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

February 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

Coming soon...

## 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 [**?**]). 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

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

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 cardinalty 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, $\Gamma : \mathcal{G}(N) \to \mathbb{R}^q$ , $G \mapsto (\Gamma_1(G), \dots, \Gamma_q(G))^T$ 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 [**?**]).

## 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 $\theta$? 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 $\theta$ 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 most cases (see van Duijn et all ?citevanDuijn. 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 $\theta$, it results in a *maximum pseudo-likelihood estimation* (MPLE). This technique tends to underestimate the standard error. However, Desmarais and Cranmer [**?**] 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 $\theta$. 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 around 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 can not 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 $\Theta$ 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 $\theta$ 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 Marcov Chain Monte Carlo Maximum Likelihood Estimator (MCMCMLE) approaches the MLE as the number of networks simulated to approximate the likelihood goes to infinity. When it comes to the interpretation of the coefficients one can show that

$$\frac{\mathbb{P}_\theta(Y_{ij} = 1 | Y_{ij}^c = G_{ij}^c)}{\mathbb{P}_\theta(Y_{ij} = 0 | Y_{ij}^c = G_{ij}^c)} = \exp(\theta_1(\Delta_1 G)_{ij}) \cdot ... \cdot \exp(\theta_q(\Delta_q G)_{ij}) \tag{9}$$

where the condition $Y_{ij}^c = G_{ij}^c$ is short for: $Y_{pq} = g_{pq}$ for all $(p,q) \in \{1,\ldots,N\}^2$ with $(p,q) \neq (i,j)$ and refers to the rest of the network. The expression $(\Delta G)_{ij} := \Gamma(G_{ij}^+) - \Gamma(G_{ij}^-)$ is called the *change statistic*. $G_{ij}^+$ emerges from $G$, while assuming $g_{ij} = 1$ and $G_{ij}^-$ emerges from $G$, while assuming $g_{ij} = 0$. This means that the $k$th component of $(\Delta G)_{ij}$ captures the difference between the networks $G_{ij}^+$ and $G_{ij}^-$ on the $k$th integrated statistic in the model.

Equation (9) now enables a *ceteris-paribus analysis* of the parameters in the model: If the $k$th change statistic $(\Delta_k G)_{ij}$ increases one unit to $(\Delta_k G)_{ij} + 1$, while all the other change statistics remain unchanged, the odds of occurrence of edge $(i,j)$, conditional on the rest of the network, is multiplied by the factor $\exp(\theta_k)$. This means that for $\theta_k$, $k \in \{1,\ldots,q\}$ the conditional odds of occurence increase if $\theta_k > 0$, decrease if $\theta_k < 0$ and stay the same if $\theta_k = 0$. Therefore, the interpretation of the parameter happens very similar to logistic regression analysis. Notice that for the sake of simplicity we did not condition on exogenous network statistics in this chapter.

## Efficiency of MPLE/MCMCMLE

Even though the MCMCMLE 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. Furthermore, 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 Strauss et al /cite, Hyvarinen /cite, Desmarais and Cranmer [**?**]). 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.
However, another major advantage of the MCMCMLE method is, as shown by van Duijn et al. [**?**], that the MCMCMLE is a more efficient estimator than the MPLE, meaning that the variance of the MCMCMLE is in general lower than the variance of the MPLE. Nevertheless, as shown by Desmarais and Cranmer [**?**] the MPLE outperforms the MCMCMLE 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 MCMCMLE, 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 MCMCMLE 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 MCMCMLE approach.

In order to demonstrate this disadvantage of the MCMCMLE 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 [**?**].

For both networks we first calculate the MCMCMLE and treat the estimated coefficients as the network's true values $\theta$. 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)( [**?**]) distribution. We simulate 500 new networks using the 'true' coefficients and estimate the MPLE as well as the MCMCMLE of these simulated networks. For every single simulated network the MCMCMLE 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 MCMCMLE 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 MCMCMLE to the MPLE. As a result, a negative value indicates a better MCMCMLE performance, while a positive value indicates a better MPLE performance. Figure **??** 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 samplesize of simulated networks in order for the MCMCMLE to outperform the MPLE. While the

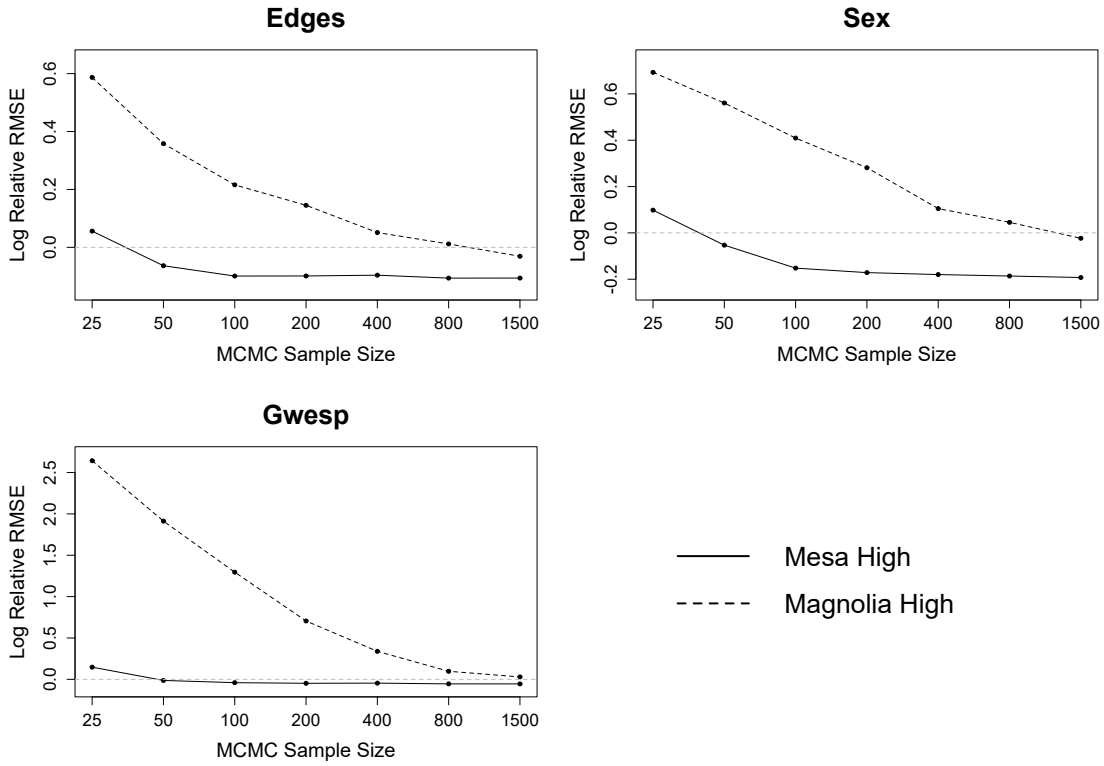


Figure 1: The log of the ratio of the RMSE for the MCMCMLE to the MPLE for different samplesizes and two different networks, Faux Mesa High and Faux Magnolia High

fairly small Faux Mesa High network only requires a samplesize of about 50 networks the larger Faux Magnolia High network already requires a samplesize of at least 1500 networks for the MCMCMLE to surpass the MPLE. These results propose especially for very large networks (e.g. social media data) the question, of how large the samplesize has to be set in order to justify the approximately exact, but comutationally expensive MCMCMLE method. The identification of the required samplesize is not only becoming more difficult as the size of the network increases, but an increase of the samplesize will also extends the mere computation time.

## Bootstrapped MPLE

- Describe method

- describe advantage over MPLE (better coverage) and MCMCMLE (faster)
- still consistent
- better efficiency than MPLE and similar results as MCMCMLE
- explain coverage probability and inlcude coverage plots
- can easily be parallelized/ using multiple cores -> increases speed
- this method is also able to detect degeneracy

## Results/Discussion

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

## To determine

1. where to include Boltzmann machine application