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

March 19, 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 [12], Robins et al. [9]). 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 [3]).

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 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 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 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 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 Marcov Chain Monte Carlo Maximum Likelihood Estimator (MCMCMLE) 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 [7]). As a result, sub-sequences of $(A_t)_{t\in\mathbb{N}}$ which are sufficiently large enough can be used for approximation (7).

Snijders [**?**] 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 orignial 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/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 [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.
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 [1] 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 [8].

For both networks we first calculate the MCMCMLE 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)([5]) 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 2 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.

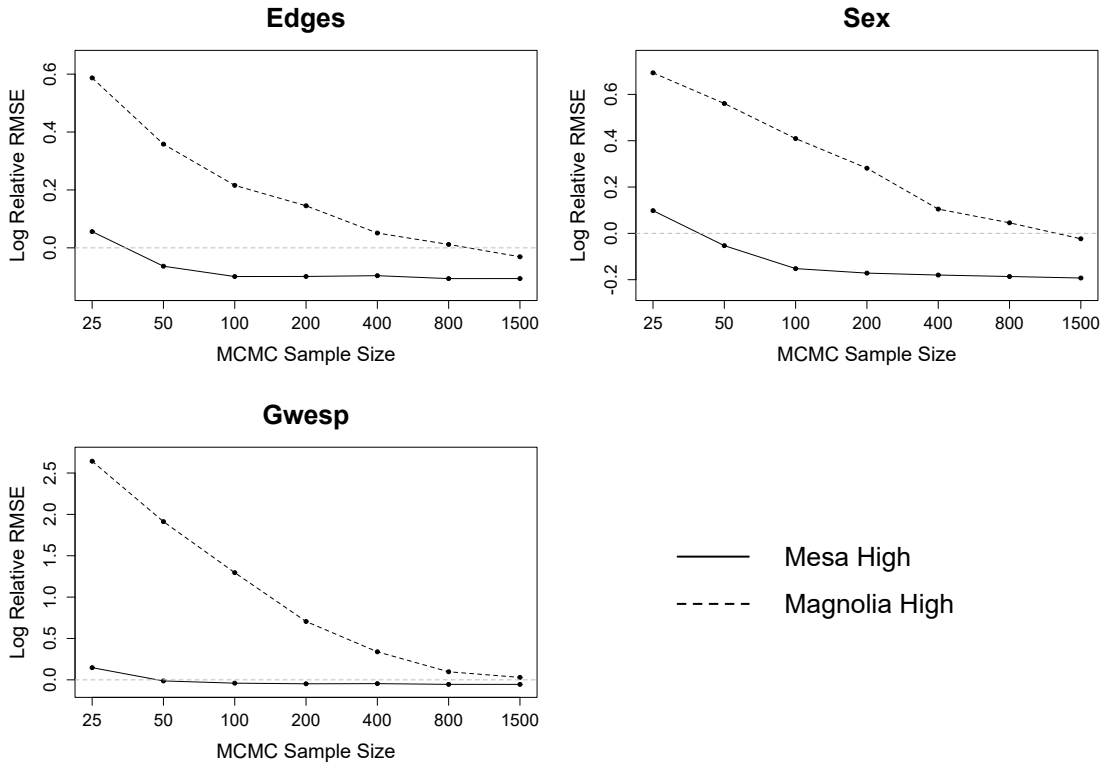

Figure 1: The log of the ratio of the RMSE for the MCMCMLE to the MPLE for different sample sizes and two different networks, Faux Mesa High and Faux Magnolia High

The plots support the fact that larger networks require a larger sample size of simulated networks in order for the MCMCMLE 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 MCMCMLE 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 MCMCMLE 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.

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 MCMCMLE if the sample size is not chosen large enough. The main reason why the MCMCLE is still widely preferred is that in contrast to the MPLE, it does not underestimate the variance of its estimates [11]. 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 [11], resulting in an underestimate of the width of the confidence intervals. 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.

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 [4], a special class of robust esti-

mators, 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 the confidence intervals based on the MCMCMLE 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 MPLE estimates to obtain 95% bootstrap confidence intervals. Similar as for the MCMCMLE and the logistic regression, we verify whether the 'true' parameter value can be found in the bootstrapped confidence interval. Figures ref1 and ref2 visualize the coverage probabilities for each of the three methods for both networks.

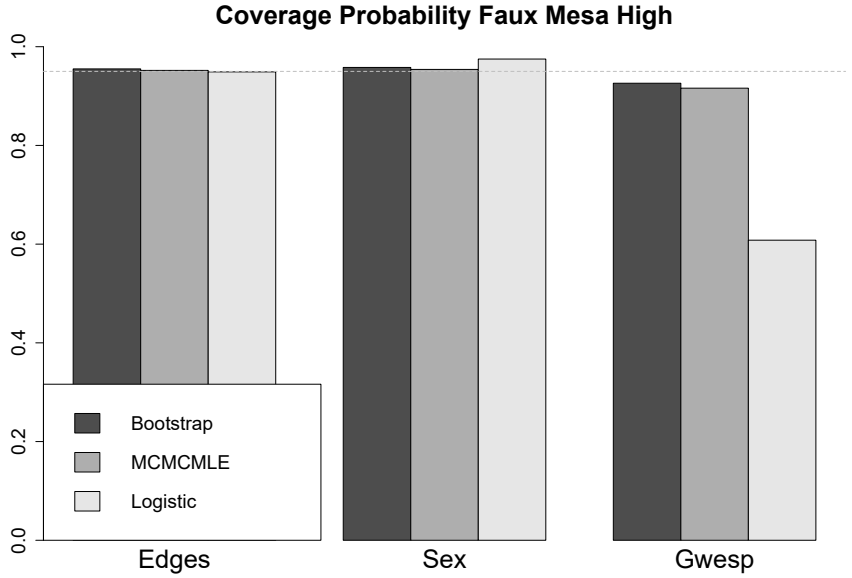

Figure 2: The Coverage Probability results of the Faux Mesa High network for bootstrapped MPLE, MCMCMLE and logistic regression

- include magnolia plot
- Describe plots
- describe advantage over MPLE (better coverage) and MCMCMLE (faster)
- can easily be parallelized/ using multiple cores -> increases speed
- this method is also able to detect degeneracy

This simualtion 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 provide a computational benefit, especially if the size of the examined network is extremely large.

Supreme Court Citation Network Data

In order to demonstrate the computational advantage of the bootstrapped MPLE to the commonly used MCMCMLE we apply both approaches to the U.S. Supreme Court Citation Network developed by Fowler and Jeon [2] and Fowler et al. [6]. The citation network we use consists of 9223 majority opinions written by the U.S. Supreme Court from 1954 to 2002. In this undirected network an edge occurs between two verticies (i.e. between two majority opinions) if one majority opinion cites the other. The endogenous statistics we include into this model are the number of edges, the edge-wise 0-shared partner (esp(0)) statistic and the dyad-wise 0-shared partner statistic (dsp(0)) (Snijders et al. [10], Hunter and Handcock [5]). The esp(k) statistic counts the number of pairs which are directly linked and share exactly k partners. The dsp(k) statistic is quite similar to esp(k), but does not require the two vertices to be directly linked. (*add explanation for adding esp(0) and dsp(0)*). 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. Beside these two nodal attributes we furthermore include the difference between the years of publication as an edge attribute. Since we do expect a non-linear effect fot this edge attribute, we include this variable as a second-degree polynomial.

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

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 and Sangick Jeon. The authority of supreme court precedent. 30(1):16–30, 2008.

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

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

[5] 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.

[6] James F. Spriggs Sangick Jeon James H. Fowler, Timothy R. Johnson and Paul J. Wahlbeck. Network analysis and the law: Measuring the legal importance of precedents at the u.s. supreme court. *Political Analysis*, 15(03):324—-346, 2007.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.