation of the normalizing constant $\psi_n(\boldsymbol{\theta})$, by maximizing $T(\widetilde{w}) - I(\widetilde{w})$. Suppose the estimated simple function based on a known T and $\boldsymbol{\theta}^{(t)}$ is $\hat{w}_m^{(t)}$, then the approximated normalizing constant is

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

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

$$
\log \hat{p}_n(\boldsymbol{\theta}; G, \hat{w}_m^{(t)}) \stackrel{\text{def}}{=} n^2 \Big[T_{\boldsymbol{\theta}}(\widetilde{G}) - \hat{\psi}_n(\boldsymbol{\theta}; \hat{w}_m^{(t)}) \Big]. \qquad (3.2.5)
$$

Maximizing $\log \hat{p}_n$ provides the *graph limit-based maximum likelihood estimator* (GLMLE) of $\boldsymbol{\theta}$. 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.

We propose an iterative procedure as follows:

Algorithm 3.2 GLMLE

1. Give an initial value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(0)}$.
2. For each t,
   (a) Use simple function approximation to estimate $\widetilde{w}^{(t)}$ by maximizing $T_{\boldsymbol{\theta}^{(t)}}(\widetilde{w}) - I(\widetilde{w})$. 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 $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \hat{p}_n(\boldsymbol{\theta}; G, \hat{w}_m^{(t)})$.
3. Stop once $||\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}||$ converges. And the corresponding $\boldsymbol{\theta}^{(t+1)}$ is the GLMLE.

The 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 $\boldsymbol{\theta}$, we can find the best value of the "latent variables" w_m, as in the step 2(a); conversely, if we know the value of the "latent variables" w_m, we can find an update of the parameters $\boldsymbol{\theta}$, as in the step 2(b). These two steps make our algorithm a maximization-maximization procedure, similar to the motivation of the generalized expectation-maximization (EM) algorithm, though the latter requires an expectation step. Note that the above iterative algorithm provides $\boldsymbol{\theta}^{(t+1)}$ in each step such that

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

when n and m are large enough. Our algorithm is guaranteed to converge to the MLE in theory if it exists, since the likelihood function of ERGM follows an exponential family and is globally concave. The maximum found is also unique.

3.2.3 Practical remarks

3.2.3.1 Initial values

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

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

in which case the corresponding graph is an Erdos-Renyi graph. $\hat{c}$ is obtained by solving (2.4.3). As this optimization problem is reduced to a one-dimensional problem, we are able to compute this rough estimate as an initial value very fast.

3.2.3.2 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, (3.2.4). 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 more desirable 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.

3.2.3.3 Updating $\boldsymbol{\theta}$

ERGMs, whose distributions are in exponential families, have log-likelihood

$$\begin{aligned} \log p_n(\boldsymbol{\theta}; G) &= n^2 \Big[\sum_{i=1}^{k} \theta_i T_i(\widetilde{G}) - \psi(\boldsymbol{\theta}) \Big] \\ &= n^2 \Big[\boldsymbol{\theta}' \boldsymbol{T}(\widetilde{G}) - \psi(\boldsymbol{\theta}) \Big], \end{aligned}$$

while a very useful property of exponential family is that, for any $\boldsymbol{\theta}$,

$$\nabla \psi(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big]. \tag{3.2.6}$$

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

$$\nabla \log p_n(\boldsymbol{\theta}; G) = n^2 \left\{ \boldsymbol{T}(\widetilde{G}) - E_{\boldsymbol{\theta}}\big[\boldsymbol{T}(\widetilde{G})\big] \right\}. \tag{3.2.7}$$

Thus the problem of evaluating normalizing constants is converted to determining the expected values of ERGM statistics, $E_{\boldsymbol{\theta}}\big[\boldsymbol{T}(\widetilde{G})\big]$, which is a function in terms of graphon w. We illustrate this with some commonly used ERGM terms $d(H_i, \widetilde{G})$, homomorphism density of edges, two-stars and triangles. In fact, the equation (2.4.2) indicates that they are asymptotically equivalent to $U_i(G)$, number of edges, two-stars or triangles in G, under a linear transformation. Specifically,

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

where $\mathbf{1}\{\cdot\}$ is the indicator function.

Note that the graphon $w(x, y)$ is the probability of having an edge between nodes i and j given their coordinates $X_i = x$, $X_j = y$. Then we have

$$\begin{aligned}
E[U_1(G)] &= \binom{n}{2} \iint w(x,y)dxdy, \\
E[U_2(G)] &= 3\binom{n}{3} \iiint w(x,y)w(x,z)dxdydz, \\
E[U_3(G)] &= \binom{n}{3} \iiint w(x,y)w(x,z)w(y,z)dxdydz,
\end{aligned}$$

where detailed derivations can be found in Appendix A.2.

Therefore, using the simple function approximation $\hat{w}_m^{(t)}$ in tth step, $E_{\boldsymbol{\theta}}[\boldsymbol{T}(\widetilde{G})]$ can be easily approximated, since integrals reduce to summations as shown in (3.2.3). This gives us an approximation of the first derivative of the log-likelihood, $\nabla \log \hat{p}_n(\boldsymbol{\theta}; G)$. A host of gradient methods can hence be employed to update $\boldsymbol{\theta}$. These techniques include, for example, gradient descent, coordinate descent, conjugate gradient. 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 $\boldsymbol{\theta}$ in our algorithm.

However, when $E_{\boldsymbol{\theta}}[\boldsymbol{T}(\widetilde{G})]$ 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 not. 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 the step 2(b) in our algorithm though that of the step 2(a) is not affected and remains $O(m^3)$.

3.2.3.4 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 criterion is, in fact, the default stopping rule of `R` function `optim`.

3.2.3.5 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

scale 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 the step 2(a). Though the complexity remains undetermined for some ERGM terms, it is known for many commonly used ERGM statistics. For example, if we consider an ERGM with number of edges, two-stars and triangles as statistics, the time complexity of the initial step 1 is $O(1)$, while in each iteration, the computational complexity of the step 2(a) is $O(m^3)$.

3.2.4 Identifiability issue of graphon estimation

As pointed out by Orbanz and Roy (2013), graphons (or W-random graph models) suffer a strong identifiability issue as, for any measure preserving bijection $\sigma : [0,1] \to [0,1]$, the graphon function $w(\sigma x, \sigma y)$ is equivalent to $w(x,y)$, in that the resulting W-random graph models are the same. Simple monotonization of a graphon function does not circumvent this issue (see Remark 3.11 in Orbanz and Roy 2013). However, Bickel and Chen (2009) prove that the representation of the equivalent class $\widetilde{w}$ is unique if the mean density $p(x) = \int w(x,y)dy$ is *strictly* increasing, though this assumption is rarely satisfied during graphon estimation in practice, especially for the blockwise constant structure imposed on graphon functions, on which this thesis is relied.

However, this identifiability issue does not impact our GLMLE algorithm, because our primary goal is to maximize the likelihood function of ERGM and graphon only influences this procedure via the updates of parameters. The gradient updates method, as in (3.2.7), implies that the estimated parameters (GLMLE) that maximize the log-likelihood are the ones whose corresponding graphon has the expected value of "statistics" $\boldsymbol{T}$ equal to the observed $\boldsymbol{T}(\widetilde{G})$. In other words, with typical choices of $\boldsymbol{T}$ such as homomorphism densities of graph motifs, these motifs fre-

quencies calculated via graphons should be the same as observed frequencies from data. But according to Diaconis and Janson (2007), subgraphs frequencies are invariant and constitute intrinsic characteristics of an equivalent class of graphons. Therefore, though graphons may not be identifiable, the values of log-likelihood are identifiable, so are estimated GLMLE.

On the other side, we still assume $\hat{w}_m^{(t)}$ in each iteration of our GLMLE estimation procedure has a monotonic increasing value of $\int \hat{w}_m^{(t)}(x,y)dy$, following Bickel and Chen (2009). In practice, we reorder rows and columns of w_m when solving the optimization problem of the step 2(a), so that the convergence issue brought by the identifiability issue of graphons can be avoided. If this assumption is not imposed, for example, maximizing $T_{\boldsymbol{\theta}^{(t)}}(w_m) - I(w_m)$ may be a problem since the estimated $\hat{w}_m$ may jump between stages of two equivalent graphons and fail to converge. Correspondingly, throughout the thesis, we also assume $\widetilde{G}$ of the network data has an ascending marginal density, which has already been stated in Section 2.4.

3.3 Evaluations

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 estimation method for ERGMs.

3.3.1 Simulation study

For our simulation study, we consider an ERGM using homomorphism densities $d(H_i, \cdot)$ as sufficient statistics, where H_1 is edge, H_2 is two-star and H_3 is triangle. This model is actually asymptotically identical to the ERGM using number of edges, two-stars and triangles as statistics after a reparameterization of $\boldsymbol{\theta}$, as shown in (2.4.2). We specify the true values of the parameters $\boldsymbol{\theta}$ to be $(-2, -1, 1)$,

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.* 2008b), 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 absolute biases (absolute values of estimation biases) and standard errors of fitted value $\hat{\boldsymbol{\theta}}$. Our method outperforms MCMC method in almost all cases for all parameters (see Table 3.1), especially when the network size n is large. However, we notice that the absolute biases and standard errors 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 random graphs from an ERGM with given

Table 3.1: Absolute biases and standard errors of parameter estimates by GLMLE and MCMCMLE for random graphs of various sizes generated by the `R` function `simulate.ergm`.

	GLMLE			MCMCMLE		
Size n	$\lvert\text{Bias}(\hat{\theta}_1)\rvert$ $\text{se}(\hat{\theta}_1)$	$\lvert\text{Bias}(\hat{\theta}_2)\rvert$ $\text{se}(\hat{\theta}_2)$	$\lvert\text{Bias}(\hat{\theta}_3)\rvert$ $\text{se}(\hat{\theta}_3)$	$\lvert\text{Bias}(\hat{\theta}_1)\rvert$ $\text{se}(\hat{\theta}_1)$	$\lvert\text{Bias}(\hat{\theta}_2)\rvert$ $\text{se}(\hat{\theta}_2)$	$\lvert\text{Bias}(\hat{\theta}_3)\rvert$ $\text{se}(\hat{\theta}_3)$
100	0.017 (0.454)	0.429 (2.248)	0.929 (2.676)	0.042 (0.404)	0.496 (1.318)	9.800 (2.764)
200	0.022 (0.316)	0.137 (1.170)	0.075 (1.291)	0.033 (0.434)	1.757 (1.992)	23.780 (4.251)
500	0.490 (0.138)	0.285 (0.701)	0.079 (1.560)	0.481 (0.263)	0.598 (1.313)	9.748 (6.600)
1000	0.922 (0.114)	0.045 (0.617)	0.154 (0.574)	0.917 (0.219)	0.483 (1.631)	27.233 (10.139)
2000	1.347 (0.095)	0.209 (0.589)	0.355 (0.505)	1.346 (0.170)	0.458 (1.946)	20.266 (13.731)
4000	1.741 (0.084)	0.417 (0.554)	0.547 (0.356)	1.742 (0.152)	0.588 (2.536)	18.510 (19.477)

Table 3.2: Absolute biases and standard errors of parameter estimates by GLMLE and MCMCMLE for random graphs of various sizes generated by the W-random graph method.

	GLMLE			MCMCMLE		
Size n	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_3)\rvert}{\text{se}(\hat{\theta}_3)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_3)\rvert}{\text{se}(\hat{\theta}_3)}$
100	0.110 (0.833)	2.412 (4.079)	0.182 (3.200)	0.004 (0.387)	0.487 (1.243)	7.164 (2.931)
200	0.018 (0.212)	0.357 (0.813)	0.098 (1.508)	0.015 (0.338)	0.803 (1.061)	6.063 (4.126)
500	0.009 (0.110)	0.223 (0.253)	0.103 (0.356)	0.031 (0.261)	0.979 (0.813)	1.681 (2.876)
1000	0.009 (0.077)	0.225 (0.145)	0.125 (0.200)	0.031 (0.226)	0.962 (0.721)	0.557 (2.298)
2000	0.007 (0.055)	0.219 (0.145)	0.110 (0.212)	0.031 (0.173)	0.982 (0.554)	1.263 (2.045)
4000	0.007 (0.045)	0.212 (0.130)	0.094 (0.170)	0.035 (0.155)	1.029 (0.490)	1.452 (1.720)

values of parameters, due to the convergence issue of Markov chains. To illustrate this, we use the W-random graph approach to simulate graphs from the underlying graphon of the above ERGM and repeat the simulation comparison.

Recall that in Section 2.3, we introduce the W-random graph model, a generative model that can be used to simulate random draws from a graphon function. More importantly, the random draws $G(n, w)$ capture the property of any large graph G_n whose underlying graphon is w. Therefore, instead of using `simulate.ergm` to generate problematic large random graphs, we can first obtain the graph limit $w_{\boldsymbol{\theta}}$ of the above ERGM with the true values of $\boldsymbol{\theta}$, and then simulate the corresponding W-random graphs $G(n, w_{\boldsymbol{\theta}})$ as random draws from the ERGM. All other settings are exactly the same as above. The results are summarized in Table 3.2.

Comparing the absolute biases and standard errors of estimates in Table 3.2 and those in Table 3.1, we find that both methods generate more sensible estimates from W-random graphs when the network size 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 the actual w corresponding to true $\boldsymbol{\theta}$, and may deliver biased samples if m is too small.

Based on the more reliable results in Table 3.2, it can be seen that graphon-based 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, or better than, that of GLMLE. This is reasonable as GLMLE is built upon limiting behavior of large graphs. Especially, our algorithm is based on (2.4.3), an asymptotic formula for the normalizing constant ψ_n, which may work less effectively when n is small.

The function $T : \widetilde{\mathcal{W}} \to \mathbb{R}$ that induces an exponential random graph model of form (2.4.1) 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 graphon-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:

$$\begin{aligned} T(\widetilde{G}) &= \theta_1(\text{edges density}) + \theta_2(\text{triangle percent}) \\ &= \frac{2\theta_1(\#\text{ edges in G})}{n^2} \\ &\quad + \frac{\theta_2(\#\text{ triangles in G})}{(\#\text{ two-stars in G}) - 2 \times (\#\text{ triangles in G})}. \end{aligned} \tag{3.3.1}$$

We specify the true value of the parameters $\boldsymbol{\theta}$ to be $(-1.8, -0.2)$. Using W-random graph 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

Table 3.3: Absolute biases and standard errors of parameter estimates by GLMLE and MCMCMLE for the ERGM model of (3.3.1).

	GLMLE		MCMCMLE	
Size n	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$
50	0.109 (0.496)	0.103 (0.575)	0.007 (0.298)	0.074 (0.295)
100	0.091 (0.375)	0.071 (0.496)	0.000 (0.214)	0.075 (0.237)
200	0.061 (0.303)	0.037 (0.374)	0.046 (0.167)	0.112 (0.145)
300	0.026 (0.295)	0.045 (0.300)	0.091 (0.130)	0.177 (0.170)
400	0.028 (0.297)	0.025 (0.302)	0.126 (0.114)	0.137 (0.130)
500	0.021 (0.272)	0.011 (0.285)	0.149 (0.105)	0.209 (0.100)

previous simulation. The results in Table 3.3 indicate that GLMLE also works on models other than (2.4.2) and performs better than MCMC-based algorithm when the network size is large.

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 graphon function. 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 this tuning parameter on GLMLE, under criteria of bias, MSE and running time. We set $n = 4000$ and specify the true value of the parameters $\boldsymbol{\theta} = (-2, -1, 1)$ 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 Figure 3.2, where the measures of performances are

Bias^2 $= ||\bar{\hat{\boldsymbol{\theta}}} - \boldsymbol{\theta}||^2$, MSE $= \overline{||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||^2}$ and Variance $=$ MSE $-$ Bias^2. Clearly, the performances of GLMLE decrease as m increases, though the values of all measures are much smaller than those of MCMCMLE. Plots (a)(b)(c) also reveal that MSE seems to have a faster decreasing rate than bias, indicating variance of GLMLE decreases faster than bias since MSE $=$ bias^2 $+$ variance. This phenomenon can be

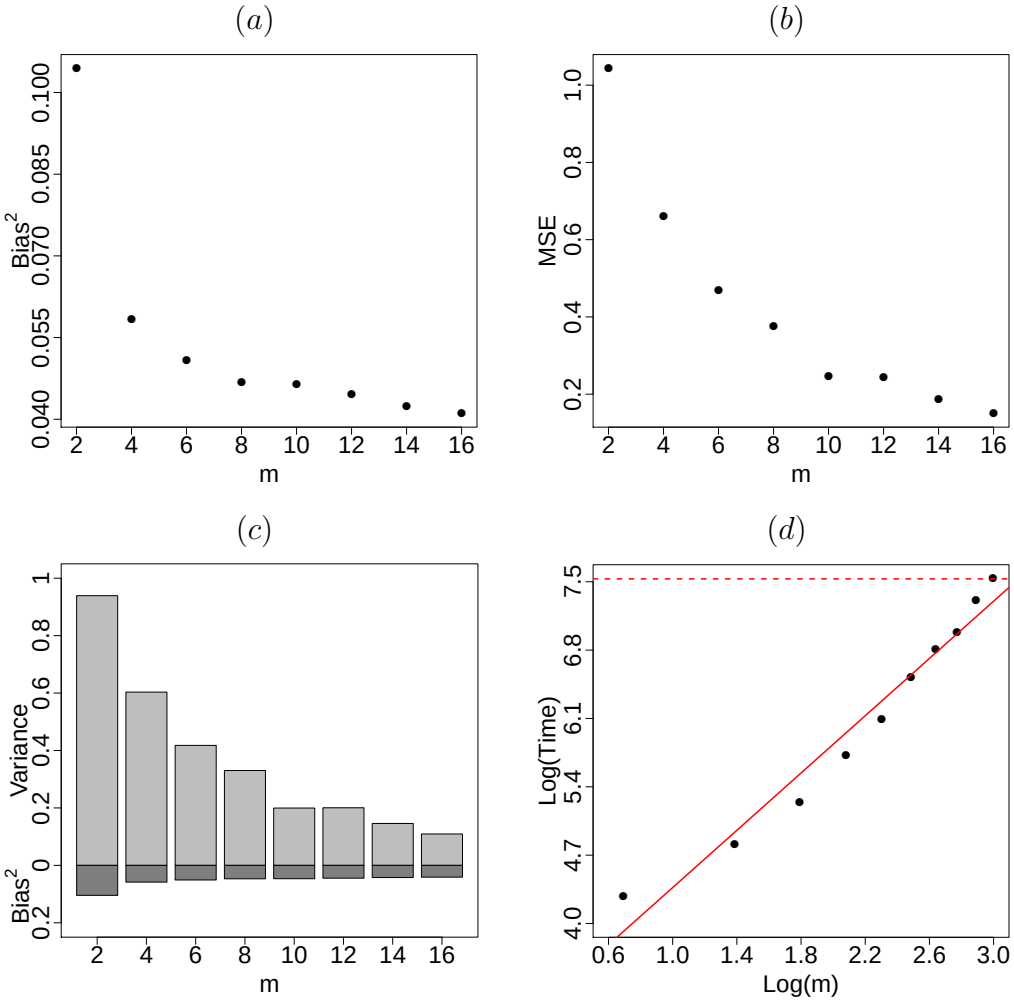

Figure 3.2: Simulation results for the impact of m on GLMLE. (a) Plot of Bias^2 vs. m. The corresponding value for MCMCMLE is 2.3333. (b) Plot of MSE vs. m. The MSE for MCMCMLE is 13.926. (c) Bar plot of Variance and Bias^2 vs. m. (d) Log-log plot of running times vs. m and the fitted line with a slope of 1.466, where the red dashed line is for the MCMC-based algorithm.

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 $\boldsymbol{\theta}$ compared with GLMLE using smaller m. Plot (d) illustrates the polynomial increase of computation time as m increases with an order of 1.466 approximately. It does not contradict with 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 the 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.

3.3.2 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 these "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 the analyses. Statistics of these two networks are shown in Table 3.4.

We first fit the ERGM in (2.4.2) to these two networks. Although MCMC-based approach works in theory for large networks, it fails in practice, primarily because

Table 3.4: Summary statistics of two networks from Slashdot.

	Number of motifs				
Data	Nodes	Edges	Two-stars	Triangles	Transtivity ratio
Slashdot0811	77,360	469,180	68,516,301	551,724	0.02416
Slashdot0902	82,168	504,230	74,983,589	602,592	0.02411

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*: $(-4.511, -1.586, 1.687)$,
2. *Slashdot0902*: $(-4.650, -1.812, 1.943)$.

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

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 $\boldsymbol{\theta}$ indicate that the log-likelihood is decreased by 9.0216 for *Slashdot0811*, while by 9.3004 for *Slashdot0902*. This implies that if we treat 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 underling 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 analyses. Hence, we apply our graphon-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 $\frac{1}{n^2} p_n(\hat{\boldsymbol{\theta}}_m)$, where we treat $\hat{\boldsymbol{\theta}}$ of $m = 16$ as the baseline to evaluate $\text{bias}^2(\hat{\boldsymbol{\theta}_m}) = ||\hat{\boldsymbol{\theta}}_m - \hat{\boldsymbol{\theta}}_{16}||^2$. The plots in Figure 3.3 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 values.

In addition, 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

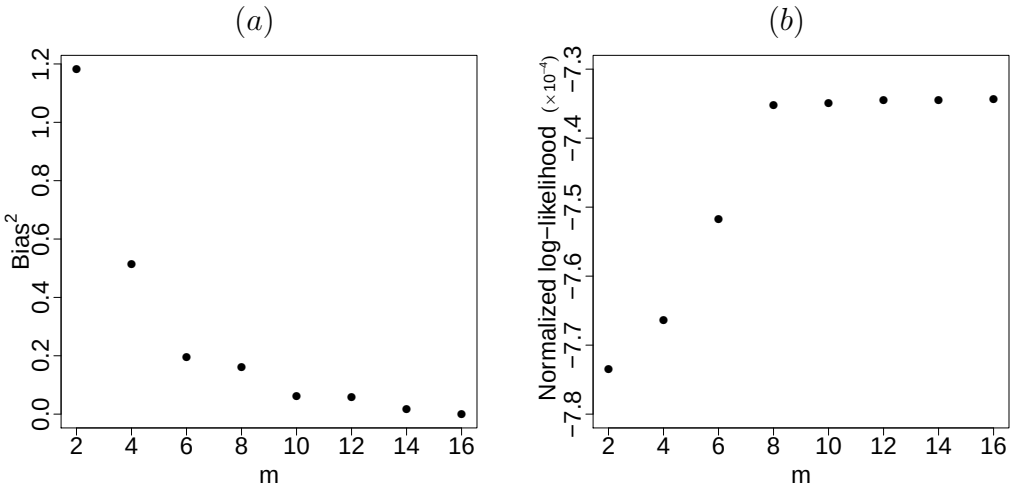

Figure 3.3: Robustness check for the impact of m on GLMLE via *Slashdot0902*. (a) Plot of Bias^2 vs. m. (b) Plot of likelihoods vs. m, where the values are normalized by a factor of 10^4.

300. In our case, the actual value of k associated with the resulting subnetwork is $k = 2$. The resulting subnetwork is a much smaller graph such that both algorithms can be applied on, for an illustration. This subnetwork G_{sub} contains 376 nodes, $1,609$ edges, $48,915$ two-stars and $1,661$ triangles.

Again consider two different ERGMs: one uses homomorphism densities of edges, two-stars and triangles as model terms (referred to as model 1) while the other one uses homomorphism densities of edges and triangle percents (referred to as model 2). For either model, we apply both MCMCMLE and GLMLE algorithms to the subnetwork G_{sub}. Then we obtain the graphon corresponding to the estimated $\hat{\boldsymbol{\theta}}$ through the maximization problem in order to approximate the value of

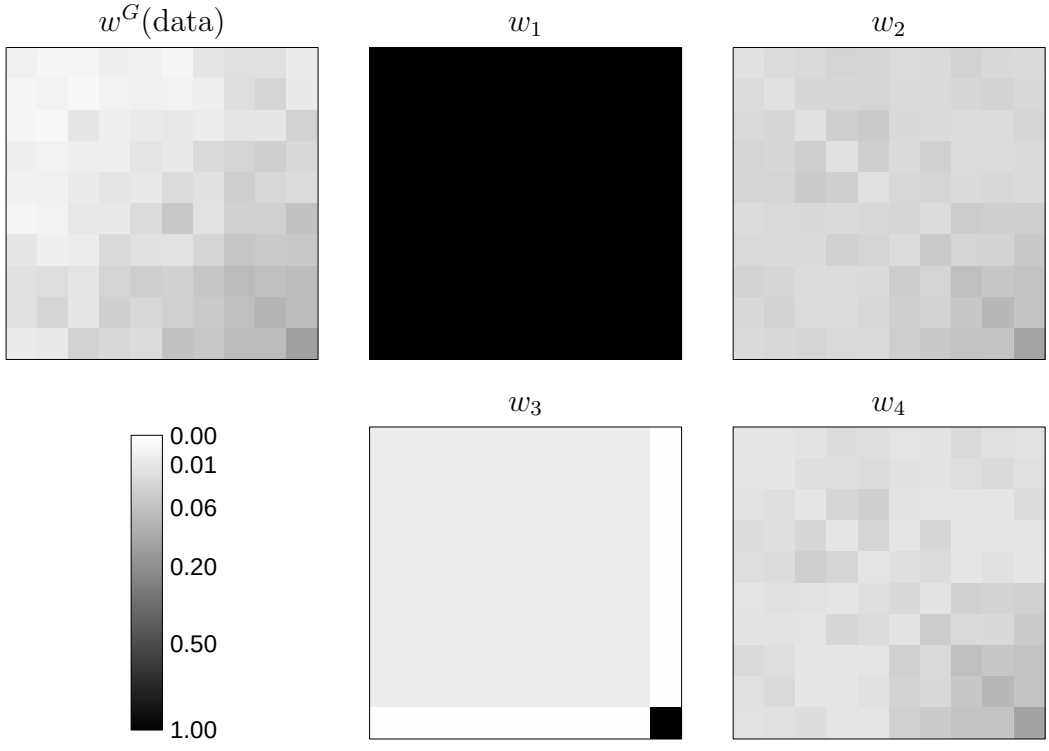

Figure 3.4: Application to the Slashdot network data. The plots are the heat maps of graphons w_1,w_2,w_3,w_4 and the graphon representation of G_{sub}, w^G, as in Table 3.5. The different shades of gray represent the values of $w(x, y) \in [0, 1]$, with black being 1 and white 0.

Table 3.5: Estimates by MCMCMLE and GLMLE for two ERGMs applied to a sub-network of *Slashdot0902*.

Method	$\hat{\boldsymbol{\theta}}$	Corresponding w	$\frac{1}{n^2}\log(p_n)$
Model 1			
MCMCMLE	$(-2.516, 3.392, 43.238)$	w_1	-44.1442
GLMLE	$(-1.842, -0.769, 0.771)$	w_2	-0.0558
GLMLE	$(-1.842, -0.769, 0.771)$	w^G	-0.0523
Model 2			
MCMCMLE	$(-1.607, 0.121)$	w_3	-0.1408
GLMLE	$(-2.192, 0.071)$	w_4	-0.0518
GLMLE	$(-2.192, 0.071)$	w^G	-0.0497

log-likelihood. In addition, we calculate the approximated log-likelihood using the GLMLE and the w^G, the graphon representation of G_{sub} as described in (2.3.3). All corresponding estimated graphon functions are visualized in Figure 3.4. The heap maps demonstrate that the underlying graphons corresponding to GLMLE are closer to the graphon 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 3.5.

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 Figure 3.4), 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 the bias coming from simple function approximation as discussed before.

For model 2, MCMC algorithm again fails to converge after 50 updates. However, the visualization of the corresponding graphon and the value of log-likelihood indicate that MCMCMLE performs much better on this model than the performance for model 1. Moreover, the estimates of these two methods are similar and so are their corresponding graphon functions, 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 is more clustered than that of GLMLE, which is captured by the corresponding graphon function (bottom middle panel in Figure 3.4).

3.3.3 Near degeneracy issue of ERGMs

Many previous attempts to develop MCMC-based estimation for ERGMs have found that these 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) show that this issue is a function of the form of the model and algorithm used.

As Snijders *et al.* (2006) point 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 in order 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 have observed this phenomenon and showed it in Figure 3.4. On the other side, our graphon-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) propose new specifications for ERGMs that represent structural properties in order 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 the ERGM we use.

3.4 Application: likelihood ratio test on ERGMs

One of the most important applications of GLMLE is its contribution to the 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 or not. 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. Firstly, general ERGM graph is not an IID (independent and identically distributed) data. Secondly, 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.

3.4.1 Test setup

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

$$\mathcal{H}_0 : \theta_i = 0,$$
$$\mathcal{H}_a : \theta_i \neq 0,$$

where $i = 1, \cdots, k$. 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, to which is also referred as deviance, i.e.,

$$\begin{aligned} D &= -2\log\left(\frac{\text{likelihood for null model}}{\text{likelihood for full model}}\right) \\ &= -2\log\hat{p}_n\big(\hat{\boldsymbol{\theta}}\big|\mathcal{H}_0\big) + 2\log\hat{p}_n\big(\hat{\boldsymbol{\theta}}\big|\mathcal{H}_a\big). \end{aligned} \tag{3.4.1}$$

3.4.2 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 complex forms 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 (effective) 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 $\mathcal{I}$ (Kolaczyk and Krivitsky 2011). For the simple ERGM that considers only number of edges, $\mathcal{I}$ is on the order of $O(n^2)$. 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. The continuous function used in ERGM is

$$T(\widetilde{G}) = \theta_1 d(H_1, \widetilde{G}) + \theta_2 d(H_2, \widetilde{G}), \tag{3.4.2}$$

where $d(H_i, \widetilde{G})$ is defined as the homomorphism density of simple graph H_i in $\widetilde{G}$, the same as (2.4.2), where, again, H_1 represents edge and H_2 stands for two-star. The hypothesis testing is

$$\begin{aligned} &\mathcal{H}_0 : \theta_2 = 0, \\ &\mathcal{H}_a : \theta_2 \neq 0, \end{aligned} \tag{3.4.3}$$

which is to test whether or not the corresponding parameter for homomorphism density of two-stars is significant. We generate 100 networks under the null model and calculate the approximated likelihood of two models in order to show the distribution of the test statistic under the null hypothesis $\mathcal{H}_0$.

The boxplots in Figure 3.5 clearly show that the distribution of test statistics is not equally distributed as χ^2_1 over different sizes of network, contrasting to the

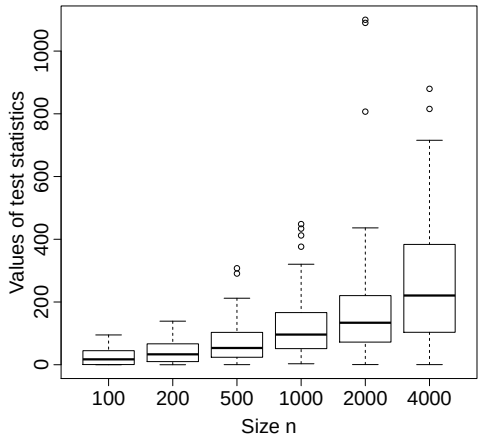

Figure 3.5: Boxplots of LRT test statistics for the hypothesis testing (3.4.3) under different settings of network size n.

Table 3.6: Mean and variance of test statistics via GLMLE for the hypothesis testing (3.4.3) under different settings of 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	32385.296	42124.496

Wilks' theorem. Instead, the distribution depends on the size n. In addition, the results in Table 3.6 indicate that the mean and variance of test statistics increase as the network size increases. In order to further examine how network size n impact the distribution of LRT test statistics, we plot logarithm of means and variances against logarithm of network sizes using 21 values of n ranging from 100 to 4000 (Figure 3.6). 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 hypothesis testing

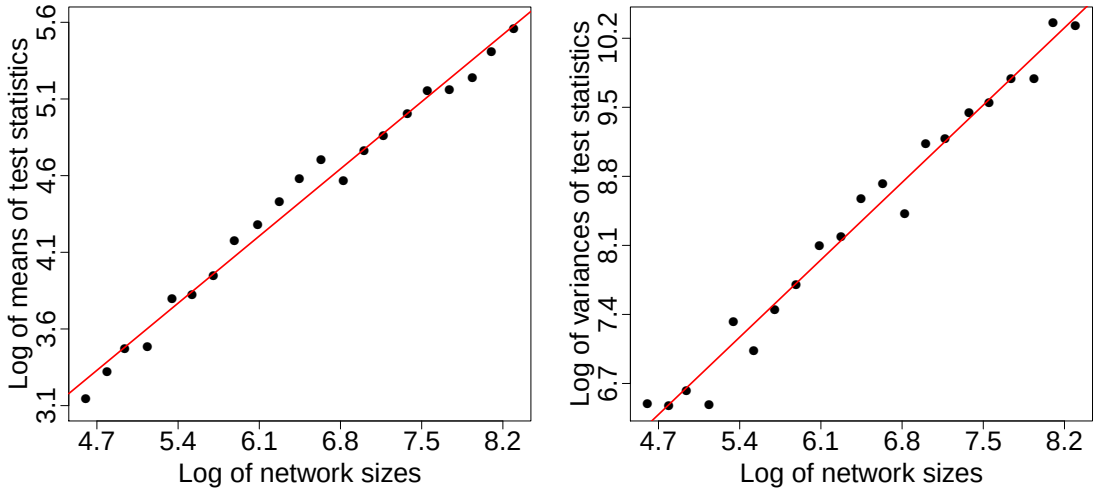

Figure 3.6: Regression plots of logarithm of mean and variance of LRT test statistics on logarithm of network size n.

(3.4.3), the test statistic is not χ^2_1-distributed, but is expected to have a mixed χ^2 distribution, where the mixing parameter depends on n. This still needs 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 Bernoulli random graph model. But N is smaller than $\binom{n}{2}$ when dependence among edges exists, which is the case for most ERGMs. However, except for extreme cases of ERGMs when dependence among edges are very strong, N is at 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 increases.

On the other hand, when 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 a same ERGM on two samples of networks with equal size and we use likelihood ratio test to test whether the parameters are the same, i.e., to test whether these two networks are from a same ERGM distribution. More precisely, if we consider the ERGM as in (2.4.2), the hypotheses are of the form:

$$\begin{aligned} \mathcal{H}_0 &: \theta_1 = \theta_1', \theta_2 = \theta_2', \theta_3 = \theta_3', \\ \mathcal{H}_a &: \theta_1 \neq \theta_1', \theta_2 \neq \theta_2', \theta_3 \neq \theta_3', \end{aligned} \tag{3.4.4}$$

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 Figure 3.7

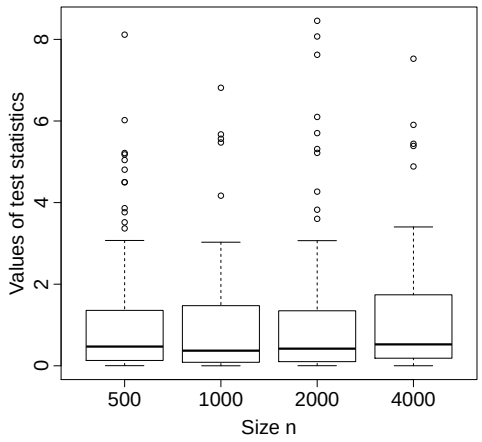

Figure 3.7: Boxplots of LRT test statistics for the hypothesis testing (3.4.4) under different settings of network size n.

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 3.7, the means and variances of test statistics. These results are in contrast to those in Figure 3.5 and Table 3.6 for the hypothesis testing studied before, when ERGM terms and the corresponding effective sample sizes are different for the full and nested models. Moreover, 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

Table 3.7: Mean and variance of test statistic via GLMLE for the hypothesis testing (3.4.4) 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

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 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.

3.4.3 LRT based on empirical p-values

Because the test statistic has an unknown exact 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,

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

where D is the test statistic (deviance) defined in (3.4.1) 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 in order 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.

Table 3.8: Analysis of deviance table for two ERGMs fitted by GLMLE to a sub-network of *Slashdot0902*.

Model	Log-likelihood	Deviance	Empirical p-value
Model 1			
NULL	−48997.19	—	—
T_1 only	−8085.31	81823.76	< 0.001
T_1 and T_2	−8019.34	131.94	0.001
model 1	−7887.76	263.16	< 0.001
Model 2			
NULL	−48997.19	—	—
T_1 only	−8085.31	81823.76	< 0.001
model 2	−7321.27	1528.08	0.007

We apply this method to the above sub-network of *Slashdot0902* in order to test whether GLMLE of each parameter is statistically significant or not.

The results shown in Table 3.8 indicate that inclusion of homomorphism density of edges, T_1, substantially improve 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, the 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 networks and model 1 is a special case in this class.

3.5 Conclusion and discussion

In this chapter, we propose a new computationally efficient method for estimating the parameters of a popular model for 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 graphons 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 $\{G, G, \cdots\}$ 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 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. In other words, when the network size is large enough, one set of parameter values correspond to a graphon function under an equivalent class. And the idea of our algorithm is that we want to use (2.4.3) to find a set of parameter

values of ERGMs, whose corresponding graphon 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 (2.4.3). 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 graphon corresponding to any values of parameters. Theoretically, this approach works when the number of bins m for simple functions, as defined in (3.2.2), 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 graphons (also the log-likelihood), we should employ larger m, which will result in a higher computational cost. Choosing a good value of m for a particular network analysis and numerical stability of m are important problems and are studied in this chapter. The result of a robustness check for m suggests that using m ranging from 8 to 16 may be appropriate in practice, considering the expensive computational cost for large m.

The comparison with existing methods using simulations and real data examples shows that our method, GLMLE, remarkably outperforms MCMCMLE, in terms of absolute biases, standard errors of estimates and values of log-likelihood. Furthermore, the computation of MCMC-based method 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 provides a computationally efficient alternative to MCMC-based algorithm for large networks. We also discover that when n is large, the MCMC-based random ERGM network

generating method fails. Thus instead, we incorporate the W-random graph generating procedure to simulate random graphs from ERGMs, which is shown to be a reliable method.

One contribution of GLMLE method is, with its help, the feasibility of likelihood ratio test (LRT) for ERGMs. Due to the special nature of a graph and the dependency among edges, the effective sample size (or Fisher information) of a network for an ERGM depends not only on model terms but also on network sizes, making the traditional result on the distribution of LRT test statistics no longer hold here. However, we propose an LRT framework based on empirical p-values to circumvent the obstacle of unknown theoretical distributions of test statistics.

Chapter 4

Graphon-based Estimation Method for ERGMs via Sampled Data

In this chapter, we extend the estimation algorithm introduced in the previous chapter to the situation when full network data are not available. Section 4.1 reveals the motivation of this extension. We then introduce a popular network sampling method in Section 4.2. Other sections are in parallel to those in the previous chapter for GLMLE. Section 4.3 presents the algorithm for sampled data in details and provides some practical remarks. Section 4.4 contains simulation results and real data analyses. Section 4.5 investigates into the likelihood ratio tests for ERGMs using only sampled data under two different hypothesis testings. We conclude with a discussion in Section 4.6.

4.1 Motivation

Despite many of the advantages that the GLMLE algorithm enjoys, there is sometimes still an obstacle of fitting ERGMs to large networks — network data. More specifically, we are often limited by data availability and the size of networks. The GLMLE method, described in the previous chapter, requires full network data in order to compute the graphon representation. However, in practice, full network data is difficult to obtain most of times with various reasons, especially for social networks. Facebook data (especially the entire network data), for example, is not public due to the privacy issue. Though Twitter data is public, it is impossible to get the entire graph structure of Twitter network via the commonly used API tool, unless we have Twitter firehose data. Besides this accessibility problem, the computational cost is another issue. For example, a subnetwork of Facebook data, scrawled by breath-first-search sampling method, is about 4 gigabytes in file size (Gjoka *et al.* 2011). It is very time consuming and memory consuming to work on this subnetwork, let alone on the entire network. Thus we turn to sampled data.

Link tracing design is one of the most widely used sampling techniques in social network analysis, especially its application on online social network data such as scrawling. However, most of times, network data come from surveys due to the inaccessibility of online network data. Link tracing designs can also be employed in surveying designs (see Salganik and Heckathorn (2004), for example). The so-called *respondent-driven sampling*, recruits respondents directly from other respondents' networks, making the sampling mechanism similar to a stochastic process on the social network (Goel and Salganik 2009). Handcock and Gile (2010) present some recent statistical advances of these designs, but they also point out that the price for this design is high, as the sampling mechanism requires physically locating the nominated respondents' network members.

Mental link tracing designs, on the other hand, collect social network data indi-

rectly, which are related to designs used in health statistics known as multiplicity sampling (Sirken 1970). In contrast to the traditional link tracing designs such as respondent-driven sampling designs, mental link tracing designs use respondents selected through standard surveys (random digit dialing telephone surveys, for example) and ask respondents questions about other people in their social network. One popular technique used in these designs is called *egocentric nominations*, which collects a specific subset of ties sent by the *ego* (respondent) and a small number of recipients, or *alters*. Typically, respondents are asked to nominate a number of relations. For each person they nominate, the interviewer then asks follow-up questions about each alter. For instance, the 2004 General Social Surveys (GSS) include a question which asks:

> From time to time, most people discuss important matters with other people. Looking back over the last six months, who are the people with whom you discussed matters important to you? Just tell me their first names or initials.

McCormick *et al.* (2012) present a thorough review of techniques used by mental link tracing designs and indicate that the egocentric nominations are in practice related to cluster sampling.

The main advantage of mental link tracing designs is that they require no special sampling techniques and are easily incorporated into standard surveys. Indirectly observed network data are, therefore, more and more popular and are feasible for a broader range of researchers, across the social science, public health and epidemiology, to implement with significantly lower cost than link tracing designs. Leskovec and Mcauley (2012) collected a sampled Facebook data, for example, from survey participants using a Facebook app, in order to study social circles in ego networks. GSS, as mentioned above, also employed mental link tracing designs in 2004. The well-known National Longitudinal Study of Adolescent to Adult Health

(AddHealth Study), as an another illustration, contained egocentric nomination questions in their questionnaires in order to collect social network data.

Motivated by the popularity of these sampled network data collected by mental link tracing designs, we extend our proposed graphon-based GLMLE algorithm to fit exponential random graph models on sampled data. The applications of ERGMs are, therefore, greatly broadened, so is our GLMLE method.

4.2 Sampled egocentric data

We focus on sampled network data collected by mental link tracing designs. Specifically, the egocentric nominations sampling technique is considered. Suppose that we randomly sample n respondents from the population (or the full social network G^{full}) of interest, where the population size (or the size of G^{full}) is N. Any standard survey methodology can be used, such as random digit dialing telephone surveys or street intercept surveys. The interviewer then asks each respondent (ego) to nominate up to k recipients (alters) and follow-up questions about relations among alters. For example, the questionnaires can be designed to include the following two questions:

- From time to time, most people discuss important matters with their friends. Looking back over the last one year, who are your best friends with whom you kept a close contact with? Please nominate up to k best friends and just tell me their names or initials.

- Do they also keep a close friendship with each other? Please list all pairs of your nominated friends who are also close friends to each other.

In most cases nominating up to k alters is not a random sample, but rather is related to the strength of the tie of interest, such as friendship.

The above egocentric nominations technique can be summarized as the following sampling algorithm:

Sampling Algorithm 4.1 Egocentric nominations

Input: The full network data $G^{\text{full}} = (V^{\text{full}}, E^{\text{full}})$

1. Random select n nodes from V^{full}.
2. For each selected node $i = 1, 2, \cdots, n$,
   (a) Select up to k nodes with top edge strength from i's neighbor in G^{full}, i.e., $\left\{ i_j, j \leq k \;\middle|\; (i, i_j) \in E^{\text{full}} \text{ and } S_{(i,i_1)} \geq S_{(i,i_2)} \geq \cdots \geq S_{(i,i_k)} \right\}$, where $S_{(i,i_j)}$ stands for the strength of (i, i_j);
   (b) Select subgraph $G_i^{(k)}$, containing nodes $\left\{ i, i_1, \cdots, i_k \right\}$, from G^{full}.

Output: Egocentric samples $\{G_i^{(k)}, i = 1, \cdots, n\}$.

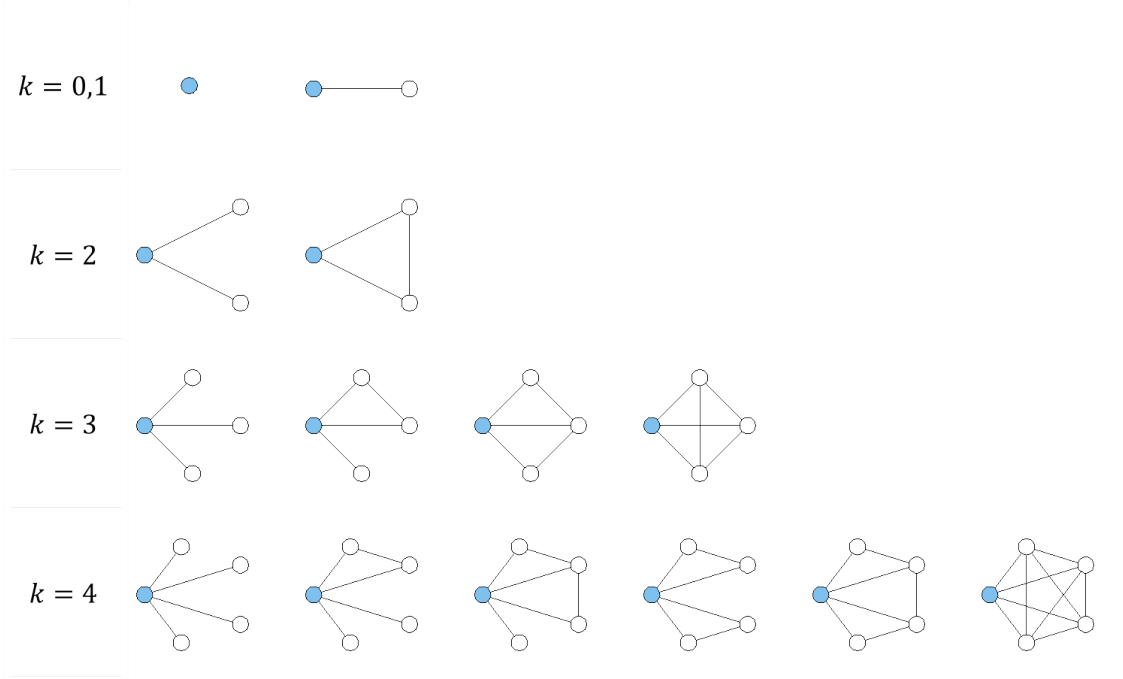

Figure 4.1: Some examples of egocentric samples for different k, where colored node represents ego while uncolored nodes are alters.

The collected sample data are n egocentric samples or more precisely, small graph motifs of size up to $k+1$, where k is the pre-specified maximum number of alters nominated. Figure 4.1 shows some examples of egocentric samples for different k.

4.3 Sample-GLMLE algorithm

As discussed in Chapter 3, graphon captures the underlying topology structure of a large network, and is an intermediate step that connects ERGM models and network data, as shown in Figure 3.1. This is also true for sampled data (see Figure 4.2). Thus, the intuition of our sample-GLMLE algorithm is to find the parameters of ERGMs that maximize the likelihood function of sampled data.

Suppose we use egocentric nomination sampling scheme to sample from a full network G^{full}. The size of G^{full} is N and the sample size is n. Denote the observed egocentric sample data as $\{G_i^{(k)}, i = 1, 2, \cdots n\}$, where k is the maximum number of alters an ego is asked to nominate. Suppose the graphon function of G^{full} is $w(\cdot, \cdot)$, which is unknown. We assume that N is much larger than n, such that egocentric samples can be regarded as independent. Thus the likelihood function of observing

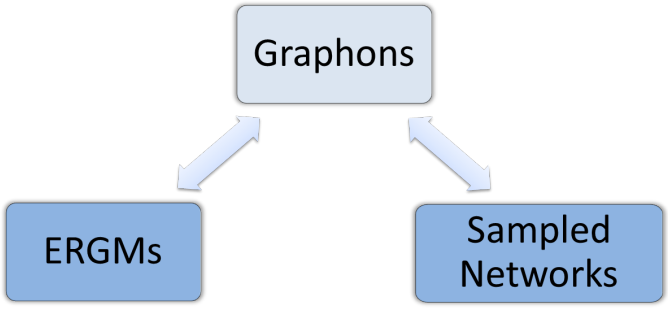

Figure 4.2: Connections among exponential random graph models, graphon functions and sampled network data. The lighter color indicates latency.

these egocentric samples is

$$L\big(w;\{G_i^{(k)}\}_{i=1}^n,N\big)\stackrel{\text{def}}{=}\prod_{i=1}^n P\big(G_i^{(k)}|w,N\big),$$

where $P\big(G_i^{(k)}|w,N\big)$ is the probability of observing an egocentric graph motif $G_i^{(k)}$ via sampling from a full network.

Note that many egocentric graph motifs are equivalent or, more precisely, isomorphic. For example, many people may only nominate one best friend, implying many $G_i^{(k)}$ are dyads. Therefore, $\{G_i^{(k)}, i=1,2,\cdots n\}$ can be categorized as distinct nonisomorphic graph motifs $\{M_j^{(k)}, j=1,2,\cdots m^{(k)}\}$ with counts n_j respectively, where $m^{(k)}$ is the number of distinct nonisomorphic graphs among $\{G_i^{(k)}\}_{i=1}^n$ and $\sum_{j=1}^{m^{(k)}} n_j = n$. Hence the likelihood function can be further expressed as

$$\begin{aligned} L\big(w;\{G_i^{(k)}\}_{i=1}^n,N\big) &= \prod_{i=1}^n P\big(G_i^{(k)}|w,N\big) \\ &= \prod_{j=1}^{m^{(k)}}\big[P\big(M_j^{(k)}|w,N\big)\big]^{n_j}, \end{aligned} \tag{4.3.1}$$

where $P\big(M_j^{(k)}|w,N\big)$ is the probability of obtaining an egocentric sample isomorphic to $M_j^{(k)}$ in terms of the latent graphon w and known N. The detailed method to calculate these probabilities will be introduced in the next section 4.3.1.

Similar to the GLMLE algorithm, we rely on the optimization problem (2.4.3) to connect ERGM models (or parameters) with their underlying graphons, and further with egocentric sample data. Again, we use simple function method proposed before to approximate graphons in the sample-GLMLE algorithm. Our proposed method can then be summarized in pseudocodes as shown in Algorithm 4.2.

Algorithm 4.2 Sample-GLMLE

Input: ERGM model, the size of the whole network N, raw egocentric sample data $\{G_i^{(k)}, i = 1, 2, \cdots n\}$.

1. Categorize $\{G_i^{(k)}, i = 1, 2, \cdots n\}$ as distinct nonisomorphic graph motifs $\{M_j^{(k)}, j = 1, 2, \cdots m^{(k)}\}$ and count the corresponding occurrence n_j.

2. Give an initial value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(0)}$.

3. Update $\boldsymbol{\theta}$, until the corresponding approximated graphon maximizes the likelihood of samples $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big) = \prod_{j=1}^{m^{(k)}} \big[P(M_j^{(k)}|\widetilde{w}, N)\big]^{n_j}$.

   - For any $\boldsymbol{\theta}^{(t)}$, use simple function approximation to estimate $\widetilde{w}^{(t)}$ by maximizing $T_{\boldsymbol{\theta}^{(t)}}(\widetilde{w}) - I(\widetilde{w})$. The corresponding simple function is $\hat{w}_m^{(t)} = \sum_{i,j=1}^{m} \hat{c}_{ij} \mathbf{1}_{A_{ij}}(x, y)$.

4. Stop once $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$ is maximized. And the corresponding $\hat{\boldsymbol{\theta}}$ is the sample-GLMLE.

Output: sample-GLMLE $\hat{\boldsymbol{\theta}}$ based on egocentric sample data and the corresponding graphon function $\hat{w}_m$

Note that the size N of G^{full} is an input. In other words, the sample-GLMLE algorithm assumes N is known. The reason is that the latent graphon is for the entire network and the probabilities of observing egocentric samples depend not only on graphons but also on network sizes. For example, the probability of a respondent having at least one friend when the population is 10 million is obviously higher than that probability when the population is only 10, assuming people have 0.5 probability of being friends with each other. In fact, the proportion $\frac{n}{N}$ impacts the performance of sample-GLMLE, which will be shown in the simulation studies (Section 4.4.1). On the other side, the assumption that full network (population) size is given is not impractical since most of times N can be obtained or estimated. For example, if we are interested in the social network in schools (like AddHealth Study), the total number of students are known. Some online networks, such as

Facebook, also provides the total number of users. If N is unknown, its value is still possible to be roughly estimated via standard population size estimation techniques.

4.3.1 Probability of egocentric graph motif

We propose the following proposition for the conditional probability of any egocentric graph motif M given a graphon function w and the full network size N, i.e., $P(M|w,N)$. The detailed derivation is included in Appendix B.1.

Proposition 4.3.1. *Suppose $V(M) = \{1,2,\ldots,m\}$ is the set of vertices of M. Without loss of generalization, always label an ego vertex as 1. Let*

$$\begin{aligned} m-1 &= \{\textit{the degree of the ego vertex in the egocentric motif } M\} \\ &= \sum_{j\in V(M)} I\{M_{1j}=1\} \end{aligned}$$

where $\{M_{ij}\}$ is the adjacency matrix of M. Obviously, $m-1 \le k$. Denote

$$\begin{aligned} A(M) &= \{\textit{alters in } M\} \\ &= V(M)\backslash\{1\}. \end{aligned}$$

Denote $\bar{M}$ as the complete graph on $V(M)$ and $p(x) = \int_{[0,1]} w(x,y)dy$.
Then, if $m-1<k$,

$$\begin{aligned} &P(\textit{observing } M|w,N) \\ =\ & \binom{N-1}{m-1}\int_{[0,1]^m}\Big\{\prod_{(i,j)\in E(M)} w(x_i,x_j)\prod_{(i,j)\in E(\bar{M})\backslash E(M)}[1-w(x_i,x_j)]\Big\}[1-p(x_1)]^{N-m}d\boldsymbol{x}. \end{aligned}$$

If $m-1=k$,

$$\begin{aligned} &P(\textit{observing } M|w,N) \\ =\ & \sum_{d=k}^{N-1}\binom{N-1}{d}\int_{[0,1]^m}\Big\{\prod_{(i,j)\in E(\bar{M})\backslash E(M)}[1-w(x_i,x_j)]\Big\}[p(x_1)]^{d-m+1}[1-p(x_1)]^{N-1-d}d\boldsymbol{x}. \end{aligned}$$

Plugging the values of these probabilities into the formula (4.3.1) yields an evaluation of the likelihood function of egocentric samples $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$.

4.3.2 Practical remarks

4.3.2.1 Nonisomorphic graph motifs

In our proposed sample-GLMLE algorithm, the first step involves categorizing $\{G_i^{(k)}, i = 1, 2, \cdots n\}$ as distinct nonisomorphic graph motifs $\{M_j^{(k)}, j = 1, 2, \cdots m^{(k)}\}$ and then counting the corresponding occurrence n_j. This falls in the realm of the well-known graph isomorphism problem, which belongs to *NP* (nondeterministic polynomial time) problems and is neither known to be solvable in polynomial time nor NP-complete (Michael and David 1979).

To make this worse, the maximum number of possible distinct nonisomorphic graph motifs for a given k may reach a formidably large number when k and n are large. Suppose we ask every respondent to nominate up to k recipients, then the collected egocentric samples are graph motifs where an ego is connected to all alters while alters may or may not connect to each other. Then the number of non-equivalent (nonisomorphic) graph motifs is just the number of nonisomorphic graphs (connected or disconnected) among k nodes. Assume n is large enough such that any nonisomorphic egocentric graph motif with up to $k+1$ nodes is possible to be observed, then $m^{(k)} \sim O\big(\frac{2^{\binom{k}{2}}}{k!}\big)$, which grows even faster than exponential rate (see Table 4.1).

Table 4.1: Maximum number of distinct nonisomorphic motifs of up to $k+1$ nodes.

k	1	2	3	4	5	6	7	8	9	$\cdots$	14
$m^{(k)}$	2	4	8	19	53	209	1,253	13,599	288,267	$\cdots$	29,104,823,811,067,332

In fact, according to Harary and Palmer (2014), there is a formula for $m^{(k)}$:

$$m^{(k)} = \sum_{n=1}^{k} \frac{2^{\binom{n}{2}}}{n!} \left(1 + \frac{n(n-1)}{2^{n-1}} + \frac{8n!(3n-7)(3n-9)}{2^{2n}(n-4)!} + O\Big(\frac{n^5}{2^{5n/2}}\Big)\right).$$

All of the above indicate that the sample-GLMLE algorithm is computationally infeasible for large k. Thus in practice, we use a small value of k, ranging from 2 to 4. For example, when $k = 2$, possible egocentric samples can only be in the form of single node, edge, two-star or triangle (the first two rows in Figure 4.1). Then, it is trivial to count the occurrences of these graph motifs among $\{G_i^{(k)}\}_{i=1}^n$. For larger value of k such as 4, we may need to check whether two egocentric graphs are isomorphic or not, and the `R` function `graph.isomorphic` from the `igraph` package (Csardi and Nepusz 2006) can be employed.

4.3.2.2 Initial values and obtaining w_m

Similar to the GLMLE algorithm, in order to obtain initial values $\boldsymbol{\theta}^{(0)}$, we apply the sample-GLMLE method by assuming the underlying graphon is a constant function, i.e., setting $m = 1$. The computational expense is very cheap since the optimization problem (2.4.3) is reduced to one-dimensional. Moreover, the practical remarks of obtaining w_m for GLMLE, as in Section 3.2.3.2, still hold here.

4.3.2.3 Maximizing the likelihood function

One important step of our sample-GLMLE algorithm is to maximize the likelihood function $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$. However, there may be computational issues for evaluating its value. For example, when n is large, the value of $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$ may be lower than the machine tolerance, making our optimization algorithm fail. Using log-likelihood may not remedy this, since $\log 0$ is not a number.

On the other hand, it should be noticed that the summation of the numbers of

observed nonisomorphic sample motifs equals the sample size,

$$\sum_{j}^{m^{(k)}} n_j = n,$$

and the probabilities of observing these motifs add up to 1,

$$\sum_{j}^{m^{(k)}} P\big(M_j^{(k)}|w, N\big) = 1.$$

Therefore, the likelihood function $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$ is actually proportional to the joint likelihood of a vector of multinomial random variables with parameters $p_j = P\big(M_j^{(k)}|w, N\big)$ and the observed occurrences are $n_j, j = 1, 2, \cdots, m^{(k)}$. With the help of Lagrange multipliers method, the maximum likelihood estimators of p_j for multinomial distribution can be easily derived,

$$\hat{p_j} = \frac{n_j}{n}, \qquad j = 1, \cdots, m^{(k)}.$$

In other words, maximizing $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$ is equivalent to finding w such that $P\big(M_j^{(k)}|w, N\big)$ matches $\frac{n_j}{n}$ for all $j = 1, \cdots, m^{(k)}$. Under L_2 norm, the criteria can also be expressed as

$$\min_w \left\{ \sum_{j=1}^{m^{(k)}} \left[P\big(M_j^{(k)}|w, N\big) - \frac{n_j}{n} \right]^2 \right\} = \frac{1}{n^2} \min_w \left\{ \sum_{j=1}^{m^{(k)}} \left[n \cdot P\big(M_j^{(k)}|w, N\big) - n_j \right]^2 \right\}.$$

Thus in practice, the above optimization criterion is more preferable than directly maximizing the likelihood or the log-likelihood function.

4.3.2.4 Computational complexity

The initial step 1 separates raw egocentric sample data into categories of distinct nonisomorphic graph motifs and counts their occurrences, requiring $O\left(n\frac{2^{\binom{k}{2}}}{k!}\right)$ times of checking isomorphism. For a fix small value of k, the computation time reduces to $O(n)$.

On the other hand, the complexity of our sample-GLMLE depends on the choice of ERGM model terms, which is the same as GLMLE. Assume the most complex ERGM term is the number of two-stars or triangles. In each iteration t of the step 3, the computational complexity of solving the maximization problem in order to obtain $\hat{w}_m^{(t)}$ is $O(m^3)$. However, unlike the GLMLE algorithm where the likelihood function to be maximized is the likelihood of ERGM (2.4.1), the likelihood function used in sample-GLMLE is the product of $P(G_i^{(k)}|\hat{w}_m^{(t)}, N)$, containing at most $\min\left(n, O\left(\frac{2^{\binom{k}{2}}}{k!}\right)\right)$ different $P(M_j^{(k)}|\hat{w}_m^{(t)}, N)$. Applying Proposition 4.3.1 yields the value of the likelihood function, whose complexity is $O\Big(\min(n, \frac{2^{\binom{k}{2}}}{k!})\Big)$. Therefore, the computation time in each iteration is $O\Big(m^3 \cdot \min(n, \frac{2^{\binom{k}{2}}}{k!})\Big)$. As discussed before, in practice we choose small k, which reduces the complexity to $O(m^3)$.

All the results above indicate that the sample-GLMLE method does not depend on the full network size N, guaranteeing its scalability to large networks.

4.4 Evaluations

Note that sample-GLMLE is computationally infeasible for a large value of k. In other words, we let respondents nominate fewer recipients when we carry out surveys. Therefore, in order to evaluate the performance of sample-GLMLE, we only consider $k = 2$ in this section. The possible egocentric motifs can hence only be in 4 forms: a single node, an edge, a two-star or a triangle. It is also easy to check egocentric samples belong to which of these four forms and count the corresponding numbers n_j, $j = 1, 2, 3, 4$.

According to Proposition 4.3.1, we have

$$P(G_i^{(2)} \text{ is a single node}|w, N) = \int_0^1 (1-p(x))^{N-1}dx,$$

$$P(G_i^{(2)} \text{ is a single edge}|w, N) = \int_0^1 (N-1)p(x)(1-p(x))^{N-2}dx,$$

$$\begin{aligned} & P(G_i^{(2)} \text{ is a two-star}|w, N) \\ = & \sum_{d=2}^{N-1} \binom{N-1}{d} \int_{[0,1]^3} \big[1-w(y,z)\big] w(x,y)w(x,z) \big[p(x)\big]^{d-2} \big[1-p(x)\big]^{N-1-d} dxdydz, \end{aligned}$$

$$\begin{aligned} & P(G_i^{(2)} \text{ is a triangle}|w, N) \\ = & \sum_{d=2}^{N-1} \binom{N-1}{d} \int_{[0,1]^3} w(y,z)w(x,y)w(x,z) \big[p(x)\big]^{d-2} \big[1-p(x)\big]^{N-1-d} dxdydz. \end{aligned}$$

Plugging these probabilities into (4.3.1) returns the value of the likelihood function of egocentric samples. Though we can choose any ERGM model for sample-GLMLE, we still consider the ERGM using the homomorphism densities of edges, two-stars and triangles as sufficient statistics, as in (2.4.2). This is simply because we want to compare the performance of sample-GLMLE with those of GLMLE and MCMCMLE, where the latter two use full network data. The comparison is reasonable only under the same ERGM model.

4.4.1 Simulation study

Recall that in Section 3.3, we have illustrated the unscalability of the traditional MCMC-based random graph generating method for ERGMs (the `R` function `simulate.ergm`), as well as the ability of generating desired large ERGM random graphs via the W-random graph approach. Thus we choose the W-random graph method here to generate large ERGM random graphs.

We specify the true values of the parameters $\boldsymbol{\theta}$ to be $(-2, -1, 1)$, the same as before. We generate ERGM graphs of different sizes $N = (500, 1000, 2000, 4000)$ for this model. In each case, we simulate 100 graphs and apply GLMLE as well as the MCMC-based procedure (`R` function `ergm`) to model these full network data. For each simulated full network, we use egocentric nominations techniques with $k = 2$ to sample $n = 100$ egocentric samples and apply the sample-GLMLE algorithm on these sampled data. For simple function approximation, we set $m = 10$.

We measure the performance of sample-GLMLE in terms of absolute biases and standard errors of the fitted value $\hat{\boldsymbol{\theta}}$. The comparison among three approaches is made under the MSE criterion. Results are included in Table 4.2 and Table 4.3. There is no improvement of the performance of sample-GLMLE as N increases. In contrast, the performance is declining. For example, when $N = 500$, almost all the values are the smallest in the table (except for the absolute bias of $\hat{\theta}_2$) and they increase for larger N, which is especially true for the standard errors of all three parameters. This phenomenon can also be verified by the corresponding values of MSE in Table 4.3. This is not surprising because we let full network size N increases while keep sample size n fixed. Hence sampled data contain fewer information of a full network as its size getting larger. In fact, the performance of sample-GLMLE

Table 4.2: Absolute biases and standard errors of parameter estimates by sample-GLMLE using sampled network data.

Full network size	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_3)\rvert}{\text{se}(\hat{\theta}_3)}$
$N = 500$	0.036 (0.182)	0.241 (0.268)	0.135 (0.308)
$N = 1000$	0.085 (0.288)	0.208 (0.279)	0.188 (0.336)
$N = 2000$	0.098 (0.387)	0.234 (0.277)	0.161 (0.365)
$N = 4000$	0.085 (0.466)	0.205 (0.355)	0.181 (0.429)

Table 4.3: MSE of parameter estimates by sample-GLMLE using sampled network data and by GLMLE and MCMCMLE using full network data.

Full network size	Sample-GLMLE MSE	GLMLE MSE	MCMCMLE MSE
$N = 500$	0.2774	0.2418	12.8713
$N = 1000$	0.3595	0.1345	7.8875
$N = 2000$	0.4500	0.1275	7.0724
$N = 4000$	0.6092	0.1023	6.3936

is positively correlated with the proportion $\frac{n}{N}$ of information captured by samples, who decreases as N increases.

It is also reasonable to find that sample-GLMLE performs slightly worse than GLMLE according to the values of MSE, primarily because the former uses only sampled data while the latter is applied on full network data. Surprisingly, our sample-GLMLE still outperforms the MCMC-based algorithm significantly, even though the former uses much fewer data (sampled egocentric data).

We also examine how well a graphon function is estimated via sample-GLMLE. Recall that in the evaluation part for GLMLE (Section 3.3), heat maps are used to directly visualize the graphons, such as Figure 3.4, in order to compare the graphons corresponding to GLMLE and MCMCMLE and the truth w^G (graphon representation of the data). However, obtaining w^G from egocentric samples is impossible. To circumvent this issue, we use motif frequencies, that are invariant and constitute intrinsic characteristics of a graphon function.

We compare the probabilities of observing egocentric graph motifs based on observed sample data, the graphon estimated via sample-GLMLE and the theoretical graphon, i.e., the $w_{\boldsymbol{\theta}}$ corresponding to the ERGM with parameters $\boldsymbol{\theta}$ (which is also the graphon used to generate W-random graphs). Explicitly, for observed sample data, the probability of an egocentric motif is just the proportion of its occurrence

in $n = 100$ samples, i.e., $P(M_j^{(2)}|\text{ samples}) = \frac{n_j}{n}$. For others, the probability of an egocentric motif can be calculated via Proposition 4.3.1 using estimated $\hat{w}_m$ and $w_{\boldsymbol{\theta}}$, respectively. All probabilities are averaged over 100 iterations and results are shown in Table 4.4. The probabilities based on sample-GLMLE are close to those obtained from the other two, indicating that the graphon estimated via sample-GLMLE captures the characteristics of sampled data and is close to the theoretical graphon. However, the relative bias of probabilities are larger for large N, which echoes the results before (see Table 4.2 and Table 4.3).

Table 4.4: Probabilities of egocentric graph motifs based on theoretical graphon, observed sample data and estimated graphon via sample-GLMLE.

$N = 500$	Single node	Single edge	Two-star	Triangle
Theoretical	2.065×10^{-4}	1.767×10^{-3}	9.812×10^{-1}	1.683×10^{-2}
Sample data	2.000×10^{-4}	1.200×10^{-3}	9.822×10^{-1}	1.640×10^{-2}
Sample-GLMLE	2.295×10^{-4}	1.700×10^{-3}	9.773×10^{-1}	2.076×10^{-2}
$N = 1000$	Single node	Single edge	Two-star	Triangle
Theoretical	4.190×10^{-8}	7.179×10^{-7}	9.831×10^{-1}	1.686×10^{-2}
Sample data	0	0	9.851×10^{-1}	1.490×10^{-2}
Sample GLMLE	2.941×10^{-5}	2.770×10^{-4}	9.823×10^{-1}	1.750×10^{-2}
$N = 2000$	Single node	Single edge	Two-star	Triangle
Theoretical	1.726×10^{-15}	5.918×10^{-14}	9.831×10^{-1}	1.686×10^{-2}
Sample data	0	0	9.836×10^{-1}	1.640×10^{-2}
Sample GLMLE	1.160×10^{-5}	1.151×10^{-4}	9.823×10^{-1}	1.754×10^{-2}
$N = 4000$	Single node	Single edge	Two-star	Triangle
Theoretical	2.930×10^{-30}	2.009×10^{-28}	9.831×10^{-1}	1.686×10^{-2}
Sample data	0	0	9.853×10^{-1}	1.470×10^{-2}
Sample GLMLE	4.919×10^{-6}	5.251×10^{-5}	9.847×10^{-1}	1.526×10^{-2}

The above simulations examine the performance of sample-GLMLE for different population size N while sample size is fixed. We then conduct a simulation study from another perspective, where we fix N but let n vary, in order to learn how sample size (or the proportion $\frac{n}{N}$) impacts the performance of the estimation procedure. We first generate an ERGM random graphs of size $N = 4000$, via W-random graph generator using $w_{\boldsymbol{\theta}}$. Then we use egocentric nominations to sample from this full network with different sample sizes $n = (100, 200, 500, 1000, 2000)$ and apply sample-GLMLE on these data. All other settings are the same as before ($m = 10$ and the number of iterations is 100).

The performance is measured under the criteria of absolute biases, standard errors and MSE. Results are summarized in Table 4.5, where those of GLMLE based on full network are also included to be compared with. Even though sample-GLMLE uses only sampled data, its MSE is comparable to that of GLMLE for full network, especially when sample size n is large. It is not surprising to find that biases, standard errors and MSE decrease as $\frac{n}{N}$ increases, since more data are input to the estimation procedure.

Table 4.5: Absolute biases, standard errors and MSE of parameter estimates by sample-GLMLE under different settings of sample size n.

Sample size	$\frac{\lvert\text{Bias}(\hat{\theta}_1)\rvert}{\text{se}(\hat{\theta}_1)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_2)\rvert}{\text{se}(\hat{\theta}_2)}$	$\frac{\lvert\text{Bias}(\hat{\theta}_3)\rvert}{\text{se}(\hat{\theta}_3)}$	MSE
$n = 100$	0.126 (0.511)	0.188 (0.374)	0.124 (0.451)	0.6710
$n = 200$	0.105 (0.345)	0.204 (0.285)	0.120 (0.371)	0.4049
$n = 500$	0.051 (0.227)	0.202 (0.200)	0.126 (0.302)	0.2414
$n = 1000$	0.019 (0.144)	0.196 (0.195)	0.109 (0.279)	0.1873
$n = 2000$	0.008 (0.094)	0.208 (0.184)	0.114 (0.265)	0.1692
Full network GLMLE	0.006 (0.043)	0.195 (0.121)	0.091 (0.163)	0.0894

4.4.2 Real data analysis

We illustrate the application of our sample-GLMLE method on sampled network data sets from the National Longitudinal Study of Adolescent Health (AddHealth Study). The widely studied AddHealth data come from a stratified sample of schools in the United States containing students in grades 7-12 (Hunter *et al.* 2008a). In order to collect friendship network data, Addhealth staff conducted surveys in these schools and employed egocentric nomination techniques. More precisely, they first constructed a roster of all students in a school from school administrators and then provided this roster to students and asked them to select up to five close male friends as well as five close female friends. Complete details of this data collection procedure have been given by Resnick *et al.* (1997) and Udry and Bearman (1998). The full data set contains 86 schools, $90,118$ students questionnaires and $578,594$ friendship nominations. However, the full AddHealth network data set is a restricted-use set and is not available to the public. On the other hand, there is a public-use data set containing information of only $6,504$ students, with some summarizing statistics (e.g. the size and the total number of edges) of egocentric networks provided. Therefore, this public-use data set can be regarded as a sampled network data, making it a perfect example to illustrate the usefulness of our sample-GLMLE algorithm as discussed at the beginning of this chapter (see Section 4.1), i.e., accessibility of data and the popularity of egocentric sampling techniques.

The network data on friendships that we study in this section was collected during the first wave (1994-1995) of AddHealth. Though it is a directed network, we ignore the directness of edges in our analysis. We process the data as follows. We exclude missing data, the students who skipped the nomination step, and consider nominations to be valid only when respondents' names were on the roster. We then set $k = 2$ and obtain egocentric samples, by choosing up to two best friends of each

Table 4.6: Summary statistics of sampled network data from AddHealth Study via egocentric nominations.

		Number of egocentric motifs			
Full network size N	Sample size n	Single node	Edge	Two-star	Triangle
$75,871$	$4,397$	692	309	$2,259$	$1,137$

student as alters. The resulting data is the same as randomly sampling students from schools and asking them to nominate up to two recipients. Some statistics of this egocentric sampled network data are summarized in Table 4.6.

Again, we consider two same ERGMs studied before: model 1 uses the homomorphism densities of edges, two-stars and triangles as model terms (2.4.2), while model 2 uses the homomorphism densities of edges and triangle percents (3.3.1). We fix $m = 10$. The estimated sample-GLMLE are as follows:

Model 1: $(-1.687, -0.776, 0.704)$,

Model 2: $(-1.915, 0.183)$,

and the estimated latent graphons for the sample network data are visualized by the tool of heat map (see Figure 4.3). The heat maps clearly reveal that the in-school friendship network of AddHealth is more dense and clustered than the online social network of Slashdot, studied in Section 3.3.2. This can also be verified by comparing the above values of the fitted sample-GLMLE with GLMLE for Slashdot. The estimated coefficients of the homomorphism density of edges have smaller values in both models, indicating a higher overall edge density, while the larger value of estimate associated with triangle density in model 2 implies the higher transitivity of the AddHealth network. One explanation of this observation is that students are more likely to befriend with each other in school because they know each other well. And it is reasonable that strong social circles are easier to be formed in school,

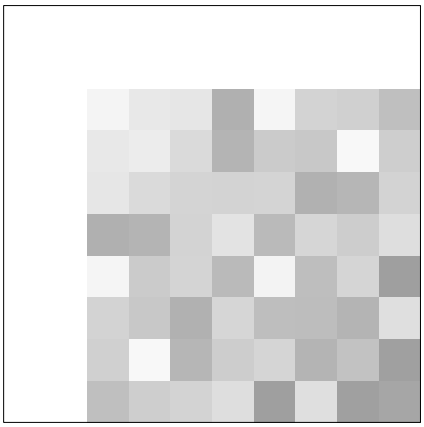

Figure 4.3: Heat map of the estimated graphon function underlying sample network data from AddHealth. The different shades of gray represent the values of w, with darker color for larger value, as shown in the legend of Figure 3.4.

such as students in the same class or in the same sport team. On the other hand, online users are anonymous and are not very familiar with each other, resulting in a weaker tendency to befriend with others or to form social circles.

The running time for the estimation of ERGM model 1 via sample-GLMLE on a 2.66 GHz processor is 756 seconds, a little longer than the total running time of fitting the same model on Slashdot data via GLMLE. This may be due to the fact that the likelihood function for sampled data does not explicitly contain ERGM parameters but is connected to them indirectly via a latent graphon function. Therefore, there is no optimal direction for updating $\boldsymbol{\theta}$, while GLMLE algorithm has the gradient direction, making the likelihood reach maximum faster.

4.5 Application: likelihood ratio test using sampled data

In Section 3.4, we illustrate the application of GLMLE on the likelihood ratio test (LRT) on ERGMs, which is feasible primarily because likelihood functions

can be approximated resorting to graphons estimated by the GLMLE algorithm. Similarly, sample-GLMLE also provides an approximation of the likelihood function of sampled data, making LRT for parameter estimates possible. In this section, we present how LRT can be conducted for ERGMs using egocentric samples.

The test statistic of LRT is twice the difference in two log-likelihoods, which expresses how many times the data are more likely to be fitted under the full model than the nested one, i.e.,

$$\begin{aligned} D &= -2\log\left(\frac{\text{likelihood for null model}}{\text{likelihood for full model}}\right) \\ &= -2\log L\big(w;\{G_i^{(k)}\}_{i=1}^n, N\big|\mathcal{H}_0\big) + 2\log L\big(w;\{G_i^{(k)}\}_{i=1}^n, N\big|\mathcal{H}_a\big). \quad (4.5.1) \end{aligned}$$

Note that the likelihood function here $L\big(w;\{G_i^{(k)}\}_{i=1}^n, N\big)$ is not the same as (3.4.1) for GLMLE — the likelihood of the ERGM model. Instead, it is the product of probabilities of egocentric samples, resulting in the distribution of the LRT test statistic unknown and more complicated than that under a full network setting.

4.5.1 LRT on two different hypothesis tests

In parallel to those for GLMLE in Section 3.4, we begin by studying two different likelihood ratio tests.

The first one considers an ERGM with model terms of the homomorphism densities of edges and two-stars, the same as (3.4.2). And the hypotheses are

$$\begin{aligned} &\mathcal{H}_0 : \theta_2 = 0, \\ &\mathcal{H}_a : \theta_2 \neq 0, \end{aligned} \quad (4.5.2)$$

testing whether or not the corresponding parameter for the homomorphism density of two-stars is significant.

We generate an ERGM random graph of size N under the null model, i.e. $\theta_2 = 0$, and obtain $n = 100$ egocentric samples from it. Applying sample-GLMLE gives

us the approximated likelihood under two models so that the value of the LRT test statistic D can be calculated. We iterate the whole process 100 times in order to reveal the distribution of D under the null hypothesis $\mathcal{H}_0$. Moreover, different values of $N = (500, 1000, 2000, 4000)$ are taken into consideration such that we are able to learn the influence of full network size on the test statistic.

The boxplots in Figure 4.4 implies more variabilities on the LRT test statistic D as network size N increases. However, different from the same plot for GLMLE (Figure 3.5), the boxplots for sample-GLMLE show that the values of D heavily concentrate around 0, indicating the approximated values of likelihood are the same under the null and alternative hypotheses. This observation may be due to the fact that sampled-GLMLE is not fitting ERGMs on the entire network data but only on a small portion of it, where only some local information is characterized. Thus the likelihood function may have already reached its maximum with the estimated parameters of the nested ERGM model. More data may be needed to distinguish the difference between the likelihoods of the full model and the nested one.

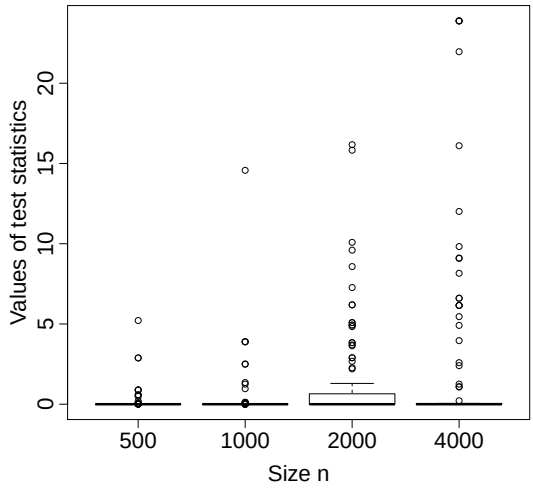

Figure 4.4: Boxplots of LRT test statistics for the hypothesis testing (4.5.2) via sample-GLMLE under different settings of network size N.

The second likelihood ratio test, in parallel to that for GLMLE in Section 3.4, considers an ERGM with the homomorphism densities of edges, two-stars and triangles as model terms. We fit this same ERGM on two sets of egocentric samples from ERGM networks generated under the null model. Likelihood ratio test can then be exploited to test whether the parameters are the same or not, so that we can find out whether these data are from networks whose underlying ERGM models are the same or not. The hypotheses are:

$$\begin{aligned} &\mathcal{H}_0 : \theta_1 = \theta_1', \theta_2 = \theta_2', \theta_3 = \theta_3', \\ &\mathcal{H}_a : \theta_1 \neq \theta_1', \theta_2 \neq \theta_2', \theta_3 \neq \theta_3'. \end{aligned} \tag{4.5.3}$$

The same as before, in order to have a general idea about distributions, we plot the boxplots of values of the LRT test statistic D (Figure 4.5). Clearly, the distributions of D for different N are not as similar to each other as those for GLMLE shown in Figure 3.7. Two reasons account for this observation. Firstly, the likelihood function $L\big(w; \{G_i^{(k)}\}_{i=1}^n, N\big)$ does not contain parameters $\boldsymbol{\theta}$ directly but

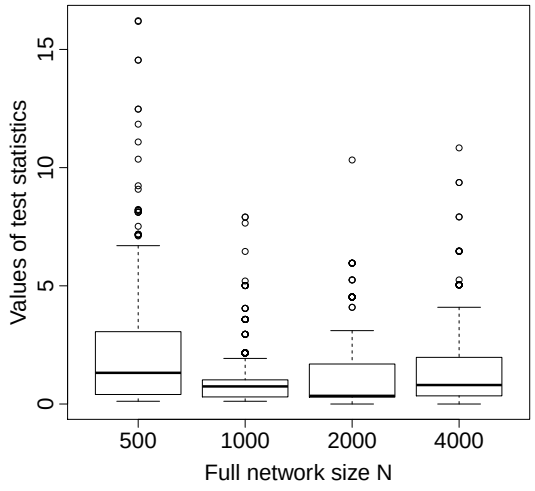

Figure 4.5: Boxplots of LRT test statistics for the hypothesis testing (4.5.3) via sample-GLMLE under different settings of network size N.

instead, is linked to parameters indirectly via the bridge of an underlying graphon. Therefore, the LRT test statistic does not contain parameters in its expression as well and traditional results do not work here. Secondly, recall the discussion for the hypothesis testings for GLMLE, why the distribution of D depends on the network size N is because the degrees of freedom of ERGM terms (or effective sample sizes) are related to N. Thus if we control the ERGM models (terms) to be the same for both null and alternative hypotheses, the distribution of the LRT test statistic should not be impacted by the network size, which is shown in Figure 3.7. But the situation for sample-GLMLE is not exactly the same, since the likelihood function is the product of probabilities of egocentric samples, which depends on $\frac{n}{N}$. Nevertheless, this argument is still partially true here, since we also notice that the distributions of the test statistic are more uniform than those of the first hypothesis testing (Figure 4.4). Using the same ERGM terms for the null and alternative models reduces, to some extend, the influence of N on the distribution of D.

Figure 4.6 provides a closer scrutiny on the distributions of the LRT test statistics for different N via histograms. Though they look similar to the distributions of χ^2 random variables, they are actually not, according to the numerical results in Table 4.7, especially the p-values of the Kolmogorov-Smirnov test. This is not surprising because the likelihood function is not the ordinary likelihood of models

Table 4.7: Mean and variance of test statistic via sample-GLMLE for the hypothesis testing (4.5.3) and the corresponding KS test p-value under different settings of network sizes N.

Network size	$N = 500$	$N = 1000$	$N = 2000$	$N = 4000$
Mean	2.156	0.982	1.085	1.231
Variance	6.057	1.160	1.497	1.837
KS test p-values	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$

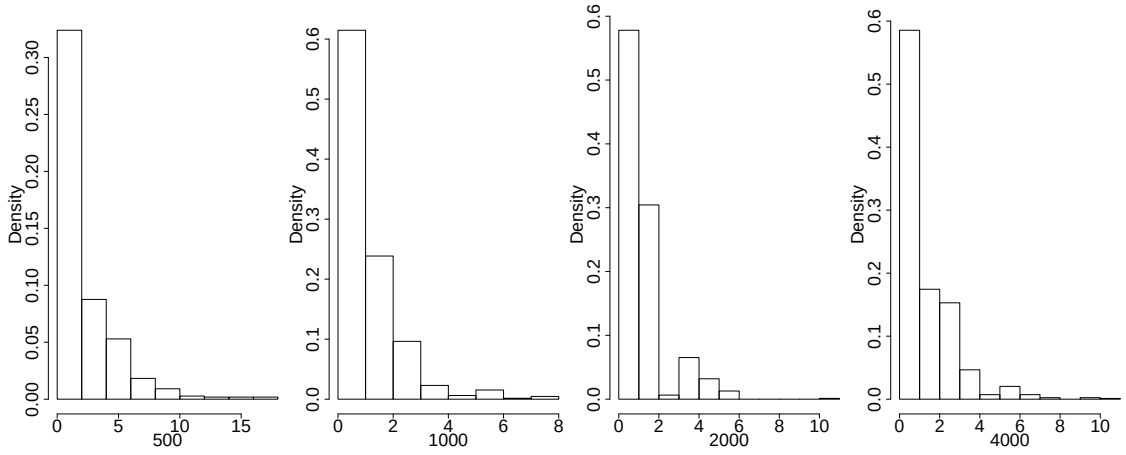

Figure 4.6: Histograms of LRT test statistics for the hypothesis testing (4.5.3) via sample-GLMLE under different settings of network size N.

generally used by LRT, and we should not expect a χ^2 distribution. Again, we observe a high occurrence of D equal to 0, as discussed before.

4.5.2 LRT based on empirical p-values

In parallel to that for GLMLE (see Section 3.4.3), though the unknown distribution of LRT test statistic is much more complicated than that of GLMLE, 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. Note that

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

where D is the test statistic (deviance) defined in (4.5.1) and D_{obs} is the observed value of it. Monte Carlo procedure is used to obtain such empirical p-values. Explicitly, we simulate a large number of full networks from ERGMs in the null. For each network, we obtain egocentric sample motifs and compute the sample-GLMLE under the null model as well as the alternative one in order to calculate D. The whole process gives us an empirical distribution of the test statistic under the null

Table 4.8: Analysis of deviance table for two ERGMs fitted by sample-GLMLE to the egocentric sample network data from AddHealth Study.

Model	log-likelihood	Deviance	Empirical p-value
Model 1			
NULL	−15758.02	—	—
T_1 only	−9035.78	13444.48	< 0.001
T_1 and T_2	−9017.58	36.4	0.003
model 1	−9016.61	1.94	0.021
Model 2			
NULL	−15758.02	—	—
T_1 only	−9035.78	13444.48	< 0.001
model 2	−8534.46	1002.64	< 0.001

hypothesis, so that we can determine the empirical p-value for the observed one, i.e., the percentage in the set of replications that the value of deviance is at least the observed value. We apply this method to the two ERGMs fitted to the AddHealth sample data, as studied in Section 4.4.2, in order to test whether sample-GLMLE of each parameter is statistically significant or not. The results are shown in Table 4.8. The empirical p-values imply that two ERGM models fitted to the AddHealth sample data are valid, while the estimated parameter for the homomorphism density of triangles in model 1 is less significant than other parameter estimates.

4.6 Conclusion and discussion

In this chapter, we propose an estimation method for ERGMs using sampled network data, referred to as sample-GLMLE algorithm, by generalizing the GLMLE method proposed in the previous chapter. This extension is primarily due to the fact that there are many practical issues associated with large network analyses

in the real world, such as the accessibility of full network data and the computational cost for analyses. Moreover, our sample-GLMLE is motivated by a popular network sampling technique via surveys, that has already been exploited in the General Social Survey, the AddHealth Survey and etc.

The nature of our estimation algorithm for sampled data is the same as that for full network data, i.e., we treat the graphon function as a bridge that connects data and ERGM models. The practical tactics are also the same as those in the previous chapter, such as the simple function approximation of a graphon. Simulation studies are presented in order to investigate into the performance of our sample-GLMLE as well as the impact of different configurations on its performance. A real data analysis with sampled network data from the AddHealth Study illustrates the application of our algorithm.

Since it is still a graphon-based method, sample-GLMLE has a theoretical limitation of a positive graphon (i.e., dense graph), though this limitation does not hurt the application on the empirical large graphs, as discussed in Section 3.5 for GLMLE. Moreover, our approach is based on an assumption of the independence of sampled ego-networks and more importantly, two asymptotically consistent approximations inherited from GLMLE. Precisely, sample-GLMLE requires the full network size N to be large and sample size n to be relatively small, while simulation results also show that a large value of the proportion $\frac{n}{N}$ improves the estimation performance, simply because there is more information captured by samples. On the other hand, the number of bins, m, for simple function approximation also has an impact on sample-GLMLE, which has been studied in the previous chapter for GLMLE. Throughout this chapter, we fix $m = 10$, though choosing a good value of m that balances the accuracy and computational cost is important for a particular network analysis in practice.

In parallel to the previous chapter, we examine the likelihood ratio test for

parameter estimates of ERGMs obtained via the sample-GLMLE approach. The likelihood function is no longer the likelihood of ERGMs but instead, the joint likelihood for samples, which has a multinomial form. Unlike the LRT for GLMLE, simulations show that the distribution of the test statistic also depends on the sample size. Therefore, the distribution of the LRT test statistic is more complicated and no theoretical results hold. However, the empirical p-values method is still capable of providing an approximated p-value such that the significance of parameters estimated via sample-GLMLE can be checked via likelihood ratio test.

In summary, sample-GLMLE is an alternative estimation procedure to the GLMLE method when only sampled network data are available.

Chapter 5

Graphon-based Likelihood Framework for Network Data with Nodal Attributes

In this chapter, we develop a graphon-based framework for modeling large networks that contain nodal attributes, based on a generalization of two-dimensional graphon functions to high-dimensional. We start with presenting the motivation of this development in Section 5.1. The extensions of W-random graph models and graphons that incorporate nodal impact are included in Section 5.2. Section 5.3 introduces ERGMs from the perspective of generalized graphon and presents some theoretical results. Several inference methods for this framework are proposed in Section 5.4. Section 5.5 demonstrates the advantages of our algorithms with comparisons to the existing estimation method via both simulations and real data examples. We conclude with a discussion in Section 5.6.

5.1 Motivation

In the previous two chapters, we propose computational algorithms for fitting exponential random graph models to network data as well as to egocentric sampled data. The key tool we employ is the graphon function, which serves as a bridge connecting ERGM models and network data, no matter whether the data are full networks or sampled ones. However, recall the definition of graphons introduced in Section 2.3, a graphon function only captures the underlying topology structure of a large network. Limited by this nature of graphons, the previous two chapters only consider statistical inference based on ERGM models that use geometric properties of networks, such as homomorphism densities or clustering coefficient, as model terms. This has undeniably impacted the application of our algorithms on fitting more general ERGMs, since the main advantage of ERGMs is their flexibilities on the choice of model terms — any statistics of networks (such as topology properties, nodal attributes or even edge attributes) can be used.

Correspondingly, the ERGM models studied in the previous two chapters are not suffice to solve many real-world questions on network data, primarily because real-world networks contain much more information than merely geometric characteristics. For instance, one popular question that many social scientists are interested in is that whether or not *homophily* (the tendency for actors to form relationships with similar others) plays a significant role in the formation of friendship among people. In other words, are people more likely to befriend with others of the same sex or not? Are rich people more likely to have rich people in their personal networks/circles? In order to answer this kind of questions, we need to consider nodal effect into network modeling and inference. A generalization of graphons is required in order to take nodal attributes into consideration, so is an extension of our GLM-LE algorithm based on a proposed generalized graphon function. The following sections will show these generalizations.

5.2 W*-random graph models

We develop W*-random graph models, the generalization of W-random graph models.

Recall that W-random graph models are characterized by graphon function $w \in W$, where W is the graphon space and $w(x, y)$ is the probability of having an edge between two nodes with latent coordinates x and y (both taking values in $[0, 1]$) respectively. Because of very weak assumptions about a graphon w (can be any two-dimensional symmetric function), W-random graph models are very flexible, resulting in a large variety of network topologies. On the other hand, as noticed from the model setups, W-random graph models assume conditional independence among edges when w and coordinates $\{x_i, i = 1, \cdots, n\}$ are given, making this class of models actually inhomophilous (exchangeable) random graph models (Diaconis and Janson 2007). However, most real-world networks are homophilous and many questions in network analysis are related to homophily as well. Thus a lack of homophily limits the application of W-random graph models in real data analysis.

Note that this inhomophily comes from two parts:

- $\{x_i\}_{i=1}^n$, serving as "locations" of nodes on the axe of w, is a realization of random samples from $U(0, 1)$.
- Once $\{x_i\}_{i=1}^n$ has been realized and w specified, W-random graph reduces to a set of $\binom{n}{2}$ Bernoulli$(w(x_i, x_j))$ trials that are conditionally independent given $\{x_i\}_{i=1}^n$.

Therefore, breaking either of these two assumptions adds dependence among nodes, leading to a generalized class of models — W*-random graph models. Particularly, when the dependency is associated with nodal information, W*-random graphs can also be homophilous based on nodal attributes. Depending on how we incorporate

nodal dependency, W*-random graph models can be proposed under two different frameworks.

5.2.1 Two general frameworks

Denote a network with nodal attributes as $G = (V, E, \boldsymbol{Z})$, where V is the set of vertices, E represents the set of edges and $\boldsymbol{Z}$ stands for the set of nodal attributes. Again, we only consider simple graphs, that are undirected graphs with no loops or multiple edges. Let w be the underlying graphon function of G and X be the random variables of the latent coordinates of nodes, whose distribution is denoted as F_X. Further, we use $F_{\boldsymbol{Z}}$ to denote the joint distribution of nodal attributes $\boldsymbol{Z}$, which can be binary (e.g. sex, infection status), categorical (e.g. grades in school or a person's race) or continuous (e.g. wealth, GPA). Throughout this chapter, we assume each dimension of nodal variables is independent of other dimensions, i.e.,

$$F_{\boldsymbol{Z}}(\boldsymbol{z}) = \prod_{l=1}^{d} F_{Z^{(l)}}(z^{(l)}),$$

where $Z^{(l)}$ represents the lth dimension of nodal attributes and d is the number of dimensions of $\boldsymbol{Z}$ (there are d different nodal variables for each node). X can be either dependent of $\boldsymbol{Z}$ or not, resulting in two different frameworks of W*-random graph models.

Framework 1 W*-random graph models with a hierarchical structure

$$\begin{aligned} \boldsymbol{Z_i} &\sim F_{\boldsymbol{Z}} \qquad i = 1, \cdots, n \\ X_i | \boldsymbol{Z_i} = \boldsymbol{z_i} &\sim F_X(\cdot | \boldsymbol{z_i}) \\ w(\cdot, \cdot) &\quad \text{is a graphon function} \\ P(e_{ij} = 1) &= w(x_i, x_j) \end{aligned}$$

Recall that W-random graph models assume X to be uniformly distributed on the unit interval, while this restriction on the distribution is not necessary (see

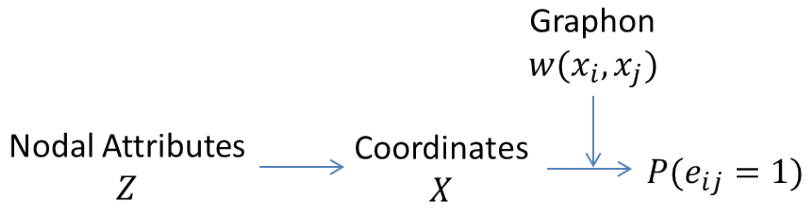

Figure 5.1: Illustration of the framework 1 of W*-random graph models.

Borgs *et al.* 2011). Our first framework assumes X to be dependent on nodal attributes $\boldsymbol{Z}$ and thus the framework itself has a hierarchical structure. The idea of this framework is illustrated in Figure 5.1, where e_{ij} is the indicator of whether nodes i and j are connected. In contrast with the W-random graph model, an exchangeable random graph models, this framework of W*-random graph model brings a new concept of conditional exchangeable random graph models.

Framework 2 W*-random graph models with kernel functions

$$\begin{aligned}
\boldsymbol{Z_i} &\sim F_{\boldsymbol{Z}} \qquad i = 1, \cdots, n \\
X_i &\sim F_X \qquad (\text{Generally, } F_X = U(0,1)) \\
w(\cdot,\cdot) &\quad \text{is a graphon function} \\
k(\cdot,\cdot) &\quad \text{is a similarity kernel function} \\
P(e_{ij}=1) &= k(\boldsymbol{z_i}, \boldsymbol{z_j}) w(x_i, x_j)
\end{aligned}$$

The second framework does not rely on a hierarchical structure but instead on a kernel function that measures the similarity of two nodes. It assumes that X and $\boldsymbol{Z}$ are independent and there is a kernel function $k(\cdot,\cdot)$ serving as weights on w. Without loss of generality, we assume $k(\cdot,\cdot)$ is separable (recall that $Z^{(l)}$ are independent of each other) and has a value in $[0,1]$:

$$\begin{aligned}
&k(\boldsymbol{z_i}, \boldsymbol{z_j}) = \prod_{l=1}^{d} k^{(l)}(z_i^{(l)}, z_j^{(l)}), \\
&k^{(l)}(z_i^{(l)}, z_j^{(l)}) \in [0,1]. \qquad\qquad l = 1, \cdots, d.
\end{aligned} \tag{5.2.1}$$

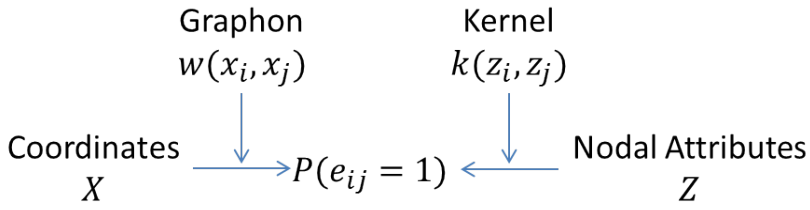

Figure 5.2: Illustration of the framework 2 of W*-random graph models.

Therefore, the probability of having an edge between two nodes is not simply determined by $w(x_i, x_j)$ but combinedly by $w(x_i, x_j)$ and $k(\boldsymbol{z_i}, \boldsymbol{z_j})$, where $k(\boldsymbol{z_i}, \boldsymbol{z_j})$ contains nodal information. Figure 5.2 illustrates this idea.

We give two examples to demonstrate that both proposed frameworks of W*-random graph models are capable of generating homophilous random graphs. Take sex as the only nodal attribute, i.e., $d = 1$, and $Z = Z^{(1)}$ is a vector of n random variables, where $Z_i = \begin{cases} 1 & \text{male} \\ 0 & \text{female} \end{cases}$ for any node $i = 1, 2, \cdots, n$.

Example 5.2.1. Assume Z_i independently and identically follows a Bernoulli distribution with $p = 0.5$. Let $F_X(\cdot|z_i) = \begin{cases} U(0, 0.5) & z_i = 1 \\ U(0.5, 1) & z_i = 0 \end{cases}$ and $w(x_i, x_j) = \begin{cases} 1 & \text{both } x_i, x_j \in (0, 0.5) \text{ or both } x_i, x_j \in (0.5, 1) \\ 0 & \text{elsewhere} \end{cases}$. The w is visualized in the left panel of Figure 5.3. With these configurations, the associated W*-random graph model of the first framework generates an extremely homophilous graph, i.e., people connect completely to others of the same sex but not to ones of a different sex.

Example 5.2.2. Again assume Z_i independently and identically follows a Bernoulli distribution with $p = 0.5$. Let $F_X(\cdot) = U(0, 1)$ and $w(x_i, x_j) = \frac{1}{2}$ for any $x_i, x_j \in (0, 1)$. The w is visualized in the right panel of Figure 5.3. Though the

Figure 5.3: Heat maps of graphons w used in two examples. The left panel is for Example 5.2.1, while the right panel is for Example 5.2.2.

flatness of w indicates an Erdos-Renyi model, if we define $k(z_i, z_j) = \begin{cases} 1 & z_i = z_j \\ 0 & z_i \neq z_j \end{cases}$, the corresponding W*-random graph model of the second framework generates a homophilous random graph where people are connected to others of the same sex with probability 0.5 but in 0 probability to people of an opposite sex.

5.2.2 Connection with stochastic block models

In fact, W-random graph models have a natural connection with stochastic block models. Revealing this connection will help us better understand the intrinsic idea behind the development of W*-random graph models.

The *stochastic block model* (SBM), first introduced by Holland *et al.* (1983) and Wang and Wong (1987), is a popular random graph models in a large variety of domains, including biology, social science and computer science. In its most basic version, the SBM states that each node belongs to a certain class (in finite number) and assumes that the probability of two nodes being connected depends on the classes they belong to. Precisely, suppose n nodes are spread into K groups with proportions $\boldsymbol{\alpha} = (\alpha_1, \cdots \alpha_K)$, i.e., the label z_i (a random number) for node i is

independently and identically drawn from $\{1, \cdots, K\}$ with probability $\boldsymbol{\alpha}$. Then the probability for i and j to be connected is determined by a $K \times K$ block matrix (or connectivity matrix) B, where B_{kl} is the connection probability between a node from group k and a node from group l. Specifically, $P(e_{ij} = 1) = B_{z_i z_j}$.

After a close scrutiny, we may find that SBM is actually equivalent to W-random graph model when w is blockwise constant, with rectangular blocks of size $\alpha_q \times \alpha_l$ and value B_{ql}. More precisely, define a binning function

$$\rho(u) = 1 + \sum_{k=1}^{K} \mathbf{1}(\sigma_k \leq u),$$

where $\mathbf{1}(\cdot)$ is the indicator function and σ_k is the cumulative proportion $\sigma_k = \sum_{j=1}^{k} \alpha_j$. Then the above SBM model corresponds to a W-random graph model with an underlying graphon function

$$w(x, y) = B_{\rho(x),\rho(y)}.$$

Therefore, W-random graph models can be regarded as a generalization of stochastic block models where the "block" matrix is no longer blockwise constant but any measurable function in the graphon space W.

Thus the intuition behind the first framework of W*-random graph model is the same as adding a hyperprior on $\boldsymbol{\alpha}$ in SBM. On the other hand, we develop the second framework by generalizing W-random graph models in the same way that the original SBM is generalized to weighted SBM whose block matrix is multiplied by a weight matrix.

5.2.3 Generalized graphon

Any W-random graph model has an underlying graphon function. Similarly, a W*-random graph model is also fully characterized by a function, to which we refer as

generalized graphon (or generalized graph limit). From this perspective, the proposal of the generalized W*-random graph models is equivalent to the development of a generalization of graphon function.

Use the same notations in Section 5.2.1. Under either framework, denote the $2(d+1)$-dimensional symmetric function

$$w^*(x_i, x_j; \boldsymbol{z_i}, \boldsymbol{z_j}) : [0,1]^2 \times [0,1]^{2d} \to [0,1]$$

as the generalized graphon, where d is the dimension of $\boldsymbol{Z}$, i.e., there are d different nodal attributes (e.g. sex, grade, GPA) for each node. Here, we constrain the range of $z_{il}, l = 1, \cdots, d$ to be $[0,1]$, for the reason of Theorem 5.3.1. This constraint is reasonable since all nodal attributes are finite in the real world (such as sex, GPA, wealth) and a simple normalization guarantees the range of z_{il} to be $[0,1]$. Furthermore, the symmetry of w^* is defined on each dimension of $(X, \boldsymbol{Z})$.

$$\begin{aligned} w^*(x_i, x_j; \boldsymbol{z_i}, \boldsymbol{z_j}) &= w^*(x_j, x_i; \boldsymbol{z_i}, \boldsymbol{z_j}), \\ w^*(x_i, x_j; \boldsymbol{z_i}, \boldsymbol{z_j}) &= w^*(x_i, x_j; z_i^{(1)}, z_j^{(1)}; \ldots; z_j^{(l)}, z_i^{(l)}; \ldots; z_i^{(d)}, z_j^{(d)}) \qquad l = 1, 2, \cdots d. \end{aligned}$$

The intuition here is that we add two dimensions for each nodal variable to graphon, so that the traditional w (without nodal variables) is generalized to higher dimensional symmetric function w^* (with d nodal variables).

With above definitions, it is trivial to conclude that $w^*(x_i, x_j; \boldsymbol{z_i}, \boldsymbol{z_j}) = w(x_i, x_j)$ for the first framework, where the nodal information $\boldsymbol{Z}$ is latent and is incorporated in the distribution of X. In other words, this framework looks more like a W-random graph model with a special distribution on X rather than a generalization, though in fact it is one. Therefore, for the rest of this chapter, we only work on the second framework of W*-random graph models, since the corresponding w^* has an explicit form and its interpretability is better — w^* can be regarded as a mixture of $d+1$ layers of a graph, with one layer containing the latent geometric information captured by $w(\cdot, \cdot)$, and d layers for the nodal attributes characterized by $k(\cdot, \cdot)$.

Similar to the graph limit theory introduced in Section 2.3, for each W*-random graph model (induced by w, k and the set of $\boldsymbol{Z}$), we can define its graphon space W^* and introduce distance as well as equivalent class such that $\widetilde{W^*}$ is a quotient space. Precisely, the generalized cut distance for two functions $f^*, g^* \in W^*$ is defined as

$$d^*_{\square}(f^*, g^*) \stackrel{\text{def}}{=} \sup_{\substack{S_1,S_2\subseteq[0,1]\\ \boldsymbol{Z_1},\boldsymbol{Z_2}\subseteq[0,1]^{2d}}} \left| \int_{S_1\times S_2\times \boldsymbol{Z_1}\times \boldsymbol{Z_2}} \Big[f^*(x_1, x_2, \boldsymbol{z_1}, \boldsymbol{z_2}) - g^*(x_1, x_2, \boldsymbol{z_1}, \boldsymbol{z_2}) \Big] dF_{X,\boldsymbol{Z}}(\boldsymbol{x}, \boldsymbol{z}) \right|.$$

And the equivalence relation is introduced in W^* as $f^* \sim g^*$ if $f^* = g^*_\sigma$, where $g^*_\sigma(x_1, x_2; \boldsymbol{z_1}, \boldsymbol{z_2}) \stackrel{\text{def}}{=} g^*(\sigma x_1, \sigma x_2; \sigma \boldsymbol{z_1}, \sigma \boldsymbol{z_2})$ and σ is a measure preserving bijection. With a quotient map $g^* \to \widetilde{g^*}$, where $\widetilde{g^*}$ is the closure of the equivalent class $\{g^*_\sigma\}$, the space $\widetilde{W^*}$ is a quotient space. Then one can define on $\widetilde{W^*}$ the natural distance $\delta^*_{\square}$ by

$$\delta^*_{\square}(\widetilde{f^*}, \widetilde{g^*}) \stackrel{\text{def}}{=} \inf_{\sigma} d^*_{\square}(f^*, g^*_\sigma) = \inf_{\sigma} d^*_{\square}(f^*_\sigma, g^*)$$

such that $(\widetilde{\mathcal{W}^*}, \delta^*_{\square})$ is a metric space.

Moreover, we can define the homomorphism density of a simple graph H^* (with nodal attributes) in w^*:

$$d(H^*, w^*) = \int_{[0,1]^{(d+1)|V(H^*)|}} \prod_{(i,j)\in E(H^*)} w^*(x_i, x_j, \boldsymbol{z_i}, \boldsymbol{z_j}) dF_{X,\boldsymbol{Z}}(\boldsymbol{x}, \boldsymbol{z}).$$

For the aforementioned reason, we are now only considering W*-random graph models of the second framework. Any graph G^* with nodal attributes (including the above simple graph H^*) can be decomposed into $d+1$ "graphs", one traditional graph G and d nodal attributes "graphs" $G^{(l)}, l = 1, \cdots, d$. Recall that we assume $k(\cdot,\cdot)$ is separable and $\boldsymbol{Z}$ are independent, then each component $k(z^{(l)}_i, z^{(l)}_j)$ is the underlying graphon function of the corresponding $G^{(l)}$. For example, suppose $d = 1$ and the nodal variable is sex. Then the observation of a triangle G^* with 3 male nodes can be thought of observing a triangle without nodal properties (traditional graph G) whose graphon function is $w(x_i, x_j)$ as well as a graph consisting of 3 male

nodes and 3 male-male edges (nodal attributes "graph" $G^{(1)}$) whose underlying graphon is $k(z_i^{(1)}, z_j^{(1)})$.

Motivated by this decomposition, the homomorphism density of a simple graph H^* in a graph G^* can be defined as the product of the homomorphism densities of the decomposed graphs of H^* in the corresponding decomposed graphs of G^*. Precisely,

$$t(H^*, G^*) = t(H, G) \times \prod_{l=1}^{d} t(H^{(l)}, G^{(l)}),$$

where $t(H, G)$ is defined in (2.3.1) and $t(H^{(l)}, G^{(l)})$ is defined similarly. A sequence of graphs $\{G_n^*\}$ is said to converge to a limit object w^* if for every finite simple graph H^*,

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

The following theorem guarantees that the above generalization is valid, in that w^* indeed captures W*-random graph models. The proof is included in Appendix C.1.

Theorem 5.2.1. *The graph sequence $G(n, w^*)$ drawn from W*-random graph models is convergent with probability 1, and its limit is the function w^*.*

5.3 ERGMs with generalized graphon

With the help of the generalized graphon developed in the previous section, we can define ERGMs on the space of $\widetilde{W^*}$ for networks containing nodal attributes.

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

$$p_n^*(G^*) \stackrel{\text{def}}{=} \exp\Big\{n^2\big(T^*(\widetilde{G^*}) - \psi_n^*\big)\Big\}, \tag{5.3.1}$$

where $\widetilde{G^*}$ is the image of G^* in the quotient space $\widetilde{W^*}$, and ψ_n^* is the normalizing constant:

$$\psi_n^* = \frac{1}{n^2} \log \sum_{G^* \in \mathcal{G}_n^*} \exp\{n^2 T^*(\widetilde{G^*})\}.$$

Since a linear combination of continuous function is still continuous, $T^*(\widetilde{G^*})$ can be written as $T^*(\widetilde{G^*}) = \sum_{i=1}^{k} \theta_i T_i^*(\widetilde{G^*})$, where k features $(T_1^*(\widetilde{G^*}), \cdots, T_k^*(\widetilde{G^*}))$ are of interest and the parameters are $(\theta_1, \cdots, \theta_k)$. Here, $T^*(\cdot)$ is also a generalization of $T(\cdot)$. For example, $T^*(\cdot)$ can be the homomorphism density of same-gender edges in G^*. In fact, this new generalized-graphon-based definition of ERGMs (5.3.1), with model terms $T^*(\widetilde{G^*})$, is equivalent to the traditional definition of ERGMs as in (2.1.1), with model terms $U(G^*)$. We provide a list of ERGM statistics for nodal effects in Appendix C.2 in order to illustrate the equivalence between $T^*(\widetilde{G^*})$ and $U(G^*)$ and further, the equivalence of these two different definitions of ERGMs.

The main theoretical results of this chapter are actually the generalizations of the theorems in Chatterjee *et al.* (2013) and Chatterjee and Varadhan (2011), the theorems that lay the theoretical foundations of our GLMLE as mentioned in Chapter 3. In fact, it is not hard to extend their theorems to high-dimensional case of graphon $w^*(x_i, x_j, \boldsymbol{z_i}, \boldsymbol{z_j}) = w(x_i, x_j)k(\boldsymbol{z_i}, \boldsymbol{z_j})$, while the only constraint is that $k(\cdot, \cdot)$ has to take values in $[0, 1]$, which is the reason why we add this constraint in the introduction of kernel functions in (5.2.1).

Define function $I : [0, 1] \to \mathbb{R}$ the same as before

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

and extend I to $\widetilde{W^*}$ in the usual manner:

$$I^*(\widetilde{w^*}) = \int_{[0,1]^{2(d+1)}} I\big(w^*(x_1, x_2, \boldsymbol{z_1}, \boldsymbol{z_2})\big) dx_1 dx_2 d\boldsymbol{z_1} d\boldsymbol{z_2}.$$

We state the theorems without proofs, since the proofs in Chatterjee *et al.* (2013) still hold here with minor modifications.

Theorem 5.3.1. *(Generalization of Theorem 3.1 in Chatterjee* et al. *(2013))* *If $T^* : \widetilde{W^*} \to \mathbb{R}$ is a bounded continuous function and ψ_n^* and I^* are defined as above, then*

$$\lim_{n\to\infty} \psi_n^* = \sup_{\widetilde{w^*}\in\widetilde{W^*}} \left(T^*(\widetilde{w^*}) - I^*(\widetilde{w^*}) \right). \tag{5.3.2}$$

Theorem 5.3.2. *(Generalization of Theorem 3.2 in Chatterjee* et al. *(2013))* *Let $\widetilde{F^*}$ be the subset of $\widetilde{W^*}$ where $T^*(\widetilde{w^*}) - I^*(\widetilde{w^*})$ is maximized. Denote G_n^* as a random graph of size n drawn from the ERGM defined by T^*. Let $\mathbb{P}$ be the probability measure on the underlying probability space on which G_n^* is defined. Then for any $\eta > 0$ there exist $C, \gamma > 0$ such that for any n,*

$$\mathbb{P}\Big(\delta_\square^*\big(\widetilde{G_n^*}, \widetilde{F^*}\big) > \eta \Big) \le Ce^{-n^2\gamma}.$$

Similar to the discussion in Section 2.4, these two theorems guarantee that random graphs drawn from ERGM induced by T^* have an underlying generalized graphon function w^*, where w^* can be obtained by maximizing the right hand side of (5.3.2). This sheds light on the parameter estimation of ERGMs with model terms containing nodal attributes (such as the number of same-gender edges in a graph).

5.4 Model inference

This section presents estimation methods for ERGMs on large networks containing nodal information. We start with proposing a special method for a special case that does not rely on the generalized graphon introduced above. Then we present our GLMLE* method, a generalization of GLMLE to more complex ERGMs resorting to the generalized graphons. We further develop sample-GLMLE* method, an extension of the sample-GLMLE algorithm, or equivalently, an extension of the GLMLE* method to egocentric sample data.

Moreover, without loss of generalization, we assume the dimension of nodal attributes is 1 ($\boldsymbol{Z}$ reduces to Z) and its distribution is Bernoulli, i.e., we only consider one binary nodal attribute such as sex. Precisely, $Z_i = \begin{cases} 1 & \text{male} \\ 0 & \text{female} \end{cases}$ for any node $i = 1, 2, \cdots, n$; $F_Z = \text{Bernoulli}(p)$ where p is the probability for male; and $F_X = U(0, 1)$. Though the above assumptions lead to simpler notations and derivations, all the results in this section can be generalized to higher-dimensional $\boldsymbol{Z}$ as well as more complicated distributions of X and $\boldsymbol{Z}$.

5.4.1 A special case for concordant and discordant graphs

One motivation of taking nodal attributes into consideration is to study the phenomenon of homophily, the tendency of people to be linked with similar others. In the framework of ERGMs, homophily can be examined by comparing the coefficients of models terms such as the number of *concordant* edges and the number of *discordant* edges. Here, concordant means that nodes have the same nodal attributes, while discordant are opposite. Figure 5.4 illustrates these two concepts.

Furthermore, in order to investigate into the impact of homophily, ERGMs should contain model terms for more complicated structure of motifs, such as concordant motifs and discordant ones, so that their coefficients can be compared. Some examples are shown in Figure 5.5.

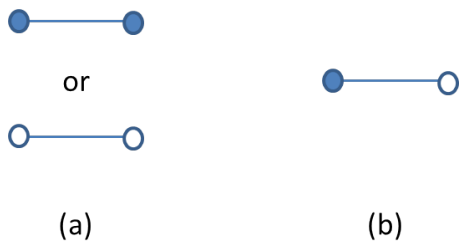

Figure 5.4: Examples of edges with nodal attributes. (a) Concordant edge; (b) discordant edge. Different colors represent different sexes.

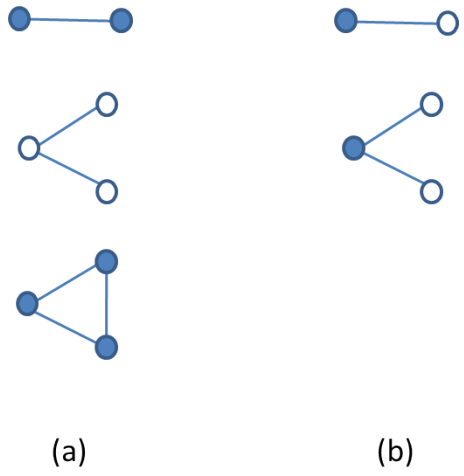

Figure 5.5: Examples of network motifs with nodal attributes. (a) Concordant motifs; (b) discordant motifs. Different colors represent different sexes.

An interesting finding is that the estimation of the ERGM with concordant motifs and discordant motifs is equivalent to the estimation of two ERGMs containing only concordant motifs or discordant motifs. In other words, we can divide a graph into two graphs, one with only concordant edges while the other with only discordant edges.

Let $G^* = (V, E, Z) \in \mathcal{G}^*$ be a full network. Define $G^+ = (V, E^+) \in \mathcal{G}$ as the associated concordant graph whose edges are only concordant, i.e., for any $(i, j) \in E^+$, we have $(i, j) \in E$ and $Z_i = Z_j$. Similarly, define $G^- = (V, E^-) \in \mathcal{G}$ as the discordant graph whose edges are only discordant, i.e., $(i, j) \in E^-$ implies $(i, j) \in E$ and $Z_i \neq Z_j$. Obviously, G^* can be divided into two graphs defined on $\mathcal{G}$,

$$G^* = G^+ + G^-,$$

with $E = E^+ \cup E^-$ and $G_{ij} = G^+_{ij} + G^-_{ij}$ where G_{ij} (or G^+_{ij}, G^-_{ij}) is the $(i, j)th$ entry of adjacency matrix of G^* (or G^+, G^-), respectively.

Let $\boldsymbol{T}^*(H^*, \cdot)$ be the vector of the homomorphism densities of concordant motifs and discordant motifs. $\boldsymbol{T}^*$ induces an exponential random graph model on $\mathcal{G}^*$ and

the probability mass function is defined as:

$$\begin{aligned} P(G^*|\boldsymbol{\theta}) &= \exp\Big\{n^2\big(\boldsymbol{\theta}\boldsymbol{T}^* - \psi_n^*(\boldsymbol{\theta})\big)\Big\} \\ &= \exp\Big\{n^2\big(\boldsymbol{\theta}^+\boldsymbol{T}^+ - \psi_n(\boldsymbol{\theta}^+)\big)\Big\}\exp\Big\{n^2\big(\boldsymbol{\theta}^-\boldsymbol{T}^- - \psi_n(\boldsymbol{\theta}^-)\big)\Big\} \\ &= P(G^+|\boldsymbol{\theta}^+)P(G^-|\boldsymbol{\theta}^-) \end{aligned} \tag{5.4.1}$$

where $\boldsymbol{T}^+$ ($\boldsymbol{T}^-$) is the vector of the homomorphism densities of concordant (discordant) motifs and $\boldsymbol{\theta}^+$ ($\boldsymbol{\theta}^-$) are the related coefficients. A proof of this decomposition is given in Appendix C.3. The decomposition of probability mass function for ERGM indicates that two different ERGMs can be fitted on the concordant graph G^+ and the discordant graph G^- separately in order to compare the estimates of $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$. Therefore, our graphon-based algorithm GLMLE proposed in Chapter 3 can be applied to obtain these estimators, since G^+ (G^-) does not contain nodal attributes, neither do the corresponding ERGM model terms.

5.4.2 GLMLE* algorithm based on generalized graphon

It should be noticed that the method in the previous section can only be used to compare the coefficients of motif densities from concordant and discordant graphs. And the ERGM model considered in the above special case does not include terms for motifs that are neither from a concordant graph nor a discordant graph. Take a triangle that consists of two discordant edges and one concordant edge as an example to illustrate, which is a counter example of the discordant transitivity. Including this model term, however, will help us examine the transitivity phenomenon in a large network. In order to tackle this problem, we rely on the generalized graphon w^* previously developed in Section 5.2.3, as well as the framework of ERGMs based on w^* (see Section 5.3).

Clearly, the GLMLE, based on the simple function approximation, can be easily extended to the estimation of ERGMs via the generalized graphon functions.

Suppose an observed graph G^* has an underlying generalized graphon function w^* (or equivalently, is from a W*-random graph model), which is captured by a latent w and a specified kernel function k, as discussed before. Then the optimization procedure (5.3.2) provides us an estimated $\hat{w^*} = \hat{w} \cdot \hat{k}$ as well as an approximation of the normalizing constant, using the simple function approximation technique. More precisely, suppose the estimated generalized graphon based on a known T^* and $\boldsymbol{\theta}^{(t)}$ is $\hat{w}_m^{*(t)}$, the approximated normalizing constant is

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

Plugging the approximated $\hat{\psi}_n^*$ into (5.3.1) yields the approximated log-likelihood function of $\boldsymbol{\theta}$:

$$\log \hat{p}_n^*(\boldsymbol{\theta}; G^*, \hat{w}_m^{*(t)}) \stackrel{\text{def}}{=} n^2 \Big[T_{\boldsymbol{\theta}}^*(\widetilde{G^*}) - \hat{\psi}_n^*(\boldsymbol{\theta}; \hat{w}_m^{*(t)}) \Big]. \tag{5.4.2}$$

Maximizing $\log \hat{p_n^*}$ gives us $\boldsymbol{\theta}^{(t+1)}$, an update of $\boldsymbol{\theta}$, and a new estimated $\hat{w}_m^*$ can then be obtained based on the updated $\boldsymbol{\theta}^{(t+1)}$. This leads to an iterative procedure of two steps until estimates stabilized. Our proposed algorithm is summarized as follows:

Algorithm 5.1 GLMLE*

1. Give an initial value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(0)}$.
2. For each t,
   (a) Use simple function approximation to estimate $\widetilde{w}^{*(t)}$ by maximizing $T_{\boldsymbol{\theta}^{(t)}}^*(\widetilde{w}^*) - I^*(\widetilde{w}^*)$. The corresponding estimated generalized graphon is $\hat{w}_m^{*(t)}$.
   (b) Set $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \hat{p}_n^*(\boldsymbol{\theta}; G^*, \hat{w}_m^{*(t)})$.
3. Stop once $||\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}||$ converges. And the corresponding $\boldsymbol{\theta}^{(t+1)}$ is the GLMLE*.

It should be noted that the estimation of w^* is not just a simple function estimation procedure for w but is also dependent on the evaluation of k. For example, under the assumption stated at the beginning of this section (Section 5.4), $F_Z = \text{Bernoulli}(p)$. Suppose $k(z_i, z_j) = \begin{cases} 1 & z_i = z_j \\ r & z_i \neq z_j \end{cases}$ is the specified kernel function, which measures the nodal similarity between any two nodes and thus introduces homophily. And smaller r implies stronger homophily. As a result, there are $\frac{1}{2}m(m+1)+2$ parameters to be estimated in order to get $\hat{w}^*_m$. Specifically, $\frac{1}{2}m(m+1)$ for the simple function approximation of $w(x,y) = \sum_{i,j=1}^{m} c_{ij}\mathbf{1}_{A_{ij}}(x,y)$ and 2 parameters (p and r) for the kernel function $k(\cdot,\cdot)$. In this regard, we are more willing to use simpler kernel functions rather than complicated ones, in order to avoid the computational issue and the identifiability issue that may occur in practice.

5.4.3 Extension to sampled data

The GLMLE* algorithm can be further generalized to the situation when only sampled egocentric data are available but not the full network data, in the same way and for the same reason that GLMLE is extended to sample-GLMLE.

Egocentric sampling method lets each respondent name up to k recipients. Again, we specify $k = 2$, and the possible egocentric motifs can only be a single node, an edge, a two-star or a triangle. The algorithm for $k = 2$ can be generalized for larger k (in fact, the Algorithm 5.2 works for any value of k).

In this section, we denote n as the egocentric sample size and N as the full network size. Denote observed egocentric samples with nodal attributes as $G_i^* = G_i^{*(2)} = (G_i, \boldsymbol{V}_{\boldsymbol{Z}i}), i = 1, 2, \ldots n$, where G_i is a simple graph motif, which represents the topology structure of G_i^* such as an edge or a two-star. $\boldsymbol{V}_{\boldsymbol{Z}i}$ is a vector of length $|G_i|$ containing nodal attributes in G_i^*, i.e., $\boldsymbol{V}_{\boldsymbol{Z}i} = (Z_1, \cdots, Z_{|G_i|})$. Suppose

$w^*(\cdot,\cdot)$ is the generalized graphon function. We assume that N is much larger than n, such that egocentric samples can be regarded as independent samples. Then the likelihood function of observed data is as follows:

$$L\big(w^*|\{G_i^*\}_{i=1}^n, N\big) = \prod_{i=1}^{n} P\big(G_i^*|w^*, N\big) \tag{5.4.3}$$

Similar to Proposition 4.3.1, we have the following formulas: (the derivations are included in Appendix C.4)

$$\begin{aligned}
& P(G_i^* \text{ is a single node with } Z_1|w^*, N) \\
= \; & \int_{[0,1]} \left(1 - \int_{[0,1]^2} w^*(x, y, z_1, z_2) dF_X(y) dF_Z(z_2)\right)^{N-1} dF_X(x) \cdot P(Z_1)
\end{aligned}$$

$$\begin{aligned}
& P(G_i^* \text{ is a single edge with} \boldsymbol{V}_{\boldsymbol{Z}1}|w^*, N) \\
= \; & \int_{[0,1]^2} w^*(x_1, x_2, z_1, z_2) \left[1 - \int_{[0,1]^2} w^*(x_1, x_3, z_1, z_3) dF_X(x_3) dF_Z(z_3)\right]^{N-2} \\
& dF_X(x_1) dF_X(x_2) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1})
\end{aligned}$$

$$\begin{aligned}
& P(G_i^* \text{ is a two-star with} \boldsymbol{V}_{\boldsymbol{Z}1}|w^*, N) \\
= \; & \sum_{d=3}^{N} \binom{N-3}{d-3} \int_{[0,1]^3} w^*(x_1, x_2, z_1, z_2) w^*(x_1, x_3, z_1, z_3) \left[1 - w^*(x_2, x_3, z_2, z_3)\right] \\
& \left[\int_{[0,1]^2} w^*(x_1, x_4, z_1, z_4)\right]^{d-3} \left[1 - \int_{[0,1]^2} w^*(x_1, x_N, z_1, z_N) dF_X(x_N) dF_Z(z_N)\right]^{N-d} \\
& dF_X(x_1) dF_X(x_2) dF_X(x_3) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1})
\end{aligned}$$

$$\begin{aligned}
& P(G_i^* \text{ is a triangle with } \boldsymbol{V}_{\boldsymbol{Z}1}|w^*, N) \\
= \; & \sum_{d=3}^{N} \binom{N-3}{d-3} \int_{[0,1]^3} w^*(x_1, x_2, z_1, z_2) w^*(x_1, x_3, z_1, z_3) w^*(x_2, x_3, z_2, z_3) \\
& \left[\int_{[0,1]^2} w^*(x_1, x_4, z_1, z_4)\right]^{d-3} \left[1 - \int_{[0,1]^2} w^*(x_1, x_N, z_1, z_N) dF_X(x_N) dF_Z(z_N)\right]^{N-d} \\
& dF_X(x_1) dF_X(x_2) dF_X(x_3) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1})
\end{aligned}$$

Plugging these probabilities into (5.4.3) provides the value of the likelihood for samples. Therefore, the sample-GLMLE can be generalized with no effort to the following algorithm for fitting ERGMs on sampled data containing nodal information:

Algorithm 5.2 Sample-GLMLE*

Input: ERGM model, the size of the whole network N, raw egocentric sample data $\{G_i^*, i = 1, 2, \cdots n\}$.

1. Categorize $\{G_i^*, i = 1, 2, \cdots n\}$ as distinct nonisomorphic graph motifs $\{M_j^*\}$ and count the corresponding occurrences n_j.

2. Give an initial value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(0)}$.

3. Update $\boldsymbol{\theta}$, until the corresponding approximated generalized graphon maximizes the likelihood of samples $L\big(w^*|\{G_i^*\}_{i=1}^n, N\big) = \prod_j \big[P(M_j^*|\widetilde{w}^*, N)\big]^{n_j}$.

    - For any $\boldsymbol{\theta}^{(t)}$, use simple function technique to estimate $\widetilde{w}^{*(t)}$ by maximizing $T^*_{\boldsymbol{\theta}^{(t)}}(\widetilde{w}^*) - I^*(\widetilde{w}^*)$.

4. Stop once $L\big(w^*|\{G_i^*\}_{i=1}^n, N\big)$ is maximized. And the corresponding $\hat{\boldsymbol{\theta}}$ is the sample-GLMLE*.

Output: the sample-GLMLE* $\hat{\boldsymbol{\theta}}$ based on egocentric sampled data and the corresponding generalized graphon $\hat{w}_m^*$

5.5 Evaluations

In this section, we present the results of examining the performances of the proposed GLMLE* algorithm and the special method in Section 5.4.1 that relies on GLMLE. Again, throughout the whole section, we only consider one binary nodal attribute $Z_i, i = 1, \cdots, n$, such as sex, which follows a Bernoulli distribution with parameter p. Moreover, we assume the kernel function (or the graphon function for nodal

attribute "graph") is

$$k(z_i, z_j) = \begin{cases} 1 & z_i = z_j \\ r & z_i \neq z_j \end{cases}, \tag{5.5.1}$$

which measures the similarity of nodes and the parameter $r \in [0, 1]$ measures the scale of homophily.

5.5.1 Simulation study

In our simulation study, we consider an ERGM with model terms of the homomorphism densities $d(H_i^*, \widetilde{G^*})$ of concordant edges H_1^* and discordant edges H_2^* (see Figure 5.4 for illustrations). This model is equivalent to the ERGM using the homomorphism densities or the numbers (in the traditional definition of ERGM) of edges and concordant edges as statistics under a reparameterization of $\boldsymbol{\theta}$, i.e.,

$$\begin{aligned} T^*(\widetilde{G^*}) &= \theta_1 d(H_1^*, \widetilde{G^*}) + \theta_2 d(H_2^*, \widetilde{G^*}) \\ &= \theta_2 d(\text{edge}, \widetilde{G^*}) + (\theta_1 - \theta_2) d(H_1^*, \widetilde{G^*}) \\ &= \frac{2\theta_2(\#\text{ edges in G})}{n^2} + \frac{2(\theta_1 - \theta_2)(\#\text{ concordant edges in G})}{n^2}. \end{aligned} \tag{5.5.2}$$

Let $p = 0.5$, i.e., people are equally likely to be a male or a female. We specify the true values of the parameters $\boldsymbol{\theta}$ to be $(-1, -2)$, indicating a network with homophily, since the larger value of θ_2 discourages the formation of discordant edges. In fact, the underlying generalized graphon corresponding to the specified values of $\boldsymbol{\theta}$, obtained via (5.3.2), is $w_{\boldsymbol{\theta}}^* = w_{\boldsymbol{\theta}} \cdot k_{\boldsymbol{\theta}}$, where

$$w_{\boldsymbol{\theta}}(x_i, x_j) = 0.1192, \ \forall x_i, x_j \in (0, 1) \qquad \text{and} \qquad k_{\boldsymbol{\theta}}(z_i, z_j) = \begin{cases} 1 & z_i = z_j \\ 0.15 & z_i \neq z_j \end{cases}.$$

Note that $w_{\boldsymbol{\theta}}$ is a constant, due to the fact that the ERGM reduces to an Erdos-Renyi model when only the number of edges as well as weighted edges (such as concordant edges) are used as model terms. However, because of the kernel function

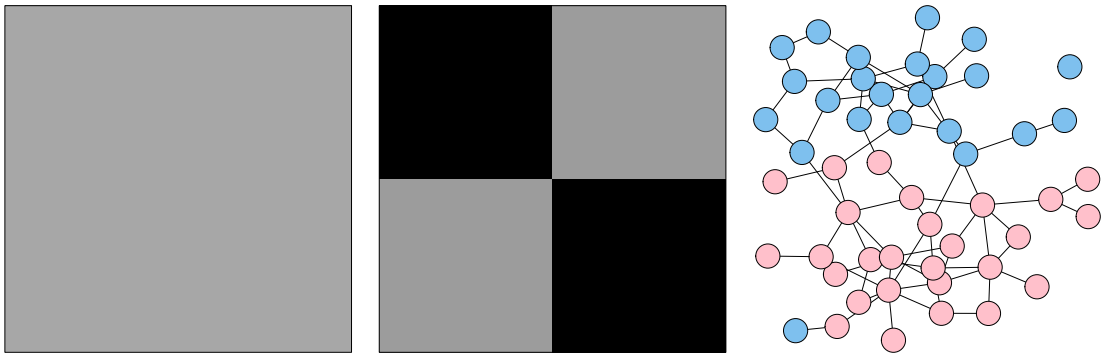

Figure 5.6: Illustration of the ERGM used in the simulation study. The left panel is the visualization of the corresponding graphon. The middle panel is the visualization of the kernel function. The right panel is the plot of a simulated network of size 50 from W*-random graph model, where male nodes are colored blue and female are pink.

of nodal attributes, fluctuations are added such that the probability of two nodes to be connected is not a constant. And $r = 0.15$ indicates that the network drawn from this ERGM model displays the phenomenon of homophily (see Figure 5.6 for an illustration).

We draws random samples from the above ERGM via simulating W*-random graphs $G^*(n, w^*_{\boldsymbol{\theta}})$ of different sizes $n = (100, 200, 500, 1000, 2000, 4000)$. In each case, we generate 100 graphs and apply three algorithms to the data — the GLMLE* algorithm as in Section 5.4.2, the GLMLE for decomposed concordant and discordant graphs as in Section 5.4.1, and the MCMC-based algorithm (`R` function `ergm` from the `ergm` package). Again, we set $m = 10$ for the simple function approximation of graphons though the actual value of m should be 1.

We measure the performances of these three approaches in terms of absolute biases and standard errors of fitted parameters $\hat{\boldsymbol{\theta}}$. Table 5.1 presents the results. Clearly, our graphon-based methods outperform MCMC-based algorithm in both criteria except for small n, such as $n = 100$, in line with the results of the comparison between GLMLE and MCMCMLE in Section 3.3.1. Similarly, the poor

Table 5.1: Absolute biases and standard errors of parameter estimates by GLMLE*, GLMLE on decomposed graphs and MCMCMLE (all values in the table are already multiplied by a factor of 10^3).

	GLMLE*		GLMLE		MCMCMLE	
Size n	$\lvert \text{Bias}(\hat\theta_1) \rvert$ $\text{se}(\hat\theta_1)$	$\lvert \text{Bias}(\hat\theta_2) \rvert$ $\text{se}(\hat\theta_2)$	$\lvert \text{Bias}(\hat\theta_1) \rvert$ $\text{se}(\hat\theta_1)$	$\lvert \text{Bias}(\hat\theta_2) \rvert$ $\text{se}(\hat\theta_2)$	$\lvert \text{Bias}(\hat\theta_1) \rvert$ $\text{se}(\hat\theta_1)$	$\lvert \text{Bias}(\hat\theta_2) \rvert$ $\text{se}(\hat\theta_2)$
100	5.372 (39.340)	14.083 (67.130)	5.372 (39.307)	14.083 (67.233)	8.069 (32.704)	5.436 (52.810)
200	3.605 (18.899)	2.918 (30.515)	3.605 (18.900)	2.918 (30.510)	6.320 (21.096)	5.532 (43.125)
500	0.491 (7.565)	0.231 (9.495)	0.491 (7.564)	0.231 (9.496)	0.599 (9.752)	0.812 (19.765)
1000	0.178 (3.636)	0.400 (4.940)	0.178 (3.636)	0.400 (4.940)	0.722 (4.347)	0.923 (9.382)
2000	0.213 (2.011)	0.255 (2.594)	0.213 (2.011)	0.255 (2.594)	0.491 (2.206)	0.519 (4.474)
4000	0.006 (0.954)	0.099 (1.251)	0.006 (0.954)	0.099 (1.251)	0.130 (1.105)	0.229 (2.205)

performance of graphon-based methods for small n is due to the fact that these algorithms are built upon asymptotic theoretical results. Moreover, Table 5.1 also reveals that, under the specific ERGM with model terms of concordant and discordant edges densities, GLMLE* is equivalent to the special GLMLE method relying on the decomposition of concordant and discordant graphs, since the likelihoods maximized are equivalent as shown in (5.4.1) and the estimation procedures of GLMLE* and GLMLE algorithms are essentially the same. On the other hand, unlike the results in Table 3.2, MCMCMLE performs much better here. This is not surprising, since the ERGM used here is much simpler, only relying on edges or weighted edges (concordant edges) where the underlying geometric graphon is a constant. In other words, the ERGM reduces to an Erdos-Renyi model. With such a simple ERGM model, MCMCMLE performs reasonably well, because there is no degeneracy issue. Furthermore, it should be noted that the values in the table are already multiplied by a factor of 10^3, implying all three methods perform very

well for this specific ERGM model. Particularly, the monotonic decrease of the absolute biases and standard errors when n increases, indicates that the parameter estimators of all three methods are consistent, while the graphon-based methods have a faster rate.

5.5.2 Real data analysis

We apply our GLMLE* method to an in-school friendship network from the AddHealth Study, Wave I (Resnick *et al.* 1997). This network contains data from two school communities, which are large and located in the southern US. According to Handcock *et al.* (2008), the two schools in question (a junior and senior high school in the same community) were combined into a single network dataset. Students who did not take the AddHealth survey or who were not listed on the schools' student rosters were eliminated, and an undirected link was established between any two individuals who both named each other as a friend. This network consists of $1,461$ nodes (students, in this case) and 974 undirected edges (mutual friendships). The vertex attributes are grade (categorical), sex (binary) and race (categorical), though we are only interested in the binary nodal attribute, i.e., sex, and assume it follows a Bernoulli distribution.

We consider an ERGM model using the homomorphism densities of edges, male-male edges and female-female edges as model terms, slightly more complicated than the ERGM examined in the previous section. Specifically,

$$T^*(\widetilde{G^*}) = \theta_1 d(\text{edge}, \widetilde{G^*}) + \theta_2 d(\text{male-male edge}, \widetilde{G^*}) + \theta_3 d(\text{female-female edge}, \widetilde{G^*}).$$

Moreover, the kernel function is different from (5.5.1), where we remove the assumption that the probabilities of same-gender friendships are the same for male

Table 5.2: Estimates by MCMCMLE and GLMLE* applied to an in-school network with gender information from AddHealth Study. The network statistics are averaged numbers of (edges, male-male edges, female-female edges) of simulated networks.

Method	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\frac{1}{n^2}\log(p_n)$	Network statistics
MCMCMLE	-3.766	0.351	0.502	-0.0121	$(967.38, 258.86, 425.75)$
GLMLE*	-3.765	0.350	0.501	-0.0120	$(973.96, 259.50, 430.07)$
Real data					$(974, 259, 430)$

and for female, i.e.

$$k(z_i, z_j) = \begin{cases} 1 & z_i = z_j = 0 \\ r_1 & z_i = z_j = 1 \\ r_2 & z_i \neq z_j \end{cases}.$$

We fit the above ERGM to the network data via the GLMLE* and MCMC-based algorithms in order to compare their performances. Table 5.2 presents the estimation results. The positive values of the estimated coefficients for same-gender edges indicate that the phenomenon of homophily exists in this real-world social network. In fact, our GLMLE* algorithm also provides the estimated $\hat{r}_2 = 0.367$, the ratio of the probability of the formation of a heterophilous tie (a female-male tie) to that of a female-female homophilous tie. The $\hat{r}_2$ quantifies the level of homophily, resulting a better interpretability of the fitted ERGM, while MCMCMLE fails to do so. Moreover, the larger value of $\hat{\theta}_3$ associated with the female-female edge density reveals the phenomenon of high school girls' being more likely to form homophilous ties compared with boys, which can also be verified and quantified by the estimate $\hat{r}_1 = 0.740$.

In addition, the fitted estimators of MCMCMLE and GLMLE* are very close, though the larger value of the log-likelihood for GLMLE* implies the superiority of our method. We further measure the performance of two algorithms under the

criteria of their reconstruction powers, by comparing the averaged network statistics of simulated random graphs with those of the original data. More precisely, we generate 100 random graphs from ERGM with the fitted parameters and obtain network statistics (the number of edges, male-male and female-female edges) for each random graph. W*-random graph models are used to simulate random samples for GLMLE*, while the MCMC-based algorithm (`simulate.ergm` from the `ergm` package) is employed for MCMCMLE. Averaged statistics for 100 samples are presented in Table 5.2. Through the comparison with the statistics of the AddHealth data, we find that the GLMLE* framework produces more similar results than the MCMC-based framework, implying the reconstructed networks via our method are closer to the true network. This, from another perspective, reflects the better performance of GLMLE*, compared with MCMCMLE.

Furthermore, we compare GLMLE* and GLMLE with this data set, in order to investigate into the impact of adding nodal attributes terms to ERGM models. Two different ERGMs are considered. The first ERGM, referred to as model 1, has model terms of the homomorphism densities of edges, two-stars and triangles, the same as (2.4.2), which is studied in Chapter 3 for GLMLE. Another model, referred to as model 2, adds one more term containing gender information, i.e., the homomorphism density of same-gender edges. Precisely, the model 2 is induced by the following T^*:

$$\begin{aligned} T^*(\widetilde{G^*}) &= \theta_1 d(\text{edge}, \widetilde{G^*}) + \theta_2 d(\text{two-star}, \widetilde{G^*}) \\ &\quad + \theta_3 d(\text{triangle}, \widetilde{G^*}) + \theta_4 d(\text{same-gender edge}, \widetilde{G^*}). \end{aligned}$$

The model 1 contains no statistics related to nodal information, on which the GLMLE algorithm can be applied, while GLMLE* method has to be employed for the estimation of the model 2. Results are included in Table 5.3.

Recall that in Figure 3.4, heat maps are used to examine the performance of different estimation procedures by visualizing the corresponding graphons. This

Table 5.3: Estimates by GLMLE and GLMLE* for two ERGMs applied to an in-school network with gender information from AddHealth Study.

Method	$\hat{\boldsymbol{\theta}}$			Corresponding w	
Model 1 with GLMLE	$(-3.496, -1.543, 1.546)$			w_1	
Model 2 with GLMLE*	$(-3.772, -1.507, 1.723, 0.452)$			w_2	
Real data				w^G	
	Average statistics in simulated networks				
Method	H_1	H_2	H_3	H_1^*	H_2^*
Model 1 with GLMLE	973.48	1816.20	163.50	484.42	489.06
Model 2 with GLMLE*	979.13	1841.40	175.26	693.62	285.51
Real data	974	1821	169	689	285

visualization tool is still useful here since the real network has its graphon representation w^G and GLMLE has its associated graphon function w_1, though it does not work for GLMLE*, since the underlying generalized graphon w^* is four-dimensional — two for geometric coordinates and two for nodal attributes. However, w^* can be projected onto the two-dimensional graphon space via marginalization, i.e., $w_2 = \int_{[0,1]^2} w^* dF(\boldsymbol{z})$. In fact, w^G and w_1 can also be regarded as projecting the related w^* onto the two-dimensional space, by incorporating nodal information into the geometric space. Thus w_1, w_2 and w^G are comparable (see Figure 5.7). Both w_1 and w_2 are similar to the data w^G, which means complicated ERGM models are able to capture the topology structure of an observed network, even though nodal attributes are ignored. However, the averaged network statistics of simulated random graphs tell a different story. Though the geometric properties of the random graphs generated from the GLMLE method are very close to the true network (such as number of edges H_1, two-stars H_2 and triangles H_3), nodal information is missed (such as number of same-gender edges H_1^* or cross-gender edges H_2^*), while

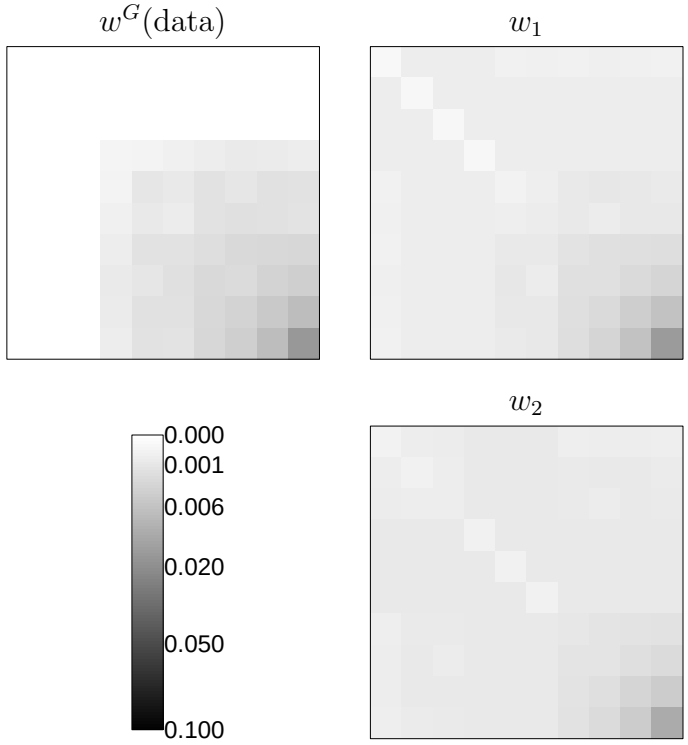

Figure 5.7: Application to the AddHealth network data. Heat maps of the graphon representation of network data w^G, the estimated graphon w_1 via GLMLE, and the projection of estimated graphon w_2 via GLMLE*, as in Table 5.3. The different shades of gray represent the values of $w(x, y) \in [0, 1]$, with black being 0.1 and white 0.

GLMLE*, along with the model 2, successfully captures these nodal characteristics.

In summary, in order to gain a better understanding of the structure of a network with nodal attributes, GLMLE* and ERGMs with node-effect terms are preferable, since they are not only able to characterize topology structures but also capture nodal impact.

5.6 Conclusion and discussion

In this chapter, we develop a complete framework of modeling network data containing nodal information, with the tool of generalized graphon, an extension of the

traditional two-dimensional graphon function to higher-dimensional. A nonparametric model, W*-random graph model, is proposed, with two different general frameworks. The generalized graphon is then defined and is proved to be the latent characteristic of W*-random graph model. It also serves as the bridge that connects network data and exponential random graph models (with node-effect model terms), which can be regarded as an extension to Chatterjee *et al.* (2013)'s work where they consider ERGMs without nodal terms. Furthermore, we propose the GLMLE* estimation method for ERGMs based on the generalized graphons. Simulation studies and real data analyses are presented to compare the performance of this algorithm with that of the MCMC-based algorithm or GLMLE. A generalization of GLMLE*, sample-GLMLE*, is also noted in this chapter for the situation when only sampled data are available, and this extension is done in the same way as how GLMLE is generalized to sample-GLMLE in Chapter 4.

Again, one theoretical limitation of GLMLE* is the positive limiting density, though our method does not suffer from this limitation in real-world applications. On the other hand, two approximations influence the performance of the GLMLE* algorithm. The first one is associated with the evaluation of the likelihood function, as shown in (5.3.2), while the other one evolves the simple function approximation of a graphon function. Simulation results show that the asymptotic results are valid for n as small as 100, by comparing with the non-asymptotic MCMC-based method.

The comparisons with the MCMC-based algorithm using simulations and real data examples indicate that our GLMLE* algorithm is more accurate, under the criteria of absolute biases and standard errors of estimates as well as values of log-likelihood, especially when the network size is large. The GLMLE* is also more computationally efficient for large data. Moreover, the comparison between GLMLE* and GLMLE for different ERGMs shows that adding node-effect terms

to an ERGM may not improve too much on capturing the topology structure of a graph, since graphons are very flexible. But it is able to characterize the nodal attributes' impact on the formation of an edge, which helps us better understand the connectivity of a network.

To summarize, the GLMLE* method is an alternative and, more importantly, superior method to the MCMC-based algorithm for fitting ERGMs on large networks with vertex attributes. It is also an extension of our previously proposed GLMLE algorithm, motivated by the observation that many real-world networks contain nodal information while traditional graphon function is only defined for the geometric properties of a network. Thus addressing this problem benefits our graphon-based methods with broadened applications on many network data in practice and on many popular fields such as link prediction.

Chapter 6

Conclusions and Future Work

This thesis focuses on developing new methods to fit exponential random graph models (ERGMs), one of the most widely used parametric models for networks, on large networks and establishing a general framework to incorporate nodal attributes into modeling. Inspired by the discovery of Chatterjee *et al.* (2013), the emerging tool in graph theory — graphon (or graph limit) — connects ERGMs and large graphs, providing a new way of fitting these traditional models to large networks. Based on this idea, we propose a graphon-based computational estimation algorithm with practical innovations. The base method is then extended to the estimation of ERGMs using sampled egocentric data that contain rich information of locally structured networks. Moreover, we develop W*-random graph models, with a non-parametric framework for networks, to incorporate nodal attributes into modeling. And the proposed generalized graphon functions make it possible to fit ERGMs on large networks containing vertex information. The new models and tools have greatly broadened the application of ERGMs, along with our proposed estimation algorithms, to real-world data of large networks. As shown in many simulations and data applications presented in this thesis, the proposed modeling frameworks and estimation methods are capable of modeling topology structures and nodal effects

as well as improving the goodness of fit of ERGMs, especially as compared to the main competing MCMC-based algorithms that are widely used in the literature.

However, the graphon-based framework for modeling large networks is a relatively new concept and is still in its early stage of development. Some theoretical works are needed for the proposed estimation algorithms based on this framework. There are also many possible ways of improving and generalizing these computational estimation procedures for better practicality, which need further research. Some directions are listed below.

To begin with, no proof is yet available for the consistency and asymptotic normality of the GLMLE or its derivatives, which are intuitively plausible based on the results of simulation studies. Table 3.2, for example, indicates that GLMLE are consistent and Table 3.7 supports the expectation that the estimators are asymptotically normal, otherwise the test statistic will not be χ^2-distributed. On one hand, it is trivial to prove the asymptotic normality of the GLMLE under certain assumptions, such as using number of edges as the model term and assuming there exist no dependence among edges. On the other hand, nonstandard assumptions (lack of independence or using complicated ERGM terms) imply that a proof is quite difficult. The unknown asymptotic distribution of GLMLE renders the determination of the distribution of likelihood ratio test (LRT) test statistic also very hard, though LRT is still feasible using empirical p-value, which is an approximation to the p-value via the Monte Carlo procedures. Another related problem is the evaluation of the variance or standard error of GLMLE. Since graphon-based algorithm is deterministic rather than MCMC-simulation-based, there is no straightforward standard error estimate for GLMLE, while the traditional method of determining the theoretical values of standard errors does not hold, e.g., the inverse of the Fisher information matrix. Therefore, we may need to employ resampling techniques such as bootstrap in order to approximate the variance of GLMLE, which requires

further investigations.

A theoretical limitation of graphon-based method is inherited from the graph limit theory. A graphon function is defined to be a positive limiting function, indicating the graph limit theory works for a sequence of dense graphs. However, this does not limit the application of the graphon tool on empirical large networks, since any finite graph has a positive underlying graphon function. On the other hand, some recent work incorporates sparsity of a network into graphon functions by adding a weight measuring sparseness (see Wolfe and Olhede 2013). This sheds light on the generalization of our proposed methods to the situation when the analysis of a sequence of large sparse networks is needed. For example, understanding the dynamic change or the evolution of a large real-world network may be of interest.

In this thesis, we use simple functions to estimate two-dimensional latent graphon functions, by first dividing the coordinate axis (unit interval) into m bins of equal size and then assuming a constant value on each block. However, the requirement for the bin size to be the same is not necessary and can be relaxed. Recall the connection between graphon function and the block matrix in stochastic block models, it is reasonable that different groups have different sizes. Thus, the simple function approximation for graphons (Algorithm 3.1) can be generalized by adding weights on the sizes of bins, leading to $m - 1$ more parameters to be estimated. Moreover, though different methods have been developed in the literature for accurately approximating an underlying graphon function for an observed network, many of which contain smoothing steps, our simple function approximation technique relies on a fairly simple blockwise constant structure without smoothing. This is primarily because we are interested in estimating parameters for ERGMs via solving an optimization problem (2.4.3), which evolves the evaluation of an integral of complicated functions of graphons. If there is no restriction imposed on the structure of graphons, this task is computationally infeasible, while under the

blockwise constant structure, the integral reduce to a summation, whose value is trival to calculate. However, if a flexible functional class can be found for approximating two-dimensional functions such that the computational cost of calculating the corresponding integral is acceptable, they are preferable since the accuracy of the graphon approximation will be improved, which in return, will improve the performance of GLMLE. The generalization of our algorithms in this direction still needs further study.

In addition, the thesis examines several exponential random graph models based on homomorphism densities in the thesis. But more research is needed for applying our algorithms to many other and more general ERGMs, such as the new specifications of ERGMs using alternating k-star, k-triangle, and k-twopath as model terms (Snijders *et al.* 2006). We give the expressions for these model terms in terms of graphons in Appendix A.2.2. Theoretically, our algorithm can be generalized to any ERGM induced by a bounded continuous function defined on the metric space of graphons.

Last but not least, the developments of W*-random graph models and generalized graphons incorporate nodal attributes into modeling. We believe this will open a new field to study. Many existing works for graphons can be extended to this realm, including the theoretical properties of graphons and the estimation of graphons from observed networks, so that graphon-based frameworks are capable of handling many real-world large networks for many complicated applications.

Bibliography

Airoldi, E. M., Costa, T. B. and Chan, S. H. (2013) Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, 692–700.

Besag, J. (1975) Statistical analysis of non-lattice data. *The statistician*, 179–195.

Bhamidi, S., Bresler, G. and Sly, A. (2008) Mixing time of exponential random graphs. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, 803–812. IEEE.

Bickel, P. J. and Chen, A. (2009) A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, **106**, 21068–21073.

Bollobás, B. and Riordan, O. (2011) Sparse graphs: metrics and random models. *Random Structures & Algorithms*, **39**, 1–38.

Borgs, C., Chayes, J., Lovász, L., Sós, V. and Vesztergombi, K. (2011) Limits of randomly grown graph sequences. *European Journal of Combinatorics*, **32**, 985–999.

Borgs, C., Chayes, J. T., Lovász, L., Sós, V. T. and Vesztergombi, K. (2008) Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, **219**, 1801–1851.

Chan, S. H. and Airoldi, E. M. (2014) A consistent histogram estimator for exchangeable graph models. *arXiv preprint arXiv:1402.1888*.

Chatterjee, S., Diaconis, P. *et al.* (2013) Estimating and understanding exponential random graph models. *The Annals of Statistics*, **41**, 2428–2461.

Chatterjee, S. and Varadhan, S. (2011) The large deviation principle for the erdős-rényi random graph. *European Journal of Combinatorics*, **32**, 1000–1017.

Csardi, G. and Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal, Complex Systems*, **1695**, 1–9.

Diaconis, P. and Janson, S. (2007) Graph limits and exchangeable random graphs. Tech. rep., Department of Mathematics, Uppsala University.

Erdős, P. and Rényi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, **5**, 17–61.

Frank, O. and Strauss, D. (1986) Markov graphs. *Journal of the american Statistical association*, **81**, 832–842.

Geyer, C. J. and Thompson, E. A. (1992) Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 657–699.

Gjoka, M., Kurant, M., Butts, C. T. and Markopoulou, A. (2011) Practical recommendations on crawling online social networks. *Selected Areas in Communications, IEEE Journal on*, **29**, 1872–1892.

Goel, S. and Salganik, M. J. (2009) Respondent-driven sampling as markov chain monte carlo. *Statistics in medicine*, **28**, 2202–2229.

Handcock, M. S. and Gile, K. J. (2010) Modeling social networks from sampled data. *The Annals of Applied Statistics*, **4**, 5–25.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M. and Morris, M. (2008) statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, **24**, 1548.

Handcock, M. S., Robins, G., Snijders, T. A., Moody, J. and Besag, J. (2003) Assessing degeneracy in statistical models of social networks. Tech. rep., Working paper.

Hanneke, S., Fu, W. and Xing, E. P. (2010) Discrete temporal models of social networks. *Electronic Journal of Statistics*, **4**, 585–605.

Harary, F. and Palmer, E. M. (2014) *Graphical enumeration*. Elsevier.

Holland, P. W., Laskey, K. B. and Leinhardt, S. (1983) Stochastic blockmodels: First steps. *Social networks*, **5**, 109–137.

Hunter, D. R., Goodreau, S. M. and Handcock, M. S. (2008a) Goodness of fit of social network models. *Journal of the American Statistical Association*, **103**.

Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. and Morris, M. (2008b) ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, **24**, nihpa54860.

Kolaczyk, E. D. and Krivitsky, P. N. (2011) On the question of effective sample size in network modeling. *arXiv preprint arXiv:1112.0840*.

Latouche, P. and Robin, S. (2013) Bayesian model averaging of stochastic block models to estimate the graphon function and motif frequencies in a w-graph model. *arXiv preprint arXiv:1310.6150*.

Leskovec, J., Lang, K. J., Dasgupta, A. and Mahoney, M. W. (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, **6**, 29–123.

Leskovec, J. and Mcauley, J. J. (2012) Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, 539–547.

Liu, J. S. (2008) *Monte Carlo strategies in scientific computing*. Springer Verlag.

Lovász, L. and Szegedy, B. (2006) Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, **96**, 933–957.

McCormick, T. H., He, R., Kolaczyk, E. and Zheng, T. (2012) Surveying hard-to-reach groups through sampled respondents in a social network. *Statistics in Biosciences*, **4**, 177–195.

Michael, R. G. and David, S. J. (1979) Computers and intractability: a guide to the theory of np-completeness. *WH Freeman & Co., San Francisco*.

Nelder, J. A. and Mead, R. (1965) A simplex method for function minimization. *The computer journal*, **7**, 308–313.

Okabayashi, S. and Geyer, C. J. (2012) Long range search for maximum likelihood in exponential families. *Electronic Journal of Statistics*, **6**, 123–147.

Orbanz, P. and Roy, D. M. (2013) Bayesian models of graphs, arrays and other exchangeable random structures. *arXiv preprint arXiv:1312.7857*.

Palla, G., Lovász, L. and Vicsek, T. (2010) Multifractal network generator. *Proceedings of the National Academy of Sciences*, **107**, 7640–7645.

Pattison, P. and Robins, G. (2002) Neighborhood–based models for social networks. *Sociological Methodology*, **32**, 301–337.

Pattison, P. and Wasserman, S. (1999) Logit models and logistic regressions for social networks: Ii. multivariate relations. *British Journal of Mathematical and Statistical Psychology*, **52**, 169–193.

Resnick, M. D., Bearman, P. S., Blum, R. W., Bauman, K. E., Harris, K. M., Jones, J., Tabor, J., Beuhring, T., Sieving, R. E., Shew, M. *et al.* (1997) Protecting adolescents from harm: findings from the national longitudinal study on adolescent health. *Jama*, **278**, 823–832.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *The Annals of Mathematical Statistics*, 400–407.

Robins, G., Pattison, P., Kalish, Y. and Lusher, D. (2007a) An introduction to exponential random graph (p^*) models for social networks. *Social networks*, **29**, 173–191.

Robins, G., Pattison, P. and Wasserman, S. (1999) Logit models and logistic regressions for social networks: Iii. valued relations. *Psychometrika*, **64**, 371–394.

Robins, G., Snijders, T., Wang, P., Handcock, M. and Pattison, P. (2007b) Recent developments in exponential random graph (p^*) models for social networks. *Social networks*, **29**, 192–215.

Salganik, M. J. and Heckathorn, D. D. (2004) Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological methodology*, **34**, 193–240.

Sirken, M. G. (1970) Household surveys with multiplicity. *Journal of the American statistical Association*, **65**, 257–266.

Snijders, T. A. (2002) Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, **3**, 1–40.

Snijders, T. A., Pattison, P. E., Robins, G. L. and Handcock, M. S. (2006) New specifications for exponential random graph models. *Sociological Methodology*, **36**, 99–153.

Strauss, D. and Ikeda, M. (1990) Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, **85**, 204–212.

Team, R. C. (2012) R: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria, 2012.

Udry, J. R. and Bearman, P. S. (1998) New methods for new research on adolescent sexual behavior. In *New Perspectives on Adolescent Risk Behavior*, 241–269. Cambridge University Press.

van Duijn, M. A., Gile, K. J. and Handcock, M. S. (2009) A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, **31**, 52–62.

Wang, Y. J. and Wong, G. Y. (1987) Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, **82**, 8–19.

Wasserman, S. and Pattison, P. (1996) Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp. *Psychometrika*, **61**, 401–425.

Wasserman, S. and Robins, G. (2005) An introduction to random graphs, dependence graphs, and p^*. *Models and methods in social network analysis*, **27**, 148–161.

Wilks, S. S. (1938) The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, **9**, 60–62.

Wolfe, P. J. and Olhede, S. C. (2013) Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*.

Appendix A

Appendix to Chapter 3

A.1 Some probabilities based on graphon w

Throughout this section, we denote $p(x) = \int_{[0,1]} w(x,y)dy$.

A.1.1 Degree distribution

Under the framework of W-random graph models (or graphons), the probability of a node i having a degree of k is

$$\begin{aligned}
& P(\text{degree}(i) = k) \\
= & \int_{[0,1]^n} P(\text{degree}(i) = k | X_1 = x_1, \cdots, X_n = x_n) \prod_{j=1}^{n} p(X_j = x_j) d\boldsymbol{x} \\
= & \int_{[0,1]^n} \sum_{(i_1,\cdots,i_k)\in\{1,\cdots,n\}\setminus\{i\}} \prod_{l=1}^{k} w(x_i, x_{i_l}) \prod_{m\notin(i_1,\cdots,i_k)} (1 - w(x_i, x_m)) d\boldsymbol{x} \\
= & \sum_{(i_1,\cdots,i_k)\in\{1,\cdots,n\}\setminus\{i\}} \int_{[0,1]} \prod_{l=1}^{k} (\int_{[0,1]} w(x_i, x_{i_l}) dx_{i_l}) \prod_{m\notin(i_1,\cdots,i_k)} (1 - \int_{[0,1]} w(x_i, x_m) dx_m) dx_i \\
= & \sum_{(i_1,\cdots,i_k)\in\{1,\cdots,n\}\setminus\{i\}} \int_{[0,1]} p^k(x_i)(1 - p(x_i))^{n-1-k} dx_i \\
= & \int_{[0,1]} \binom{n-1}{k} p^k(x)(1 - p(x))^{n-1-k} dx.
\end{aligned}$$

A.1.2 Probability of graph motifs for arbitrary several nodes

With the help of graphon, we can derive the probability of observing an edge, a two-star or a triangle from a W-random graph.

$$\begin{aligned} p(\text{edge}) &= \int_{[0,1]^2} w(x,y)dxdy \\ p(\text{two-star}) &= \int_{[0,1]^3} w(x,y)w(x,z)dxdydz \\ p(\text{triangle}) &= \int_{[0,1]^3} w(x,y)w(x,z)w(y,z)dxdydz \end{aligned}$$

More generally, for any simple graph H with $V(H) = \{1, 2 \ldots, k\}$, we have

$$p(H) = d(H,w) = \int_{[0,1]^k} \prod_{(i,j)\in E(H)} w(x_i, x_j)dx_1, \cdots, dx_k \tag{A.1.1}$$

where $E(H)$ denotes the edge set of H.

A.1.3 Some conditional probabilities

When it is given that $X_i = x_i$ and node i is connected to j, the conditional density of "position" (latent coordinate) of X_j is

$$\begin{aligned} & p(X_j = x_j | X_{ij} = 1, X_i = x_i) \\ = & \frac{p(X_j = x_j, X_{ij} = 1 | X_i = x_i)}{p(X_{ij} = 1 | X_i = x_i)} \\ = & \frac{p(X_{ij} = 1 | X_i = x_i, X_j = x_j) p(X_j = x_j | X_i = x_i)}{\int_{[0,1]} p(X_{ij} = 1 | X_i = x_i, X_j = x_j) p(X_j = x_j | X_i = x_i) dx_j} \\ = & \frac{w(x_i, x_j)}{\int_{[0,1]} w(x_i, x_j) dx_j} \\ = & \frac{w(x_i, x_j)}{p(x_i)}. \end{aligned}$$

When it is given that $X_i = x_i$ and node i is connected to j and k, the conditional density of the existence of a link between j and k is as follows. This is also the

conditional density for the third tie when a two-star is observed.

$$
\begin{aligned}
& p(X_{jk}=1|X_{ij}=1,X_{ik}=1,X_i=x_i) \\
= & \int_{[0,1]^2} p(X_{jk}=1|X_{ij}=1,X_{ik}=1,X_i=x_i,X_j=x_j,X_k=x_k)\times \\
& p(X_j=x_j,X_k=x_k|X_i=x_i,X_{ij}=1,X_{ik}=1)dx_jdx_k \\
= & \int_{[0,1]^2} w(x_j,x_k)\frac{p(X_{ij}=1,X_i=x_i,X_j=x_j,X_k=x_k|X_{ik}=1)}{p(X_{ij}=1,X_i=x_i|X_{ik}=1)}dx_jdx_k \\
= & \int_{[0,1]^2} w(x_j,x_k)\times \\
& \frac{p(X_{ij}=1|X_i=x_i,X_j=x_j,X_k=x_k,X_{ik}=1)p(X_i=x_i,X_j=x_j,X_k=x_k|X_{ik}=1)dx_jdx_k}{\int_{[0,1]^2} p(X_{ij}=1|X_i=x_i,X_j=x_j,X_k=x_k,X_{ik}=1)p(X_i=x_i,X_j=x_j,X_k=x_k|X_{ik}=1)dx_jdx_k} \\
= & \int_{[0,1]^2} w(x_j,x_k)\frac{w(x_i,x_j)p(X_j=x_j)\frac{w(x_i,x_k)}{p(x_k)}\frac{p(x_k)}{\int_{[0,1]^2}w(x_i,x_k)dx_idx_k}}{\int_{[0,1]^2} w(x_i,x_j)p(X_j=x_j)\frac{w(x_i,x_k)}{p(x_k)}\frac{p(x_k)}{\int_{[0,1]^2}w(x_i,x_k)dx_idx_k}dx_jdx_k}dx_jdx_k \\
= & \int_{[0,1]^2} w(x_j,x_k)\frac{w(x_i,x_j)w(x_i,x_k)}{\int_{[0,1]^2} w(x_i,x_j)w(x_i,x_k)dx_jdx_k}dx_jdx_k \\
= & \int_{[0,1]^2} w(x_j,x_k)\frac{w(x_i,x_j)}{p(x_i)}\frac{w(x_i,x_k)}{p(x_i)}dx_jdx_k
\end{aligned}
$$

The conditional probability of a tie between nodes j and k when it is given that node i is connected to j and k is as follows:

$$
\begin{aligned}
& P(X_{jk}=1|X_{ij}=1,X_{ik}=1) \\
= & \int_{[0,1]^3} p(X_{jk}=1|X_{ij}=1,X_{ik}=1,X_i=x_i,X_j=x_j,X_k=x_k)\times \\
& p(X_i=x_i,X_j=x_j,X_k=x_k|X_{ij}=1,X_{ik}=1)dx_idx_jdx_k \\
= & \int_{[0,1]^3} w(x_j,x_k)\frac{p(X_{ij}=1,X_i=x_i,X_j=x_j,X_k=x_k|X_{ik}=1)}{p(X_{ij}=1|X_{ik}=1)}dx_idx_jdx_k \\
= & \int_{[0,1]^3} w(x_j,x_k)\frac{w(x_i,x_j)w(x_i,x_k)}{\int_{[0,1]^3} w(x_i,x_j)w(x_i,x_k)dx_idx_jdx_k}dx_idx_jdx_k \\
= & \frac{\int_{[0,1]^3} w(x_j,x_k)w(x_i,x_j)w(x_i,x_k)dx_idx_jdx_k}{\int_{[0,1]^3} w(x_i,x_j)w(x_i,x_k)dx_idx_jdx_k}
\end{aligned}
$$

Actually, the above conditional probability can be regarded as the probability of a third tie when a two-star is observed. Thus it can be calculated from another

perspective.

$$\begin{aligned} & P(X_{jk}=1|X_{ij}=1,X_{ik}=1) \\ = & \frac{P(X_{jk}=1,X_{ij}=1,X_{ik}=1)}{P(X_{ij}=1,X_{ik}=1)} \\ = & \frac{p(\text{triangle})}{p(\text{two-star})} \\ = & \frac{\int_{[0,1]^3} w(x_j,x_k)w(x_i,x_j)w(x_i,x_k)dx_i dx_j dx_k}{\int_{[0,1]^3} w(x_i,x_j)w(x_i,x_k)dx_i dx_j dx_k} \end{aligned}$$

A.2 Gradient and Hessian matrix

Denote graphon-based ERGM terms as T_i. In gradient-based method for solving optimization problem, we need to calculate the gradient, the first moment of T_i. The Hessian matrix, i.e., the second moment of T_i and T_j, is also required if we employ optimization algorithm that uses Hessian matrix, such as Newton-Raphson method. Moreover, evaluation of Hessian matrix is also helpful for deriving the confidence interval of estimated GLMLE.

On the other hand, typical choice of T_i is the homomorphism density of simple graph H_i in $\widetilde{G}$, i.e., $T_i = d(H_i, \widetilde{G})$, which is equivalent to $U_i(G)$, the number of H_i in G after a linear transformation. This has been shown, for example, in (2.4.2). Therefore, in this section, we derive the gradient and hessian matrix based on $U_i(G)$, for easier interpretation. We start with some examples, such as commonly used number of edges, two-stars and triangles, and then generalize to arbitrary simple graph motif H_i.

A.2.1 Some examples

A.2.1.1 Number of edges

Note that

$$\begin{aligned} U_1(G) &\stackrel{\text{def}}{=} \{\text{number of edges in } G\} \\ &= \sum_{i=1}^{n-1}\sum_{j>i} G_{ij} \\ &= \sum_{i=1}^{n-1}\sum_{j>i} \mathbf{1}\{(i,j)\in E(G)\} \\ &= \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n} \mathbf{1}\{(i,j)\in E(G)\} \end{aligned}$$

where $\{G_{ij}\}$ is the adjacency matrix of G and $\mathbf{1}(\cdot)$ is the indicator function. Then

$$\begin{aligned} E(U_1) &= E[E(U_1|x_1,\ldots,x_n)] \\ &= E\left[E\Big(\sum_{i=1}^{n-1}\sum_{j>i}\mathbf{1}\{(i,j)\in E(G)\}|x_1,\ldots,x_n\Big)\right] \\ &= \sum_{i=1}^{n-1}\sum_{j>i} E\Big[E\big(\mathbf{1}\{(i,j)\in E(G)\}|x_1,\ldots,x_n\big)\Big] \\ &= \sum_{i=1}^{n-1}\sum_{j>i}\int_{[0,1]^n} w(x_i,x_j)dx_1,\cdots,dx_n \\ &= \sum_{i=1}^{n-1}\sum_{j>i}\int_{[0,1]^2} w(x,y)dxdy \\ &= \binom{n}{2} p(\text{edge}) \end{aligned}$$

When $n \geq 4$, which is obviously true in most networks, the second moment is:

$$\begin{aligned}
& E(U_1^2) \\
= \; & E[E(U_1^2|x_1,\ldots,x_n)] \\
= \; & E\left[E\Big[\Big(\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\mathbf{1}\{(i,j)\in E(G)\}\Big)^2|x_1,\ldots,x_n\Big]\right] \\
= \; & \frac{n(n-1)}{2}\int_{[0,1]^2} w(x,y)dxdy + \frac{n(n-1)(n-2)(n-3)}{4}\left(\int_{[0,1]^2} w(x,y)dxdy\right)^2 + \\
& n(n-1)(n-2)\int_{[0,1]^3} w(x,y)w(x,z)dxdydz \\
= \; & \binom{n}{2}p(\text{edge}) + \frac{4!}{4}\binom{n}{4}[p(\text{edge})]^2 + 3!\binom{n}{3}p(\text{two-star})
\end{aligned}$$

A.2.1.2 Number of two-stars

Similarly, note that

$$\begin{aligned}
U_2(G) \; & \stackrel{\text{def}}{=} \; \{\text{number of two-stars in } G\} \\
& = \; \frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} G_{ij}G_{ik} \\
& = \; \frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}
\end{aligned}$$

Then

$$
\begin{aligned}
E(U_2) &= E[E(U_2|x_1,\ldots,x_n)] \\
&= E\Big[E\Big(\frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n}\mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}|x_1,\ldots,x_n\Big)\Big] \\
&= \frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n}E\Big[E\Big(\mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}|x_1,\ldots,x_n\Big)\Big] \\
&= \frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n}\int_{[0,1]^n} w(x_i,x_j)w(x_i,x_k)dx_1,\cdots,dx_n \\
&= \frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n}\int_{[0,1]^3} w(x,y)w(x,z)dxdydz \\
&= \frac{n(n-1)(n-2)}{2}p(\text{two-star}) \\
&= 3\binom{n}{3}p(\text{two-star})
\end{aligned}
$$

When $n \geq 6$, the second moment is:

$$
\begin{aligned}
& E(U_2^2) \\
= \;& E[E(U_2^2|x_1,\ldots,x_n)] \\
= \;& E\Big[E\Big[\Big(\frac{1}{2}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}\Big)^2|x_1,\ldots,x_n\Big]\Big] \\
= \;& 3\binom{n}{3}\int w(x,y)w(x,z)dxdydz + n(n-1)(n-2)\int w(x,y)w(x,z)w(y,z)dxdydz + \\
& 6\binom{n}{3}\int [w(x_1,x_2)w(x_1,x_3)w(x_1,x_4)+w(x_1,x_2)w(x_1,x_3)w(x_2,x_4)]\,dx_1dx_2dx_3dx_4 + \\
& 6\binom{n}{3}\int \Big[2w(x_1,x_2)w(x_1,x_3)w(x_1,x_4)w(x_2,x_3)+\frac{1}{2}w(x_1,x_2)w(x_1,x_3)w(x_2,x_4)w(x_3,x_4)\Big]\times \\
& dx_1\cdots dx_4 + \\
& 4!\binom{n}{4}\int \Big[\frac{1}{4}w(x_1,x_2)w(x_1,x_3)w(x_1,x_4)w(x_1,x_5)+\frac{1}{2}w(x_1,x_2)w(x_1,x_3)w(x_1,x_4)w(x_4,x_5)+ \\
& \frac{1}{2}w(x_1,x_2)w(x_1,x_3)w(x_2,x_4)w(x_2,x_5)+w(x_1,x_2)w(x_1,x_3)w(x_2,x_4)w(x_4,x_5)\Big]dx_1\cdots dx_5 + \\
& \frac{5!}{4}\binom{n}{5}\Big(\int w(x,y)w(x,z)dxdydz\Big)^2 \\
= \;& 3\binom{n}{3}p(\text{two-star}) + 6\binom{n}{3}p(\text{triangle}) + 4!\binom{n}{4}\Big[p(\text{3-star})+p(H_1)+2p(H_2)+\frac{1}{2}p(H_3)\Big] + \\
& 5!\binom{n}{5}\Big[\frac{1}{4}p(\text{4-star})+\frac{1}{2}p(H_4)+\frac{1}{2}p(H_5)+p(H_6)\Big] + 6!\binom{n}{6}\frac{1}{4}\big[p(\text{two-star})\big]^2
\end{aligned}
$$

A.2.1.3 Number of triangles

Again, note that

$$
\begin{aligned}
U_3(G) &\stackrel{\text{def}}{=} \{\text{number of triangles in } G\} \\
&= \frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} G_{ij}G_{ik}G_{jk} \\
&= \frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}\cdot\mathbf{1}\{(j,k)\in E(G)\}
\end{aligned}
$$

Then

$$
\begin{aligned}
& E(U_3) \\
= \; & E[E(U_3|x_1,\ldots,x_n)] \\
= \; & E\Big[E\Big(\frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}\cdot\mathbf{1}\{(j,k)\in E(G)\}|x_1,\ldots,x_n\Big)\Big] \\
= \; & \frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} E\Big[E\Big(\mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}\cdot\mathbf{1}\{(j,k)\in E(G)\}|x_1,\ldots,x_n\Big)\Big] \\
= \; & \frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \int_{[0,1]^n} w(x_i,x_j)w(x_i,x_k)w(x_j,x_k)dx_1,\cdots,dx_n \\
= \; & \frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \int_{[0,1]^3} w(x,y)w(x,z)w(y,z)dxdydz \\
= \; & \frac{n(n-1)(n-2)}{6}\int_{[0,1]^3} w(x,y)w(x,z)w(y,z)dxdydz \\
= \; & \binom{n}{3}p(\text{triangle})
\end{aligned}
$$

When $n \geq 6$, the second moment of U_3 is:

$$
\begin{aligned}
& E(U_3^2) \\
= \; & E[E(U_3^2|x_1,\ldots,x_n)] \\
= \; & E\Big[E\Big[\Big(\frac{1}{6}\sum_{\substack{i,j,k=1\\ i\neq j\neq k}}^{n} \mathbf{1}\{(i,j)\in E(G)\}\cdot\mathbf{1}\{(i,k)\in E(G)\}\cdot\mathbf{1}\{(j,k)\in E(G)\}\Big)^2|x_1,\ldots,x_n\Big]\Big] \\
= \; & \binom{n}{3}\int w(x,y)w(x,z)w(y,z)dxdydz \\
& +\frac{4!}{2}\binom{n}{4}\int w(x_1,x_2)w(x_1,x_3)w(x_2,x_3)w(x_2,x_4)w(x_3,x_4)dx_1dx_2dx_3dx_4 \\
& +\frac{5!}{4}\binom{n}{5}\int w(x_1,x_2)w(x_1,x_3)w(x_2,x_3)w(x_2,x_4)w(x_2,x_5)w(x_4,x_5)dx_1\cdots dx_5 \\
& +\frac{6!}{36}\binom{n}{6}\Big(\int w(x,y)w(x,z)w(y,z)dxdydz\Big)^2 \\
= \; & \binom{n}{3}p(\text{triangle})+4!\binom{n}{4}\frac{1}{2}p(H_7)+5!\binom{n}{5}\frac{1}{4}p(H_8)+6!\binom{n}{6}\frac{1}{36}[p(\text{triangle})]^2
\end{aligned}
$$

A.2.2 Generalization

Denote $U_i = \{\text{number of } H_i \text{ in } G\}, i = 1, \cdots, k,$ where $(H_1, \cdots, H_k)$ are graph motifs such as edges, two-stars or triangles etc. Then

$$E(U_i) = C_i \binom{n}{m_i} p(H_i),$$

where C_i is a constant depending on the way we count the number of H_i; n is the size of the network G; m_i is the size of graph motif H_i and $p(H_i)$ is defined in (A.1.1).

For any two ERGM graph motif H_1, H_2, denote the set of all possible attached graphs of these two simple graphs as $AG = \{H_{12} : H_{12} = H_1 \oplus H_2\}$. And $\forall H_a \in AG$, define P_a as the mappings such that $P_a(H_1, H_2) = H_a$. Then the second order $E(U_1U_2)$ is a summation of many terms, where each term is the probability of attached graph of H_1 and H_2 multiplying its coefficient. Specifically,

$$E(U_1U_2) = \sum_{H_a \in AG} |P_a| m_a! \binom{n}{m_a} p(H_a),$$

where $|P_a|$ is the number of mappings such that $P_a(H_1, H_2) = H_a$, m_a is the size of H_a.

A.2.2.1 Evaluation of $|P_a|$

There exists a formula that $|P_a|$ satisfies,

$$\sum_{H_a \in AG} |P_a| = \sum_{i=0}^{\min(m_1, m_2)} \binom{m_1}{i} \binom{m_2}{i} i!$$

However, this formula is not suffice to calculate each $|P_a|$. Therefore, we derive a computational algorithm for $|P_a|$:

Algorithm 3.B Computational algorithm for $|P_a|$

1. For any two simple graph motifs H_1 and H_2 with adjacency matrices M_1 and M_2, expand them into $(m_1 + m_2) \times (m_1 + m_2)$ matrices M_1^+ and M_2^+
2. Fix M_1^+, permute M_2^+ such that there are $\sum_{i=0}^{\min(m_1,m_2)} \binom{m_1}{i}\binom{m_2}{i} i!$ different permutations $PM = \{M_{2p}^+ : M_{2p}^+ \text{ is permuted from } H_2^+\}$
3. Attach M_1^+ and M_{2p}^+, such that $AM = \{M_{12}^+ = M_1^+ + M_{2p}^+, \forall M_{2p}^+ \in PM\}$
4. Cluster elements from AM into different graph isomorphism, and the count is $|P_a|$ where the corresponding graph isomorphism is H_a.

A.2.2.2 Evaluation of $p(H_a)$

Though (A.1.1) gives us a formula for $p(H_a)$, it is still difficult to be used in practice because it consists integrals. To address this problem, we use simple function approximation to convert integrals into summations. Then for any H with matrix representation M, where the indices of M are $i_1, \cdots i_k$ and k is the number of nodes in H, we have the following computable formula:

$$p(H) = \sum_{i_1,\cdots i_k=1}^{m} (\frac{1}{m})^k \prod_{\substack{p,q=1 \\ p\leq q}}^{k} w_{i_p,i_q}^{M_{i_p,i_q}}$$

where m is the configuration of simple function approximation. The time complexity of this method is $O(m^k k^2)$.

Moreover, we have derived a simpler method to obtain $p(H)$ for some specific graph motifs, especially the motifs in popular alternating k-triangle, k-two-paths model. Suppose the simple function approximation of w is w_m, which is an $m \times m$

matrix.

$$\begin{aligned} p(\text{k-triangles}) &= \frac{1}{m^{k+2}} \sum_{i,j=1}^{m} (w_m^2)_{ij}^k (w_m)_{ij}, \\ p(\text{k-two-paths}) &= \frac{1}{m^{k+2}} \sum_{i,j=1}^{m} (w_m^2)_{ij}^k. \end{aligned}$$

A.2.3 Gradient and Hessian matrix for ERGMs

ERGM, whose distribution is in exponential families, has log-likelihood

$$\begin{aligned} \log p_n(\boldsymbol{\theta}; G) &= n^2 \Big[\sum_{i=1}^{k} \theta_i T_i(\widetilde{G}) - \psi(\boldsymbol{\theta}) \Big] \\ &= n^2 \Big[\boldsymbol{\theta}' \boldsymbol{T}(\widetilde{G}) - \psi(\boldsymbol{\theta}) \Big], \end{aligned}$$

and a very useful property of exponential family is that, for any $\boldsymbol{\theta}$,

$$E_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big] = \nabla \psi(\boldsymbol{\theta}).$$

$$Var_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big] = \nabla^2 \psi(\boldsymbol{\theta}).$$

Thus the first derivative of the log-likelihood function for an ERGM graph G is

$$\nabla \log p_n(\boldsymbol{\theta}; G) = n^2 \Big\{ \boldsymbol{T}(\widetilde{G}) - E_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big] \Big\},$$

and the second derivative (Hessian matrix) is

$$\nabla^2 \log p_n(\boldsymbol{\theta}; G) = -n^2 Var_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big].$$

Applying the method in Appendix A.2.2, we can obtain $\nabla \log p_n(\boldsymbol{\theta}; G)$ as well as $\nabla^2 \log p_n(\boldsymbol{\theta}; G)$. Note that

$$\Big\{ Var_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big] \Big\}_{ij} = E[T_i T_j] - E[T_i] E[T_j].$$

Then the Fisher information is

$$I \stackrel{\text{def}}{=} -E_{\boldsymbol{\theta}} \nabla^2 \log p_n(\boldsymbol{\theta}; G) = n^2 Var_{\boldsymbol{\theta}} \big[\boldsymbol{T}(\widetilde{G}) \big]$$

and

$$I_{ij} = n^2\big\{E[T_iT_j] - E[T_i]E[T_j]\big\}.$$

Note that maximum likelihood estimators typically satisfy

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \to N(0, I^{-1}),$$

though this has not been proved for our GLMLE. However, asymptotically normality is intuitively plausible based on the results of Table 3.7, otherwise the test statistic will not be χ^2-distributed. Suppose that the above result holds, the convergence in distribution means roughly that the asymptotic distribution of GLMLE $\hat{\boldsymbol{\theta}}$ is $N\big(\boldsymbol{\theta}, (nI)^{-1}\big)$ for large n. Based on this assumption, we can derive the confidence interval of any element of estimated parameters, $\hat{\theta}_i$, for the level α, i.e.,

$$\big[\hat{\theta}_i - z_{\alpha/2}(nI_{ii})^{-1/2}, \quad \hat{\theta}_i + z_{\alpha/2}(nI_{ii})^{-1/2}\big].$$

And the joint $100(1-\alpha)\%$ confidence region for $\boldsymbol{\theta}$ is:

$$\Big\{\boldsymbol{\theta} : (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'(nI)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \le \chi^2_p(1-\alpha)\Big\}$$

Appendix B

Appendix to Chapter 4

B.1 Proof of Proposition 4.3.1

Suppose $V(M) = \{1, 2, \ldots, m\}$ is the set of vertices of M. Without loss of generalization, an ego vertex is always labeled as 1. Let

$$\begin{aligned} m-1 &= \{\text{the degree of the ego vertex in the egocentric motif } M\} \\ &= \sum_{j \in V(M)} I\{M_{1j} = 1\} \end{aligned}$$

where $\{M_{ij}\}$ is the adjacency matrix of M. Obviously, $m - 1 \leq k$. Denote

$$\begin{aligned} A(M) &= \{\text{alters in } M\} \\ &= V(M) \backslash \{1\}. \end{aligned}$$

Denote $\bar{M}$ as the complete graph on $V(M)$ and $p(x) = \int_{[0,1]} w(x, y) dy$.

Under the above notations, if $m-1<k$,

$$
\begin{aligned}
& P(\text{observing } M|w,N) \\
= & \sum_{d=0}^{N-1} P(M|w,N,d_1=d)P(d_1=d|w,N) \\
= & P(M|w,N,d_1=m-1)P(d_1=d|w,N) \\
= & \int_{[0,1]} \frac{\int_{[0,1]^{m-1}} \prod_{(i,j)\in E(M)} w(x_i,x_j) \prod_{(i,j)\in E(\bar{M})\setminus E(M)} [1-w(x_i,x_j)]dx_2\cdots dx_m}{\int_{[0,1]^{m-1}} \prod_{j\in A(M)} w(x_1,x_j)dx_2\cdots dx_m} \times \\
& \binom{N-1}{m-1}[p(x_1)]^{m-1}[1-p(x_1)]^{N-m}dx_1 \\
= & \binom{N-1}{m-1}\int_{[0,1]^m} \left\{ \prod_{(i,j)\in E(M)} w(x_i,x_j) \prod_{(i,j)\in E(\bar{M})\setminus E(M)} [1-w(x_i,x_j)] \right\} \times \\
& [1-p(x_1)]^{N-m}d\boldsymbol{x}.
\end{aligned}
$$

If $m-1=k$,

$$
\begin{aligned}
& P(\text{observing } M|w,N) \\
= & \sum_{d=k}^{N-1} \int_{[0,1]} P(M|w,N,d_1=d,X_1=x_1)P(d_1=d|w,N,X_1=x_1)P(X_1=x_1)dx_1 \\
= & \sum_{d=k}^{N-1} \int_{[0,1]} P(M|\text{a k-star, } w,N,X_1=x_1)P(d_1=d|w,N,X_1=x_1)dx_1 \\
= & \sum_{d=k}^{N-1} \int_{[0,1]} \frac{\int_{[0,1]^{m-1}} \prod_{(i,j)\in E(M)} w(x_i,x_j) \prod_{(i,j)\in E(\bar{M})\setminus E(M)} [1-w(x_i,x_j)]dx_2\cdots dx_m}{\int_{[0,1]^{m-1}} \prod_{j\in A(M)} w(x_1,x_j)dx_2\cdots dx_m} \times \\
& \binom{N-1}{d}[p(x_1)]^{d}[1-p(x_1)]^{N-1-d}dx_1 \\
= & \sum_{d=k}^{N-1} \binom{N-1}{d} \int_{[0,1]^m} \left\{ \prod_{(i,j)\in E(\bar{M})\setminus E(M)} [1-w(x_i,x_j)] \right\} [p(x_1)]^{d-m+1} \times \\
& [1-p(x_1)]^{N-1-d}d\boldsymbol{x}.
\end{aligned}
$$

Appendix C

Appendix to Chapter 5

C.1 Proof of Theorem 5.2.1

Because of the joint independence of the distributions of X and $Z_l, l = 1, \cdots, d$ as well as the separable condition of w^*, we can decompose any graph G^* into $d+1$ layers, denoting as G and $G^{(l)}, l = 1, \cdots, d$, where $w(x_i, x_j)$ and $k^{(l)}(z_i^{(l)}, z_j^{(j)})$ are the corresponding graphons, respectively.

Let H^* be any simple graph of size n_{H^*} and $G(n, w^*)$ be a W*-random graph. According to Theorem 2.5 in Lovász and Szegedy (2006), for every $0 < \varepsilon < 1$, we have

$$P\Big(\big|t(H, G(n, w)) - d(H, w)\big| > \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2 n}{18 n_{H^*}^2}\Big),$$

where $G(n, w)$ is the geometric layer of $G(n, w^*)$. Similarly, we have the same result for nodal attribute graphs, i.e., for any $l = 1, \cdots, d$,

$$P\Big(\big|t(H^{(l)}, G(n, k^{(l)})) - d(H^{(l)}, k^{(l)})\big| > \varepsilon\Big) \leq 2\exp\Big(-\frac{\varepsilon^2 n}{18 n_{H^*}^2}\Big),$$

where $G(n, k^{(l)}) = [G(n, w^*)]^{(l)}$, the lth nodal attribute graph. In other words, $t(H, G(n, w))$ and $t(H^{(l)}, G(n, k^{(l)}))$ converge to $d(H, w)$ and $d(H^{(l)}, k^{(l)})$ in probability, respectively.

Note that if two sequences of random variables are convergent in probability, the product of them converges to the product of their corresponding limits. It is trivial to generalize this result to multiple sequences of random variables from two sequences. Thus, for any $0 < \varepsilon < 1$,

$$\lim_{n\to\infty} P\left(\left|t(H, G(n, w)) \cdot \prod_{l=1}^{d} t(H^{(l)}, G(n, k^{(l)})) - d(H, w) \cdot \prod_{l=1}^{d} d(H^{(l)}, k^{(l)})\right| > \varepsilon\right) = 0. \tag{C.1.1}$$

Recall the definition of the homomorphism density of H^* in G^*,

$$t(H^*, G^*) = t(H, G) \times \prod_{l=1}^{d} t(H^{(l)}, G^{(l)}),$$

and the homomorphism density of H^* in w^*,

$$\begin{aligned} d(H^*, w^*) &= \int_{[0,1]^{(d+1)|V(H^*)|}} \prod_{(i,j)\in E(H^*)} w^*(x_i, x_j, \boldsymbol{z_i}, \boldsymbol{z_j}) dF_{X,\boldsymbol{Z}}(\boldsymbol{x}, \boldsymbol{z}) \\ &= \int_{[0,1]^{|V(H^*)|}} \prod_{(i,j)\in E(H^*)} w(x_i, x_j) dF_X(\boldsymbol{x}) \times \\ &\quad \prod_{l=1}^{d} \int_{[0,1]^{|V(H^*)|}} \prod_{(i,j)\in E(H^*)} k^{(l)}(z_i^{(l)}, z_j^{(l)}) dF_{Z^{(l)}}(\boldsymbol{z^{(l)}}) \\ &= d(H, w) \times \prod_{l=1}^{d} d(H^{(l)}, k^{(l)}). \end{aligned}$$

The equation (C.1.1) is therefore equivalent to

$$\lim_{n\to\infty} P\Big(\big|t(H^*, G(n, w^*)) - d(H^*, w^*)\big| > \varepsilon\Big) = 0,$$

which completes the proof.

C.2 List of ERGM model terms for nodal effects

Here we provide a list of common ERGM model terms (containing nodal attributes) used in `R` function `ergm` from the `ergm` package (Hunter *et al.* 2008b) and the

corresponding statistics $T^*(\widetilde{G^*})$ in terms of w^*. The `attrname` argument used in `ergm` is a character string giving the name of a quantitative attribute in the networks vertex attribute list, which is the same as specifying a nodal attribute $Z^{(l)}$ under our framework. Without loss of generality, we assume $d = 1$, i.e., $\boldsymbol{Z} = Z^{(1)} = Z$ and always set `attrname` $= Z$. Moreover, the `by` argument chooses which factor of the nodal attribute should be considered and we always set `by` $= c$, where c is a possible factor or value of Z. Suppose the network size is n.

- Absolute difference
  - `absdiff(attrname=Z, pow=p)`: the sum of $|Z_i - Z_j|^p$ for all edges (i, j);
  - corresponding w^*-based term:

$$\int w^*(x_i, x_j, z_i, z_j)|z_i - z_j|^p \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Concurrent node count
  - `concurrent(by=c)`: the number of nodes with c with degree 2 or higher;
  - corresponding w^*-based term:

$$n \int P(\text{degree} > 2)\mathbf{1}(z = c) \cdot dF_Z(z).$$

- Degree range
  - `degrange(from=a, to=b, by=c)`: the number of nodes with c with degree ranging in $[a, b)$;
  - corresponding w^*-based term:

$$n \int P(\text{degree} \in [a, b))\mathbf{1}(z = c) \cdot dF_Z(z).$$

- Degree

  - `degree(d, by=c)`: the number of nodes with c of degree d;
  - corresponding w^*-based term:

$$n \int P(\text{degree} = d)\mathbf{1}(z = c) \cdot dF_Z(z).$$

- k-Stars
  - `kstar(k, attrname=Z)`: the number of concordant k-stars;
  - corresponding w^*-based term:

$$\int \prod_{i=2}^{k+1} w^*(x_1, x_i, z_1, z_i)\mathbf{1}(z_1 = \cdots = z_{k+1}) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Main effect of a covariate
  - `nodecov(attrname=Z)`: the sum of Z_i and Z_j for all edges (i, j);
  - corresponding w^*-based term:

$$\int w^*(x_i, x_j, z_i, z_j)(z_i + z_j) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Factor attribute effect
  - `nodefactor(attrname=Z)`: adds multiple network statistics, one for each of the unique values of Z;
  - corresponding w^*-based term for $Z = c$ as an illustration:

$$n \int \mathbf{1}(z = c) \cdot dF_Z(z).$$

- Uniform homophily
  - `nodematch(attrname=Z, diff=FALSE)`: the number of concordant edges;
  - corresponding w^*-based term:

$$\int w^*(x_i, x_j, z_i, z_j)\mathbf{1}(z_i = z_j) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Differential homophily
  - `nodematch(attrname=Z, diff=TRUE)`: adds multiple network statistics about the number of concordant edges, one for each of the unique values of Z;
  - corresponding w^*-based term for $Z = c$ as an illustration:

$$\int w^*(x_i, x_j, z_i, z_j)\mathbf{1}(z_i = z_j = c) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Nodal attribute mixing
  - `nodemix(attrname=Z)`: adds multiple network statistics about the number of edges, one for each possible pairing of attribute values;
  - corresponding w^*-based term for the pairing (c_1, c_2) as an illustration:

$$\int w^*(x_i, x_j, z_i, z_j)\mathbf{1}(z_i = c_1)\mathbf{1}(z_j = c_2) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Number of ties between actors with similar attribute values
  - `smalldiff(attrname=Z, cutoff=p)`: the number of edges with $|Z_i - Z_j| < p$;
  - corresponding w^*-based term:

$$\int w^*(x_i, x_j, z_i, z_j)\mathbf{1}(|z_i - z_j| < p) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}).$$

- Triangles
  - `triangle(attrname=Z)`: the number of concordant triangles;
  - corresponding w^*-based term:

$$\int w^*_{(i,j)} w^*_{(i,k)} w^*_{(j,k)} \mathbf{1}(z_i = z_j = z_k) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}),$$

    where $w^*_{(i,j)} = w^*(x_i, x_j, z_i, z_j)$ for simpler notation.

- Triangle percentage
  - `tripercent(attrname=Z)`: 100 times the ratio of the number of concordant triangles to the number of connected concordant triplets;
  - corresponding w^*-based term:

$$\frac{100 \int w^*_{(i,j)} w^*_{(i,k)} w^*_{(j,k)} \mathbf{1}(z_i = z_j = z_k) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z})}{\int w^*_{(i,j)} w^*_{(i,k)} \mathbf{1}(z_i = z_j = z_k) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z})}.$$

The above list illustrates the equivalence of graphon-based definition of ERGMs as in (5.3.1), with model terms $T^*(\widetilde{G^*})$, and the traditional definition of ERGMs as in (2.1.1), with model terms $U(G^*)$.

C.3 Proof of the decomposition in (5.4.1)

Recall that we assume a binary nodal attribute Z is distributed as Bernoulli(p) and X is from $U(0,1)$. In order to study the concordant and discordant graphs, it is natural to define the kernel function as $k(z_i, z_j) = \begin{cases} 1 & z_i = z_j \\ r & z_i \neq z_j \end{cases}$, since there are only two situations, i.e., either two nodes have the same nodal attribute or different ones. Moreover, for simpler notation, throughout the proof, we drop the tilde symbol from $\widetilde{w^*}$ by assuming w^* is the unique representation of its equivalent class.

We start the proof by first considering two simple model terms for concordant and discordant graphs, i.e., the homomorphism densities of concordant edges and

discordant edges. Precisely,

$$\begin{aligned}
\boldsymbol{\theta T}^* &= \theta^+ d(\text{concordant edge}, w^*) + \theta^- d(\text{discordant edge}, w^*) \\
&= \theta^+ \int w^*(x_1, x_2, z_1, z_2)\mathbf{1}(z_1 = z_2) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) + \\
&\quad \theta^- \int w^*(x_1, x_2, z_1, z_2)\mathbf{1}(z_1 \neq z_2) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) \\
&= \theta^+ \int w^+(x_1, x_2)\mathbf{1}(z_1 = z_2) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) + \\
&\quad \theta^- \int w^-(x_1, x_2)\mathbf{1}(z_1 \neq z_2) \cdot dF_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) \\
&= \theta^+ T^+ + \theta^- T^-,
\end{aligned}$$

where w^+ (w^-) is the underlying graphon function for concordant (discordant) graph, i.e. $\begin{cases} w^+ = w \\ w^- = rw \end{cases}$ according to the specified kernel structure. Moreover, we have

$$\begin{aligned}
I^*(w^*) &= \int I(w^*(x_1, x_2, z_1, z_2))dx_1 dx_2 dz_1 dz_2 \\
&= \frac{1}{2}\int \big[w^* \log w^* + (1 - w^*)\log(1 - w^*)\big]dx_1 dx_2 dz_1 dz_2 \\
&= \frac{1}{2}\int \big[w^+ \log w^+ + (1 - w^+)\log(1 - w^+)\big]d\boldsymbol{x} + \\
&\quad \frac{1}{2}\int \big[w^- \log w^- + (1 - w^-)\log(1 - w^-)\big]d\boldsymbol{x} \\
&= I(w^+) + I(w^-).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\lim_{n\to\infty} \psi_n^*(\boldsymbol{\theta}) &= \sup_{w^* \in W^*} (\boldsymbol{\theta T}^* - I^*(w^*)) \\
&= \sup_{w^* \in W^*} \big(\theta^+ T^+ + \theta^- T^- - I(w^+) - I(w^-)\big) \\
&= \sup_{w^+ \in W} \big(\theta^+ T^+ - I(w^+)\big) + \sup_{w^- \in W} \big(\theta^- T^+ - I(w^-)\big) \\
&= \lim_{n\to\infty} \psi_n(\theta^+) + \lim_{n\to\infty} \psi_n(\theta^-)
\end{aligned}$$

Thus for very large n, $\psi_n^*(\boldsymbol{\theta})$ equals to the sum of $\psi_n(\theta^+)$ and $\psi_n(\theta^-)$ asymptotically and

$$\begin{aligned} P(G^*|\boldsymbol{\theta}) &= \exp\left\{n^2\big(\boldsymbol{\theta}\boldsymbol{T}^* - \psi_n^*(\boldsymbol{\theta})\big)\right\} \\ &= \exp\left\{n^2\big(\boldsymbol{\theta}^+\boldsymbol{T}^+ - \psi_n(\boldsymbol{\theta}^+)\big)\right\}\exp\left\{n^2\big(\boldsymbol{\theta}^-\boldsymbol{T}^- - \psi_n(\boldsymbol{\theta}^-)\big)\right\} \\ &= P(G^+|\boldsymbol{\theta}^+)P(G^-|\boldsymbol{\theta}^-). \end{aligned}$$

It is trivial to generalize the above deduction to complicated motifs on G^+ and G^-, such as concordant and discordant two-stars, which will complete the proof.

C.4 Derivation of the probabilities of egocentric motifs based on generalized graphons for $k = 2$

If an ego does not nominate any alter, then an egocentric motif of a single node is observed. Assume its nodal attribute is Z_1, then the probability is

$$\begin{aligned} & P(G_1^* \text{ is a single node with } Z_1|w^*, N) \\ = \; & P(G_1 \text{ is a single node}, Z_1|w^*, N) \\ = \; & P(G_1 \text{ is a single node } |Z_1, w^*, N)\cdot P(Z_1|w^*, N) \\ = \; & \sum_{d=0}^{N-1} P(G_1 \text{ is a single node } |Z_1, w^*, N, d_1 = d)\cdot P(d_1 = d|Z_1, w^*, N)\cdot P(Z_1) \\ = \; & P(G_1 \text{ is a single node } |Z_1, w^*, N, d_1 = 0)\cdot P(d_1 = 0|Z_1, w^*, N)\cdot P(Z_1) \\ = \; & P(d_1 = 0|Z_1, w^*, N)\cdot P(Z_1) \\ = \; & \int_{[0,1]} \left(1 - \int_{[0,1]^2} w^*(x, y, z_1, z_2)dF_X(y)dF_Z(z_2)\right)^{N-1} dF_X(x)\cdot P(Z_1). \end{aligned}$$

If an ego nominates only one alter, then an egocentric motif of a single edge is observed. Assume nodal attributes of these two nodes are $\boldsymbol{V}_{\boldsymbol{Z}1}$, then the probability

of observing a single edge is

$$
\begin{aligned}
& P(G_1^* \text{ is a single edge with } \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & P(G_1 \text{ is a single edge}, \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & P(G_1 \text{ is a single edge } | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & \int_{[0,1]^{2N-2}} P(G_1 \text{ is a single edge } | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N, \boldsymbol{X} = \boldsymbol{x}, Z_3, \cdots, Z_N) \times \\
& dF_X(\boldsymbol{x}) dF_Z(z_3) \cdots dF_Z(z_N) P(\boldsymbol{V}_{\boldsymbol{Z}1}) \\
= \; & \int_{[0,1]^2} w^*(x_1, x_2, z_1, z_2) \left[1 - \int_{[0,1]^2} w^*(x_1, x_3, z_1, z_3) dF_X(x_3) dF_Z(z_3) \right]^{N-2} \times \\
& dF_X(x_1) dF_X(x_2) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1}).
\end{aligned}
$$

If an ego nominates two alters and these two are not friends of each other, then an egocentric motif of a two-star is observed. The probability of observing such an egocentric motif with $\boldsymbol{V}_{\boldsymbol{Z}1}$ is

$$
\begin{aligned}
& P(G_1^* \text{ is a two-star with } \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & P(G_1 \text{ is a two-star}, \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & P(G_1 \text{ is a two-star } | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & \sum_{d=3}^{N} \int_{[0,1]^N} P(G_1 \text{ is a two-star } | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N, d_1 = d-1, \boldsymbol{X} = \boldsymbol{x}, Z_4, \cdots, Z_N) \times \\
& P(d_1 = d-1 | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N, d_1 = d, \boldsymbol{X} = \boldsymbol{x}, Z_4, \cdots, Z_N) dF_X(\boldsymbol{x}) \times \\
& dF_Z(z_4) \cdots dF_Z(z_N) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1}) \\
= \; & \sum_{d=3}^{N} \int_{[0,1]^N} P(G_1 \text{ is a two-star } | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N, 1 \text{ is connected to 2,3,...,d}, \boldsymbol{X}, Z_4, \cdots, Z_N) \times \\
& \binom{N-3}{d-3} P(1 \text{ is connected to 2,3,...,d} | \boldsymbol{V}_{\boldsymbol{Z}1}, w^*, N, \boldsymbol{X} = \boldsymbol{x}, Z_4, \cdots, Z_N) dF_X(\boldsymbol{x}) \times \\
& dF_Z(z_4) \cdots dF_Z(z_N) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1}) \\
= \; & \sum_{d=3}^{N} \binom{N-3}{d-3} \int_{[0,1]^3} w^*(x_1, x_2, z_1, z_2) w^*(x_1, x_3, z_1, z_3) \left[1 - w^*(x_2, x_3, z_2, z_3) \right] \times \\
& \left[\int_{[0,1]^2} w^*(x_1, x_4, z_1, z_4) \right]^{d-3} \left[1 - \int_{[0,1]^2} w^*(x_1, x_N, z_1, z_N) dF_X(x_N) dF_Z(z_N) \right]^{N-d} \times \\
& dF_X(x_1) dF_X(x_2) dF_X(x_3) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1}).
\end{aligned}
$$

Similarly, if an ego nominates two alters and these two are friends of each other, then an egocentric motif of a triangle is observed. The probability of observing such an egocentric motif with $\boldsymbol{V}_{\boldsymbol{Z}1}$ is

$$
\begin{aligned}
& P(G_1^* \text{ is a triangle with } \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & P(G_1 \text{ is a triangle}, \boldsymbol{V}_{\boldsymbol{Z}1} | w^*, N) \\
= \; & \sum_{d=3}^{N} \binom{N-3}{d-3} \int_{[0,1]^3} w^*(x_1, x_2, z_1, z_2) w^*(x_1, x_3, z_1, z_3) w^*(x_2, x_3, z_2, z_3) \times \\
& \left[\int_{[0,1]^2} w^*(x_1, x_4, z_1, z_4) \right]^{d-3} \left[1 - \int_{[0,1]^2} w^*(x_1, x_N, z_1, z_N) dF_X(x_N) dF_Z(z_N) \right]^{N-d} \times \\
& dF_X(x_1) dF_X(x_2) dF_X(x_3) \cdot P(\boldsymbol{V}_{\boldsymbol{Z}1}).
\end{aligned}
$$