University of Alberta

A simulation-based approach to assess the goodness of fit of Exponential Random Graph Models

by

Yin Li

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Biostatistics

Department of Mathematical and Statistical Science

Fall 2010
Edmonton, Alberta

Examining Committee

Dr. Keumhee Carriere Chough, Mathematical and Statistical Science (Supervisor)

Dr. P. Hooper , Mathematical and Statistical Science.

Dr. S. Sentil, Public Health.

Abstract

Exponential Random Graph Models (ERGMs) have been developed for fitting social network data on both static and dynamic levels. However, the lack of large sample asymptotic properties makes it inadequate in assessing the goodness-of-fit of these ERGMs. Simulation-based goodness-of-fit plots were proposed by Hunter et al (2006), comparing the structured statistics of observed network with those of corresponding simulated networks. In this research, we propose an improved approach to assess the goodness of fit of ERGMs. Our method is shown to improve the existing graphical techniques. We also propose a simulation based test statistic with which the model comparison can be easily achieved.

Keywords: Social Networks, Exponential Random Graph Models, Goodness of fit.

Acknowledgements

I am deeply indebted to my supervisor Professor Keumhee Carriere Chough, for the excellent guidance and great encouragement and generous support.

I am truly grateful to Department of Mathematical & Statistical Sciences, University of Alberta for its support through my master program.

I appreciate to my parents, Xinyuan Li and Hua Yan. Without their constant and selfless love and support, I could not have the opportunity to have and enjoy such joyous and great life in Canada.

Special thanks go to my thesis committee members: Dr. P. Hooper and Dr. S. Sentil for their time spent in revising my thesis.

Last but not least, I thank all my friends, in particular, Xiaoqing Niu, Ying Or and Jian Deng for their invaluable help.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A network is one of the most important forms of dependent data and is especially useful to describe social relationships. A social network is a social structure among individuals according to its specific types of interdependency such as friendship, kinship, and academic paper co-authorship. The application of social network analysis has been emerged to many areas, to name a few, the behavior of epidemics in public health, protein interactions in biological science, and coalition formation dynamics in political science.

In early years of statistical network analysis, researchers mainly dealt with the distribution of various network statistics. In the 1980's, a series of literatures about network modeling random networks allowed deeper exploration in network structures and brought social network analysis into a new page. The attention is mainly focused on statistical inferences on both static and dynamic networks and assessing the goodness of fit of the models.

In 1981, Holland and Leinhardt applied a log-linear statistical model, which is named the p_1 model, to directed networks under dyadic independence assumption, that is the dyads are independent of one another in random networks. Three types of effects which can be evaluated by p_1 model are sender effect (popularity), receiver effect (expansiveness) and dyadic effect (reciprocation).

To extend the p_1 model in terms of the dependency structure in networks, Frank and Strauss (1986) proposed exponential random graph models (ERGMs) under the Markov dependency assumption in random networks. Markov dependence is a more realistic dependence structure in random networks. In Markov graph, two possible network ties are said to be conditionally dependent only when they are connected by a common node. Frank and Strauss proved that the counts of various triangles and k-stars are sufficient statistics for Markov graphs and regarded the count effects as parameters in ERGMs. Modifications of ERGMs are discussed in Snijders (2004) and Hunter (2005), which generalize the dependence assumption and include more structure statistics in ERGMs.

Since the calculation of maximum likelihood estimate involves maximizing the sum of log-likelihood for all possible networks, MLE method is not applicable to large networks. Strauss and Ikeda (1990) introduced pseudo-likelihood estimation for social network models. In their work, it was shown that maximum pseudo-likelihood estimate (MPLE) is equal to MLE for independent network models and MPLE provides a reasonable approximation for estimating ERGMs in the form of logit models.

Due to the lack of good understanding of MPLE and the inefficiency of MPLE in some known cases, MPLE is often used as a starting value in some iterative procedures of estimation, such as Markov Chain Monte Carlo maximum likelihood estimation. Details can be found in Geyer and Thompson (1992) and Snijders (2002).

Since there is no standard large sample asymptotic theory for networks, it is difficult to assess the goodness of fit of the models. Hunter (2006) proposed simulation based goodness-of-fit plots for static ERGMs, comparing the structured statistics of observed network with those of corresponding simulated networks. The idea is that a fitted model should reproduce network statistics similar to the observed one. The choice of the network statistics for constructing these goodness-of-fit plots is made according to the interest of the study.

Our work focuses on the extension of Hunter's Goodness of fit test on static ERGMs. Based on a simulated sample of networks generated from the fitted ERGM, we approximate the distributions of the structure statistics in the random network instead of the distribution of the random network itself. Although the distributions have not been proved theoretically, we obtain the approximate distribution for some of the structure statistics based on simulation. The goodness of fit tests could be improved with this assumption of the distribution.

In section 2 of this paper, we introduce the modeling, the estimation and goodness of fit tests of the ERGMs. From section 3.1 to 3.3, we describe our idea of using the approximate distributions of the network structure statistics to improve the goodness of fit tests. Then, we employ the degree counts, a common type of structure statistics of a random network defined in section 3.2, to indicate the improvement of goodness of fit tests with two examples (section 3.4). The simulations are done in section 3.5 to check the Poisson assumption of the distribution of the degree counts in random networks. Conclusions and further discussions are made in section 4.

Chapter 2

Literature Review

2.A Random graphs and ERGMs

A network consists of vertices and edges. In social networks, the number of vertices is called the network size. The edges, which are also called ties, can either be directed or undirected. A network with directed edges is called a directed network or a directed graph. If a network contains only undirected edges, it is an undirected network or an undirected graph.

To introduce the notation, consider a network with n vertices. Let N be the index set of vertices $\{1,2,\cdots,n\}$. A network G on N is a subset of the set of pairs of elements in N. In other words, G is the set of edges of a network.

For a random network with n vertices, the set of possible edges can be denoted by $G_0 \subseteq N \times N$. G is random and $G \subseteq G_0$.

In order to study the feature of a social network, we tend to generalize it to a random network with the same network size. A random network is obtained by starting with n vertices and adding edges between them at random. Each possible edge in the random network is a random variable, which has a value of 1 if the edge is present in G and 0 otherwise. For example, we can use Y_{ij} to denote the possible edge between node i and node j, where $i < j$ for undirected networks and $i \neq j$ for directed networks. Let Y denote the sequence of the random variables. An instance of an undirected network is $Y = \{Y_{12}, Y_{13}, \cdots, Y_{n-1,n}\}$, and y denote the observation of the random network.

Hence, both Y and G refer to a random network and we have y to denote a realization of the random network in terms of an observation of Y.

2.A.1 Dependence structure

As mentioned before, networks are dependent data. This means that in a general network Y, each variable Y_{ij} is related to the other variables in the network. The modeling of networks is complicated and the computation is burdensome. By defining a dependence structure in Y, it imposes some constraints on the dependence of the network and it simplifies the modeling. The dependence structure of a network indicates which edges in the network are conditionally dependent. Two edges are conditionally dependent if the conditional probability that the edges both are present, given the other edges in the network, does not equal to the product of their marginal conditional probabilities.

Frank and Strauss (1986) proposed a dependence graph to describe the dependence structure in a network. A dependence graph D on a random network G is a fixed graph. The vertices of D are the possible edges in G, and the edges of D are the pairs of edges in G that are conditionally dependent. A clique of D is defined as a nonempty subset A of the possible edges of G such that either A contains only one edge or each pair of edges in A are connected in D, that is, the pairs of edges in A are conditionally dependent.

Bernoulli networks

Bernoulli networks have the simplest dependence structure which is obtained if all edge variables Y_{ij} are mutually independent and follow the Bernoulli(p_{ij}) distributions.

$$Y_{ij} \sim \text{Bernoulli}(p_{ij}), \forall i \neq j \quad . \tag{2.1}$$

The joint probability of the Bernoulli network is,

$$P(Y = y) = \prod_{i \neq j} p_{ij}^{y_{ij}} (1 - p_{ij})^{1 - y_{ij}} \quad . \tag{2.2}$$

Under the Bernoulli dependence assumption, there does not exist any pairs of edges which are conditionally dependent. The dependence graph D contains a collection of isolated vertices, with each vertex corresponds to an edge in the Bernoulli network. In other words, the clique A can only be a single edge in G.

Markov networks

Markov dependence is a more realistic dependence assumption in random networks. When Markov dependence is assumed, the networks are called Markov networks. In Markov networks, two edges are conditionally dependent if and only if they have a vertex in common. The dependence graph D contains edges between a pair of vertices if the vertices denote two edges which are conditionally dependent in G. This implies that the clique A can be either triangles or k-stars in Markov networks.

$$\text{triangles:} \quad A_1 = \{\{i, j\}, \{j, k\}, \{i, k\}\} \quad ,$$

$$\text{k-stars:} \quad A_2 = \{\{i_0, i_1\}, \{i_0, i_2\}, \cdots, \{i_0, i_k\}\} \quad .$$

Note that the clique A can also be a single edge as a 1-star in the Markov network.

More complex dependence assumptions

More complex ERGMs that go beyond Markov random graphs have been developed in Pattison and Robins (2002), Schweinberger and Snijders (2003) and Sniijders (2004). The motivation is that Markov dependence seems inefficient for some social networks, especially for large networks.

2.A.2 Model construction

Frank and strauss (1986) presented the Exponential Random Graph Models (ERGMs) applied to both undirected and directed networks.

The general form of ERGMs is as follows,

$$Pr(Y = y) = c^{-1} exp \sum_{A \subseteq G} \theta_A^T g_A(y) \quad , \tag{2.3}$$

where,

$$c = \sum_{G} exp \sum_{A \subseteq G} \theta_A^T g_A(y) \quad ,$$

and A is the clique of the dependence graph; $g_A(y)$ is a vector of statistics of G which describes the network structure statistics on A.

The ERGMs can be applied to any social network. A particular form of the models is chosen according to the set of cliques, that is $\{A\}$, and the structure statistics of interest.

The set of cliques in a network is determined by the dependence structure proposed before the model construction. For example, if the network is considered as a Bernoulli network, then independence is assumed among edges. The set of cliques $\{A\}$ contains only the single edges in the network. In another case, if Markov dependence is proposed to be the structure assumption of the network, then $\{A\}$ contains triangles and k-stars, where $k = 1, 2, \cdots, n-1$. With more general dependence assumption on the network, the ERGMs could be more complex in which $\{A\}$ contains more types of cliques.

Frank and Strauss proposed the ERGMs by starting from the most general situation, and then extending the results to Markov networks. They showed that for any network, the probability of a realization y has the form,

$$Pr(Y = y) = c^{-1} exp \sum_{A \subseteq G} \theta_A \quad , \tag{2.4}$$

where,

$$c = \sum_{G} exp \sum_{A \subseteq G} \theta_A \quad ,$$

and θ_A is an arbitrary constant if A is a clique of D and $\theta_A = 0$ otherwise. Each of the terms in the exponential function in formula (4) is according to a clique of the dependence graph D. The most general dependence hypothesis could be an assumption that the dependence graph D is a complete graph. It means that any two edges in the random network are conditionally dependent. For this dependence structure, a clique A can be any single edges and any order of combination of edges.

ERGMs for Bernoulli networks

For Bernoulli networks, the set of cliques $\{A\}$ contains only single edges.

According to (5), the ERGMs can be constructed as

$$Pr(Y = y) = c^{-1}exp\{\sum_{\{i,j\}} \theta_{ij}y_{ij}\} = c^{-1}exp\{\sum_{\{i,j\}\subseteq G} \theta_{ij}\} \quad , \tag{2.5}$$

where,

$$c = \sum_{G} exp\{\sum_{\{i,j\}\subseteq G} \theta_{ij}\} \quad .$$

Under the independence assumption, the normalizing constant can be simplified to ,

$$\begin{aligned} c &= \sum_{G} exp\{\sum_{\{i,j\}\subseteq G} \theta_{ij}\} \\ &= \sum_{G} \prod_{\{i,j\}\subseteq G} exp\theta_{ij} \\ &= \prod_{i,j}(1 + exp\theta_{ij}) \end{aligned} \tag{2.6}$$

Plugging (7) into (6), we get a simple form of likelihood function of a realization network y. That is,

$$\begin{aligned} Pr(Y = y) &= \frac{\prod_{\{i,j\}\in G} exp\ \theta_{ij} \prod_{\{i,j\}\notin G} exp(0)}{\prod_{i,j}(1 + exp\theta_{ij})} \\ &= \prod_{\{i,j\}\in G} \frac{exp\ \theta_{ij}}{1 + exp\ \theta_{ij}} \prod_{\{i,j\}\notin G} \frac{1}{1 + exp\ \theta_{ij}} \quad . \end{aligned} \tag{2.7}$$

In (8), θ_{ij} is the effect of the edge {i,j}. Compared with the equation (2), we can see the effect of the edge is actually the log odds of the edge, which is $\theta_{ij} = log\frac{p_{ij}}{1-p_{ij}}$. When homogeneity assumption is imposed so that the effect for each edge is identical, the model becomes,

$$Pr(Y = y) = c^{-1}exp(\theta y_{++}) \quad , \tag{2.8}$$

where θ is the common effect for edges and y_{++} is the total number of edges in graph G.

ERGMs for Markov networks

For Markov graphs, the set of cliques $\{A\}$ contains triangles and k-stars. The ERGMs under Markov assumption are,

$$Pr(Y=y) = c^{-1}exp[\sum_{u,v,w} \tau_{uvw} y_{A_1} + \sum_{k=1}^{n-1} \sum_{v_0,\cdots,v_k} (\sigma_{v_0 \cdots v_k}/k!) y_{A_2}] \quad , \tag{2.9}$$

where $y_{A_1} = y_{uv}y_{vw}y_{wu}$, $y_{A_2} = y_{v_0}y_{v_1} \cdots y_{v_k}$, and $u, v, w, v_0, v_1, \cdots, v_k$ are vertices in the graph. With homogeneity assumed, the model can be simplified to,

$$Pr(Y=y) = c^{-1}exp[\tau t + \sum_{k=1}^{n-1} (\sigma_k s_k)] \quad , \tag{2.10}$$

where τ is the common triangle effect, σ_k is the common k-star effect, t and s_k are numbers of triangles and k-stars in the graph, respectively.

Frank and Strauss also introduced a model only consider the edge effect, triangle effect and 2-star effect. It is called the $\rho\sigma\tau$ model,

$$Pr(Y=y) = c^{-1}exp(\rho r + \sigma s + \tau t) \quad , \tag{2.11}$$

where ρ, σ, τ refer to the common effects of an edge, a 2-star and a triangle, respectively, with r, s, t represent numbers of the corresponding cliques.

At last, let's impose the homogeneity assumption to the general form of the ERGMs (3). The homogeneous ERGMs are of the form,

$$Pr(Y=y) = c^{-1}exp\{\theta^T g(y)\} \quad , \tag{2.12}$$

where,

$$c = \sum_{y} exp\{\theta^T g(y)\} \quad .$$

2.B Estimation

Maximum likelihood estimation (MLE) is not feasible to ERGMs for relatively large networks. Consider a network with n vertices. The random network has $2^{\frac{n(n-1)}{2}}$ realizations. The log-likelihood of the model (13) is,

$$\begin{aligned} l(\theta) = logL(\theta) &= log(c(\theta)^{-1} exp\{\theta^T g(y)\}) \\ &= -log(c(\theta)) + \theta^T g(y) \quad . \end{aligned} \tag{2.13}$$

Since the normalizing constant c is a summation of $2^{\frac{n(n-1)}{2}}$ terms, the maximizing of the log-likelihood function is really burdensome. For example, when $n = 34$, the number of terms is $7.55 * 10^{168}$.

An approximation of the estimation was proposed by Strauss and Ikeda (1990). They introduced maximum pseudo-likelihood estimation method for estimating the parameters of ERGMs.

2.B.1 Maximum pseudo-likelihood estimation (MPLE)

As presented by Strauss and Ikeda in 1990, ERGMs is rewritten in its logit form. First, the probability of $Y_{ij} = 1$ conditioning on the rest of the variables in the network can be expressed without the normalizing constant in the following form,

$$Pr(Y = y) = c^{-1} exp\{\sum_{A} \theta_A^T g_A(y)\} \quad ,$$

which implies,

$$\begin{aligned} Pr(Y_{ij}=1|Y_{ij}^{c}) &= \frac{Pr(Y=y_{ij}^{1})}{Pr(Y=y_{ij}^{1})+Pr(Y=y_{ij}^{0})} \\ &= \frac{exp\{\theta^{T}g(y_{ij}^{1})\}}{exp\{\theta^{T}g(y_{ij}^{1})\}+exp\{\theta^{T}g(y_{ij}^{0})\}} \quad . \end{aligned}$$

Then, with the conditional probability, the log odds of an edge Y_{ij} can be constructed, as,

$$\begin{aligned} \frac{Pr(Y_{ij}=1|Y_{ij}^{c})}{Pr(Y_{ij}=0|Y_{ij}^{c})} &= \frac{exp\{\theta^{T}g(y_{ij}^{1})\}}{exp\{\theta^{T}g(y_{ij}^{0})\}} \\ &= exp\{\theta^{T}[g(y_{ij}^{1})-g(y_{ij}^{0})]\} \\ &= exp\{\theta^{T}\Delta_{y_{ij}}\} \quad , \end{aligned}$$

where,

$$\Delta_{y_{ij}} = g(y_{ij}^{1})-g(y_{ij}^{0}) \quad .$$

We can interpret the parameters in the ERGMs as the effects of the statistic changes on the log odds of an edge, when the edge variable is changed from 0 to 1.

The pseudolikelihood function is defined by,

$$PL(\theta) = \prod_{ij} Pr(y_{ij}|y_{ij}^{c}) \quad ,$$

and the Maximum Pseudolikelihood estimator is obtained by maximizing this function.

Let $P_{ij} = Pr(y_{ij} = 1|y_{ij}^{c})$ and $Q_{ij} = 1 - P_{ij}$. Then, to maximize the

Pseudolikelihood function, we need to solve,

$$\frac{\partial}{\partial\theta} logPL = \sum_{ij}\{\frac{y_{ij}}{P_{ij}}\frac{\partial P_{ij}}{\partial\theta} + \frac{1-y_{ij}}{Q_{ij}}(-\frac{\partial P_{ij}}{\partial\theta})\} = 0 \quad , \tag{2.14}$$

and it implies,

$$\Rightarrow \sum_{ij}\frac{1}{P_{ij}Q_{ij}}(y_{ij}-P_{ij})\frac{\partial P_{ij}}{\partial\theta} = 0 \quad . \tag{2.15}$$

The parameters can be estimated by logistic regression.

2.B.2 Markov chain Monte Carlo maximum likelihood estimation

Recently, some results have been reported about the inefficiency of MPLE in ERGMs for some of the well known network data. Since various Monte Carlo estimation techniques are now developed, the Markov chain Monte Carlo maximum likelihood estimation (MCMCMLE) has been proposed to be a better approximation of the MLE. This approach has been presented and reviewed by a number of authors (see Snijders,2002; Handcock et al., 2006; Wasserman and Robins, 2005).

Simulation is the core method of Monte Carlo maximum likelihood estimation. To obtain the MCMCMLE of an ERGM, a starting set of parameter values would be assigned first. Then, based on the Monte Carlo Markov Chain simulation, new data is sampled from the current ERGM and the parameters are re-estimated. This process is repeated until the parameter estimates stabilize.

The start point of the parameters is usually chosen as the MPLE. Simulation of the network distribution could be achieved by a number of algorithms, such as the Metropolis algorithm.

Snijders et al. (2006) indicated that the MCMCMLE is inadequate in ERGMs for some datasets, for example, in which the transitivity effects are strong. The estimation process would not stabilize in this situation. This implies that the ERGMs are inappropriate for the data.

2.C Goodness of fit for ERGMs

When we obtain the estimated ERGMs, it is important to evaluate how well the models fit the observed networks. In this section, we introduce some of the traditional methods and the goodness of fit diagnosis plots presented by Hunter et al. (2008).

2.C.1 Traditional Methods (AIC, BIC)

Traditional methods such as AIC (Akaike, 1973) or BIC (Schwarz, 1978) have been used to assess the goodness of fit of ERGMs.

In the general case, the AIC is,

$$AIC = 2k - 2\ln(L)\,,$$

where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

The BIC is,

$$BIC = k\ln(n) - 2\ln(L)\,,$$

where k and L are the same as those in the AIC and n is the effective sample size.

Both of the methods are appropriate when the observations are an independent and identically distributed sample. However, this assumption is not valid for network data since the variables in a random network are usually considered to be dependent.

2.C.2 Goodness of fit plots

In Hunter et al. (2008), he proposed his approach of assessing the goodness of fit tests of ERGMs. The idea is that a fitted model should, more or less, reproduce network statistics seen in the original. In Hunter's presentation material in 2007, he explained the goodness of fit intuition with the following figure.

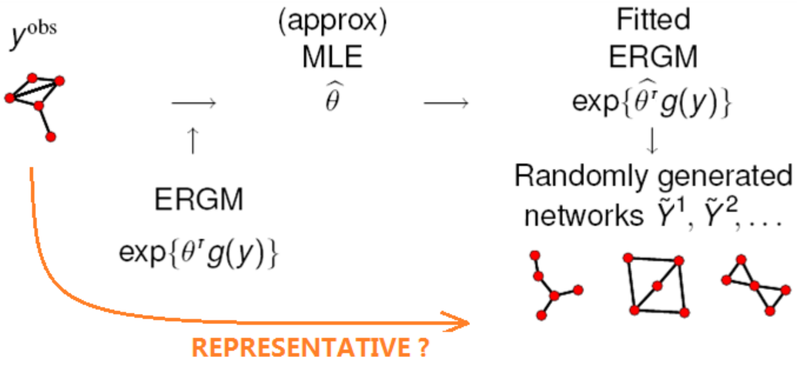

y^{obs}
(approx) MLE $\widehat{\theta}$
Fitted ERGM $\exp\{\widehat{\theta}^{\tau} g(y)\}$
ERGM $\exp\{\theta^{\tau} g(y)\}$
Randomly generated networks $\tilde{Y}^1, \tilde{Y}^2, \ldots$
REPRESENTATIVE ?

Figure 2.1: Motivation of Hunter's goodness of fit plots

In Figure 1, the observed network is denoted as y^{obs}. We can try any ERGM to fit y^{obs}. After the estimation process, we obtain a fitted ERGM which is denoted by $exp\{\hat{\theta}^T g(y)\}$. Hunter et al.(2008) indicates that in order to evaluate how well the ERGM fits the data, the structural statistics of the observed network should be compared with the corresponding statistics on networks simulated from the fitted model, such as $\{\tilde{y}^1, \tilde{y}^2, \ldots\}$. If the observed structure statistics can represent the simulated structure statistics, it implies a good fit of the ERGM.

In order to compare the structure statistics between the observed and the simulated, Hunter et al.(2008) presented the goodness of fit plots.

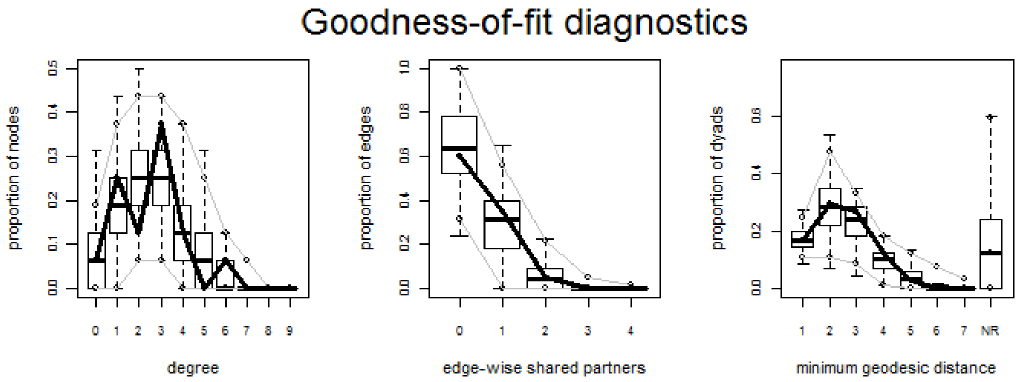


Figure 2.2: Hunter's diagnostics plots

In Figure 2, the observed network is compared to the networks simulated from the fitted ERGM in terms of the below structure statistics:

- Degree: the number of ties of a vertex;
- Edgewise shared partners: a vertex connecting both ends of a tie;
- Minimum geodesic distance: the minimum number of connected ties via which two vertices are related.

The diagnosis is not necessarily achieved through these three structure statistics. Any other structure statistics can be chosen to evaluate the goodness of tests. The question of how many and what structure statistics should be included in the tests are determined by the interests of the researchers.

Although there are various structure statistics that can be used in the diagnostic plots, they are all based on the central approach: simulation of a sample of random networks from a fitted ERGM, and comparison of the observed structure statistics and the simulated structure statistics summarized from the sample of simulated networks.

The boxplots in Hunter's goodness of fit plots represent the distributions of the simulated structure statistics which depend on the ERGM. The connected lines in the plots represent the structure statistics in the observed network. The broken lines represent the 2.5% and 97.5% quantiles in the distribution of the simulated structure statistics after removing the outliers. These 95% confidence intervals are connected to form the boundaries of the reasonable values of the structure statistics. Thus, if the connected lines are within the boundaries, it implies the simulated sample of the structure statistics are well represented by the observed structure statistics. This also has the implication that the ERGM which the simulation is based on has a good fit.

Compared with the traditional methods, the goodness of fit plots provide more information. The plots provide separate views of the goodness of fit of the model according to each of the considered structure statistics.

Chapter 3

A simulation based approach applied to the goodness of fit tests on ERGMs

3.A Instruction

Since network is a dependent data, it lacks of the properties of asymptotics. Traditional methods of goodness of fit test, like AIC and BIC, are based on the assumption of independent samples and thus are not appropriate for network data. This makes it difficult to assess the goodness of fit tests and also raises some questions.

Firstly, since MLE is not appropriate for ERGMs, the likelihood of the network cannot be used to assess goodness of fit test. Even if we can calculate the likelihood of the network with moderate size, the likelihood could be small considering the large number of terms contained in the normalizing constant. Secondly, there is no efficient way to compare two ERGMs. In Hunter's plots, the fact that the observed statistics are within the boundaries implies a good fit of the model. Otherwise, it implies a bad fit of the model. However, between two good fits, it is hard to justify which one is better. Thirdly, without a good answer for the second question, we do not have an approach to select the best model from a class of ERGMs.

In this paper, we propose an approximating likelihood approach on network structure statistics in order to solve the first two questions. In the discussion, we will talk about the further work which can be done to answer the third question.

Origin of the idea

Consider simulating a network from a certain ERGM. The result is regarded as an observed network G_{obs}. Then we fit a new ERGM to G_{obs}. We hope the fitted ERGM can be as close as possible to the original model. However, there is often a discrepancy between the original model and the fitted model.

Is the discrepancy avoidable? Let us look at a special case. Assume the original model is the independent model, $Bernoulli(0.5)$ distribution, on a n-vertex random network. Actually, with the original model, any realization of the random network is generated with equal probability. This means the observed network can be any n-vertex network, which might lead to any fitted model among ERGMs.

The discrepancy itself is not the problem. The real question behind is that without any constraints on the dependence assumptions of the network, we are not able to even reject that any network comes from an independent graph model, $Bernoulli(0.5)$ distribution. In terms of the probability $Pr(Y = y)$, we may fail to reduce to a class of ERGMs which fit the observed data well. To solve this problem, we consider some structure statistics and their distributions.

The reason why we can consider the likelihood of the network statistics instead of the likelihood of the network is threefold. 1. Usually in a social network, the order of the vertices is not important. Altering the index of the vertices without moving any edges would not change the network, or to say the pattern of connections. This explains why the homogeneity is often assumed. In this situation, the likelihood of the observed structure statistics, which define the pattern of connections, is equivalent to the likelihood of the network. 2. Generating from any ERGM, even the independent model $Bernoulli(0.5)$, the likelihood of the network structure statistics will not be equal. In other words, there always exist more likely patterns and less likely patterns of the network. The discrepancy for the structure statistics from the most common patterns gives us a clue about the chance that the observed network comes from a certain model. 3. The distribution of the structure statistics can be approximated based on the simulation of the networks.

In Hunter's goodness-of-fit plots, several structural statistics are employed to build the plots. For each structural statistic, a good fit should reproduce a number of networks which the statistic in the simulated networks can be represented by the counterpart in the observed network.

In our work, we try to extend Hunter's work by investigating the distributions of the structural statistics. In Figure 3, Hunter's diagnostics plots are built based on a sample of networks. For each structure statistic, a boxplot is constructed according to the sample of simulated statistic counts. To find out the distribution of the statistics, we take some of the boxplots as examples (in the boxes) and show the histograms for the counts according to each boxplot. It seems that the statistics have certain distributions. The closed form of the

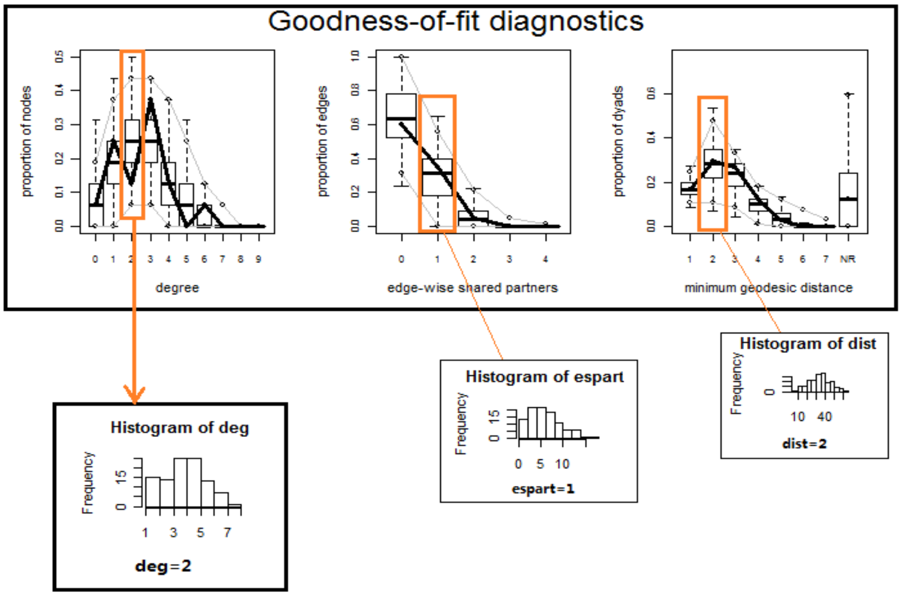


Figure 3.1: Exploring the distribution of the structure statistics

distributions cannot be proved. However, an approximated form of the distributions can be obtained based on the simulation. With this approximated distribution, we can extend Hunter's goodness of fit tests. In the following sections, degree counts distribution showed in the left plot in Figure 3 is used to demonstrate the idea.

3.B Degree count distribution of a network

From this section, the distribution about the degree counts in a random network is discussed to illustrate our idea of extending the goodness of fit test. First, we give the definition of this distribution. Then, we show the approach to use the approximated distribution of degree counts to calculate the likelihood of the statistics and the representativeness of the pattern of connection. Finally, we explain how to use the representativeness to test the goodness of fit.

In the study of graphs and networks, the degree of a vertex in a network is the number of edges it possesses. The degree distribution $P(k)$ is the probability distribution of a vertex to have exact k degrees. Let us define the distribution of the degree counts in a random network.

A degree count D_k is a random variable representing the number of vertices in a random network which have exact k degrees.

The distribution of the degree count is referred to the distribution of D_k in a random network generated from an ERGM.

In social networks, dependence is usually assumed among the variables. Because of this dependency, the distribution of degree counts is very complex and has not been fully discussed. But for independent random network, the degree distribution and the distribution of degree counts have been studied. Before we illustrate the idea of the goodness of fit test, we need to study the distributions in both independent and dependent random networks.

Degree counts distribution for independent networks

Bollobas's work in 1981 was the first detailed discussion about the degree distribution of an independent random network. In the paper, it was shown that the degree distribution is binomial distribution for independent networks.

For an independent random network (a Bernoulli network) in which any two vertices are connected independently with a common probability p, we know that the model is (9) and the common edge effect is $\theta = log\frac{p}{1-p}$. The probability of a vertex to have exact k degrees in this random network is denoted by $P(k)$, and it follows $binomial(n-1,p)$ distribution, i.e.,

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \quad . \tag{3.1}$$

In our work, instead of considering the probability distribution of a vertex's degree, we focus on the distribution of the number of vertices having a certain amount of degrees.

Let $Deg = (D_0, D_1, \cdots, D_{n-1})$ be a sequence of the count numbers of these degrees over a random network, that is, D_k denotes the number of vertices in G which have exact k connections with other vertices.

Bollobas further discussed that the distribution of D_k approaches a Poisson distribution,

$$Pr(D_k = j) = \frac{e^{-\lambda_k}\lambda_k^j}{j!} \quad ,$$

where $\lambda_k = nP(k)$.

Degree distribution for dependent networks

Binomial degree distribution is only valid for independent random networks. For dependent networks, the degree distribution could be quite complex and there is no complete form of the distribution according to the literatures.

In this paper, we do not make any assumptions on the degree distribution of dependent random networks. However, we assume the distribution of D_k can still be approximated by a Poisson distribution. We can not provide a theoretical proof for this assumption; but in section 3.4, some simulation results will be shown to illustrate its efficiency.

From a certain ERGM, we can produce a sample of networks. After summarizing the degree counts in the sampled networks, we obtain a sample of counts for each degree. Based on this simulation, we can check that the distribution of the degree counts can still be approximated by a Poisson distribution. The details are presented in section 3.5.

3.C Goodness of fit test based on the approximated distribution of degree counts

With Poisson distribution assumed for each variable d_k, we can fit a Poisson model from the sample of counts of d_k and the estimated parameter is $\hat{\lambda_k}$,

$$Pr(D_k = j) = \frac{e^{-\hat{\lambda}_k}\hat{\lambda}_k^j}{j!} \quad .$$

Then let d_k^{obs} be the count of vertices with k degrees in the observed network, we calculate the likelihood of the observed count in the fitted Poisson distribution. Equivalently, we can say that we obtain the likelihood of that d_k^{obs} are generated from the certain ERGM. Let Lik be the likelihood of the observed statistics in each D_k distribution.

Hunter's goodness of fit tests can be improved in two ways:

One development is using p values of the observed degree counts in D_k distributions to test the goodness of fit of a certain ERGM. Similarly, in Hunter's plots, observed structure statistics are compared with the boundaries which are constructed by the 2.5% and 97.5% quantiles of the sample of simulations. Since we have the approximate distribution of the structure statistics, we can construct the 95% confidence intervals based on the fitted Poisson distributions. Also, we can use a p value for each of the degree counts in the corresponding distributions equivalently.

Another development is to construct a new measurement to evaluate the discrepancy of the observed network from the simulated sample of networks generated from the ERGM. Let $Rep(k)$ be the measure of discrepancy of the observed count in the D_k distribution calculated by

$$Rep(k) = \frac{\text{Likelihood of } Obs_k \text{ in } D_k \text{ distribution}}{\text{Maximum Likelihood in } D_k \text{ distribution}} \quad .$$

This measurement $Rep(k)$ takes values between 0 and 1. The closer $Rep(k)$ is to 1, the less discrepancy there is between the observed count and the most probable count in its distribution. When $Rep(k)$ equals to 1, it implies that d_k^{obs} reaches the maximum of the likelihood in D_k distribution. The Hunter's rule of goodness of fit tests is that a fitted model should reproduce network statistics seen in the original. Another way around is to say that, the simulated network structure statistics should be represented by the observed network statistics. Similarly, the measurement of discrepancy $Rep(k)$ gives us a quantitative tool to check the representativeness of an observed count to the corresponding sample of simulated counts. Define REP as the measurement of the representativeness of an observed network to the sample of simulated networks generated from a certain ERGM,

$$REP = \prod_k Rep(k) \quad ,$$

where k takes values of all the degrees presented in the observed network. Although this REP is not well interpreted, it is a simple and useful criterion to compare two ERGMs. Consider we have two ERGMs and generate two samples of simulated networks, with which the REP can be calculated. Then, it implies the ERGM with the larger REP is better, as it can generate a sample of simulated networks which can be better represented by the observed network.

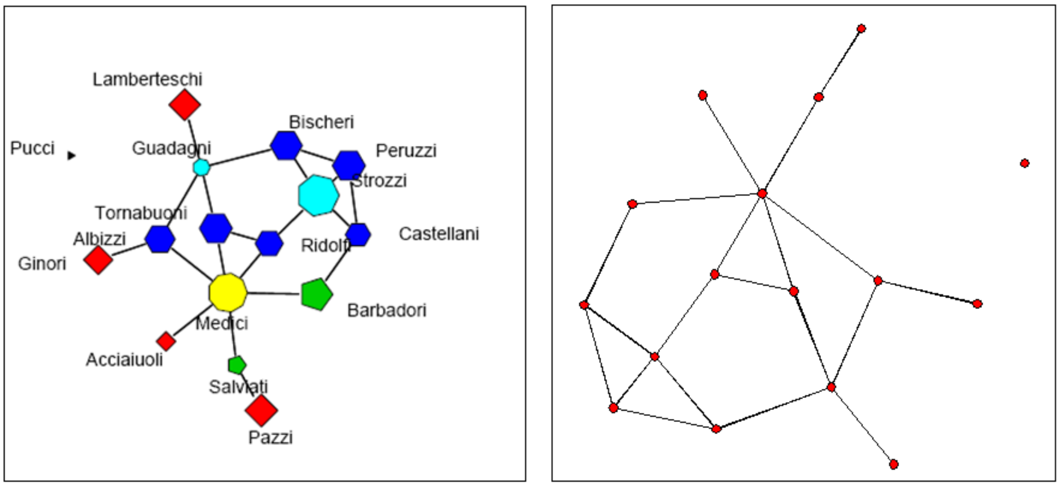


Figure 3.2: Marriage ties among Renaissance Florentine families

With these two developments on the goodness of fit tests, we can solve the first two problems listed at the beginning of this section.

3.D Examples

We have introduced the two developments on the goodness of fit tests of ERGMs. In this section, we use 2 examples to illustrate how these modifications improve the goodness of fit tests and solve the first two problems listed in the instruction of this section. For each example, we will first introduce the network data. Then, with the assumption that the distribution of degree counts is approximate Poisson, we will show the extended goodness of fit tests. We check the validation of the assumption by simulation in next section.

Example A

In this example, the network data is about the marriage ties among Renaissance Florentine families, see Figure 4.

Data description: Breiger & Pattison (1986), in their discussion of local role analysis, use a subset of data on the social relations among Renaissance Florentine families (person aggregates) collected by John Padgett from historical documents. In the network, edges indicate the marriage ties between 16 families. Families are presented by vertices with particular shape and size. The number of sides of each vertex denote the degree of the vertex plus 3; and the size of the vertex indicates the wealth of the family, see the left of Figure 4.

The goal of this example is to obtain a distribution-based 95% confidence intervals for the $D_k's$. So, we do not consider the information of wealth. The original network can be transformed to the right of Figure 4, a simple graph which keeps the pattern of connections. In other words, the two networks have the same network structure.

In this example, the ERGM that we use to fit the network data is

$$Model\ 1 : Pr(Y = y) = c^{-1}exp\{\rho r + \sigma s\}.$$

The estimated parameters are $\hat{\rho} = -1.664, \hat{\sigma} = 0.012$. To assess the goodness of fit test of Model 1, a sample of simulated networks need to be generated from Model 1 by MCMC procedure. The Hunter's goodness of fit plots and our approach of goodness of fit test can be constructed based on the simulated structure statistics in the sample of networks. In terms of the degree counts, the Hunter's diagnostic plot is in Figure 5. We study the distribution of D_k for $k \in \{1, 2, \cdots, 7\}$ and fit each distribution with Poisson distribution. With these approximated distribution, we can give the confidence intervals and p values for each of these degree counts.

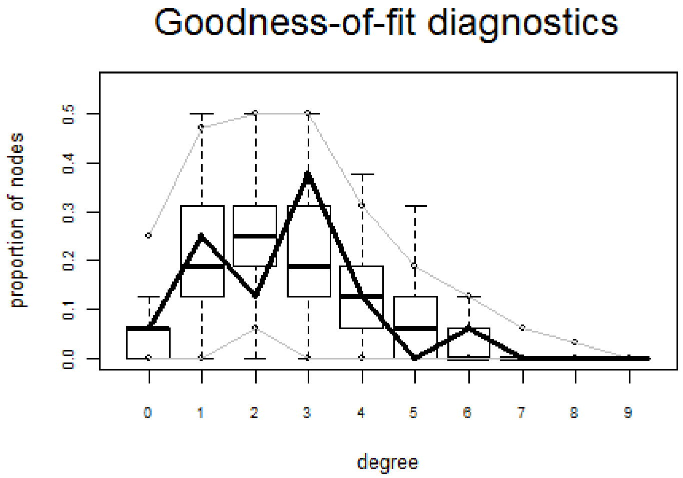


Figure 3.3: Hunter's diagnosis plot via degree

k	1	2	3	4	5	6	7
$\hat{\lambda}_k$	3.17	4.34	3.79	2.18	0.97	0.34	0.10
Obs_k	4	2	6	2	0	1	0
CI_k	[0,7]	[1,9]	[1,8]	[0,5]	[0,3]	[0,2]	[0,1]
P_k	0.39	0.19	0.18	0.63	0.38	0.29	0.90

Table 3.1: Results in Example A

The results of Example A are recorded in Table 1. For $k \in \{1, 2, \cdots, 7\}$, λ_k is the estimated Poisson parameter for the D_k distribution. Based on the approximate distribution, $Poisson(\lambda_k)$, 95% confidence interval can be constructed as $(2.5\% quantile, 97.5\% quantile)$. Since Poisson distribution is a discrete variable distribution, the coverage of the confidence intervals may not be exactly 95%. Equivalently, the P value of each of the observed degree counts can be calculated. For $k \in \{1, 2, \cdots, 7\}$, $P_k = min\{Pr(D_k \geq Obs_k), Pr(D_k \leq Obs_k)\}$. A larger P value of an observed degree count means a bigger chance of the observed coming from the corresponding Poisson distribution.

In this example, all the observed degree counts are included in the 95% confidence intervals and all the P values are larger than 0.05. It implies a good fit of the ERGM. Based on Hunter's plot, we can also construct the 95% confidence intervals without the assumption of approximate Poisson distributions of the degree counts. However, the distribution-based confidence intervals are more reasonable when the assumption is valid and would show its advantage especially when we want to identify and remove the outliers in the sample of simulated degree counts.

Example B

Data description: This data set, formerly called 'fauxhigh', represents an in-school friendship network. The school community is in the rural western US, with a student body that is largely Hispanic and Native American. In the network, each vertex represents a student and each edge represents the friendship between two students. See Figure 6, the shapes of nodes denote sex: circles for female, squares for male, and triangles for unknown. Labels denote the units digit of grade (7 through 12), or $'-'$ for unknown. Therefore, in this network data, there are two covariates, sex and grade.

In Hunter's work of introducing the R package 'ergm' in 2008, two different models were used to fit the network. In R language, the two models can be written as,

- Model 2.1: model = ergm(faux.mesa.high $\sim$ edges + nodematch($'$Grade$'$, diff = TRUE) + nodefactor($'$Sex$'$))

- Model 2.2: model = ergm(faux.mesa.high $\sim$ edges + nodematch($'$Grade$'$) + gwesp(0.5, fixed=TRUE), verbose=TRUE)

In the two models, the terms following $\sim$ symbol represent the structure statistics $g(y)$ in the exponential function in ERGMs. The statistics concerned in Model 2.1 include

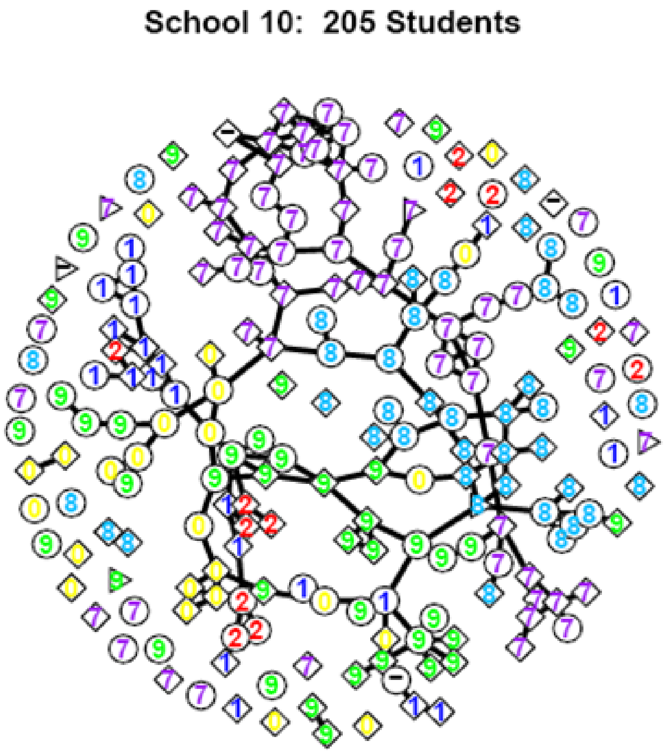


Figure 3.4: Mutual Friendships in Fauxhigh School 10

- the number of edges in y [ergm code: edges];

- the number of edges between students of the same grade, counted separately for each possible grade [ergm code: nodematch($'$Grade$'$, diff = TRUE)];

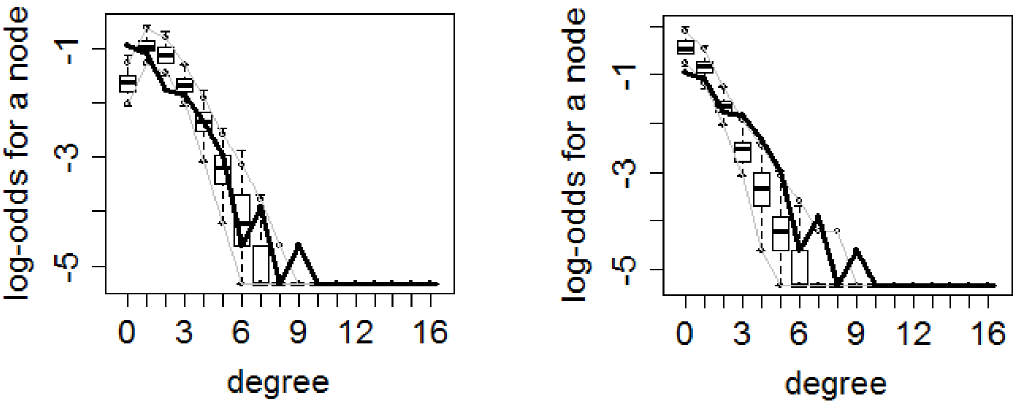
- the number of edges involving males, with male-male edges counted twice [ergm code: nodefactor($'$Sex$'$)].

The statistics concerned in Model 2.2 include the first two in Model 2.1 and replace the third term with the number of geometrically weighted edgewise shared partner [ergm code: gwesp(0.5, fixed=TRUE)].

Our goal in this example is to calculate and compare the measure of representativeness 'REP' for the two models and also to find the one that can generate a sample of networks which is better represented by the original data.

Hunter's plots in Figure 7 show that both of the models fit well but are not perfect. It is not easy to tell from the plots which one fits better.

In our approach, two samples of simulated networks are generated from the two models, respectively. With the assumption of Poisson distribution of the degree counts, the degree counts in the three samples can be fitted by Poisson distributions. Then, we can calculate the likelihood of the observed degree counts in the fitted Poisson distributions. As mentioned in section 3.3, the measurement of discrepancy $Rep(k)$ and the measurement of the representativeness REP can be constructed based on the likelihoods of the observed degree counts and the fitted Poisson distributions. With the results in Table 2, we can compare the two models via representativeness.



a. Model 2.1

b. Model 2.2

Figure 3.5: Hunter's diagnosis plot via degree for a. Model 2.1 and b. Model 2.2

In the results, the $Rep(k)$ for $k = 0, 2, 9$ takes very small values for both models. It means that for degrees equal to 0,2 and 9, there is a large discrepancy of the observed degree counts from the simulated sample, which can be seen from the Hunter's plots in Figure 7.

To compare the goodness of fit for the two models, we can compare the $Rep(k)$ separately for each k considered in the example. Also, we can compare the REP instead. As both of the models do not fit perfectly, the REP for the two models are very small. Albeit, the larger REP implies a better representativeness of the model. Thus, in terms of the degree counts, Model 2.1 fits better since it can generate a sample of networks in which the simulated degree counts can be better represented by the original degree counts. We can take the log-transformation of the ratio of REP for the two models,

$$log(\frac{REP_{Model2.1}}{REP_{Model2.2}}) = 1.459 \ .$$

Measure	Model 2.1	Model 2.2
Rep(0)	0.0009	0.0006
Rep(1)	0.8359	0.8553
Rep(2)	0.0104	0.0086
Rep(3)	0.7041	0.5833
Rep(4)	0.9938	0.9822
Rep(5)	0.7833	0.7272
Rep(6)	0.8336	0.8849
Rep(7)	0.1007	0.0874
Rep(8)	0.4450	0.5500
Rep(9)	0.0102	0.0050
REP	1.686e-09	3.920e-10

Table 3.2: Measure of the discrepancy $Rep(k)$ and the representativeness REP

If the new measure is larger than 0, then it implies the Model in the nominator has a better fit than the Model in the denominator.

3.E Simulations

This section includes two parts of simulations. The first part is regarding checking the assumption of degree counts distribution for Example A and B in section 3.4. The second part is related to checking the assumption for a general class of dependent ERGMs.

Simulation A

In this part, we check the Poisson assumption on the distribution of degree counts in Examples A and B. The procedures are similar for the two example and are listed as follows.

1. Generate N networks from $Pr(y) = c^{-1}exp(\theta g(y))$ using MCMC procedure. This can be implemented by running 'ergm' Package in R.

2. Obtain a matrix of degree counts, i.e. $Deg.matrix = \{x_{ij}\}, i = 1, 2, \cdots, N; j =$

$1, 2, \cdots, n$, where x_{ij} is the number of vertices have $j-1$ degrees in the ith network.

3. Fit a Poisson model for each column of $Deg.matrix$, i.e. $Poisson(\lambda_k)$ for the kth column referring to the $(k-1)$th degree counts. Estimate the $\lambda_k's$. Use Pearson χ^2 test to assess the goodness of fit of the Poisson models.

4. Repeat the above steps for K replicates, calculate the mean p values and the standard errors.

We only need to consider the degrees which show up at least in 1 simulated network for all K replicates. Because for those degrees which are never present in N simulated networks, the counts are all 0; even if the degrees are present in other replicates, the counts would be highly suppressed to 0 and the Poisson models would be good fittings anyway. For this reason, we consider degree counts for the degree of $\{0, 1, 2, \cdots, 7\}$ in the Example A and $\{0, 1, 2, \cdots, 9\}$ in Example B, respectively.

Simulation A.1

Simulation setup: $N = 100, n = 16, K = 20, Model1 : model = ergm(flo \sim edges + kstar(2))$.

k	0	1	2	3	4	5	6	7
p	0.91	0.88	0.65	0.79	0.93	0.84	0.94	0.95
$se(p)$	0.05	0.05	0.05	0.05	0.04	0.06	0.05	0.05

Table 3.3: P values of checking the Poisson assumption for Example A

The p values in Table 3 correspond to the Pearson χ^2 tests of checking Poisson assumption for each degree count distribution for Example A. The p

values are quite large and imply the assumption is valid for the distribution of degree counts.

Simulation A.2

Simulation setup: $N = 100, n = 205, K = 20$.

- Model 2.1: model = ergm(faux.mesa.high ∼ edges + nodematch(′Grade′, diff = TRUE) + nodefactor(′Sex′))

- Model 2.2: model = ergm(faux.mesa.high ∼ edges + nodematch(′Grade′) + gwesp(0.5, fixed=TRUE), verbose=TRUE)

k	0	1	2	3	4	5	6	7	8	9
$p_{model2.1}$	1.00	1.00	1.00	1.00	1.00	1.00	0.79	1.00	1.00	1.00
$se(p)_{model2.1}$	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00
$p_{model2.2}$	1.00	1.00	1.00	1.00	1.00	1.00	0.85	0.95	1.00	1.00
$se(p)_{model2.2}$	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.05	0.00	0.00

Table 3.4: P values of checking the Poisson assumption for Example B

The p values in Table 4 correspond to the Pearson χ^2 tests of checking Poisson assumption for each degree count distribution for Example B. The results include p values for Model2.1 and Model2.2. Most of the p values are close to 1, and the rest are also quite large. It indicates that the assumption is valid for the distribution of degree counts for both of the models.

Simulation B

Simulation A checked the Poisson assumption for the models in examples. Is this assumption valid for other models? In this part, we check the Poisson assumption on the distribution of degree counts for general ERGMs defining dependent random networks. We indicate it by using only one class of ERGMs which is very common and representative. The class of ERGMs is the $\rho\sigma\tau$

models,

$$Pr(Y = y) = c^{-1}exp(\rho r + \sigma s + \tau t) \ .$$

In R language, the model can be written as,

$$model_{\rho\sigma\tau} = ergm(data \sim edges + kstar(2) + triangles) \ .$$

The $\rho\sigma\tau$ models have coefficients for edges, 2 stars and triangles. With different coefficients, we can specify different models. In this simulation, we check the Poisson assumption for these models.

The procedure is as follows:

1. Generate a large number of networks from independent random network model using MCMC procedure. This guarantees we have networks with various network patterns.

2. Fit each of the networks with the $\rho\sigma\tau$ models and record the coefficients. Construct a coefficient space (3 dimensions for the $\rho\sigma\tau$ models).

3. Check the Poisson assumption for the models with all the coefficient setups in the space.

For independent random network model, we use $Pr(y_{ij}) = Bernoulli(p)$, where the density p takes values in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. 200 networks with 16 vertices are generated for each value of density.

Figure 8 shows the distributions for the three coefficients separately. For each coefficient setup in this constructed space, we can obtain a model. After generating a sample of networks from the model, we check the Poisson assumption for the distribution of degree counts. The R code and part of the

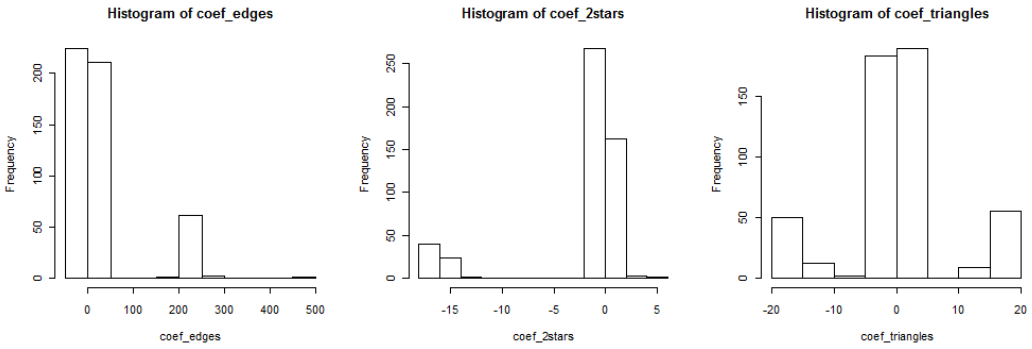


Figure 3.6: Coefficient space in Simulation B

result are shown in Appendix D.

The space consists of 500 coefficient setups. Appendix D only shows 20 results out of the 500. Each results have 19 columns. First three columns are the coefficients according to edges, 2 stars and triangles, respectively. The last 16 columns are the p values of checking the Poisson assumption for the distributions of the degree counts. The p values are not always available for all the degrees, because the degree counts may all be 0 in the simulated sample. If this happens, the unavailable p values are replaced by $'\text{NaN}'$ in the results.

For all the available p values in the 500 results in this simulation, 92% of p values are larger than 0.5 and almost 70% of p values are larger than 0.8. Therefore, the Poisson assumption is valid for a large scale of the coefficients in this class of ERGMs. Further, the approach based on the Poisson assumption on the distribution of degree counts could be generalized to many other dependent ERGMs.

Chapter 4

Conclusions and Further discussions

The standard large sample asymptotics is missing for networks. Therefore, the traditional tests to assess the goodness of fit of models like AIC, BIC are not appropriate for the ERGMs. There is no standard criteria for assessing goodness of fit of ERGMs and model selection.

So far, the best test we can do is the diagnostic plots by Hunter et al. Hunter collected several network structure statistics of common interest, for each of which he constructed the goodness of fit plots with original structure statistics in the target network and those in the simulated networks that generated from the fitted ERGM. A good fit of the ERGM is indicated by that the simulated structure statistics are represented by the original statistics.

In our work, we follow Hunter's idea and try to investigate the distributions of the network structure statistics. Based on a sample of simulated networks from a certain ERGM, we can obtain a sample for each of the structure statistics, based on which we tried several distributions to fit the statistics. In fact, some of the structure statistics have distributions that can be well approximated by Poisson distribution, such as the degree counts and the counts of edge-wise shared partners. This approach can also be applied to other network structure statistics if their distributions can be proven or well approximated by simulation.

Two improvements have been done for the goodness of fit tests of the ERGMs. With ′the marriage ties in florentine families′ example, we showed that the 95% confidence intervals of the degree counts could be constructed as distribution-based statistics. These intervals would be more precise when the assumption of distribution is valid. With ′the friendships in high school′ example, the measure of discrepancy and the measure of representativeness are regarded as the criterion of comparing two ERGMs. Based on the distribution of the degree counts, Hunter's diagnostic plots are quantified into these measures, which could be easily used to compare different ERGMs.

Assessing goodness of fit tests of ERGMs is still at its early stage. Further work should be focused on a number of aspects. Some of them are mentioned below.

- Efforts should be made to find the closed form of distribution of network structure statistics.

  Our work is built on assuming the Poisson distribution of some of the structure statistics in random networks. Other distributions we tried to fit the distribution of degree counts were binomial distribution, multinomial distribution and Poisson log-normal distribution. The Poisson and binomial distributions gave the best fits. In the future work, the closed form of the structure statistics distributions could be proved to make the approach more reliable. Otherwise, as an alternative, a better approximation of the distribution could also improve the performance.

- A standard measure of the goodness of fit of ERGMs should be constructed.

  There is no good interpretation of the measurements used in the example B. Since the $Rep's$ and REP are constructed based on the likelihood of the structure statistics, a modification can be made as an addition to the $Rep's$ and REP of a measure of model complexity. Then, a standard criterion could be constructed like AIC, BIC.

- Model selection might be available after the standard measure is developed.

  If the standard criterion mentioned above is constructed, then we are not only able to compare two different ERGMs conveniently, but also to use the criterion in the procedure of model selection. Although the model selection might not locate the best model within all of the ERGMs, it is possible to find the best model within a certain class of ERGMs defined by interests.

Chapter 5

Appendix

5.A R code

Appendix A: R Code for Example A

```
observed=c(4,2,6,2,0,1,0)
lambda=c(3.17,4.34,3.79,2.18,0.97,0.34,0.10)
CIP=function(observed,lambda)
{
m=length(observed)
CI=matrix(nrow=2,ncol=m)
p=c()
for (i in 1:m)
{
obs=observed[i]
lam=lambda[i]
CI[1,i]=qpois(0.025,lam)
CI[2,i]=qpois(0.975,lam)
if (obs==0) p[i]=dpois(0,lam) else p[i]=min(ppois(obs,lam),1-ppois(obs-1,lam))
}
list=list(CI=CI, pvecter=p)
return(list)
}
CIP(observed,lambda)
```

Appendix B: R Code for Example B

```
library(”ergm”)
data(”faux.mesa.high”)
```

```
fx=faux.mesa.high
n=205
# begin{modeling}
model3 = ergm(faux.mesa.high ∼ edges + nodematch(”Grade”, diff = TRUE)
+ nodefactor(”Sex”))
summary(model3)
model4 = ergm(faux.mesa.high ∼ edges + nodematch(”Grade”) + gwesp(0.5,
fixed=TRUE), verbose=TRUE, seed = 789)
summary(model4)
# end{modeling}
# begin{goodness of fit plots}
# Model 2.1
m3gof = gof(model3, GOF = ∼ distance + espartners + degree + triadcensus,
verbose = TRUE, interval = 5e+4, seed=111)
par(mfrow = c(2,2))
plot(m3gof, cex.lab=1.6, cex.axis=1.6, plotlogodds = TRUE)
# Model 2.2
m4gof = gof(model4, GOF = ∼ distance + espartners + degree + triadcensus,
verbose = TRUE, interval = 5e+4, seed=111)
par(mfrow = c(2,2))
plot(m4gof, cex.lab=1.6, cex.axis=1.6, plotlogodds = TRUE) # end{goodness
of fit plots}
```

```
    set.seed(333)
N=100
n=205
```

```
K=20
pto2.pois=matrix(nrow=K,ncol=15)
p.pois=matrix(nrow=K,ncol=15)
for (k in 1:K)
{
# begin{network generation}
# for model 2.1
m3gof = gof(model3, GOF = ∼ distance + espartners + degree + triadcensus,
verbose = TRUE, interval = 5e+4)
deg=m3gof$sim.deg

   # for model 2.2, use following 3 lines instead of the 3 lines above
#m4gof = gof(model4, GOF = ∼ distance + espartners + degree + triadcensus, # verbose = TRUE, interval = 5e+4)
#deg=m4gof$sim.deg
# end{network generation}

   # begin{REP}
for (j in 1:15)
{
a=deg[,j]
b=rep(0,(n+1))
c=c(0:n)
for (i in 1:(n+1))
{
b[i]=length(a[a==(i-1)])
}
p.pois[k,j]=sum(a)/N
d.pois=dpois(c,p.pois[k,j])*N
xto2.pois=sum((b-d.pois)*(b-d.pois)/d.pois)
pto2.pois[k,j]=1-pchisq(xto2.pois,n)
}
}
round(pto2.pois,4)
pmatrix.pois=pto2.pois[,1:10]
result2=rbind(apply(pmatrix.pois,2,mean),apply(pmatrix.pois,2,sd)/sqrt(K))
round(result2,4)
p.mat=p.pois[,1:10]
result2.p=rbind(apply(p.mat,2,mean),apply(p.mat,2,sd)/sqrt(K))
round(result2.p,4)
pp=p.pois[,1:10]
ppp=apply(pp,2,mean)
degobs=m3gof$obs.deg[1:10]
rep=c()
for (i in 1:length(degobs))
{
rep[i]=dpois(degobs[i],ppp[i])/max(dpois(floor(ppp[i]),ppp[i]),dpois(floor(ppp[i])+1,ppp[i]))
}
rep
(prod(rep))
# end{REP}
```

Appendix C: R Code for Simulation A

```
# Simulation 1.A library(”ergm”)
data(”florentine”)
flo=flomarriage
model=ergm(flo ∼ edges+kstar(2))
set.seed(321)
N=100
n=16
K=20
pto2.pois=matrix(nrow=K,ncol=n)
p.pois=matrix(nrow=K,ncol=n)
for (k in 1:K)
{
gof1=gof(model,nsim=N)
deg=gof1$sim.deg
for (j in 1:n)
{
a=deg[,j]
b=rep(0,(n+1))
c=c(0:n)
for (i in 1:(n+1))
{
b[i]=length(a[a==(i-1)])
}
p.pois[k,j]=sum(a)/N
d.pois=dpois(c,p.pois[k,j])*N
xto2.pois=sum((b-d.pois)*(b-d.pois)/d.pois)
pto2.pois[k,j]=1-pchisq(xto2.pois,n)
}
}
round(pto2.pois,4)
pto2.mat=pto2.pois[,1:8]
result1.p=rbind(apply(pto2.mat,2,mean),apply(pto2.mat,2,sd)/sqrt(K))
lambda.mat=p.pois[,1:8]
result2.lambda=rbind(apply(lambda.mat,2,mean),apply(lambda.mat,2,sd)/sqrt(K))
round(result2.lambda,4)

# Simulation 1.B
library(”ergm”)
data(”faux.mesa.high”)
fx=faux.mesa.high
N=100
K=20
n=205
set.seed(321)
model2.1 = ergm(faux.mesa.high ∼ edges + nodematch(”Grade”, diff = TRUE)
+ nodefactor(”Sex”))
model2.2 = ergm(faux.mesa.high ∼ edges + nodematch(”Grade”) + gwesp(0.5,
fixed=TRUE), verbose=TRUE, seed = 789)
pto2.pois=matrix(nrow=K,ncol=10)
p.pois=matrix(nrow=K,ncol=10)
for (k in 1:K)
```

```
{
gof2.1=gof(model2.1,nsim=N)
deg=gof2.1$sim.deg
# For model 2.2, use following 2 lines instead of above 2 lines
#gof2.2=gof(model2.2,nsim=N)
#deg=gof2.2$sim.deg
for (j in 1:10)
{
a=deg[,j]
b=rep(0,(n+1))
c=c(0:n)
for (i in 1:(n+1))
{
b[i]=length(a[a==(i-1)])
}
p.pois[k,j]=sum(a)/N
d.pois=dpois(c,p.pois[k,j])*N
stat.pois=c()
for (ii in 1:length(b))
{if (b[i]==0) stat.pois[ii]=0 else stat.pois[ii]=(b[i]-d.pois[i])*(b[i]-d.pois[i])/d.pois[i]}
xto2.pois=sum(stat.pois)
pto2.pois[k,j]=1-pchisq(xto2.pois,n)
if (j¿6)
{ a=deg[,j]
b=rep(0,11)
c=c(0:9)
for (i in 1:10)
{
b[i]=length(a[a==(i-1)])
}
p.pois[k,j]=sum(a)/N
d.pois=dpois(c,p.pois[k,j])*N
stat.pois=c()
for (ii in 1:length(b))
if (b[i]==0) stat.pois[ii]=0 else stat.pois[ii]=(b[i]-d.pois[i])*(b[i]-d.pois[i])/d.pois[i]
xto2.pois=sum(stat.pois)
pto2.pois[k,j]=1-pchisq(xto2.pois,10)
}
}
}
round(pto2.pois,4)
pto2.mat=pto2.pois[,1:10]
result1.p=rbind(apply(pto2.mat,2,mean),apply(pto2.mat,2,sd)/sqrt(K))
result1.p
lambda.mat=p.pois[,1:10]
result2.lambda=rbind(apply(lambda.mat,2,mean),apply(lambda.mat,2,sd)/sqrt(K))
round(result2.lambda,4)
```

Appendix D: R Code for Simulation B

```
library(”ergm”)
n=16
set.seed(333)
```

```
theta.mat=matrix(nrow=500,ncol=3)
test.p=c()
pset=c(0.1,0.3,0.5,0.7,0.9)
count=0
for (pi in 1:5)
{
p=pset[pi]
for (j in 1:100)
{
array=c(rbinom(16*15/2,1,p))
mat=matrix(0,nrow=n,ncol=n)
count1=0
for (irow in 1:(n-1))
{
for (icol in (irow+1):n)
{
count1=count1+1
mat[irow,icol]=array[count1]
}
}
net=network(mat)
net$gal$directed=F
model=ergm(net~edges+kstar(2)+triangles)
count=count+1
theta.mat[count,]=model$coef
}
```

```
}

# simulation B (continued)
set.seed(333)
result=matrix(nrow=1,ncol=19)
for (j in 1:length(theta.mat[,1]))
{
theta0=theta.mat[j,]
model=ergm(flo~edges+kstar(2)+triangles)
theta.mat[test,]=model$coef
set.seed(321)
N=100
n=16
K=4
pto2=matrix(nrow=K,ncol=n)
pto2.pois=matrix(nrow=K,ncol=n)
p=matrix(nrow=K,ncol=n)
p.pois=matrix(nrow=K,ncol=n)
for (k in 1:K)
{
gofly3=gof(model,nsim=N,theta0=theta0)
deg=gofly3$sim.deg
for (j in 1:n)
{
a=deg[,j]
b=rep(0,(n+1))
c=c(0:n)
for (i in 1:(n+1))
{
b[i]=length(a[a==(i-1)])
}
p.pois[k,j]=sum(a)/N
d.pois=dpois(c,p.pois[k,j])*N
xto2.pois=sum((b-d.pois)*(b-d.pois)/d.pois)
pto2.pois[k,j]=1-pchisq(xto2.pois,n)
}
}
ans=round(pto2.pois,4)
ans2=rbind(apply(ans,2,mean),apply(ans,2,sd)/sqrt(K))
setup=matrix(c(rep(c(theta0),c(2,2,2))),2,3)
ans3=cbind(setup,ans2)
result=rbind(result,ans3)
}
```

# results for Simulation B

```
-1.03 -1.10 -13.07 0.94 0.00 0.94 1.00 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
-1.75 -0.19 -15.57 0.70 0.48 0.78 0.85 0.91 1.00 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
-2.34 0.02 -15.73 0.57 0.52 0.89 0.72 0.78 0.98 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
-3.20 0.29 -15.66 0.38 0.13 0.84 0.76 1.00 1.00 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
-0.90 0.03 -0.08 1.00 0.80 0.92 0.74 0.87 0.87 0.94 0.70 0.81 1.00 0.75 1.00 NaN NaN NaN NaN
-0.75 0.01 -0.33 1.00 1.00 0.97 0.96 0.82 0.82 0.93 0.98 1.00 1.00 NaN NaN NaN NaN NaN NaN
0.52 -0.25 0.41 NaN 1.00 0.99 0.80 0.70 0.78 0.96 1.00 1.00 1.00 NaN NaN NaN NaN NaN NaN
1.14 -0.32 0.04 1.00 1.00 0.96 0.23 0.91 0.80 1.00 1.00 1.00 NaN NaN NaN NaN NaN NaN NaN
-0.83 0.13 -0.20 NaN NaN NaN 1.00 0.78 1.00 0.97 0.97 0.94 0.86 0.67 0.98 0.80 1.00 1.00 NaN
0.17 -0.13 0.43 NaN 1.00 1.00 0.91 0.56 0.91 0.74 1.00 0.95 0.96 0.64 0.89 1.00 0.53 NaN NaN
-0.98 0.10 -0.08 NaN NaN 1.00 1.00 0.51 0.66 0.99 0.98 0.91 0.86 0.66 0.35 0.74 1.00 0.95 NaN
3.08 -0.14 -0.20 NaN NaN NaN NaN 1.00 0.93 1.00 0.86 0.60 0.87 0.99 0.99 1.00 NaN NaN NaN
5.99 -0.32 0.28 NaN NaN NaN NaN NaN NaN NaN 1.00 1.00 0.93 1.00 0.90 0.55 0.89 0.94 0.91
14.40 -0.75 0.29 NaN NaN NaN NaN NaN NaN NaN NaN 0.71 1.00 0.75 0.49 0.70 0.90 1.00 NaN
2.22 0.06 -0.41 NaN NaN NaN NaN NaN NaN 1.00 1.00 1.00 0.89 0.73 0.73 0.99 0.99 1.00 NaN
7.90 -0.57 0.56 NaN NaN NaN NaN NaN NaN 1.00 1.00 0.68 0.89 0.61 0.95 0.61 1.00 1.00 NaN
221.6 -15.99 16.40 NaN NaN NaN NaN NaN NaN NaN NaN 1.00 1.00 0.98 0.94 0.57 0.97 0.75 0.16
-5.25 0.48 -0.32 0.00 0.00 0.80 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
-5.34 0.63 -0.76 0.00 0.00 0.91 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
229.1 -16.03 15.81 NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN 1.00 1.00 0.84 0.89 0.54 0.73
```

5.B Table of terminology

Terminology	Description
A social network	is a social structure made up of individuals (or organizations) called "nodes," which are tied (connected) by one or more specific types of interdependency, such as friendship, kinship, common interest, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige.
Vertices or nodes	are the fundamental units in networks, representing objects which possibly have associations with each other.
Edges or ties	are connections or associations between vertices. In social networks, they represent relationships between objects.
Size of a network(n)	is the number of vertices in a social network.
Density or network density	is the proportion of ties in a network relative to the total number of possible ties.
Structure statistics $g(y)$	are a collection of statistics built on a network, representing the structure of the network.
Clustering coefficient	is a measure of the likelihood that two associates of a vertex are associates themselves.
Edge variable(Y_{ij})	is a binary random variable of an edge being present or absent.
G	denotes the set of edges in a social network. It is also used to denote a network.
Y	denotes the vector of edge variables in a random network.
y	A realization of Y.
Dependence graph D	is a fixed graph for a particular random network G. The vertices in D are the possible edges in G, and the edges in D are the pairs of edges in G that are conditionally dependent.
A	is defined as a nonempty subset of the possible edges of G such that either the subset represents a singe edge or the subset contains edges that are conditionally dependent in G.
Y_{ij}^c	denotes the compliment of Y_{ij} in Y.
Y_{ij}^1 or Y_{ij}^0	denotes a network with Y_{ij} set to 1 or 0.

Table 5.1: Table of terminologies

Bibliography

1. O. Frank and D. Strauss. Markov graphs. Journal of the American Statistical Association, 81(395):832-842, 1986.

2. C. J. Geyer and E. A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data (with discussion). Journal of the Royal Statistical Society, Series B, 54:657-699, 1992.

3. S. M. Goodreau, M. S. Handcock, D. R. Hunter, C. T. Butts, and M. Morris. A statnet tutorial. Journal of Statistical Software, 24(9):1-26, 2008.

4. M. S. Handcock, G. L. Robins, T. A. B. Snijders, J. Moody, and J. Besag. Assessing degeneracy in statistical models of social networks. Journal of the American Statistical Association, 76:33-50, 2003.

5. P. W. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs (with discussion). Journal of the American Statistical Association, 76(373):33-65, 1981.

6. D. R. Hunter and M. S. Handcock. Inference in curved exponential family models for networks. Journal of Computational and Graphical Statistics, 15(3):565-583, 2006.

7. D. R. Hunter, S. M. Goodreau, and M. S. Handcock. Goodness of fit of social network models. Journal of the American Statistical Association, 103(481):248-258, 2008.

8. D. R. Hunter, M. S. Handcock, C. T. Butts, S. M. Goodreau, and M. Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. Journal of Statistical Software, 24(3), 2008. http://www.jstatsoft.org/v24/i03/paper.

9. M. Morris, M. S. Handcock, and D. R. Hunter. Specification of exponential-family random graph models: Terms and computational aspects. Journal of Statistical Software, 24(4), 2008. http://www.jstatsoft.org/v24/i04.

10. P. E. Pattison and S. S. Wasserman. Logit models and logistic regressions for social networks: II. Multivariate relations. British Journal of Mathematical and Statistical Psychology, 52(2):169-193, 1999.

11. G. L. Robins, T. A. B. Snijders, P. Wang, M. S. Handcock, and P. E. Pattison. Recent developments in exponential random graph models for social networks. Social Networks, 29(2):192-215, 2007.

12. T. A. B. Snijders. The statistical evaluation of social network dynamics. Sociological Methodology, 31:361-395, 2001.

13. T. A. B. Snijders. Markov Chain Monte Carlo estimation of exponential random graph models, Journal of Social Structure, 3, (2002).

14. T. A. B. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock (2004), New specifications for exponential random graph models, Center for Statistics and the Social Sciences working paper no. 42, University of Washington.

15. T. A. B. Snijders. Models for longitudinal network data. In P. J. Carrington, J. Scott, and S. S. Wasserman, editors, Models and Methods in Social Network Analysis, chapter 11. Cambridge University Press, New York, 2005.

16. T. A. B. Snijders. Statistical methods for network dynamics. In S. R. Luchini et al., editors, Proceedings of the XLIII Scientific Meeting, Italian Statistical Society, pages 281-296, Padova: CLEUP, 2006.

17. T. A. B. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock. New specifications for exponential random graph models. Sociological

Methodology, 36:99-153, 2006.

18. D. Strauss and M. Ikeda. Pseudolikelihood estimation for social networks. Journal of the American Statistical Association, 85(409):204-212, 1990.

19. S. S. Wasserman and P. E. Pattison. Logit models and logistic regression for social networks: I. An introduction to Markov graphs and p*. Psychometrika, 61(3):401-425, 1996.

20. P.E. Pattison, G.L. Robins, 2002. Neighbourhood-based models for social networks. Sociological Methodology 32,301C337.

21. D. R. Hunter, R. Hummel. Exponential-Family Random Graph Models for Social Networks. Presentation material on PRIMES, Apr. 26, 2007.