

# Local Dependence in Exponential Random Network Models

---

A Thesis  
Presented to  
The Division of Mathematics and Natural Sciences  
Reed College

---

In Partial Fulfillment  
of the Requirements for the Degree  
Bachelor of Arts

---

Nicholas Solomon

May 2017



Approved for the Division  
(Mathematics - Statistics)

---

Albyn Jones



# Acknowledgements

First and foremost, this thesis is due to all of the professors at Reed who have taught, encouraged, and guided me during my time here. Obviously, my adviser, Albyn Jones, had the largest impact on this document. Thank you for putting up with my silly mistakes and keeping me on track. I also have to thank you for teaching me statistics in the first place. Math 392 is the reason I am where I am today.

I also have to thank Andrew Bray, Chester Ismay, and Jim Fix for their contributions to this project. On more than one occasion, each of them took the time out of their busy schedules to listen to me complain about my thesis. Without them, I would not have been able to get as far as I did.

Finally, I owe a debt to Jerry Shurman, who, as my sophomore year adviser, insisted I have some useful skill upon graduating from Reed. Without him, I never would have taken a statistics class and this thesis would be about complex analysis.

Equally as important as my professors, this project is due to my family. My parents have provided unwavering love and support during my time here. They also paid my tuition.

Last but not least, I want to thank all the friends I made here. You know who you are, and if you think this is meant for you, then it is.



# Table of Contents

<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Random graphs	1
1.2 Network modeling	2
<b>Chapter 2: Exponential random graph models</b>	<b>5</b>
2.1 Finding parameter estimates	6
2.2 Examples	11
<b>Chapter 3: Locally dependent exponential random network models</b>	<b>15</b>
3.1 Definitions and notation	16
3.2 Preliminary theorems	17
3.3 Consistency under sampling	18
3.4 Asymptotic normality of statistics	20
<b>Chapter 4: Discussion and examples</b>	<b>27</b>
4.1 Examples of convergence	27
4.2 Standard error estimation	29
4.3 Discussion of future work	30
<b>Appendix A: Code</b>	<b>33</b>
A.1 Network simulation code	33
A.2 Jackknife standard error code	36
<b>References</b>	<b>39</b>





# List of Tables

2.1	The results of fitting model 2.13 using MPLE. . . . .	12
2.2	The results of fitting model 2.13 using MCMC-MLE. . . . .	12
2.3	The results of fitting model 2.13 using Bayesian methods. . . . .	12
4.1	Standard error estimates using the jackknife at the vertex level. . . .	30
4.2	Standard error estimates using the jackknife at the neighborhood level.	30



# List of Figures

1.1	An undirected graph on four vertices. . . . .	1
1.2	A directed graph on four vertices . . . . .	2
2.1	A plot of the Florentine marriage network where node size indicates wealth in thousands of lira. . . . .	11
2.2	Posterior density plots of the Florentine marriage model parameter estimates. . . . .	12
2.3	Goodness-of-fit assessment for the Bayesian model fit to the Florentine marriage data. These plots compare simulated distributions of graph statistics that were not modeled to the observed values (in red). . . .	13
3.1	A locally dependent random network with neighborhoods $A_1, A_2, \dots, A_K$ and two binary node attributes, represented as gray or black and circle or diamond. . . . .	15
3.2	The projection mapping from $\mathcal{B} = \{A_1, A_2\}$ to $\mathcal{A} = \{A_1\}$ . . . . .	19
4.1	A simulated network with 100 vertices, colored according to neighborhood. The vertex fill indicates the presence or absence of the simulated attribute. . . . .	28
4.2	Density plots of simulated statistics of locally dependent random networks. . . . .	28
4.3	Normal Q-Q plots for simulated statistics of locally dependent random networks. . . . .	29



# Abstract

Graph representations are used across disciplines for the analysis and visualization of relational data. Exponential random graph models allow for a general method of modelling the underlying stochastic process that has generated the observed data conditional on observer attributes of the vertices, or nodes. Recent developments in ERGMs have introduced the notions of local dependence and the exponential random network model, or ERNM. Local dependence enforces the assumption of independence between edges that connect nodes that are, in some sense, “far apart.” This is formalized by the introduction of a neighborhood structure on the graph: a partition of the vertices with the property that edges between two neighborhoods are stochastically independent of all other edge variables. This independence allows for the proof of a desirable consistency condition and a central limit theorem for statistics of the graph. The random network model allows for the joint modelling of both the graph and random attributes of the vertices. This has useful applications in network analysis, as it allows researchers to make inferences about how graphical features affect vertices and vice versa. This thesis combines these two developments to show that ERNMs with the local dependence property have the same useful consistency property and that a similar central limit theorem also holds.



# Chapter 1

## Introduction

### 1.1 Random graphs

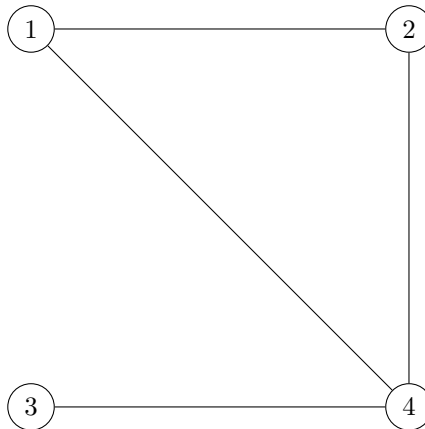


Figure 1.1: An undirected graph on four vertices.

The mathematical object we call a graph is a set of vertices and edges, so for a graph  $G$ , we write  $G = (V, E)$  where  $V$  is the set of vertices and  $E$  are ordered or unordered pairs  $(i, j)$  for  $i, j \in V$ . These objects can be represented visually by pictures like Figure 1.1. If the pairs in  $E$  are ordered, then we say that  $G$  is directed (or sometimes that  $G$  is a digraph), otherwise we say that  $G$  is undirected. The example in Figure 1.1 is undirected; when drawing directed graph, we can indicate the direction of the the edge  $(i, j)$  by drawing an arrow from node  $i$  to node  $j$ , as in Figure 1.2. If we only consider the presence or absence of an edge between vertices, then we say that the graph  $G$  is unweighted. In a weighted graph, each edge is assigned a numeric value, called the weight. The case of an unweighted graph is equivalent a weighted graph where each edge weight  $w_{ij} \in \{0, 1\}$ .

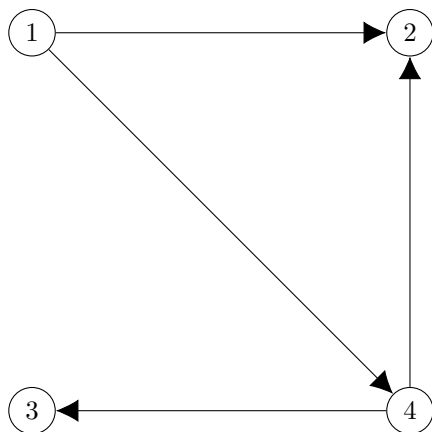


Figure 1.2: A directed graph on four vertices

Generally, the most convenient way to represent a graph is as an adjacency matrix. For a graph with  $n$  vertices, the adjacency matrix is the  $n \times n$  matrix with the weight of edge  $(i, j)$  in the  $i, j$  position. Note that if a graph  $G$  is undirected, then its adjacency matrix is symmetric and if  $G$  is an unweighted graph, then all its entries are either 1 or 0. For example, the graph in Figure 1.1 has adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \quad (1.1)$$

Here, we have adopted the convention that vertices do not have an edge with themselves, so the diagonal elements of this matrix are all zero. Sometimes we will also represent a graph as an edge list, which is simply the set of pairs in  $E$  with their weights.

If we assume that a graph  $G$  is generated by some kind of stochastic process, then we consider the whole graph as a random variable. This is equivalent to considering the random matrix  $Y$ , where  $Y$  is the adjacency matrix of  $G$ . We can consider the  $n^2$  random variables  $Y_{ij}$  for  $i, j \in \{1, \dots, n\}$ , taking values in the set of possible weights for the random graph in consideration. For example, if  $G$  is an undirected binary random graph, then this can be simplified to the  $n^2/2 - n$  Bernoulli random variables  $Y_{(i,j)}$  for  $1 \leq i \leq j \leq n$ . By breaking the random graph into these simpler parts, we can begin to apply familiar statistical ideas to these complex objects.

## 1.2 Network modeling

Graphs are a convenient way to represent relational data, or data that contain information about a relationship between actors. This brings us to the area of network analysis, where random graphs are used to model many different kinds of relationships. For example, in sociology, networks are commonly used to model relationships among people, like marriages within a group of people. Political scientists use random graphs to represent things like the network of trade agreements between nations. Finally,



in epidemiology, random networks can model an underlying contact network, say of people who shake hands or otherwise come into contact in a way that could spread a disease, over which a transmissible infection spreads. This allows the researcher to estimate how the medical features of the disease and the sociological features of the population interact to promote or inhibit the spread of disease. As an example of this, see Groendyke, Welch, & Hunter (2012), where, among other things, they examine if closing schools would have had an effect on the spread of an 1861 measles outbreak.

To formalize these notions, we begin with a random graph  $Y$  on  $n$  vertices (called nodes in the field of network analysis) considered as a random matrix with sample space  $\mathcal{Y}$ , where  $\mathcal{Y}$  is the set of all possible graphs on  $n$  vertices. There may be some restrictions on  $\mathcal{Y}$ , depending on the application being considered. Usually, we prohibit edges that connect a vertex to itself by forcing the diagonal entries of  $Y$  to be 0, as in (1.1). However, specific contexts may require nonobvious constraints put in place by the researcher; for example, when using a network to model the romantic relationships within a group of people, the researcher must exclude networks with ties between nodes that do not have the appropriate gender, depending on the respective preferences of each pair of nodes.

This thesis explores two classes of network models: the exponential random graph model (ERGM) and the exponential random network model (ERNM). These models are quite similar in that they both model the structural features of the network, such as node degree and triangle formation, as a function of nodal covariates, say age or gender. However, the ERNM also models the effect of network features on the nodes. For example, Fellows & Handcock (2012) show that teens with friends who smoke or drink are more likely to smoke or drink themselves.

The notion of local dependence was introduced by Schweinberger & Handcock (2015) in the context of ERGMs. Here we extend this idea to ERNMs and provide new proofs of their two main theorems in this context. The first theorem shows that the stochastic process of sampling smaller subnetworks from a large network satisfies a desirable consistency condition. This means that as we sample larger and larger subnetworks, the model parameter estimates become better and better approximations of the true parameter values. Shalizi & Rinaldo (2013) showed that this is not the case for most ERGMs. The second theorem shows that certain types of statistics of locally dependent random networks have an asymptotically normal distribution, which is also desirable from the point of view of making inferences based on these models.



## Chapter 2

# Exponential random graph models

The first model of interest to this thesis is the exponential random graph model (ERGM), introduced by S. Wasserman & Pattison (1996). Broadly, given a network, this model allows us to specify structural features that are of interest and test in a principled way if there are more of those kinds of features present than would be expected if the network were formed by chance. This allows the researcher to draw conclusions about the latent process that generated the network.

This model can be expressed by the equation

$$P(Y = y|\theta) = \frac{e^{\theta \cdot g(y)}}{C(\theta, \mathcal{Y})}, \quad (2.1)$$

where  $g(y)$  is a vector of statistics that depend on the network  $y$  and  $\theta$  is a vector of parameters. The function  $C(\theta, \mathcal{Y})$  is a normalizing constant. We may easily extend the model to include nodal covariates (like demographic information) by introducing the fixed  $n \times p$  matrix  $x$  which is used to calculate some of the components of  $g$ .

Note that the constant in the denominator depends both on the support of the random variable  $Y$ , which is fixed and—in principle—known, and the unknown parameter  $\theta$ . In general, this constant will have the form

$$C(\theta, \mathcal{Y}) = \sum_{y \in \mathcal{Y}} e^{\theta \cdot g(y)}, \quad (2.2)$$

which is impossible to compute in all but the most trivial cases, as the sum is over all possible networks. For the simplest undirected binary network with  $n$  nodes, we have  $|\mathcal{Y}| = 2^{n(n-1)/2}$ , so for 10 nodes, the constant  $C$  will be a sum of  $2^{45}$  terms. As each term requires a non-trivial calculation, we are unable to calculate the value of  $C$  directly for a network of any reasonable size.

With the model in place, we can begin to estimate the parameter  $\theta$  via maximum likelihood estimation. To do that, we define the likelihood function, which takes the parameter space into the reals. Our goal is to find the value of  $\theta$  that maximizes this function given the observed data. We call this  $\theta$  the maximum likelihood estimator, or MLE. Intuitively, we are looking for the value of  $\theta$  that makes the network we have observed most likely according to our model. In our case this function is

$$L(\theta|y) = \frac{e^{\theta \cdot g(y)}}{C(\theta, \mathcal{Y})}. \quad (2.3)$$

Note that  $C$  also varies with  $\theta$ .

The fact that this term varies with  $\theta$  means that a naive optimization algorithm will be prohibitively slow. Exploring the parameter space almost always involves evaluating the function at many different points. For example, the common gradient descent algorithm involves evaluating the function many times in the neighborhood of a point to approximate the gradient. This then allows the algorithm to choose a new point where the function is larger. This process then repeats. Even when given relatively simple functions, this algorithm can require hundreds of function calls. In our case, that is not feasible, as we cannot even evaluate our function once.

## 2.1 Finding parameter estimates

The intractable normalizing constant (2.2) makes fitting this model directly extremely difficult. With no general analytic method to maximize the likelihood, we must turn to numerical approximations. However, these algorithms involve evaluating  $L(\theta)$  at many different points. When  $\theta$  varies the value of  $C$  also changes, so each function evaluation must recompute this enormous sum. This makes standard numerical approaches useless for the problem at hand. Despite this there are several approaches that make use of a variety of approximations to (we hope) find reasonable parameter estimates. Here we will present both frequentist and Bayesian methods for estimating  $\theta$ . To begin, we introduce another piece of notation.

**Definition 2.1.** The change statistic is the vector

$$\delta_g(y)_{ij} = g(y_{ij}^+) - g(y_{ij}^-), \quad (2.4)$$

where  $g$  is the vector of statistics, as before, and  $y_{ij}^+$  and  $y_{ij}^-$  are the observed network  $y$  with the edge from  $i$  to  $j$  taken to be present and absent, respectively.

Now, we are able to show that, in the case of an unweighted graph

$$\text{logit} \left( P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c) \right) = \theta \cdot \delta_g(y)_{ij},$$

where  $Y_{ij}^c$  is the random variable without considering the tie from  $i$  to  $j$  and the function

$$\text{logit}(x) = \log \left( \frac{x}{1-x} \right).$$

This follows from noting that

$$\begin{aligned} \text{logit} \left[ P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c) \right] &= \log \left[ \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{1 - P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)} \right] \\ &= \log \left[ \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} \right], \end{aligned}$$

as the complement of the set of graphs where  $Y_{ij} = 1$  is the set of graphs where  $Y_{ij} = 0$ . Now,

$$\begin{aligned} \log \left[ \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} \right] &= \log[P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)] - \log[P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)] \\ &= \log \left[ e^{\theta \cdot g(y_{ij}^+)} \right] - \log \left[ e^{\theta \cdot g(y_{ij}^-)} \right] \\ &= \theta \cdot (g(y_{ij}^+) - g(y_{ij}^-)) \\ &= \theta \cdot \delta_g(y)_{ij}, \end{aligned}$$

as desired.

Thus, each component of  $\theta$  represents the increase in the log-odds of a tie from  $i$  to  $j$  being present in the network that is associated with a change in  $y_{ij}^c$  that increases the corresponding component of  $g(y)$  by one unit. An example of this kind of interpretation is presented in Section 2.2.

### 2.1.1 Frequentist methods

The change statistic allows us to define the pseudolikelihood, introduced by Strauss & Ikeda (1990). The pseudolikelihood function represents the model where the probability that a tie from  $i$  to  $j$  exists is independent of all the other ties. Even when this assumption is known to be false, the hope is that the maximum pseudolikelihood estimator will be a good approximation of the more general MLE. This yields the likelihood function

$$\text{PL}(\theta) = \prod_{i \neq j} P(Y_{ij} = y_{ij} | \theta).$$

Taking the logit transforms this into a standard logistic regression where the response variable is the vector containing a 1 or a 0 depending on the existence of the tie between nodes  $i$  and  $j$  and the explanatory variables are the vectors of change statistics. This can then easily be fit using widely available generalized linear modeling tools. The estimator returned by this logistic regression is called the maximum pseudolikelihood estimator (MPLE).

However, this method exhibits a few problems. In many cases, as already discussed, the assumption of dyadic independence is not justifiable, so the coefficient estimates cannot be expected to be accurate in general. The class of models called dyadic independence models are those in which the form of  $g$  makes the likelihood exactly equal to the pseudolikelihood, and so we can fit them exactly using this method. When the dyadic independence assumption does not hold, we are no longer maximizing the likelihood associated with our model, so there is no reason to expect that we have the properties that make maximum likelihood estimators so nice. Specifically, there is no reason to believe that the parameter estimates actually asymptotically approach the true values or that the estimates are approximately normally distributed with the reported standard errors. An example of the trouble with standard errors will be seen in Section 2.2.

It is clear that a better method for computing the maximum likelihood estimator is highly desirable. This brings us to the Markov chain Monte Carlo MLE (MCMC-MLE) algorithm developed by Geyer & Thompson (1992). They introduce a method with applications to intractable distributions beyond ERGMs, but we will remain within that specific context. The crux of the algorithm lies in choosing a fixed value  $\theta_0$  (typically the MPLE) that we hope is close to the true value of  $\theta$  and then maximizing the likelihood ratio

$$\log \left[ \frac{L(\theta)}{L(\theta_0)} \right]. \quad (2.5)$$

We begin with the log likelihood of model (2.1)

$$\ell(\theta) = \theta \cdot g(y) - \log(C(\theta, \mathcal{Y})). \quad (2.6)$$

We can rewrite the ratio (2.5) as

$$\ell(\theta) - \ell(\theta_0) = (\theta - \theta_0) \cdot g(y) - \log \left( \frac{C(\theta, \mathcal{Y})}{C(\theta_0, \mathcal{Y})} \right). \quad (2.7)$$

Clearly, this is maximized at the same value of  $\theta$  as the original likelihood, but now we can show that

$$\frac{C(\theta, \mathcal{Y})}{C(\theta_0, \mathcal{Y})} = \mathbb{E}_{\theta_0} \left[ e^{(\theta - \theta_0) \cdot g(y)} \right]. \quad (2.8)$$

This follows from noting that

$$\begin{aligned} \frac{C(\theta, \mathcal{Y})}{C(\theta_0, \mathcal{Y})} &= \frac{\sum_{z \in \mathcal{Y}} e^{\theta \cdot g(z)}}{\sum_{w \in \mathcal{Y}} e^{\theta_0 \cdot g(w)}} \\ &= \frac{\sum_{z \in \mathcal{Y}} e^{\theta \cdot g(z)} e^{(\theta_0 - \theta) \cdot g(z)}}{\sum_{w \in \mathcal{Y}} e^{\theta_0 \cdot g(w)}} \\ &= \sum_{z \in \mathcal{Y}} \left[ e^{\theta \cdot g(z)} e^{-\theta_0 \cdot g(z)} \left( \frac{e^{\theta_0 \cdot g(z)}}{\sum_{w \in \mathcal{Y}} e^{\theta_0 \cdot g(w)}} \right) \right]. \end{aligned}$$

Recognizing the term in parentheses as  $P_{\theta_0}(Y = z)$  makes it clear that we have

$$\frac{C(\theta, \mathcal{Y})}{C(\theta_0, \mathcal{Y})} = \sum_{z \in \mathcal{Y}} e^{(\theta - \theta_0) \cdot g(z)} P_{\theta_0}(Y = z) = \mathbb{E}_{\theta_0} \left[ e^{(\theta - \theta_0) \cdot g(Y)} \right],$$

as desired. Now we need only estimate this expectation using the weak law of large numbers. We do this by generating a Markov chain with stationary distribution  $P_{\theta_0}$  and sampling sufficiently many times to get a good approximation of (2.8).

Hence, the problem is reduced to constructing a Markov chain with  $P_{\theta_0}$  as its stationary distribution. We use the algorithm developed by Morris, Handcock, & Hunter (2008). It is essentially a Metropolis-Hastings algorithm where the proposal distribution chooses a pair of nodes, also called a dyad, then if they are connected, removes that tie, and if they are not connected, adds a tie between them. The obvious proposal distribution chooses two vertices at random, producing a simple random

walk on  $\mathcal{Y}$ . As an alternative to this naïve proposal distribution, Morris et al. (2008) and M. S. Handcock et al. (2016) have developed the tie-no-tie, or TNT, proposal distribution. In this method, the dyad is chosen by first selecting whether we will toggle a dyad with or without a tie in the original network. Then a pair of nodes from within the set of those which are either connected or not connected (depending on the previous step) is chosen at random. According to its inventors, this modification of the random walk on  $\mathcal{Y}$  causes the chain to mix better, especially in sparse networks.

This allows us to approximate samples from the probability distribution on  $\mathcal{Y}$  implied by the parameter  $\theta_0$ . Then we can estimate the expected value in (2.8) and calculate what we hope is a good estimate for the actual MLE. However, this method is not without its caveats. It is shown in Hunter, Handcock, Butts, Goodreau, & Morris (2008) that this algorithm can be sensitive to the initial parameter  $\theta_0$  and a poor choice of this parameter can cause the approximation of the likelihood function (2.7) to never achieve a maximum on the parameter space. This happens when the function that we are optimizing, an approximation of the true log likelihood ratio, is bad enough that it becomes unbounded and so numeric optimization routines fail to find the maximum.

### 2.1.2 Bayesian methods

The parameter  $\theta$  may also be estimated by Bayesian methods. However, this introduces the issue of sampling from a “doubly intractable” posterior distribution, where the problem of incalculable normalizing constants in the posterior is compounded by the functional form of our model (2.1). Markov chain Monte Carlo methods for these distributions have been studied by Murray, Ghahramani, & MacKay (2012), where the easy to implement exchange algorithm was introduced, and Caimo & Friel (2011), where this algorithm was applied to ERGMs. The algorithm very cleverly avoids the intractable constants in (2.1) by augmenting the posterior with an auxiliary variable from the same sample space as the parameter of interest. By doing this in just the right way, we are able to cancel all intractable constants from the acceptance probability (2.10).

To be precise, let the observed network  $y$  be taken from the distribution  $P_\theta$  and let the prior for  $\theta$  have distribution  $p(\theta)$ . We must construct a Markov chain with stationary distribution equal to the posterior given by

$$\pi(\theta|y) = \frac{P_\theta(Y = y)p(\theta)}{\int P_\theta(Y = y)p(\theta) d\theta}. \quad (2.9)$$

In a standard Metropolis-Hastings implementation with proposal distribution  $\theta' \sim q(\cdot|\theta)$  we would have the acceptance probability

$$a = \min \left[ 1, \frac{P_{\theta'}(Y = y)p(\theta')q(\theta|\theta')}{P_\theta(Y = y)p(\theta)q(\theta|\theta')} \right], \quad (2.10)$$

where the likelihoods in the numerator and the denominator are evaluated at different values of  $\theta$ . For the ERGM, the normalizing constants will not cancel. To get around

this, we take  $w \sim P_{\theta'}$  and  $\theta' \sim q(\cdot|\theta)$ . Here,  $w$  is another network that we sample from the ERGM distribution with parameter  $\theta'$ , while  $\theta'$  is drawn from an arbitrary proposal distribution that can depend on the current value of  $\theta$ . Now the distribution we are sampling from is

$$\pi(\theta, \theta', w|y) = \frac{P_{\theta}(Y = y)P_{\theta'}(Y = w)q(\theta'|\theta)p(\theta)}{\int P_{\theta}(Y = y)P_{\theta'}(Y = w)q(\theta'|\theta)p(\theta) d\theta}. \quad (2.11)$$

The conditional distribution of  $\theta$  is the posterior we are after. To begin, we draw  $\theta'$  from the (arbitrary) distribution  $q$ , which can depend on  $\theta$ . Then we draw  $w$  from the distribution implied by  $\theta'$  (using already developed MCMC routines) and we propose an exchange of the generated data  $w$  and the observed data  $y$  between the parameters  $\theta$  and  $\theta'$  which is accepted with probability

$$a = \min \left[ 1, \frac{P_{\theta'}(Y = y)P_{\theta}(Y = w)p(\theta')q(\theta'|\theta)}{P_{\theta}(Y = y)P_{\theta'}(Y = w)p(\theta)q(\theta|\theta')} \right]. \quad (2.12)$$

Now all the incalculable constants in (2.12) cancel, and we can approximate the posterior distribution as desired.

It was also demonstrated by Caimo & Friel (2011) that in most cases the use of adaptive direction sampling (ADS) allows for a more thorough exploration of the state space in fewer iterations. This improvement involves running multiple chains (say twice as many as the number of parameters in the model) that, at each iteration  $i$ , are updated separately. When updating the  $j$ -th chain, two other chains,  $k$  and  $\ell$  are randomly selected, then we propose  $\theta_j^i = \gamma(\theta_k^i - \theta_\ell^i) + \epsilon$  where  $\gamma$  is a fixed tuning parameter and  $\epsilon$  is a small random quantity, both chosen to achieve a higher level of mixing than is usually seen when running a single chain in isolation. This improves the overall mixing of the separate chains and allows the process to thoroughly explore the sample space in fewer iterations. This is the default method in the **Bergm** package.



## 2.2 Examples

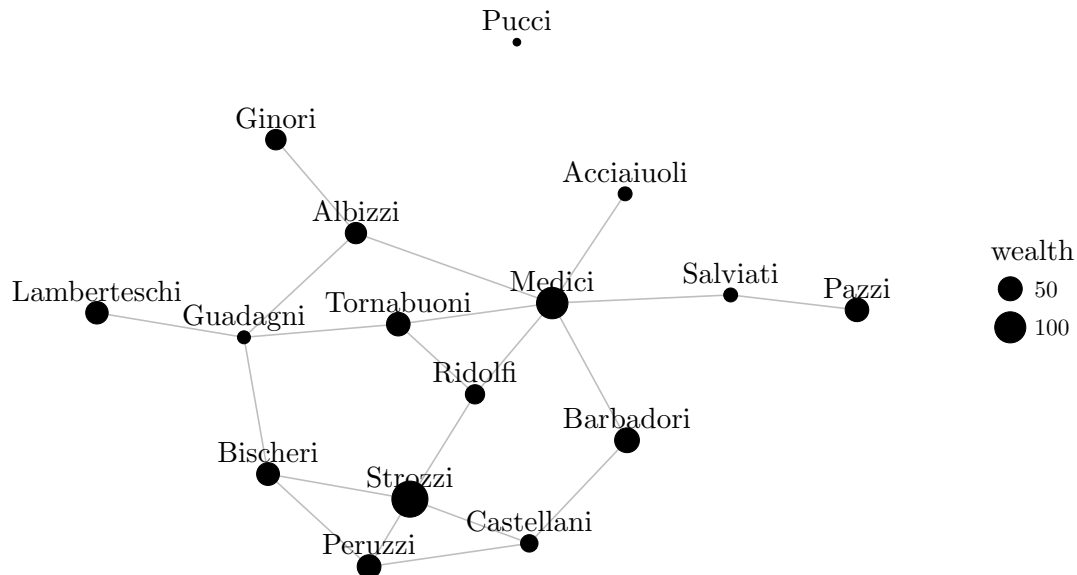


Figure 2.1: A plot of the Florentine marriage network where node size indicates wealth in thousands of lira.

For illustrative purposes, we will use the Florentine wedding data included in the `R ergm` package by M. S. Handcock et al. (2016), which we also use to fit MPLE and MCMC-MLE models. We will use the `Bergm` package by Caimo & Friel (2014) to fit the Bayesian models. These data consist of an undirected network of marriages between Florentine families during the Renaissance, along with several nodal covariates, including the family wealth in thousands of lira in the year 1427. This network is drawn in Figure 2.1. We will fit the ERGM with network statistics

$$g(y) = \left( \sum_{i < j} y_{ij}, \sum_{i < j < k} y_{ij} y_{jk} y_{ki}, \sum_{i < j} y_{ij} |x_i - x_j| \right), \quad (2.13)$$

where  $y$  is the network and  $x$  is the corresponding vector of wealth measurements. Simply put, this creates a term for the number of edges, the number of triangles, and the difference in wealth between connected nodes in the network. The edge term acts as a sort of intercept in the model by measuring the overall propensity of actors to form ties, the triangle term measures the propensity of actors in the graph to form triangles, and the wealth difference term accounts for how difference in family fortune affects the probability of tie formation. Tables 2.1, 2.2, and 2.3 show the coefficient estimates and standard errors. Bayesian estimation is done using a very flat multivariate normal prior centered on the origin with variance-covariance matrix

$$\Sigma = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}.$$

Table 2.1: The results of fitting model 2.13 using MPLE.

term	estimate	std.error
edges	-2.36	0.44
triangle	0.16	0.44
absdiff.wealth	0.02	0.01

Table 2.2: The results of fitting model 2.13 using MCMC-MLE.

term	estimate	std.error
edges	-2.27	0.45
triangle	-0.04	0.59
absdiff.wealth	0.02	0.01

Figure 2.2 shows posterior density estimates for the Bayesian model.

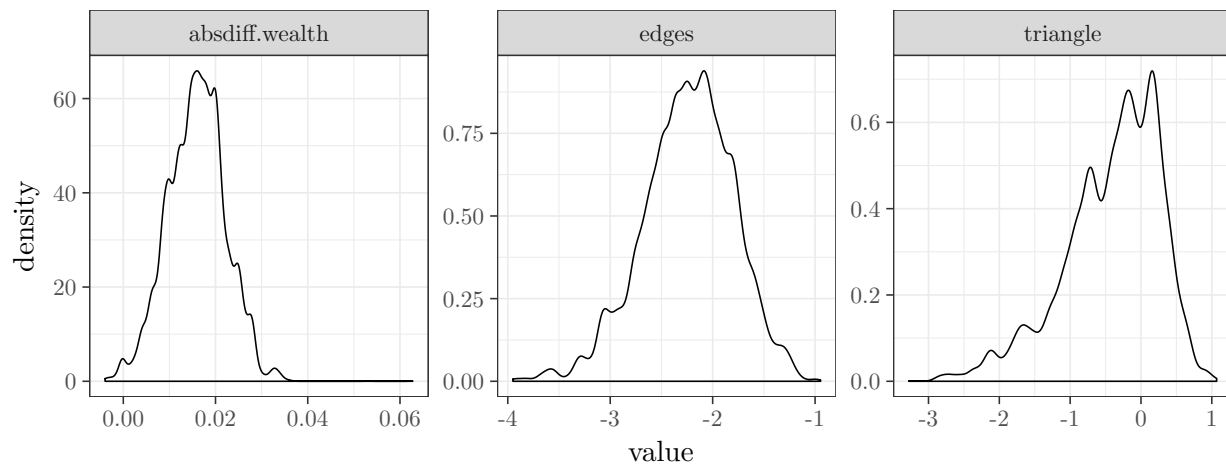


Figure 2.2: Posterior density plots of the Florentine marriage model parameter estimates.

Table 2.3: The results of fitting model 2.13 using Bayesian methods.

term	posterior mean	posterior s.d.
edges	-2.24	0.44
triangle	-0.41	0.67
absdiff.wealth	0.02	0.01

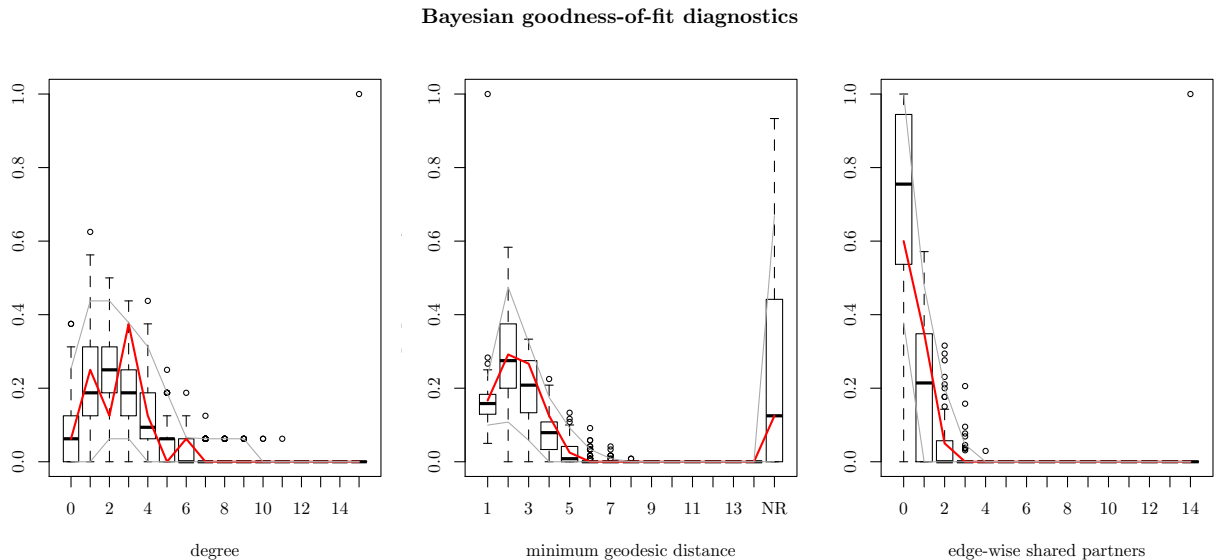


Figure 2.3: Goodness-of-fit assessment for the Bayesian model fit to the Florentine marriage data. These plots compare simulated distributions of graph statistics that were not modeled to the observed values (in red).

We can see that all of these methods produce similar outcomes, with a clearly nonzero edge term, which makes sense as this is akin to an intercept term in a standard linear model, and a small but significant term corresponding to the difference in wealth. Notice that the standard error of the triangle term (which creates dependence between dyads) is much larger when the model is fit using MCMC-MLE. This supports the notion that standard errors reported by MPLE are unreliable in dyadic dependence models, as was discussed in Section 2.1.1. Interpreting these coefficients allows us to infer that a difference in wealth of 1 unit (in this case 1000 lira), changes the log-odds of forming a tie (according to the MCMC-MLE model) by a factor of  $\theta_3 \approx -0.042$ . The `ergm` and `Bergm` packages also provide tools for assessing the goodness-of-fit of exponential random graph models. These tools simulate many networks from the distribution implied by the model with the parameter estimate and then compare the distributions of several network statistics that *were not* modeled to the observed network. These statistics are by default minimum geodesic distance, edgewise shared partners, and degree distribution. Figure 2.3 shows plots of these comparisons for the Bayesian model. As we can see, our model matches the simulated distributions fairly well.



# Chapter 3

## Locally dependent exponential random network models

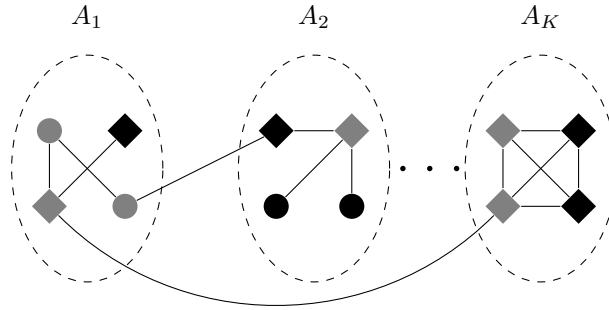


Figure 3.1: A locally dependent random network with neighborhoods  $A_1, A_2, \dots, A_K$  and two binary node attributes, represented as gray or black and circle or diamond.

To begin, we define a random network, developed by Fellows & Handcock (2012). By way of motivation, note that in the ERGM the nodal variates are fixed and are included in the model as explanatory variables in making inferences about network structure. Furthermore, there is a class of models that we do not discuss here that consider the network as a fixed explanatory variable in modeling (random) nodal attributes. It is not difficult to come up with situations where a researcher would like to *jointly* model both the network and the node attributes. Thus we define a class of networks in which both the network structure and the attributes of the individual nodes are modeled as random quantities.

**Definition 3.1** (Random network). Let  $\mathcal{N}$  be a countable collection of nodes (which we take to be a subset of  $\mathbb{N}$ ). Let  $Y$  be the random graph on the nodes  $\mathcal{N}$  with support  $\mathcal{Y}$ . Then for each element  $n \in \mathcal{N}$ , let there be a corresponding random vector of node attributes  $X_n \in \mathbb{R}^q$  and collect them into the  $n \times q$  random matrix  $X$  with support  $\mathcal{X}$ . The *random network* is the random variable  $Z = (Y, X)$  with support  $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$ .

Now we wish to model these objects, so we follow the ERGM and turn to the exponential family (Fellows & Handcock, 2012). We write

$$P(Y = y, X = x|\eta) \propto e^{\eta \cdot g(y,x)}. \quad (3.1)$$

This looks very similar to the ERGM, but note the explicit dependence on the quantity  $x$ . More concretely, we can include terms that depend *only* on  $x$ , which would have no place in an ERGM. We can further express the difference of the two models by rewriting the left hand side of 3.1 as

$$P(X = x, Y = y|\eta) = P(Y = y|X = x, \eta)P(X = x|\eta),$$

where the first term on the right hand side is the ERGM and the second term is

$$P(X = x|\eta) = \frac{C(\mathcal{Z}, \eta, x)}{C(\mathcal{Z}, \eta)},$$

where

$$C(\mathcal{Z}, \eta, x) = \int_{\{(v,u) \in \mathcal{Z}: u=x\}} P(X = x|\eta).$$

Roughly, this is the proportion of the total sample space  $\mathcal{Z}$  that is possible with  $x$  fixed. This is not, in general, equal to one, so the ERNM is not equal to the ERGM (Fellows & Handcock, 2012).

### 3.1 Definitions and notation

We will consistently refer to a set of nodes,  $A_k$ , as the  $k$ -th neighborhood, with an uppercase  $K$  representing the total number of neighborhoods and a lowercase  $k$  representing a specific neighborhood. The variable  $\mathcal{N}$  will refer to the domain of a random network, usually the union of a collection of neighborhoods. Nodes within the network will be indexed by the variables  $i$  and  $j$ , with  $Z_{ij} = (\{Y_{ij}, X_i, X_j\})$ , where  $Y_{ij}$  is referring to the edge between nodes  $i$  and  $j$ , and  $X_i$  and  $X_j$  refer to the random vectors of node attributes. Abstracting this further,  $i$  and  $j$  will also refer to tuples of nodes, so we will write  $\vec{i} = (i_1, i_2, \dots, i_q) \in \mathcal{N} \times \mathcal{N} \times \dots \times \mathcal{N}$ . The variables  $Z$  and  $Y$  will also often carry a subscript of  $W$  or  $B$  (for example  $Y_{Bij}$ ) which emphasizes that the edge from  $i$  to  $j$  is within or between neighborhoods, respectively. Finally, for lack of a better notation, the indicator function  $\mathbb{I}_B(i, j)$  (where  $B$  is for between) is one if  $i \in A_l$  and  $j \in A_p$  where  $l \neq p$ , and zero otherwise.

**Definition 3.2** (Local dependence property). Extending the definition in Schweinberger & Handcock (2015), a random network model satisfies *the local dependence property* if there is a partition of the node set  $\mathcal{N}$  into neighborhoods  $A_1, A_2, \dots, A_K$  for  $K \geq 2$  such that the network variables  $Z_{ij}$  are dependent when  $i, j \in A_\ell$  for some  $\ell$  and independent otherwise. We also require that nodal attributes depend only on the attributes of nodes within the same neighborhood. Thus, the probability measure can be written as

$$P(Z \in \mathbf{Z}) = \prod_{k=1}^K \left[ P_{kk}(Z_{kk} \in \mathbf{Z}_{kk}) \prod_{\ell=1}^{k-1} P_{k\ell}(Z_{k\ell} \in \mathbf{Z}_{k\ell}, Z_{\ell k} \in \mathbf{Z}_{\ell k}) \right], \quad (3.2)$$

where  $Z_{mn}$  is the subnetwork consisting of the random graph ties from nodes in  $A_m$  to those in  $A_n$  and the appropriate node variables and  $\mathbf{Z}_{mn}$  is a subset of the sample space of  $Z_{mn}$ . Furthermore, the measures  $P_{kk}$  can induce dependence between dyads while the measures  $P_{kl}$  induce independence.

**Definition 3.3** (Sparsity). Also from Schweinberger & Handcock (2015), we say a locally dependent random network is  $\delta$ -sparse if there is some  $\delta > 0$  and some  $C > 0$  such that

$$\mathbb{E}(|Y_{Bij}|^p) \leq Cn^{-\delta}, \quad (p = 1, 2) \quad (3.3)$$

where  $n = |\mathcal{N}|$  and  $Y_{Bij}$  signifies the tie between neighborhoods from node  $i \in A_l$  to node  $j \in A_m$  where  $l \neq m$ .

## 3.2 Preliminary theorems

In proving our theorems, we will make use of several other central limit theorems, all of which can be found in Billingsley (1995). The first is the Lindeberg-Feller central limit theorem for triangular arrays. The second is Lyapounov's condition, which gives a convenient way to show that the Lindeberg-Feller theorem holds. Finally, we make use of a central limit theorem for dependent random variables. For the sake of brevity, in this section we state each of these without proof.

**Theorem 3.1** (Billingsley, 1995 Theorem 27.2). *For each  $n$  take  $X_{n1}, \dots, X_{nr_n}$ , independent with  $\mathbb{E}(X_{ns}) = 0$  for all  $n$  and  $s$  (where no generality is lost in this assumption). Then we have  $\sigma_{ns}^2 = \text{Var}(X_{ns}) = \mathbb{E}(X_{ns}^2)$ . Next, set  $s_s^2 = \sum_{s=1}^{r_n} \sigma_{ns}^2$ . Now set*

$$S_n = X_{n1} + \dots + X_{nr_n}.$$

*If the Lindeberg condition,*

$$\lim_{n \rightarrow \infty} \sum_{s=1}^{r_n} \frac{1}{s_n^2} \int_{|X_{ns}| \geq \epsilon s_n} X_{ns}^2 = 0 \quad (3.4)$$

*holds for all  $\epsilon > 0$ , then  $S_n \xrightarrow{d} N(0, 1)$ .*

**Theorem 3.2** (Billingsley, 1995 Theorem 27.3). *Let  $S_n$  be as before. Then if Lyapounov's condition,*

$$\lim_{n \rightarrow \infty} \sum_{s=1}^{r_n} \frac{1}{s_n^{2+\delta}} \mathbb{E}(|X_{ns}|^{s+\delta}) = 0 \quad (3.5)$$

*holds for some  $\delta > 0$ , then the Lindeberg condition also holds. Therefore  $S_n \xrightarrow{d} N(0, 1)$ .*

**Theorem 3.3** (Billingsley, 1995 Theorem 27.4). *Suppose that  $X_1, X_2, \dots$  is stationary and  $\alpha$ -mixing with  $\alpha_n = O(n^{-5})$  and that  $\mathbb{E}(X_n) = 0$  and  $\mathbb{E}(X_n^{12}) < \infty$ . Note that the condition on  $\alpha$  is stronger than what we require. Our  $X_n$  will be  $M$ -dependent, meaning that each  $X_n$  is independent of all  $X_m$  where  $|n - m| > M$ . It is true that*

an  $M$ -dependent sequence is  $\alpha$ -mixing for constant  $\alpha$ . Then, if  $S_n = X_1 + \dots + X_n$ , we have

$$\frac{\text{Var}(S_n)}{n} \rightarrow \sigma^2. \quad (3.6)$$

Then, if  $\sigma > 0$ , we have  $S_n \xrightarrow{d} N(0, 1)$ .

The final theorem is Slutsky's theorem, a classic result of asymptotic theory in statistics.

**Theorem 3.4** (Wasserman, 2004 Theorem 5.5). *Let  $X_n, X, Y_n$  be random variables and let  $c$  be a constant. Then, if  $X_n \xrightarrow{d} X$  and  $Y \xrightarrow{p} c$  we have  $X_n + Y_n \xrightarrow{d} X + c$  and  $X_n Y_n \xrightarrow{d} cX$ .*

### 3.3 Consistency under sampling

With these in place, we attempt to extend a result about locally dependent ERGMs proven by Schweinberger & Handcock (2015) to locally dependent ERNMs. In short, this theorem states that the parameters estimated by modeling a small sample of a larger network can be generalized to the overall network. It was shown by Shalizi & Rinaldo (2013) that most useful formulations of ERGMs do not form projective exponential families in the sense that the distribution of a subgraph cannot be, in general, recovered by marginalizing the distribution of a larger graph with respect to the edge variables not included in the smaller graph. Hence, we are unable to generalize parameter estimates from the subnetwork to the total network.

To show that locally dependent ERNMs do form a projective family, let  $\mathbb{A}$  be a collection of sets  $\mathcal{A}$ , where each  $\mathcal{A}$  is a finite collection of neighborhoods. Also, allow the set  $\mathbb{A}$  to be an ideal, so that if  $\mathcal{A} \in \mathbb{A}$ , every subset of  $\mathcal{A}$  is also in  $\mathbb{A}$  and if  $\mathcal{B} \in \mathbb{A}$ , then  $\mathcal{A} \cup \mathcal{B} \in \mathbb{A}$ . If  $\mathcal{A} \subset \mathcal{B}$ , think of passing from the set  $\mathcal{A}$  to the set  $\mathcal{B}$  as taking a larger sample of the (possibly infinite) set of neighborhoods in the larger network. Then let  $\{\mathcal{P}_{\mathcal{A}, \theta}\}_{\mathcal{A} \in \mathbb{A}}$  be the collection of ERNMs with parameter  $\theta$  indexed by the sets in  $\mathbb{A}$ . For each  $\mathcal{A} \in \mathbb{A}$ , let  $\mathcal{P}_{\mathcal{A}, \Theta} = \{\mathcal{P}_{\mathcal{A}, \theta}\}_{\theta \in \Theta}$  be the collection of ERNMs on the neighborhoods in  $\mathcal{A}$  with parameter  $\theta \in \Theta$  where  $\Theta \subset \mathbb{R}^p$  is open. Assume that each distribution in  $\mathcal{P}_{\mathcal{A}, \Theta}$  has the same support  $\mathcal{Z}_{\mathcal{A}}$  and that  $\mathcal{A} \subset \mathcal{B}$  if and only if  $\mathcal{Z}_{\mathcal{B}} = \mathcal{Z}_{\mathcal{A}} \times \mathcal{Z}_{\mathcal{B} \setminus \mathcal{A}}$ . Then, the exponential family  $\{\mathcal{P}_{\mathcal{A}, \Theta}\}_{\mathcal{A} \in \mathbb{A}}$  is projective in the sense of Shalizi & Rinaldo (2013 Definition 1) precisely when Theorem 3.6 holds.

This follows from a specific case of the general definition given by Shalizi & Rinaldo (2013). There, for every pair  $\mathcal{A}$  and  $\mathcal{B}$  with  $\mathcal{A} \subset \mathcal{B}$ , they define the natural projection mapping  $\pi_{\mathcal{B} \rightarrow \mathcal{A}} : \mathcal{Z}_{\mathcal{B}} \rightarrow \mathcal{Z}_{\mathcal{A}}$ . Informally, this mapping projects the set  $\mathcal{Z}_{\mathcal{B}}$  down to  $\mathcal{Z}_{\mathcal{A}}$  by simply removing the extra data. For example if  $\mathcal{B} = \{A_1, A_2\}$  and  $\mathcal{A} = \{A_1\}$  as in Figure 3.1, then the mapping  $\pi_{\mathcal{B} \rightarrow \mathcal{A}}$  is shown in Figure 3.2.



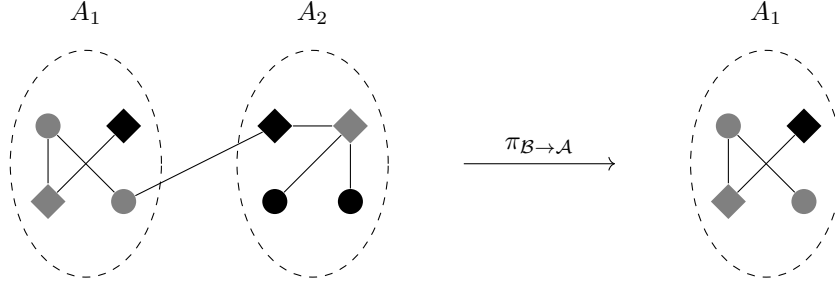


Figure 3.2: The projection mapping from  $\mathcal{B} = \{A_1, A_2\}$  to  $\mathcal{A} = \{A_1\}$ .

This is desirable because Shalizi & Rinaldo (2013) have demonstrated the following theorem.

**Theorem 3.5** (Shalizi & Rinaldo, 2013 Theorem 3). *If the exponential model family  $\{\mathcal{P}_{\mathcal{A}\Theta}\}_{\mathcal{A}\in\mathbb{A}}$  is projective and the log of the normalizing constant can be written as*

$$\begin{aligned} \log(C(\theta, \mathcal{Z})) &= \log\left(\int_{\mathcal{Z}} e^{\theta \cdot g(z)} dz\right) \\ &= r(|\mathcal{Z}|) a(\theta), \end{aligned} \quad (3.7)$$

where  $r$  is a positive, monotone increasing function of some positive measure on  $\mathcal{Z}$  and  $a$  is a differentiable function of  $\theta$ , then the maximum likelihood estimator exists and is strongly consistent, meaning that the MLE,  $\hat{\theta} \xrightarrow{a.s.} \theta$ , where  $\theta$  is the unknown parameter being estimated.

This is trivially achieved by setting  $r = 1$  for all values of  $|\mathcal{Z}|$  and setting  $a(\theta) = \log(C(\theta, \mathcal{Z}))$ . We have differentiability of  $a$  with respect to  $\theta$  by a result from multivariable calculus that follows from Fubini's theorem. From a practical perspective, this means that a researcher using this model can assume that parameters estimated from samples of a large network are increasingly good approximations for the true parameter values as the sample size increases.

**Theorem 3.6.** *Let  $A_1, A_2, \dots$  be a sequence of neighborhoods and define the sequence  $\{\mathcal{N}_K\} = \bigcup_{i=1}^K A_i$ . Then let  $Z_1, Z_2, \dots$  be the sequence of locally dependent random networks on the  $\mathcal{N}_K$ . For each  $Z_K$ , there is the corresponding set of neighborhoods  $\mathcal{A}_K$ . Let  $P_K$  be a generic probability distribution from the family  $\{\mathcal{P}_{K\Theta}\}_{\Theta \in \Theta}$ . Let the network*

$$\pi_{\mathcal{A}_{K+1} \rightarrow \mathcal{A}_K}(Z_{K+1}) = Z_{K+1 \setminus K},$$

with corresponding distribution  $P_{K+1 \setminus K}$ . Then

$$P_K(Z_K \in \mathbf{Z}_K) = P_{K+1}(Z_K \in \mathbf{Z}_K, Z_{K+1 \setminus K} \in \mathcal{Z}_{K+1 \setminus K}), \quad (3.8)$$

my\_dev where  $\mathcal{Z}_K$  is the sample space of the distribution  $P_K$  and  $\mathbf{Z}_K \subset \mathcal{Z}_K$ . This is a specific case of the definition of projectibility for a general exponential family given by Shalizi & Rinaldo (2013).

*Proof.* This follows from the definition of local dependence, in much the same way as the proof for ERGMs by Schweinberger & Handcock (2015) does. We have

$$\begin{aligned} P_{K+1}(Z_K \in \mathbf{Z}_K, Z_{K+1 \setminus K} \in \mathbf{Z}_{K+1 \setminus K}) &= P_{K+1}(Z_K \in \mathbf{Z}_K) P_{K+1 \setminus K}(Z_{K+1 \setminus K} \in \mathbf{Z}_{K+1 \setminus K}) \\ &= P_K(Z_K \in \mathbf{Z}_K)(1) \\ &= P_K(Z_K \in \mathbf{Z}_K), \end{aligned}$$

where the measure becomes  $P_k$  from the product definition of a locally dependent random network.  $\square$

### 3.4 Asymptotic normality of statistics

In this section we will prove that certain classes of statistics of locally dependent random networks are asymptotically normally distributed as the number of neighborhoods tends to infinity. The statistics we consider can be classified into three types: first, statistics which depend only on the graph structure; second, statistics that depend on both the graph and the nodal variates; and third, statistics that depend only on the nodal variates. The first class of statistics has already been considered by Schweinberger & Handcock (2015), but we will reproduce the proof here, as the second proof is very similar. The third class of statistics becomes normal in the limit by a central limit theorem for  $M$  dependent random variables in Billingsley (1995).

Before we begin to explicitly define each of these classes, we clarify the notation that will be used. A general statistic will be a function

$$S : \mathcal{N}^d \rightarrow \mathbb{R},$$

where  $\mathcal{N}^d$  is the  $d$ -fold Cartesian product of the set of nodes,  $\mathcal{N}$ , with itself:

$$\mathcal{N}^d = \underbrace{\mathcal{N} \times \cdots \times \mathcal{N}}_{d \text{ times}}.$$

Additionally, the statistic will often carry a subscript  $K$ , indicating that the statistic is of the random network with  $K$  neighborhoods.

Formally, as explained in Schweinberger & Handcock (2015), the first class of statistics contains those that have the form

$$S_K = \sum_{i \in \mathcal{N}^d} S_{Ki},$$

where

$$S_{Ki} = \prod_{l,p \in i} Y_{lp},$$

a product  $q$  of edge variables that captures the interaction desired. We will also make use of the set  $A_k^d$ , which is a similar cartesian product. When we write  $i \in A_k^d$ , we mean the every component of the  $d$ -tuple  $i$  is an element of  $A_k$ . Furthermore, by a catachrestic abuse of notation, we will write  $l, p \in i$  to mean that  $l$  and  $p$  are vertices contained in the  $d$ -tuple  $i$ . Now we are ready to prove the first case of the theorem.

**Theorem 3.7.** *Let  $A_1, A_2, \dots, A_K$  be a sequence of neighborhoods of size at most  $M$  and form the sequence of domains  $\mathcal{N}_K = \bigcup_{k=1}^K A_k$ . Then let  $Z_1, Z_2, \dots, Z_K$  be the sequence of unweighted random networks on the  $\mathcal{N}_K$ . Then, let the statistic  $S_K : \mathcal{N}_K^d \rightarrow \mathbb{R}$  be given. Furthermore, assume the statistic depends only on the graph variables of the  $Z_K$ . We also assume that the  $Z_K$  satisfy the local dependence property and that they are  $\delta$ -sparse, for some  $\delta > d$ . Finally, we require that  $\text{Var}(W_K^*) \rightarrow \infty$ , where  $W_K^*$  is defined in (3.10). Then*

$$\frac{S_K - \mathbb{E}(S_K)}{\sqrt{\text{Var}(S_K)}} \xrightarrow[K \rightarrow \infty]{d} N(0, 1). \quad (3.9)$$

*Proof.* As the networks  $Z_K$  are unweighted, all edge variables  $Y_{ij} \in \{0, 1\}$ . Let  $\mu_{ij} = \mathbb{E}(Y_{ij})$ . Then define  $V_{ij} = Y_{ij} - \mu_{ij}$ . Therefore, without loss of generality, we may work with  $V_{ij}$ , which has the convenient property that  $\mathbb{E}(V_{ij}) = 0$ . This means that we can similarly shift our statistics of interest,  $S_K$ . Therefore, call  $S_K^* = S_K - \mathbb{E}(S_K)$ , so that  $\mathbb{E}(S_K^*) = 0$ .

Note that we can write

$$S_K^* = W_K^* + B_K^*,$$

with

$$W_K^* = \sum_{k=1}^K W_{K,k}^* = \sum_{k=1}^K \sum_{i \in \mathcal{N}_K^d} \mathbb{I}(i \in A_k^d) S_{Ki}^* \quad (3.10)$$

and

$$B_K^* = \sum_{i \in \mathcal{N}_K^d} \mathbb{I}_B(i) S_{Ki}^*, \quad (3.11)$$

where the indicator functions restrict the sums to within the  $k$ -th neighborhood and between neighborhoods of the graph, respectively. Specifically,  $\mathbb{I}_B(i) = 1$  when the  $d$ -tuple of nodes  $i$  contains nodes from different neighborhoods, or exactly when  $\mathbb{I}(i \in A_k^d) = 0$  for all neighborhoods  $k$ . By splitting the statistic into the within and between neighborhood portions, we are able to make use of the independence relation between edges that connect neighborhoods. We also have  $\mathbb{E}(W_K^*) = 0$  and  $\mathbb{E}(B_K^*) = 0$ , as each quantity is a sum of random variables with mean zero.

The idea of this proof is to gain control over the variances of  $B_K^*$  and all the elements of the sequence  $W_{K,k}^*$ . We can then show that  $B_K^*$  is converging in probability to zero and that the triangular array  $W_K^*$  satisfies Lyapounov's condition, and is thus asymptotically normal. Finally, Slutsky's theorem allows us to extend the result to  $S_K^*$ .

To bound the variance of  $B_K^*$ , note that

$$\text{Var}(B_K^*) = \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}_B(i) \mathbb{I}_B(j) \text{Cov}(S_{Ki}^*, S_{Kj}^*).$$

Despite independence, some of these covariances may be nonzero if the two terms of the statistic both involve the same edge. For example, in Figure 3.1, a statistic that counted the number of edges between gray nodes plus the number of edges between

diamond shaped nodes would have a nonzero covariance term because of the edge between the two nodes that are both gray and diamond shaped. To show that, in the limit, these covariances vanish, we need only concern ourselves with the nonzero terms in the sum. That is, only those terms where  $\mathbb{I}_B(i)\mathbb{I}_B(j) = 1$ . This happens exactly when both  $S_{Ki}^*$  and  $S_{Kj}^*$  involve a between neighborhood edge variable. So, note that we have

$$\begin{aligned} \text{Cov}(S_{Ki}^*, S_{Kj}^*) &= \mathbb{E}(S_{Ki}^* S_{Kj}^*) - \mathbb{E}(S_{Ki}^*) \mathbb{E}(S_{Kj}^*) \\ &= \mathbb{E}(S_{Ki}^* S_{Kj}^*), \end{aligned} \quad (3.12)$$

as the expectation of each term is zero. Next we take  $Y_{l_1 l_2}$  to be one of the (possibly many) between neighborhood edge variables in this product (such that  $\mathbb{I}_B(i) = 1$  where  $i$  is any tuple containing  $l_1$  and  $l_2$ ) and  $V_{l_1 l_2}$  to be the recentered random variable corresponding to  $Y_{l_1 l_2}$ . Then,

$$\begin{aligned} \text{Cov}(S_{Ki}^*, S_{Kj}^*) &= \mathbb{E} \left( \prod_{m,n \in i} V_{mn} \prod_{m,n \in j} V_{mn} \right) \\ &= \mathbb{E} \left( V_{l_1 l_2}^p \prod_{m,n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} \right), \quad (p = 1, 2) \end{aligned}$$

where we must consider the case where  $p = 1$  to account for the covariance of  $S_{Ki}^*$  and  $S_{Kj}^*$  when  $i \neq j$  and the case where  $p = 2$  to account for the variance of  $S_{Ki}^*$ , which is computed in the case where  $i = j$ . So, if  $p = 1$ , then

$$\begin{aligned} \mathbb{E} \left( V_{l_1 l_2} \prod_{m,n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} \right) &= \mathbb{E}(V_{l_1 l_2}) \mathbb{E} \left( \prod_{m,n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} \right) \\ &= 0, \end{aligned}$$

by the local dependence property and the assumption that  $\mathbb{E}(V_{l_1 l_2}) = 0$ . The local dependence property allows us to factor out the expectation of  $V_{l_1 l_2}$ , as this edge is between neighborhoods, and therefore independent of every other edge in the graph. Now, if we have  $p = 2$ , then, by sparsity and the fact that the product below is at most 1,

$$\mathbb{E}(V_{l_1 l_2}^2) \mathbb{E} \left( \prod_{m,n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} \right) \leq DCn^{-\delta},$$

where  $D$  is a constant that bounds the expectation above. There exists such a constant because each of the  $V_{mn}$  are bounded by definition, so a product of them is bounded. So as  $K$  grows large, the between neighborhood covariances all become asymptotically negligible. Therefore, we can conclude that

$$\text{Var}(B_K^*) = \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}_B(i) \mathbb{I}_B(j) \text{Cov}(S_{Ki}^*, S_{Kj}^*) \leq DCn^{2d(-\delta)}. \quad (3.13)$$

So we have

$$\text{Var } B_K^* \rightarrow 0. \quad (3.14)$$

Then, for all  $\epsilon > 0$ , Chebyshev's inequality gives us

$$\lim_{K \rightarrow \infty} P(|B_K^*| > \epsilon) \leq \lim_{K \rightarrow \infty} \frac{1}{\epsilon^2} \text{Var}(B_K^*) = 0, \quad (3.15)$$

so

$$B_K^* \xrightarrow{P} 0. \quad (3.16)$$

Next, we bound the within neighborhood covariances, as we also have

$$\text{Var}(W_{Kk}^*) = \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}(i, j \in A_k) \text{Cov}(S_{Ki}^*, S_{Kj}^*).$$

As the covariance forms an inner product on the space of square integrable random variables, the Cauchy-Schwarz inequality gives us

$$\mathbb{I}(i, j \in A_k) |\text{Cov}(S_{Ki}^*, S_{Kj}^*)| \leq \mathbb{I}(i, j \in A_k) \sqrt{\text{Var}(S_{Ki}^*)} \sqrt{\text{Var}(S_{Kj}^*)}. \quad (3.17)$$

Then, as each  $S_{Ki}^*$  has expectation zero, we know that

$$\text{Var}(S_{Ki}^*) = \mathbb{E}(S_{Ki}^{*2}) - \mathbb{E}(S_{Ki}^*)^2 = \mathbb{E}(S_{Ki}^{*2}). \quad (3.18)$$

As  $S_{Ki}^{*2} \leq 1$ , we know  $\text{Var}(S_{Ki}^*) \leq 1$  for all tuples  $i$ , so we have the bound

$$\mathbb{I}(i, j \in A_k) |\text{Cov}(S_{Ki}^*, S_{Kj}^*)| \leq \mathbb{I}(i, j \in A_k). \quad (3.19)$$

Now all that remains is to apply the Lindeberg-Feller central limit theorem to the double sequence  $W_K^* = \sum_{k=1}^K W_{K,k}^*$ . To that end, first note that, as each neighborhood contains at most a finite number of nodes,  $M$ , we can show that

$$\begin{aligned} \text{Var}(W_{K,k}^*) &= \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}(i, j \in A_k^d) \text{Cov}(S_{Ki}^*, S_{Kj}^*) \\ &= \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}(i, j \in A_k^d) \mathbb{E}(S_{Ki}^* S_{Kj}^*) \\ &\leq \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} 1 \\ &\leq M^{2d}. \end{aligned} \quad (3.20)$$

Now we prove that Lyapounov's condition (3.21) holds for the constant in the exponent  $\delta = 2$ . So

$$\begin{aligned} \lim_{K \rightarrow \infty} \sum_{k=1}^K \frac{1}{\text{Var}(W_K^*)^2} \mathbb{E}(|W_{K,k}^*|^4) &= \lim_{K \rightarrow \infty} \frac{1}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \mathbb{E}(W_{K,k}^{*2}) \mathbb{E}(W_{K,k}^{*2}) \\ &\leq \lim_{K \rightarrow \infty} \frac{M^{2d}}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \mathbb{E}(W_{K,k}^*)^2 \\ &= \lim_{K \rightarrow \infty} \frac{M^{2d}}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \text{Var}(W_{K,k}^*) \\ &= \lim_{K \rightarrow \infty} \frac{M^{2d}}{\text{Var}(W_K^*)} \\ &= 0. \end{aligned} \quad (3.21)$$

where  $\text{Var}(W_K^*)$  tends to infinity by assumption. Therefore, Lyapounov's condition holds, and so by the Lindeberg-Feller central limit theorem, we have,

$$\frac{W_K^*}{\sqrt{\text{Var}(W_K^*)}} \xrightarrow[K \rightarrow \infty]{d} N(0, 1). \quad (3.22)$$

Slutsky's theorem (3.4) gives the final result for  $S_K^* = W_K^* + B_K^*$ . Then we have

$$\frac{S_K^*}{\sqrt{\text{Var}(S_K^*)}} \xrightarrow[K \rightarrow \infty]{d} N(0, 1), \quad (3.23)$$

as desired.  $\square$

The second class of statistics are those that depend on both the graph and the nodal variates. These have a very similar form as the statistics previously considered. Now we require that

$$S_K = \sum_{i \in \mathcal{N}^d} S_{Ki},$$

where

$$S_{Ki} = \prod_{l, p \in i} Y_{lp} h(X_l, X_p),$$

a product with at most  $q$  terms.

**Theorem 3.8.** *If  $S_K$  is a statistic depending on both the random graph and the random nodal attributes, the sequence of random networks are as before, and the function  $h$  is uniformly bounded in the sense that, for all  $l$  and  $m$ , there is some  $B$  such that*

$$P(|h(X_l, X_m)|^p > B) = 0, \quad (p = 1, 2)$$

*then we also have*

$$S_K \xrightarrow[K \rightarrow \infty]{d} N(0, 1).$$

*Proof.* This proof is very similar to the proof of Theorem 3.7. We write

$$S_K = WK + BK,$$

exactly as before, incorporating the function  $h$  into each  $S_{Ki}$  as we did above. Then the binary nature of the graph and the uniform boundedness of  $h$  allow us to once again recenter the  $Y_{ij}$ , meaning that we will work with  $V_{ij}h(X_i, X_j) = Y_{ij}h(X_i, X_j) - \mu_{ij}$ . We also have  $\mathbb{E}(V_{ij}h(X_i, X_j)) = 0$ , so  $\mathbb{E}(S_{Ki}^*) = 0$  as well.

For the between neighborhood covariances, we once again choose  $V_{l_1 l_2}$ , a between neighborhood network variable. Then we once again write

$$\begin{aligned} \text{Cov}(S_{Ki}^*, S_{Kj}^*) &= \mathbb{E} \left( \prod_{m, n \in i} V_{mn} h(X_m, X_n) \prod_{m, n \in j} V_{mn} h(X_m, X_n) \right) \\ &= \mathbb{E} \left( (V_{l_1 l_2} h(X_{l_1}, X_{l_2}))^p \prod_{m, n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} h(X_m, X_n) \right), \quad (p = 1, 2) \\ &= \mathbb{E}((V_{l_1 l_2} h(X_{l_1}, X_{l_2}))^p) \mathbb{E} \left( \prod_{m, n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} h(X_m, X_n) \right), \end{aligned}$$

by the local dependence property. Then, when  $p = 1$ , we have  $\mathbb{E}(V_{l_1 l_2} h(X_{l_1}, X_{l_2})) = 0$  by assumption, so the covariance is identically zero. When  $p = 2$  we have

$$\mathbb{E} \left( (V_{l_1 l_2} h(X_{l_1}, X_{l_2}))^2 \right) \leq C n^{-\delta}$$

by sparsity and

$$\mathbb{E} \left( \prod_{m, n \in (i \cup j) \setminus \{l_1, l_2\}} V_{mn} h(X_m, X_n) \right) \leq (DB)^{2q-2}$$

almost surely by uniform boundedness and the fact this product has at most  $2q - 2$  terms. This follows from the fact that  $h$  is bounded by  $B$  and that  $V_{mn}$  is bounded by some constant  $D$ , by definition. So

$$\text{Cov}(S_{Ki}^*, S_{Kj}^*) \leq B^{2q-2} C n^{-\delta},$$

which tends to zero as  $K$  grows large. So, again by Chebyshev's inequality, we have

$$B_K^* \xrightarrow[K \rightarrow \infty]{p} 0.$$

Next we bound the within neighborhood covariances. Now with each  $|S_{Ki}^*| \leq B^q$ , we have

$$\mathbb{I}(i, j \in A_k) \left| \text{Cov}(S_{Ki}^*, S_{Kj}^*) \right| \leq \mathbb{I}(i, j \in A_k) B^{2q}. \quad (3.24)$$

Now, we show that Lyaponouv's condition (3.21) holds for the same  $\delta = 2$ . Once again note that each neighborhood has at most  $M$  nodes, so

$$\begin{aligned} \text{Var}(W_{K,k}^*) &= \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}(i, j \in A_k) \text{Cov}(S_{Ki}^*, S_{Kj}^*) \\ &\leq \sum_{i \in \mathcal{N}_K^d} \sum_{j \in \mathcal{N}_K^d} \mathbb{I}(i, j \in A_k) B^{2q} \\ &\leq M^{2d} B^{2q}. \end{aligned}$$

Then Lyaponouv's condition is

$$\begin{aligned} \lim_{K \rightarrow \infty} \sum_{k=1}^K \frac{1}{\text{Var}(W_K^*)^2} \mathbb{E}(|W_{K,k}^*|^4) &= \lim_{K \rightarrow \infty} \frac{1}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \mathbb{E}(W_{K,k}^{*2}) \mathbb{E}(W_{K,k}^{*2}) \\ &\leq \lim_{K \rightarrow \infty} \frac{M^{2d} B^{2q}}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \mathbb{E}(W_{K,k}^{*2}) \\ &= \lim_{K \rightarrow \infty} \frac{M^{2d} B^{2q}}{\text{Var}(W_K^*)^2} \sum_{k=1}^K \text{Var}(W_{K,k}^*) \\ &= \lim_{K \rightarrow \infty} \frac{M^{2d} B^{2q}}{\text{Var}(W_K^*)} \\ &= 0. \end{aligned} \quad (3.25)$$

Therefore, by the Lindeberg-Feller central limit theorem and Slutsky's theorem, we have

$$\frac{S_K^*}{\sqrt{\text{Var}(S_K^*)}} \xrightarrow[K \rightarrow \infty]{d} N(0, 1). \quad \square$$

Finally, the last class of statistic is that which depends only on the nodal variates. This result follows directly from a central limit theorem for  $M$ -dependent random variables, which can be found in Billingsley (1995, p. 364). Establishing this theorem requires that we assume that the statistic in question depends only on a single variable across nodes. Therefore we assume that the statistic depends only on a single nodal covariate.

**Theorem 3.9.** *Take the sequence  $Z_K$  as before, and let  $X_K$  be the vector of nodal variates for each  $Z_K$ . Call each entry of this vector  $X_{Ki}$ , the variate corresponding to node  $i$ . Furthermore, we assume that  $\mathbb{E}(X_{Ki}^{12}) < \infty$ ,  $\mathbb{E}(X_{Ki} = 0)$ . Then,*

$$\lim_{K \rightarrow \infty} \frac{\text{Var}(\sum_{i=1}^n X_{Ki})}{n} = \sigma^2,$$

where  $n = |\mathcal{N}|$ . Furthermore, if  $\sigma > 0$ , then

$$\frac{\sum_{i=1}^n X_{Ki}}{\sqrt{n \text{Var}(\sum_{i=1}^n X_{Ki})}} \xrightarrow[K \rightarrow \infty]{d} N(0, 1).$$

*Proof.* Two random variables  $X_{Kl}$  and  $X_{Kp}$  are dependent if and only if  $l$  and  $p$  are in the same neighborhood. Without loss of generality, assume that the neighborhoods are such that all nodes within a neighborhood are indexed by consecutive integers. Then let  $M = \limsup |A_K|$ . Then the sequence  $X_{Kl}$  is  $M$ -dependent, so the result follows by application of Theorem 27.4 in Billingsley (1995).  $\square$

In practice, the hypothesis that the twelfth moment exists is satisfied for most reasonable distributional assumptions about nodal covariates. Furthermore, the assumption that all nodal variates have expectation zero can easily be satisfied by recentering the observed data. Finally, the delta method gives us an asymptotically normal distribution for a differentiable statistic of the nodal variate. The univariate nature of the statistic is a fundamental limitation of this approach, however I am unable to find an analogous multidimensional central limit theorem that would allow us to establish the asymptotic normality of a statistic of multiple nodal variates.



# Chapter 4

## Discussion and examples

### 4.1 Examples of convergence

In this section we simulate locally dependent random networks and calculate a statistic from each class discussed in Chapter 3. These networks grow larger and larger throughout the simulation, with each having  $n$  vertices and  $n/10$  neighborhoods, to explore the asymptotic properties of our statistics. The nodes have two attributes. The first is a group, coded as 0 or 1, that effects the network formation. Edges are more likely to form between vertices in the same group. The second is a random attribute, also coded as 0 or 1. For each vertex, the value of this attribute depends on the attributes of the other vertices that it is connected to. The attribute can be thought of as a contagious infection or behavior that spreads along edges of the graph. The C++ and R code used to generate these networks is shown in section A.1. This sort of complex relationship between the vertices and the edges is exactly the kind of generative process that ERNMs hope to capture.

Figure 4.1 shows an illustration of a simulated network with 100 vertices. The coloring shows the neighborhood membership of each vertex, while the presence or absence of fill indicates the value of the simulated attribute. Note the clustering of the neighborhoods and the attribute values. This is exactly the kind of joint clustering the locally dependent ERNM models. Furthermore, this also makes very intuitive sense as a realistic network structure. For example, suppose the attribute we are considering is smoking. This network shows that people who are friends with smokers are more likely to smoke themselves. This was the analysis done by Fellows & Handcock (2012) when introducing the ERNM.

Adding local dependence to this model is what allows us to show central limit theorems, however. In Figures 4.2 and 4.3, we can see the desired convergence towards a normal distribution that is guaranteed by the theorems proved in Chapter 3. In Figure 4.3, the quantile-quantile plots become very linear when the number of nodes reaches 200 and the number of neighborhoods reaches 20. All of the skewness has disappeared by this point and the tails of the distributions are almost perfectly normal. Finally, note that the slope is becoming shallower as the number of vertices grows. This corresponds to the shrinking of the statistic's variance, which is exactly what we

hope to see as we grow the sample size.

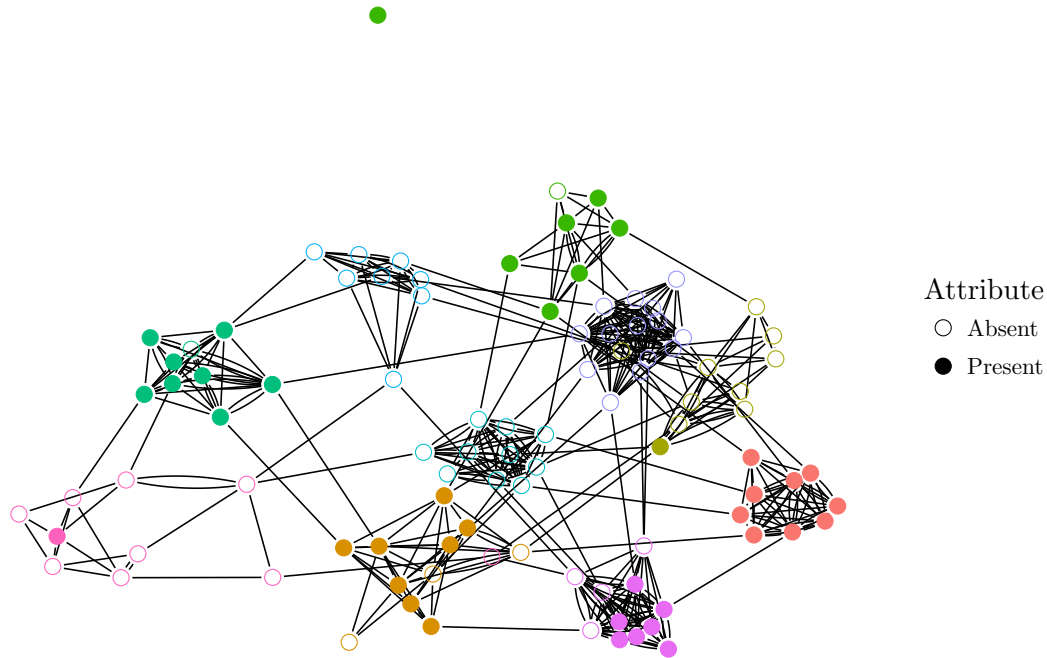


Figure 4.1: A simulated network with 100 vertices, colored according to neighborhood. The vertex fill indicates the presence or absence of the simulated attribute.

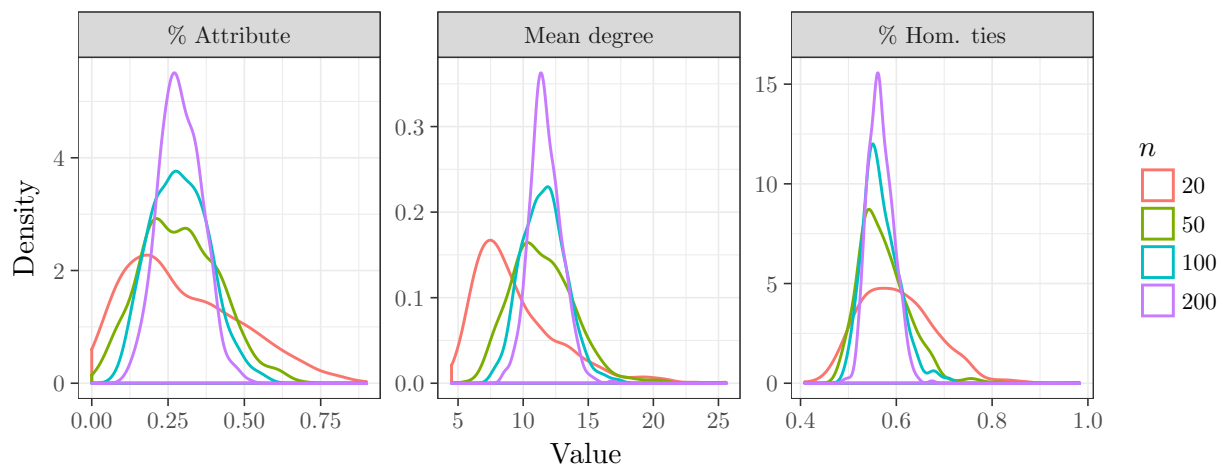


Figure 4.2: Density plots of simulated statistics of locally dependent random networks.

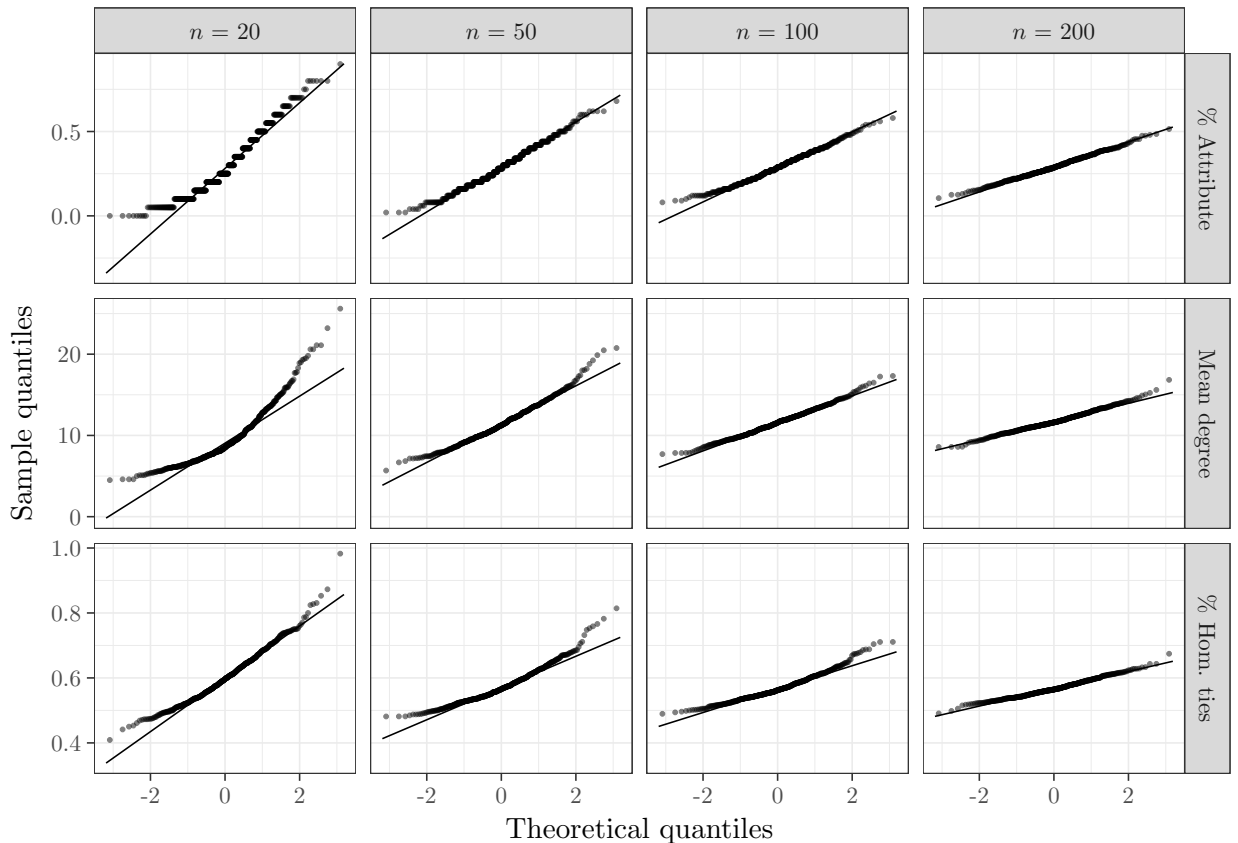


Figure 4.3: Normal Q-Q plots for simulated statistics of locally dependent random networks.

## 4.2 Standard error estimation

Given the asymptotic normality, all that remains to make these theorems useful for inference is to find a method to estimate the standard error of a statistic from a single observation. To that end, here we simulate a single network with 200 vertices by the same procedure as above and attempt to recover the standard errors of the statistics. We can compare these estimates to the standard errors of the empirical distributions we found in our previous simulation.

There are several approaches to approximating the standard errors. The first is to follow the vertex-level jackknife procedure outlined by Snijders & Borgatti (1999). This method is a modification of the standard jackknife estimator of standard deviation. At each iteration of the procedure, we remove one vertex from the network and recalculate the statistics of interest. Then we estimate the standard error with the equation

$$\widehat{\text{s.e.}} = \sqrt{\frac{n-2}{2n} \sum_{i=1}^n (g_{-i} - \bar{g})^2}, \quad (4.1)$$

where  $g$  is the quantity whose standard error we are approximating and  $g_{-i}$  is  $g$  calculated with vertex  $i$  removed. This differs from the standard jackknife estimator

Table 4.1: Standard error estimates using the jackknife at the vertex level.

Statistic	Simulated standard error	Jackknife standard error
% Attribute	0.0694576	0.0245443
Mean degree	1.2062191	0.6020049
% Hom. ties	0.0262896	0.0204720

Table 4.2: Standard error estimates using the jackknife at the neighborhood level.

Statistic	Simulated standard error	Jackknife standard error
% Attribute	0.0694576	0.0615096
Mean degree	1.2062191	1.2017300
% Hom. ties	0.0262896	0.0316870

in the multiplicative constant. Heuristically, this constant accounts for the fact that we are seeing much less variation in the jackknife networks than we would expect to see in the true distribution. This follows from the fact that the variance of a statistic of the edge variables is (approximately) inversely proportional to the number of edge variables,  $n(n-1)$ , not the number of vertices,  $n$  (Snijders & Borgatti, 1999). We can see in Table 4.1 that this procedure underestimates the variance of our statistics. This is most likely because of the dependence between vertices in the process that generated the network.

The second method is a standard jackknife, but leverages the neighborhood structure and the local dependence of the network. We do this by applying the jackknife at the level of the neighborhoods. That is to say, we remove each neighborhood in turn and recompute the statistic and then compute the jackknife estimator

$$\widehat{\text{s.e.}} = \sqrt{\frac{m-1}{m} \sum_{i=1}^m (g_{-i} - \bar{g})^2}, \quad (4.2)$$

where  $m$  is the number of neighborhoods. This process gives much more appropriate estimates, shown in Table 4.2.

### 4.3 Discussion of future work

Further development of locally dependent ERNMs for modelling complex social processes will have major impacts in the social sciences. The most pressing issue is the creation of software that allows for estimating parameters in these models. Code for fitting locally dependent ERGMs was developed by Schweinberger & Handcock (2015) and a Markov chain Monte Carlo algorithm for ERNMs was implemented by Fellows & Handcock (2012). However, it still remains to combine these two developments to work with locally dependent ERNMs.

Further mathematical work is also required to extend Theorem 3.9 to statistics involving more than one vertex attribute. This proof requires a multivariate analogue of the central limit theorem for  $M$ -dependent random variables. As my adviser told me, where there is a univariate central limit theorem, there is invariably a corresponding multidimensional theorem. However, we are unable to find a proof of this, so that result must be the topic of future work.

Finally, an investigation into the properties of standard error estimates like those produced above is much needed. Network sample sizes tend to be relatively small, so being forced to aggregate at the neighborhood level is a nontrivial issue. Furthermore, the jackknife procedure is most likely not making full use of all the information available within the observed network. However, a bootstrap procedure is, as far as my research shows, undefined for network data. Difficulty comes from being unable to sample with replacement. If one were to sample a network's vertices or edges with replacement, most likely duplicate edges would appear in the resampled networks. It is unclear how duplicates should be handled when calculating network statistics, so that sort of procedure is difficult to define.



# Appendix A

## Code

All source materials used to write this thesis are available in the Reed College Library's digital thesis archive. This appendix contains the most important code used in the final chapter.

### A.1 Network simulation code

```
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(thesisdown))
  devtools::install_github("ismayc/thesisdown")
library(thesisdown)

library(statnet)
library(Bergm)
library(tidyverse)
library(broom)
library(ggraph)
library(purrr)
library(Rcpp)
library(knitr)
```

The following C++ code, compiled into a shared library using the Rcpp package, gives function definitions used in simulating networks (Eddelbuettel & Francois, 2011).

```
#include <Rcpp.h>

using namespace Rcpp;

//[[Rcpp::export]]
IntegerMatrix make_adj(List vert_df){
```

```

NumericVector vert = vert_df["vert"];
NumericVector nbhd = vert_df["nbhd"];
NumericVector grp = vert_df["grp"];
double n = vert.size();

IntegerMatrix adj_mat(n,n);

for(int i = 0; i < n; i++){
  for(int j = 0; j < n; j++){
    if(nbhd[i] == nbhd[j]){
      if(grp[i] == grp[j]){
        adj_mat(i,j) = as<int>(rbinom(1,1,.3));
        continue;
      } else {
        adj_mat(i,j) = as<int>(rbinom(1, 1, .1));
        continue;
      }
    } else{
      adj_mat(i,j) = as<int>(rbinom(1, 1, 50/(n*n)));
      continue;
    }
  }
}

for(int i = 1; i < n/10; i++){
  for(int j = 0; j < n; j++){
    for(int k = 0; k < n; k++){
      for(int l = 0; l < n; l++){
        if(nbhd[j] == i &&
           nbhd[k] == i &&
           nbhd[l] == i){
          if(adj_mat(j, k) == 1 &&
             adj_mat(k, l) == 1 &&
             adj_mat(j, l) == 0){
            adj_mat(j, l) = as<int>(rbinom(1, 1, .75));
          }
        }
      }
    }
  }
}

return adj_mat;
}

```



```

#[[Rcpp::export]]
IntegerVector make_attr(List vert_df, IntegerMatrix adj_mat){
  IntegerVector attr = vert_df["attr"];
  IntegerVector nbhd = vert_df["nbhd"];
  int n = attr.size();
  for(int i = 0; i < n; i++){
    for(int j = 0; j < n; j++){
      if(attr[i] == 1 && adj_mat(i,j) == 1 && nbhd[i] == nbhd[j]){
        attr[j] = as<int>(rbinom(1, 1, .75));
      }else if(adj_mat(i,j) == 1 && nbhd[i] == nbhd[j]){
        attr[j] = as<int>(rbinom(1, 1, .1));
      }
    }
  }
  return attr;
}

```

This R code generates networks and computes the three chosen statistics for each one. The final result is a data set of networks and corresponding statistics used to produce density and quantile-quantile plots.

```

make_net <- function(n){
  vert_df <- tibble(vert = 1:n)
  nbhds <- 1:floor(n/10)
  vert_df <- vert_df %>%
    mutate(nbhd = sample(nbhds, n, replace = TRUE),
           grp = rbinom(n, 1, .5),
           attr = rbinom(n, 1, .3))

  adj_mat <- make_adj(vert_df)

  vert_df$attr <- make_attr(vert_df, adj_mat)

  net <- network(adj_mat)
  net %v% "group" <- vert_df$grp
  net %v% "attribute" <- vert_df$attr
  net %v% "nbhd" <- vert_df$nbhd

  return(net)
}

node_match <- function(net){
  edge_list <- as.edgelist(net)
  grps <- net %v% "group"
}

```

```

num <- 0
for(i in 1:nrow(edge_list)){
  if(grps[edge_list[i, 1]] == grps[edge_list[i,2]]) num <- num + 1
}
num/nrow(edge_list)
}
n_reps <- 500
results <- tibble(num = c(rep(20, n_reps),
                          rep(50, n_reps),
                          rep(100, n_reps),
                          rep(200, n_reps)))
results <- results %>%
  mutate(net = map(num, make_net),
         mean_deg = map_dbl(net, ~ mean(degree(.x))),
         mean_attr = map_dbl(net, ~ mean(.x %v% "attribute")),
         prop_homo = map_dbl(net, node_match))
results <- results %>%
  gather(stat, value, -(num:net))

```

## A.2 Jackknife standard error code

This code computes the neighborhood level standard error estimates for statistics of the graph.

```

n_vert <- 200
net <- make_net(n_vert)

nbhds <- net %v% "nbhd"
num_nbhds <- length(unique(nbhds))
boot3 <- numeric(num_nbhds)

jack_df <- tibble(mean_deg = numeric(num_nbhds),
                  mean_attr = numeric(num_nbhds),
                  prop_homo = numeric(num_nbhds))

for(i in 1:num_nbhds){
  temp_net <- rm_vertex(net, which(nbhds == i))
  jack_df$mean_deg[i] <- mean(degree(temp_net))
  jack_df$mean_attr[i] <- mean(temp_net %v% "attribute")
  jack_df$prop_homo[i] <- node_match(temp_net)
}

```





# References

- Billingsley, P. (1995). *Probability and measure* (3. ed., authorized reprint). New Delhi: Wiley India.
- Caimo, A., & Friel, N. (2011). Bayesian inference for exponential random graph models. *Social Networks*, 33(1), 41–55. <http://doi.org/10.1016/j.socnet.2010.09.004>
- Caimo, A., & Friel, N. (2014). Bergm: Bayesian exponential random graphs in R. *Journal of Statistical Software*, 61(2). Retrieved from <http://www.jstatsoft.org/v61/i02/>
- Eddelbuettel, D. (2013). *Seamless R and C++ integration with rcpp*. New York: Springer.
- Eddelbuettel, D., & Francois, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, (40, 8).
- Fellows, I., & Handcock, M. S. (2012). Exponential-family random network models. *ArXiv Preprint ArXiv:1208.0121*.
- Geyer, Charles J., & Thompson, E. A. (1992). Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3), 657–699. Retrieved from <http://www.jstor.org/stable/2345852>
- Groendyke, C., Welch, D., & Hunter, D. R. (2012). A network-based analysis of the 1861 Hagelloch measles data. *Biometrics*, 68(3), 755–765. <http://doi.org/10.1111/j.1541-0420.2012.01748.x>
- Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., Krivitsky, P. N., & Morris, M. (2016). *ergm: Fit, simulate and diagnose exponential-family models for networks*. The Statnet Project (<http://www.statnet.org>). Retrieved from <http://CRAN.R-project.org/package=ergm>
- Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., & Morris, M. (2008). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3).
- Ismay, C. (2017). *thesisdown: An updated R Markdown thesis template using the*

*bookdown* package.

- Morris, M., Handcock, M. S., & Hunter, D. R. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*, 24(4).
- Murray, I., Ghahramani, Z., & MacKay, D. (2012). MCMC for doubly-intractable distributions. *ArXiv Preprint ArXiv:1206.6848*.
- Robinson, D. (2016). *Broom: Convert statistical analysis objects into tidy data frames*. Retrieved from <https://CRAN.R-project.org/package=broom>
- Schweinberger, M., & Handcock, M. S. (2015). Local dependence in random graph models: Characterization, properties and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3), 647–676.
- Shalizi, C. R., & Rinaldo, A. (2013). Consistency under sampling of exponential random graph models. *The Annals of Statistics*, 41(2), 508–535. <http://doi.org/10.1214/12-AOS1044>
- Snijders, T. A. B., & Borgatti, S. P. (1999). Non-parametric standard errors and tests for network statistics. *Connections*, 22(2), 61–70.
- Strauss, D., & Ikeda, M. (1990). Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85(409), 204–212. Retrieved from <http://www.jstor.org/stable/2289546>
- Wasserman, L. (2004). *All of statistics*. New York: Springer-Verlag.
- Wasserman, S., & Pattison, P. (1996). Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p\*. *Psychometrika*, 61(3), 401–425.
- Wickham, H. (2016a). *Purrr: Functional programming tools*. Retrieved from <https://CRAN.R-project.org/package=purrr>
- Wickham, H. (2016b). *Tidyverse: Easily install and load 'tidyverse' packages*. Retrieved from <https://CRAN.R-project.org/package=tidyverse>
- Xie, Y. (2016). knitr: A general-purpose package for dynamic report generation in R (Version R package version 1.15.19).