Exponential Random Graph Models with Big Networks: Maximum Pseudolikelihood Estimation and the Parametric Bootstrap

June 24, 2017

Abstract

With the growth of interest in network data across fields, the Exponential Random Graph Model (ERGM) has emerged as the leading approach to the statistical analysis of network data. ERGM parameter estimation requires the approximation of an intractable normalizing constant. Simulation methods represent the state-of-the-art approach to approximating the normalizing constant, leading to estimation by Monte Carlo maximum likelihood (MCMLE). MCMLE is accurate when a large sample of networks is used to approximate the normalizing constant. However, as the size of the network increases, MCMLE is computationally expensive, and may be prohibitively so if the size of the network is on the order of 10,000. When the network is large, one option for estimation is maximum pseudolikelihood (MPLE). MPLE is considerably less accurate than MLE in small networks, but exhibits comparable mean squared error in large networks. The standard MPLE is simple and fast, but generally underestimates standard errors. We show that a resampling method—the parametric bootstrap—results in accurate coverage probabilities for confidence intervals. Furthermore, bootstrapped MPLE has the advantage of being embarrassingly parallel. We compare the two different approaches by applying them on U.S. Supreme Court Citation Network Data.

Introduction

The field of network science faces a double-edge sword when it comes to computational limitations to innovation. First, the networks under study are growing larger with each year. [**CHRISTIAN, WOULD YOU CHECK OUT N NODES IN *SOCIAL NETWORKS*?**]. Second, analytical methods are growing more

sophisticated, increasingly involving estimation and optimization, going beyond descriptive calculations [**Could also classify articles based on whether a statistical model is used *SOCIAL NETWORKS*?**]. In order to avoid a limiting horizon in which state-of-the-art methods cannot be used with state-of-the-art "big" data, network methodologists need to consider all available options in easing the computational burden of tools for network analysis.

The Exponential Random Graph Model

The *exponential random graph model* (ERGM) is a probability model for directed or undirected binary networks. This means neither the weighting nor the temporal change of ties is considered in the model. In literature, ERGMs are sometimes also referred to as *p-star* or p^* models (see Wassermann and Pattison [18], Robins et al. [12]). In the following, we will focus on directed networks; however, undirected networks can be introduced very similar.

The ERGM takes the adjacency matrix of an observed network G^{obs} as the manifestation of a matrix-like random variable Y. This means that a network of N nodes can be defined as a adjacency matrix $G = (g_{ij}) \in \mathbb{R}^{N\times N}$, where $g_{ij} \in \{0, 1\}$ for all $i, j \in \{1, \dots, N\}$. $g_{ij} = 1$ means that there is an edge between actors i and j, while $g_{ij} = 0$ indicates that these actors are not directly connected. Since the model does not consider loops, one has $g_{ii} = 0$ for all $i \in \{1, \dots, N\}$. Furthermore, define

$$\mathcal{G}(N) := \Big\{ G \in \mathbb{R}^{(N\times N)} : g_{ij} \in \{0, 1\},\ g_{ii} = 0 \Big\}$$

as the set of all possible networks on N nodes without loops. Note that the cardinality of set $\mathcal{G}(N)$ is increasing exponentially for every newly included actor, which results in $2^{N(N-1)/2}$ total elements. Therefore, for an already small number of actors the cardinality of $\mathcal{G}(N)$ turns out to be an astronomically large number. For this reason calculating the MLE is either extremely time-consuming or with today's technology not achievable. As a consequence, literature usually makes use of Markov Chain Monte Carlo (MCMC) methods.
With the definition of $\mathcal{G}(N)$ we define

$$Y : \Omega \to \mathcal{G}(N) \quad , \quad \omega \mapsto (Y_{ij}(\omega))_{i,j=1,\dots,N}$$

as a matrix-like random variable. As the probability function from Y to $\mathcal{G}(N_V)$ we define the ERGM as

$$\mathbb{P}_\theta(Y = G) = \frac{\exp(\theta^T \cdot \Gamma(G))}{\sum_{G^* \in \mathcal{G}(N)} \exp(\theta^T \cdot \Gamma(G^*))} \tag{1}$$

where $\theta \in \mathbb{R}^q$ is a $q-$dimensional vector of parameters, $\Gamma : \mathcal{G}(N) \to \mathbb{R}^q$, $G \mapsto (\Gamma_1(G), \ldots, \Gamma_q(G))^T$ is a q-dimensional function of different network statistics and $c(\theta) := \sum_{G^* \in \mathcal{G}(N)} \exp(\theta^T \cdot \Gamma(G^*))$ is a normalization constant which ensures that (1) defines a probability function on $\mathcal{G}$. As already mentioned, a specific network G can be considered as a manifestation of a matrix-like random variable, whose probability of occurrence can be modeled with equation (1). A key role when modeling an ERGM is played by the function $\Gamma(\cdot)$. The decision about which network statistics are incorporated into the model affects the model significantly (see Handcock [4]).

Estimation

As mentioned above calculating $c(\theta)$ is not achievable for most cases with today's technology. Therefore, the question arises how one can estimate the parameter vector θ? A first idea could be the following: One can assume that the *dyads* are independent of each other, which means that the random variables Y_{ij} inside the random matrix Y are independent of each other. In this case, one can show that

$$\text{logit}(\mathbb{P}_\theta(Y_{ij} = 1)) = \theta^T \cdot (\Delta G)_{ij}$$

This corresponds with the *logistic regression* approach, where the observations of the dependent variables are simply edge values of the observed adjacency matrix, and the observations of the covariate values are given as the scores of every single change statistic. Therefore, the estimation of θ can then be obtained as usual using straight forward maximum-likelihood estimation. The resulting likelihood function is of the following form:

$$\text{lik}(\theta) = \mathbb{P}_\theta(Y = G) = \prod_{i,j} \frac{\exp\left(\theta^T \Delta(G))_{ij}\right)}{1 + \exp\left(\theta^T \Delta(G))_{ij}\right)} \tag{2}$$

The problem with this simple estimation procedure is that the assumed hypothesis of the independence of the dyads turns out to be erroneous in many cases (see van Duijn et al. [17]). This is a systematic problem: The presence of network data is inextricably connected with the presence of *relational data*, which by definition

should not be assumed to be independent of each other. If this dependency structure is deliberately ignored and equation (2) is used to estimate θ, it results in a *maximum pseudo-likelihood estimation* (MPLE). This technique tends to underestimate the standard error. However, Desmarais and Cranmer [1] show that the pseudo-likelihood provides a consistent approximation of the maximum likelihood, meaning that the MPLE converges to the MLE as the size of the network increases.

There are several techniques to circumvent estimators, which underestimate the standard error of θ. In the following, we will introduce a technique based on *Markov Chain Monte Carlo (MCMC)* and maximum-likelihood methods.

The more rigorous technique is to estimate the parameters directly with the log-likelihood function derived from (1), which has the following form:

$$\text{loglik}(\theta) = \theta^T \cdot \Gamma(G) - \log(c(\theta)) \tag{3}$$

where G is the observed network. For the vector of network statistics, one can assume without loss of generality

$$\Gamma(G) = 0 \tag{4}$$

This means that centering the vector of network statistics does not affect the probability function of the network variable Y. Therefore, in context of the likelihood function (3) the vector of statistics can always be assumed to be centered on the observed network.
Due to assumption (4), one gets from (3) the simplified log-likelihood function

$$\text{loglik}(\theta) = -\log(c(\theta)) \tag{5}$$

The problem resulting from estimating the parameters with (3) is that the term

$$c(\theta) := \sum_{G^* \in \mathcal{G}(N_V)} \exp(\theta^T \cdot \Gamma(G^*))$$

which sums up the weighted network statistics of all possible networks of N nodes, has to be evaluated. Even for networks with small numbers of nodes this presents an enormous computational obstacle, and the necessary calculations for larger networks cannot currently be completed in any reasonable timeframe. As a result, for sufficiently large networks it is not possible to estimate the parameters directly with the likelihood function.

An expedient for this limitation is based on the following consideration: Fix any vector of parameters $\theta_0 \in \Theta$ from the underlying parameter range Θ and compute for $\theta \in \Theta$ the expected value. Then, one can show that

$$\mathbb{E}_{\theta_0}\left[\exp\left((\theta-\theta_0)^T \cdot \Gamma(Y)\right)\right] = \frac{c(\theta)}{c(\theta_0)}$$

This equation offers the following possibility: If one draws L random networks $G_1, \ldots, G_L$ out of a distribution $\mathbb{P}_{\theta_0}$ appropriately, one gets with the *law of big numbers* the following relation:

$$\frac{1}{L} \cdot \sum_{i=1}^{L} \exp\left((\theta-\theta_0)^T \cdot \Gamma(G_i)\right) \quad \xrightarrow{\mathbb{P}_{\theta_0}} \quad \mathbb{E}_{\theta_0}\left[\exp\left((\theta-\theta_0)^T \cdot \Gamma(Y)\right)\right] = \frac{c(\theta)}{c(\theta_0)} \tag{6}$$

For a big enough number, L, of random networks, the following approximation is reasonable:

$$\frac{c(\theta)}{c(\theta_0)} \approx \frac{1}{L} \cdot \sum_{i=1}^{L} \exp\left((\theta-\theta_0)^T \cdot \Gamma(G_i)\right) \tag{7}$$

One can now use equation (7) to determine an approximation of the log-likelihood function (5):

$$\begin{aligned}
\text{loglik}(\theta) - \text{loglik}(\theta_0) &= -\log(c(\theta)) + \log(c(\theta_0)) \\
&= -\log\left(\frac{c(\theta)}{c(\theta_0)}\right) \\
&= -\log\left(\mathbb{E}_{\theta_0}\left[\exp\left((\theta-\theta_0)^T \cdot \Gamma(Y)\right)\right]\right) \\
&\approx -\log\left(\frac{1}{L} \cdot \sum_{i=1}^{L} \exp\left((\theta-\theta_0)^T \cdot \Gamma(G_i)\right)\right)
\end{aligned}$$

By differentiating this equation on both sides with respect to θ one gets an approximate score function:

$$s(\theta) \approx -\frac{\partial}{\partial\theta} \log\left(\frac{1}{L} \cdot \sum_{i=1}^{L} \exp\left((\theta-\theta_0)^T \cdot \Gamma(G_i)\right)\right) \tag{8}$$

This approximate score function now can be used as usual, i.e., it can be iteratively approximately optimized with the *Newton-Raphson algorithm*. As a result, the approximate maximum-likelihood estimator for the parameters can be computed.

As pleasant as this may sound, the immediate question arises: How can a sufficient number of suitable drawings $G_1, ..., G_L$ be taken from the distribution $\mathbb{P}_{\theta_0}$?

For this purpose, the *Markov Chain Monte Carlo (MCMC)* methods can be used. This approach does not deliberately ignore the dependency structures inside the network. Furthermore, the Markov Chain Monte Carlo Maximum Likelihood Estimator (MCMLE) approaches the MLE as the number of networks simulated to approximate the likelihood goes to infinity.

No matter which kind of MCMC algorithm is used, the basic idea is the following: One constructs a *Markov chain* $(Y_t)_{t\in\mathbb{N}}$ on the set of all possible networks $\mathcal{A}(N)$ of N nodes, whose *stationary distribution* is in conformity with the distribution $\mathbb{P}_{\theta_0}$. One can show that every single realization (or *trajectory*) of this stochastic process accomplishes the convergence result (6) (for this version of the *Law of big numbers for Markov chains* we reference Meyn and Tweedie [9]). As a result, sub-sequences of $(A_t)_{t\in\mathbb{N}}$ which are sufficiently large enough can be used for approximation (7).

Snijders [14] provides a Metropolis Hastings algorithm on how to simulate networks: Choose a matrix $G^{(0)} \in \mathcal{G}(N)$ to start with (e.g., the observed network). For $k \in \{0, ..., L-1\}$ recursively proceed as follows:

1. Randomly choose an edge (i, j) where $i \neq j$ from $G^{(k)}$
2. Compute the value
$$\pi := \frac{\mathbb{P}_\theta(Y_{ij} \neq g_{ij}^{(k)} | Y_{ij}^c = G_{ij}^c)}{\mathbb{P}_\theta(Y_{ij} = g_{ij}^{(k)} | Y_{ij}^c = G_{ij}^c)}$$
3. Fix $\delta := \min\{1, \pi\}$ and draw a random number Z from $\text{Bin}(1, \delta)$. If
   - $Z = 0$, let $G^{(k+1)} := G^{(k)}$
   - $Z = 1$, define $G^{(k+1)}$ via
$$g_{pq}^{(k+1)} = \begin{cases} 1 - g_{pq}^{(k)} & \text{if } (p,q) = (i,j) \\ g_{pq}^{(k)} & \text{if } (p,q) \neq (i,j) \end{cases}$$
4. Start at step 1 with $G^{(k+1)}$.

The depicted algorithm provides a sequence of random networks $G_(0), \ldots, G_(L)$. Since the original matrix was chosen randomly and the first simulated networks are very dependent on the chosen matrix (only one edge is changed per iteration), usually the first B networks, where $N << B << L$, are discarded as the so called Burn-In.

Efficiency of MPLE/MCMLE

Even though the MCMLE is in general favored over the MPLE method there are also cases where the MPLE comes in handy. Foremost, MPLE is quick and simple, since estimation can be done by basic logistic regression and does not exert elaborate MCMC methods. As mentioned in the previous chapter the MPLE approaches the MLE as the size of the networks increase and as a consequence, is a consistent estimator (see Lindsay [8], Strauss and Ikeda [16], Hyvarinen [7], Desmarais and Cranmer [1]). This implies that for an increasing number of nodes the MPLE converges in probability to the MLE, meaning that for large enough networks the MPLE becomes an option to computationally intensive MCMC methods.
At this point we want to mention that we are familiar with the work of Shalizi and Rinaldo [13], arguing that consistency is not given in the ERGM framework. They prove that one cannot run an ERGM on a sub-network in order to make inferences about the full network. The way we use the term *consistency* in this paper is different and aligns with the way consistency is seen by Lindsay [8], i.e. instead of considering sub-networks, we argue that both, the MLE as well as the MPLE, approach the true coefficient values as the size of networks increases.
Another major advantage of the MCMLE method is, as shown by van Duijn et al. [17], that the MCMLE is a more efficient estimator than the MPLE, meaning that the variance of the MCMLE is in general lower than the variance of the MPLE.-Rewrite
Nevertheless, as shown by Desmarais and Cranmer [1] the MPLE outperforms the MCMLE if the number of simulated networks used to approximate the likelihood is not chosen large enough. It is even more remarkable that the number of simulated networks needed for the MCMLE, in order to surpass the MPLE increases as the size of the network increases. This means that for very large networks it becomes difficult to determine the number of simulated networks needed in order for the MCMLE to outperform the MPLE. In other words, the larger the network of interest is, the larger one has to choose the number of simulated networks in order to justify the time intensive MCMLE approach.

In order to demonstrate this disadvantage of the MCMLE we conduct a simulation study using Goodreau's Faux Mesa High School data, which represents a simulation of an in-school friendship network among 203 students as well as the Faux Magnolia High School data, representing an in-school friendship network among 1451 students. The data for both networks originates from Resnick et al [10].

For both networks we first calculate the MCMLE and treat the estimated coefficients

as the network's true values θ. For both networks we take the same parametrization, using the number of edges, the nodal attribute for gender and the geometrically weighted edgewise shared partners (gwesp)(see Hunter [6]) distribution were we fix the decay parameter λ at 0.25.
We simulate 500 new networks using the 'true' coefficients and estimate the MPLE as well as the MCMLE of these simulated networks. For every single simulated network the MCMLE calculation is being repeated several times using 1500, 800, 400, 200, 100, 50 and 25 simulated networks in order to approximate the likelihood. Based on these results, we compute the root mean square error, which is a measure of the accuracy of an estimator, combining both, the bias and the variance. Mathematically written, the RMSE for an estimator $\hat{\theta}$ is defined as

$$RMSE = \sqrt{\sum_{i=1}^{500}(\theta - \hat{\theta}_i)^2}$$

implying that the smaller the RMSE, the more accurate is the estimator. Since the MCMLE has higher efficiency and converges to the MLE, the RMSE decreases as the number of simulated networks used for the likelihood approximation increases. On the other hand, the RMSE of the MPLE is a constant value since no network simulations are required. In order to compare the RMSE of the two estimation techniques, we take the log of the ratio of the MCMLE to the MPLE. As a result, a negative value indicates a better MCMLE performance, while a positive value indicates a better MPLE performance. Figure 1 visualizes the results of the simulation study. The solid line illustrates the results of the log relative RMSE of the Faux Mesa High network, while the dashed line illustrates the corresponding results of the Faux Magnolia High network.
The plots support the fact that larger networks require a larger sample size of simulated networks in order for the MCMLE to outperform the MPLE. While the fairly small Faux Mesa High network only requires a sample size of about 50 networks the larger Faux Magnolia High network already requires a sample size of at least 1500 networks for the MCMLE to surpass the MPLE. These results propose especially for very large networks (e.g. social media data) the question, of how large the sample size has to be set in order to justify the approximately exact, but computationally expensive MCMLE method. The identification of the required sample size is not only becoming more difficult as the size of the network increases, but an increase of the sample size will also extends the mere computation time.

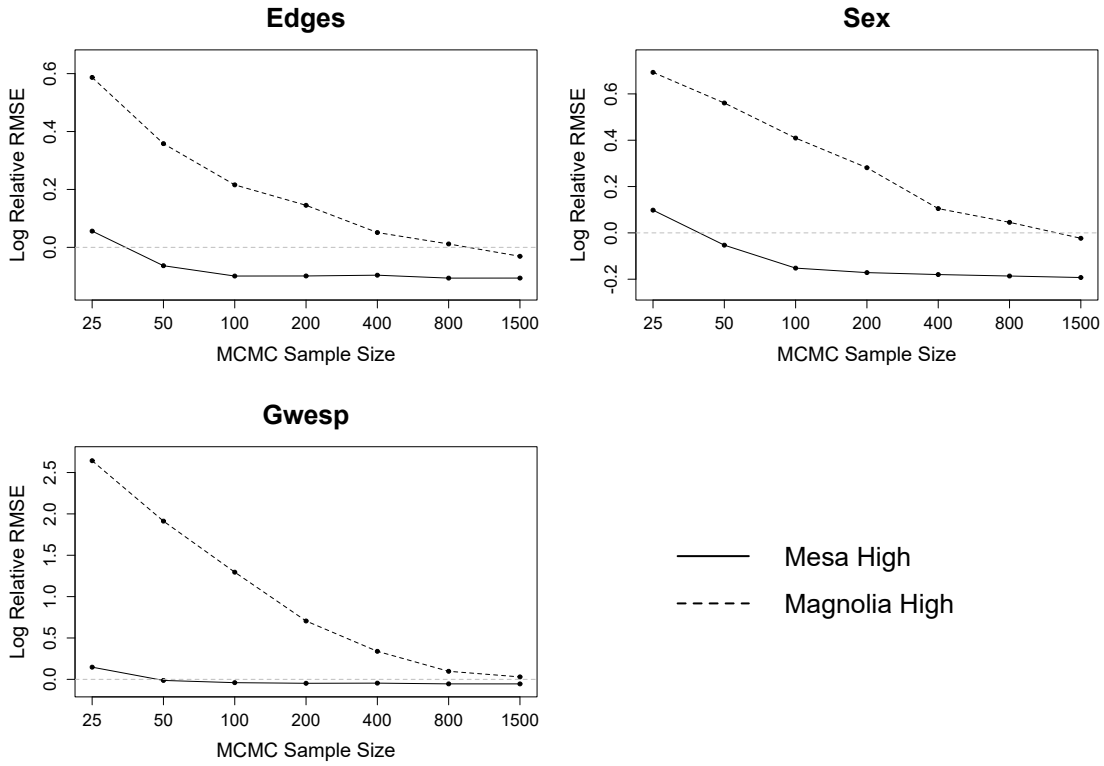

Figure 1: The log of the ratio of the RMSE for the MCMLE to the MPLE for different sample sizes and two different networks, Faux Mesa High and Faux Magnolia High

Bootstrapped MPLE

As discussed in the previous section, the MPLE converges to the MLE as the size of the network increases. Moreover, the MPLE is able to outperform the MCMLE if the sample size is not chosen large enough. The main reason why the MCMLE is still widely preferred is that in contrast to the MPLE, it does not underestimate the variance of its estimates [17]. By the definition of the ERGM it is obvious that this model is an exponential family distribution where θ is the natural parameter and $\Gamma(Y)$ is the sufficient statistic. This allows us to work in the framework of exponential family distributions and as a consequence, leads to the conclusion that the sampling distribution of the MLE is multivariate normal with mean vector equal to the MLEs and a covariance matrix equal to the inverse of the negative Hessian matrix $[-H]^{-1}$ of the likelihood function at the MLE. The problem with the MPLE is that calculating

$[-H]^{-1}$ by the pseudolikelihood function will underestimate the variance of the MPLE [17], resulting in an underestimate of the width of the confidence intervals. van Duijn et al. show that constructing 95% MPLE confidence intervals can result in intervals that only comprise the true value in less than 75% instead of the demanded 95%. In this paper, we are going to refer to the MPLE confidence intervals as *logistic regression confidence intervals* simply because the MPLE is calculated using logistic regression methods that also use the inverse of the negative Hessian matrix as an estimate for the covariance matrix.

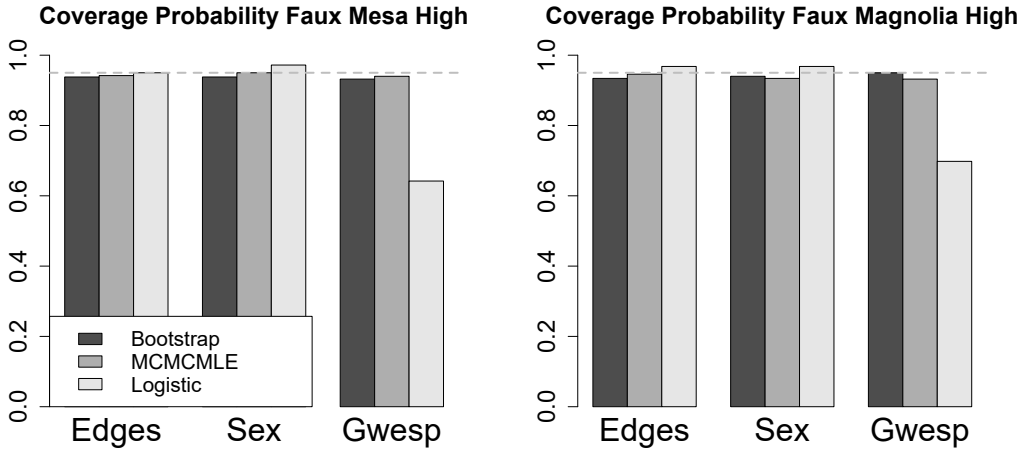

Figure 2: The Coverage Probability results of the Faux Mesa High network (left) and of the Faux Magnolia High network (right) for bootstrapped MPLE, MCMLE and logistic regression

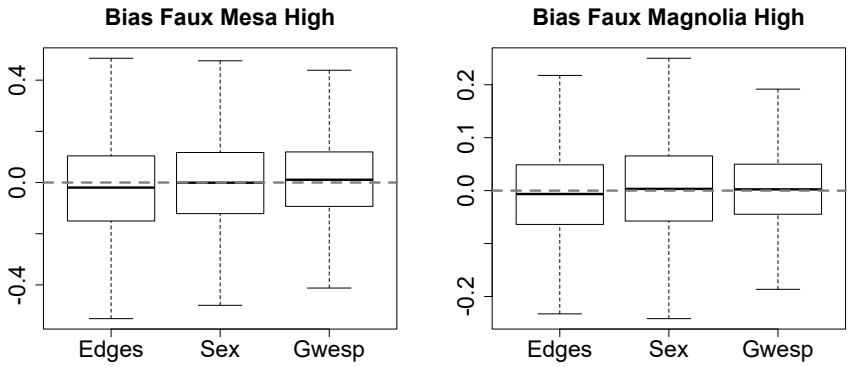

Figure 3: The boxplots visualize the bias $(\hat{\theta} - \theta)$ over the 500 iterations for the Faux Mesa High network (left) and the Faux Magnolia High network (right).

Since the MPLE has the advantage of being approximately exact and computationally inexpensive, but has the disadvantage of underestimating corresponding confidence intervals, we are going to apply a technique, referred to as *bootstrapped MPLE*, that was first introduced by Desmarais and Cranmer [1] and provides a consistent estimate of MPLE confidence intervals. Desmarais and Cranmer argue that this bootstrapped MPLE is a multivariate M-estimator [5], a special class of robust estimators, meaning that this algorithm consistently estimates the confidence intervals of the MPLE.
We verify the consistency of the bootstrapped MPLE by conducting a simulation study on the same two networks with the same paramtrization as in the previous chapter: The Faux Mesa High friendship network and the Faux Magnolia High friendship network.

For the simulation study, we determine the MPLE for the model and treat these estimates as the networks' 'true' parameter values. We then use these parameter values to simulate a sample of 1000 networks from the distribution of Y. For each of the 1000 networks, we calculate 95% confidence intervals based on the MCMLE and the logistic regression and examine whether the 'true' parameter values lie in these intervals. In addition, we determine the bootstrapped MPLE confidence intervals by sampling 500 networks for each of the originally sampled 1000 networks, by using the respective MPLE as parameter values. For every newly sampled network, we again determine the MPLE and then take the 2.5th and 97.5th percentile of the 500 MPLE estimates to obtain a 95% bootstrap confidence intervals. Similar as for the MCMLE and the logistic regression, we verify whether the 'true' parameter value

can be found in the bootstrapped confidence interval.
Figure 2 visualizes the coverage percentages for each of the three methods for both networks. The dashed line is set at 0.95 and represents the optimal value. It is in evidence that the bootstrapped MPLE performed equally well as the MCMLE, achieving results that obtain the true parameter values in approximately 95% of the cases. Additionally, a difference in the results between the smaller Faux Mesa High network and larger Faux Magnolia High network is not identifiable. Similar to the results of van Duijn et al. [17] our results for the logistic regression differ distinctively from the anticipated 95%, visualizing that the MPLE underestimates the variance of its estimates. Figure 3 illustrates the bias between the 'true' network coefficients θ and the MPLE estimates. (=> not degenerate?, maybe provide boxplots for scnetwork) This simulation study shows that bootstrapped MPLE is able to overcome the main disadvantage of the MPLE by retaining the validity of confidence intervals. Furthermore, this method also adopts the advantage of the MPLE of being consistent. However, the main advantage of the bootstrapped MPLE is the rapidity of the estimation that provides a computational benefit, especially if the size of the examined network is extremely large.

Cosponsorship Network Data

In order to demonstrate the computational advantage of the bootstrapped MPLE to the commonly used MCMLE we apply both approaches to the data on Cosponsorships in the U.S. Senate and U.S. House of Representatives for the 93rd to 108th Congresses developed by Fowler (2006) [2] [3]. The cosponsorship network consists of 2635 nodes and contains information of over 280,000 pieces of legislation proposed in the U.S. House and Senate for the time period 1973 - 2004. In this network each node indicates on cosponsor and an edge indicates the link of each cosponsor of a piece of legislation to its sponsor. The endogenous statistics we include into this model are the number of edges and the alternating k-star statistic as it was introduced by Snijders et al. [15] and modified by Hunter and Handcock (2006) [6]. The alternating k-star statistic adds one network statistic to the model equal to a weighted alternating sequence of k-star statistics with weight parameter λ and is a way to include a networks entire degree distribution as a network statistic. In this model we fix the weight paramter $\lambda = 0.4975$. Snijders et al. [15] introduced an approach involving k-star statistics $S_1(A), \ldots, S_{N-1}(A)$, where $S_k(A)$ denotes the

number of k-stars in the network, $k \in \{1, \ldots, N-1\}$. For simplicity, let us define

$$S_k(A) := \Gamma_{star(k)}(A)$$

define kstar. Note that in every network $S_1(A) = \Gamma_{edges}(A)$, i.e., $S_1(A)$ is equal to the number of edges in the network. On this basis, Snijders introduces the *alternating k-star statistics*

$$\mathfrak{S}(A, \lambda) := \sum_{k=2}^{N-1} (-\frac{1}{\lambda})^{k-2} S_k(A) = S_2(A) - \frac{S_3(A)}{\lambda} + \cdots + (-1)^{N-3} \frac{S_{N-1}(A)}{\lambda^{N-3}}$$

Models with this statistic and a fixed decay parameter turn out to be standard ERGMs and Hunter and Handcock [6] succeeded in proving that one can also rewrite alternating k-stars as a function of a network's degree distribution

$$\mathfrak{S}(A, \lambda) = \lambda \left(\lambda \sum_{j=1}^{N-1} \left(1 - \left(1 - \frac{1}{\lambda} \right)^j \right) D_j(A) + 2S_1(A) \right) \tag{9}$$

where $D_j(A) := \Gamma_{deg(j)}(A)$ is the number of nodes with and degree of j. In the next step, we define the *geometrically weighted degree* statistic as the first summand of (9)

$$\Gamma_{gwd}(A, \lambda) := \lambda \sum_{j=1}^{N-1} \left(1 - \left(1 - \frac{1}{\lambda} \right)^j \right) D_j(A) \tag{10}$$

At this point it also becomes obvious where the *geometrically* comes from. It simply refers to the geometric sequence $(1 - \frac{1}{\lambda})^j$ which appears in these statistics.

continue here The two nodal attributes we consider for this network are the year the majority opinion was drafted as well as an attribute that indicates whether a case appears on the Oxford list of salient cases. In order to include nodal covariates into the ERGM, the vector of nodal attributes is expanded into an artificial matrix C, which has the same dimensions as G. The first row of matrix C consists of the first actor's attribute, repeated N times. The second row of matrix C, consists of the second actor's attribute, repeated N times, and so on. Then, the statistics for a nodal covariate is defined as

$$\Gamma_{nodal} : \mathcal{G}(N) \to \mathbb{R} \quad , \quad G \mapsto \sum_{i<j}^{N} g_{ij} c_{ij}$$

	MCMLE		Logistic Regression		bootstrapped MPLE	
	Estimate	St. Error	Estimate	St. Error	Lower Bound	Upper Bound
Edges	-5.657	0.028	-4.568	0.038	-5.160	-4.530
Nodecov.Salience	0.592	0.012	1.482	0.009	1.413	1.890
Nodecov.Year	0.253	0.012	0.208	0.015	0.182	0.504
Absdiff.Year	-2.470	0.160	-3.121	0.194	-3.308	-2.434
Absdiff.Year.Squared	0.221	0.066	0.436	0.080	-0.028	0.517
DSP(0)	0.0005	0.0002	0.021	0.0002	0.019	0.031
ESP(0)	-3.129	0.012	-2.162	0.004	-2.179	-1.511

Table 1: Estimation results for the Supreme Court Network using MCMLE, bootstrapped MPLE and logistic regression

Beside these two nodal attributes, we furthermore include the difference between the years of publication as an edge attribute. An edge attribute can be written as a matrix E of the same dimensions as the observed adjacency matrix G. In this case entry e_{ij} is filled with the absolute difference between the year of case i and case j. Then, the corresponding statistic can be calculated the same way as nodal covariates. Since we do expect a non-linear effect for this edge attribute, we include this variable as a second-degree polynomial by creating a matrix E^2, where the matrix gets squared component wise.

We estimate the coefficient of this matrix using both techniques, the MCMLE and the bootstrapped MPLE. The MCMLE requires a sample size of at least 100000 networks to converge. The bootstrapped MPLE was estimated by using 500 simulated networks. As we described in the chapter *Estimation*, only one edge at a time is changed when simulating networks. For better comparison, we chose the same Burn-In (163840 MH-steps) and the same number of iterations (10240 MH-steps) for sampling networks. The results can be found in table 1.

The bootstrapped MPLE is not only simple and fast, it is also embarrassingly parallel, meaning that there is no problem separating the computation into several tasks. In other words, by using multiple cores, the computing time for estimating bootstrapped MPLE confidence intervals can be reduced substantially. Figure 4 illustrates the relative computing time of the bootstrapped MPLE using 500 simulated networks and the MCMLE for the three networks Faux Mesa High (205 nodes), Faux Magnolia High (1461 nodes) and Supreme Court Citation (9223 nodes) for an increasing number of computing cores. For the small network we simulate 2000 net-

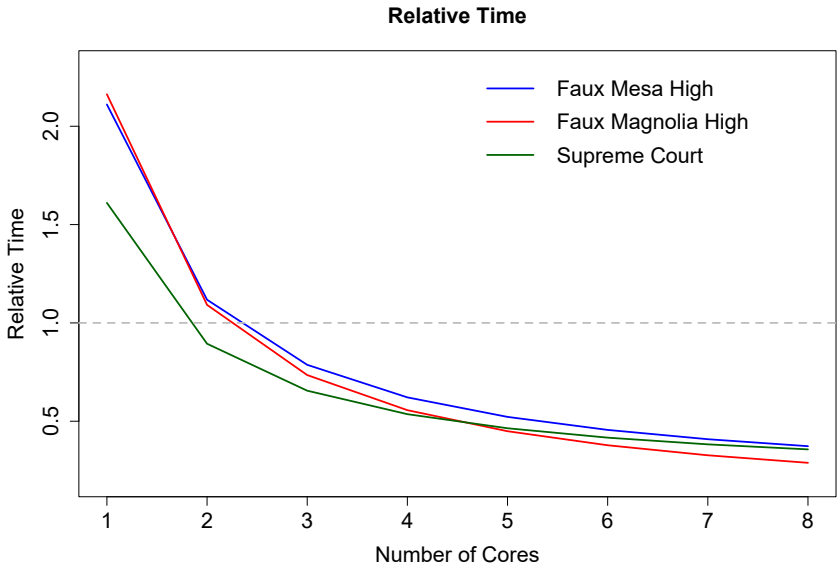

Figure 4: The y-axis gives the ratio of the bootstrapped MPLE time to that of the MCMLE time. Values below 1 indicate that the bootstrapped MPLE requires a shorter computing time.

works using a MCMC interval of 2000 steps, for the medium network we simulate 8000 networks using a MCMC interval of 5000 steps and for the large network we simulate 1 million networks using $10,000$ MCMC steps in order to approximate the likelihood appropriately. The chosen sample sizes and MCMC steps are necessary to guarantee a good model fit. We define the simulation time of the bootstrapped MPLE as a function of the number of available computing cores x:

$$\text{boostrapped MPLE time} = \text{network simulation time} + \frac{500 \cdot \text{MPLE estimation time}}{x}$$

Based on this, we define the relative computing time as

$$\text{relative computing time} = \frac{\text{bootstrapped MPLE time}}{\text{MCMLE time}}$$

This means that a relative computing time greater than 1 indicates that the MCMLE computing time is shorter, while a relative computing time smaller than 1 indicates that the bootstrapped MPLE provides faster results. Figure 4 demonstrates that the two high school friendship networks only require three cores for the bootstrapped MPLE to outperform the MCMLE. Even more interesting is that the even larger

Supreme Court Citation network only requires two cores for the bootstrapped MPLE to outperform the MCMLE.

One of the major disadvantages of MPLE over MCMLE is that it cannot assess *degeneracy*. When fitting an ERGM one usually has to deal with the problem of unreliable approximatively likelihood estimates for the model's parameters. The reason why degeneracy occurs is that the stochastic process generated by the MCMC-algorithm does not necessarily hold through the model's defined distribution of the random variable Y as stationary distribution (see Handcock [4] and Rinaldo et al. [11]). The bootstrapped MPLE, however, allows assessing degenerate models as well. In order to verify whether a model is degenerate or not, one can take a look at density and trace plots as visualized in figure ??. The trace plots on the left side depict the the attained values via MCMC simulated networks for every single statistic included into the model, centered on the statistic values of the observed network. The plots on the right side visualize the empirical density function of the respective statistic, based on the simulated networks (Hunter and Handcock [6]). For a non-degenerated model the empirical density function should be symmetrical around zero for every included centered statistic, since this corresponds with the expected value of a centered statistic (compare equation (4)). Otherwise, the values of the simulated networks systematically differ from the corresponding statistics in the observed network, making it unreasonable to assume that the simulated networks originate from the same distribution as the observed network. Furthermore, the trajectories in the trace plot should not indicate a dependence structure. This would be a signal that the constructed stochastic process violates the Markov properties.

insert density plots and trace plots

- describe advantage over MPLE (better coverage) and MCMLE (faster)
- can easily be parallelized/ using multiple cores -> increases speed
- this method is also able to detect degeneracy
- include table with boot MPLE results and MCMLE results

Other chapters

1. Fields that made fruitful use of the MPLE for networks, e.g. Boltzmann machine

2. Examples where a logit approach was used instead of an ERGM, because the network was too big and it was too time consuming using the MCMLE. We can then try to apply the bootstrapped MPLE approach.

Results/Discussion

- Bootstrap MPLE is an improvement of the MPLE and performs just as good as the MCMLE.

- Bootstrap MPLE has the advantage of being fast + multiple cores

- Bootstrap MPLE is consistent

- Conclusion: For very large networks the Bootstrap MPLE approach is more reasonable

Bibliography

[1] B. A. Desmarais and S. J. Cranmer. Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876, 2012.

[2] James H. Fowler. Connecting the congress: A study of cosponsorship networks. *Political Analysis*, 14(04):456–487, 2006.

[3] James H. Fowler. Legislative cosponsorship networks in the u.s. house and senate. 28:454–465, 2006.

[4] Mark S. Handcock. *Assessing degeneracy in statistical models of social networks: https://www.csss.washington.edu/Papers/wp39.pdf*. 2003.

[5] P.J. Huber, J. Wiley, and W. InterScience. *Robust statistics*. Wiley New York, 1981.

[6] David R. Hunter and Mark S. Handcock. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, 2006.

[7] Aapo Hyvärinen. Consistency of pseudolikelihood estimation of fully visible boltzmann machines. *Neural Computation*, 18(10):2283–2292, 2006.

[8] Bruce G. Lindsay. Composite likelihood methods. *Contemporary Mathematics*, 80(1), 1988.

[9] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Communications and control engineering series. Cambridge University Press, Cambridge and New York, 2nd ed edition, 2009.

[10] Michael D. Resnick, Peter S. Bearman, Robert Wm Blum, Karl E. Bauman, Kathleen M. Harris, Jo Jones, Joyce Tabor, Trish Beuhring, Renee E. Sieving, Marcia Shew, Marjorie Ireland, Linda H. Bearinger, and J. Richard Udry. Protecting adolescent's from harm: Findings from the national longitudinal study on adolescent health. *JAMA - Journal of the American Medical Association*, 278(10):823–832, 9 1997.

[11] Alessandro Rinaldo, Stephen E. Fienberg, and Yi Zhou. On the geometry of discrete exponential families with application to exponential random graph models. *Electronic Journal of Statistics*, 3:446–484, 2009.

[12] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29(2):173–191, May 2007.

[13] Cosma Rohilla Shalizi and Alessandro Rinaldo. Consistency under sampling of exponential random graph models. *Ann. Statist.*, (2):508–535, 04.

[14] T.A.B. Snijders. *Markov Chain Monte Carlo estimation of Exponential Random Graph Models.* Journal of Social Structure 3(2), 2002.

[15] Tom A. B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153, 2006.

[16] David Strauss and Michael Ikeda. Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85:204–212, 1990.

[17] Marijtje A. J. van Duijn, Krista J. Gile, and Mark S. Handcock. A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, 31(1):52–62, 1 2009.

[18] Stanley Wasserman and Philippa Pattison. Logit models and logistic regression for social networks: An introduction to markov graphs and p-star. *Psychometrica Vol.61*, 1996.