

Network structure from rich but noisy data

Natalia Garcia Martin & Maud Lemerrier

2018-10-18

Abstract

We have implemented an R package to estimate the parameters of exponential random graph models (ERG models), in the case where the data consists of noisy observations of the underlying hidden structure. The first part of this report focusses on the Bernoulli (or Erdos Renyi) model, performing inference via the Expectation Maximization algorithm, while the second part explores other ERG models, for which Bayesian parameter estimation is more challenging.

1 Introduction

1.1 Problem formulation and motivation

The true network structure is drawn from $P(A|\theta_A)$. We will later constrain this distribution to belong to the exponential random graph models family. The observations of the network are supposed to be noisy. The network structure and the observations are related to one another by $P(\text{data}|A, \theta_Y)$. Our aim is to infer the parameters $\theta = \{\theta_A, \theta_Y\}$ through the posterior distribution $P(\theta|\text{data})$.

$$P(\theta|\text{data}) \propto \sum_A P(\text{data}|A, \theta_Y) P(A|\theta_A) P(\theta)$$

This optimisation problem is hard since the function is often non convex, might have many local maxima and no analytical solutions. In our model, as illustrated on Figure 1, we have two unknowns: the latent variables and the parameters. We also suppose that the underlying network is probed k times.

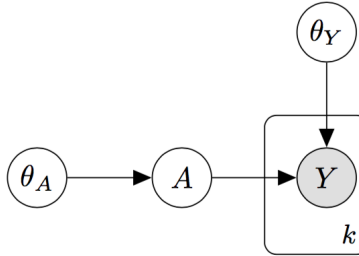


Figure 1: Graphical Model

In this report, we consider unweighted undirected networks, characterised by an $n \times n$ symmetric adjacency matrix A , having elements $A_{i,j} = 1$ if nodes i and j are connected by an edge and 0 otherwise. Similarly, we note $Y_{i,j}^{(k)}$ the edge measurement between node i and j at time k .

1.2 Network models

1.2.1 Bernoulli (Erdős–Rényi) model

The Bernoulli or Erdős–Rényi model is one of the simplest models for random networks. Starting with n isolated vertices, for any pair of nodes (i, j) an edge is added with probability $\rho > 0$. This step is repeated for the $\binom{n}{2} = n(n-1)/2$ pairs of nodes. This network is determined by the number of nodes n , the probability of edge formation ρ and the total number edges m and can be expressed as either $G(n, \rho)$ or $G(n, m)$. We will use the former notation. Since there are $\binom{n}{2}$ pairs of nodes in the network, the expected number of edges

is given by $\binom{n}{2}\rho$. Similarly, the expected number of 2-stars is $\binom{n}{3}\rho^2$ and the expected number of triangles is $\binom{n}{3}\rho^3$. The degree $\deg(j)$ of a node j is defined as the number of edges connected to it. Given the adjacency matrix A , this is

$$\deg(j) = \sum_i A_{i,j}.$$

The average degree of the graph is then given by the average of the individual nodes degrees. For a Bernoulli graph, the probability that a given node has degree d is the product of a) the probability that d links are present, b) the probability that the remaining potential links are missing, and c) the number of combinations in which we can select the d links out of the potential $n - 1$. Hence, they follow a Binomial distribution:

$$P(\deg(j) = d) = p_d = \binom{n-1}{d} \rho^d (1-\rho)^{n-1-d}$$

with expected degree $\rho(n-1)$. For large n and small ρ , this distribution can be approximated by a Poisson distribution with parameter $\rho(n-1)$. Due to all pairs of nodes having the same probability of edge formation, the Erdős-Rényi model is not appropriate to represent social networks, where edges tend to form between nodes which already share common connections.

1.2.2 ERGM model

Exponential random graph models (ERGM) are a family of probability distributions on graphs. Let \mathcal{A}_n be the set of all graphs on n vertices, and consider the following model, where θ_i , $i = 1, 2, \dots, m$ are real valued parameters, and T_i , $i = 1, 2, \dots, m$ are real valued statistics defined on \mathcal{A}_n . Possible sufficient statistics for such models are the degree of the vertices, the number of edges, the number of k-stars, the number of triangles, or the number of connected components. Altering these features allows us to construct networks which resemble social relationships. For example, triadic closure is a common occurrence in friendship networks, where as explained before, nodes with common neighbours are prone to form ties between them, resulting in a large number of triangles. The general form for an ERGM is

$$\begin{aligned} P(A = a|\theta) &= \frac{1}{Z(\theta)} \exp \left(\sum_{i=1}^m \theta_i T_i(a) \right) \\ &= \frac{1}{Z(\theta)} \exp [\theta^t T(a)], \end{aligned}$$

where $Z(\theta)$ is a normalisation constant satisfying $Z(\theta) = \sum_{a \in \mathcal{A}_n} \exp [\theta^t T(a)]$ and $\theta, T(a) \in \mathbb{R}^m$ represent the vector of parameters and the vector of sufficient statistics respectively. The 1-dimensional case with $\theta = \frac{\rho}{1-\rho}$ and $T(a) = \sum_{i,j} a_{ij}$ gives the expression for the Bernoulli graph.



Figure 2: Example of edge, 2-star, 3-star and triangle

2 Methods

2.1 Inference for the Bernoulli model

Since the underlying matrix A is symmetric, we consider the lower triangular part of the matrix only and simulate the interactions for each day k as edge observations $Y_{i,j}^{(k)}$ for entries $i < j$. These observations are assumed to be independent Bernoulli random variables conditioned on $A_{i,j}$. Given that the prior probability for a random edge is ρ , the prior model for the network is

$$P(A|\rho) = \prod_{i < j} \rho^{A_{i,j}} (1 - \rho)^{1 - A_{i,j}}.$$

The data model is specified by $P(\text{data}|A, \theta)$, where *data* corresponds to the observed $Y_{i,j}^{(k)}$. To simplify the notations, we let $E_{i,j} = \sum_k Y_{i,j}^{(k)}$. Applying Bayes' theorem,

$$P(A, \theta|\text{data}) = \frac{P(\text{data}|A, \theta)P(A)P(\theta)}{P(\text{data})}.$$

Our aim is to find the parameters θ that fix the relationship between the network and the data in order to then estimate the actual network structure. Summing over all the possible network structures A , we obtain the likelihood function

$$P(\theta|\text{data}) = \sum_A P(A, \theta|\text{data})$$

which we want to maximise to find the best estimates of θ given the observed data. Taking logarithms in both sides and making use of the Jensen inequality, we get

$$\begin{aligned} \log P(\theta|\text{data}) &= \log \sum_A P(A, \theta|\text{data}) \\ &= \log \sum_A q(A) \frac{P(A, \theta|\text{data})}{q(A)} \\ &\geq \sum_A q(A) \log \frac{P(A, \theta|\text{data})}{q(A)}, \end{aligned}$$

where $q(A)$ is any probability distribution over networks satisfying $\sum_A q(A) = 1$. The right hand side of the equation is maximised in the case of equality, that is, when

$$q(A) = \frac{P(A, \theta|\text{data})}{\sum_A P(A, \theta|\text{data})}.$$

Notice that $q(A) = P(A|\text{data}, \theta)$. We will use this result in order to estimate the parameters θ and the true underlying network A by employing an EM algorithm. We now define the true-positive rate α and the false-positive rate β respectively as the probabilities of observing an edge when it actually exists and observing it when it does not exist. We assume that the priors for α , β and ρ are uniform in the interval $[0,1]$. Then,

$$\begin{aligned} P(\text{data}|A, \theta) &= \prod_{i < j} \left(\alpha^{\sum_k Y_{i,j}^{(k)}} (1 - \alpha)^{N_{i,j} - \sum_k Y_{i,j}^{(k)}} \right)^{A_{i,j}} \left(\beta^{\sum_k Y_{i,j}^{(k)}} (1 - \beta)^{N_{i,j} - \sum_k Y_{i,j}^{(k)}} \right)^{1 - A_{i,j}} \\ &= \prod_{i < j} \left(\alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}} \right)^{A_{i,j}} \left(\beta^{E_{i,j}} (1 - \beta)^{N_{i,j} - E_{i,j}} \right)^{1 - A_{i,j}}. \end{aligned}$$

Combining this equation with the prior model for the network involving ρ that we defined above, we obtain

$$\begin{aligned} P(A, \theta | \text{data}) &= \frac{1}{P(\text{data})} \prod_{i < j} (\alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}})^{A_{i,j}} (\beta^{E_{i,j}} (1 - \beta)^{N_{i,j} - E_{i,j}})^{1 - A_{i,j}} p(A) p(\theta) \\ &= \frac{1}{P(\text{data})} \prod_{i < j} (\rho \alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}})^{A_{i,j}} ((1 - \rho) \beta^{E_{i,j}} (1 - \beta)^{N_{i,j} - E_{i,j}})^{1 - A_{i,j}}. \end{aligned}$$

The EM (expectation-maximization) algorithm

E-step

The expectation step consists in updating the posterior probabilities with the previous estimates of the parameters.

$$q^{(t)}(A) = \frac{p(A, \hat{\theta}^{(t)} | \text{data})}{\sum_A p(A, \hat{\theta}^{(t)} | \text{data})}$$

M-step

The maximisation step consists in solving the tractable optimisation problem:

$$\sum_A q^{(t)}(A) \nabla_{\theta} \log p(A, \hat{\theta}^{(t+1)} | \text{data}) = 0$$

With the Bernoulli model, the maximisation step is tractable, since we have:

$$P(A, \theta | \text{data}) = \frac{1}{P(\text{data})} \prod_{i < j} (\rho \alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}})^{A_{i,j}} ((1 - \rho) \beta^{E_{i,j}} (1 - \beta)^{N_{i,j} - E_{i,j}})^{1 - A_{i,j}}.$$

To simplify the notations, let $Q_{i,j} = P(A_{i,j} = 1 | \text{data}, \theta) = \sum_A q(A) A_{i,j}$.

Algorithm 2.1: EM($\alpha^{(0)}, \beta^{(0)}, \rho^{(0)}$)

$(\alpha, \beta, \rho) \leftarrow (\alpha^{(0)}, \beta^{(0)}, \rho^{(0)})$

while stopping criterion is not met

 E step:

$$Q_{i,j} \leftarrow \frac{\rho \alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}}}{\rho \alpha^{E_{i,j}} (1 - \alpha)^{N_{i,j} - E_{i,j}} + (1 - \rho) \beta^{E_{i,j}} (1 - \beta)^{N_{i,j} - E_{i,j}}}$$

 M step:
do $\left\{ \begin{array}{l} \alpha \leftarrow \frac{\sum_{i < j} E_{i,j} Q_{i,j}}{N \sum_{i < j} Q_{i,j}} \\ \beta \leftarrow \frac{\sum_{i < j} E_{i,j} (1 - Q_{i,j})}{N \sum_{i < j} (1 - Q_{i,j})} \\ \rho \leftarrow \frac{1}{\binom{n}{2}} \sum_{i < j} Q_{i,j} \end{array} \right.$

return (Q, α, β, ρ)

2.2 Inference for the ERGM model

The aim of this section is to discuss how a bayesian estimation of the parameters $\theta = \{\theta_A, \theta_Y\}$ can be performed. In our case, we cannot use a vanilla Metropolis Hastings algorithm with $p(\theta | Y)$ as a target distribution, for the following reasons:

- The normalizing function $Z(\theta)$ is intractable, and since it depends on θ it does not cancel out in the acceptance probability. This problem would also arise in the case where the network is completely observed.
- The likelihood appearing in the acceptance ratio is intractable, since we would have to sum over all the states that the hidden variable A can take ($2^{n(n-1)/2}$ configurations since the $n(n-1)/2$ edges can be in two states).

$$\begin{aligned}\alpha_{MH}(\theta, \theta') &= \min \left\{ 1, \frac{p(y|\theta')p(\theta')q(\theta|\theta')}{p(y|\theta)p(\theta)q(\theta'|\theta)} \right\} \\ &= \min \left\{ 1, \frac{\sum_A p(y|\theta', A)p(A|\theta')p(\theta')q(\theta|\theta')}{\sum_A p(y|\theta, A)p(A|\theta)p(\theta)q(\theta'|\theta)} \right\}\end{aligned}$$

For this report, we have decided to restrict our study to the case where the observations are supposed to be drawn from an ERGM probability distribution, without modelling any noise. In this settings we use the following notation: x a random variable drawn from an ERGM model with unnormalized probability model $h(x|\theta) = \exp(\theta^T T(x))$ and normalizing function $Z(\theta)$. Generalizing this model to the case of noisy observations is out of scope for this report.

Maximum Pseudolikelihood estimation (MPML)

Recall the general form of the ERGM:

$$P(A = a|\theta) = \frac{1}{Z(\theta)} \exp [\theta^T T(a)].$$

Maximum likelihood estimation involves the computation of the normalisation constant $Z(\theta)$, which is extremely hard to evaluate given the large number of possible structures of this network. Holland and Leinhardt (1981) were the first to develop a log-linear model for network data and tackled this problem by assuming independence in the dyads (a_{ij}, a_{ji}) , which allows parameter estimation through MLE. One of the proposed methods for estimating parameters in our ERGM network is maximum pseudolikelihood estimation (Strauss and Ikeda, 1990). The pseudolikelihood function is given by the product of the probabilities of the a_{ij} , each of these conditioned on the rest of the data. We use the notation $a_{ij} = 1$ to indicate the existence of an edge between nodes i and j and $a_{ij} = 0$ if they are not connected. We define a_{ij}^c to be the status of all pairs in a other than (i, j) . We also define a_{ij}^+ to be the same network as a in the case where $a_{ij} = 1$ and a_{ij}^- for $a_{ij} = 0$. Conditioned on $A_{ij}^c = a_{ij}^c$, A has only two possibilities: $A_{ij} = 1$ or $A_{ij} = 0$. We approximate the marginal $P(A_{ij} = 1)$ by the conditional probability $P(A_{ij} = 1|A_{ij} = a_{ij}^c)$. The joint distribution can then be approximated as

$$P(A|\theta) \approx \prod_{ij} P(a_{ij}|a_{ij}^c).$$

The conditional odds are given by

$$\frac{P(a_{ij} = 1|a_{ij}^c)}{P(a_{ij} = 0|a_{ij}^c)} = \exp [\theta^T (T(a_{ij}^+) - T(a_{ij}^-))].$$

Therefore,

$$\text{logit } P(a_{ij} = 1|a_{ij}^c) = \theta^T [T(a_{ij}^+) - T(a_{ij}^-)] = \theta^T \Delta T_{ij},$$

where $\Delta T_{ij} = T(a_{ij}^+) - T(a_{ij}^-)$ is the vector of changes in $T(a)$ when the entry a_{ij} changes from 1 to 0. These conditional probabilities do not involve $Z(\theta)$ so we can maximise the pseudolikelihood function using logistic regression. However, it is hard to quantify the uncertainty around these estimates.

Markov Chain Monte Carlo Maximum Likelihood estimation (MCMC-MLE)

Snijders (2002) introduces a method for estimating θ based on a Markov Chain Monte Carlo (MCMC) approximation of the MLE where the log-likelihood function is not evaluated directly. Instead, we take log ratios of the likelihood under proposed parameters θ and initial value of parameters θ_0 . We start by re-centering $P(A = a|\theta) = \frac{1}{Z(\theta)} \exp[\theta^t T(a)]$ around some value a for which $T(a) = 0$. Then, the log-likelihood can then be written as:

$$\begin{aligned} \text{loglik } \theta &= \log L(\theta|a) = \log \exp[\theta^t T(a)] - \log Z(\theta) \\ &= \theta^t T(a) - \log Z(\theta) \\ &= -\log Z(\theta). \end{aligned}$$

Therefore, taking log ratios and using the law of large numbers, we get

$$\begin{aligned} \text{loglik } \theta - \text{loglik } \theta_0 &= -\log Z(\theta) + \log Z(\theta_0) \\ &= -\log \frac{Z(\theta)}{Z(\theta_0)} \\ &= -\log E_{\theta_0} [\exp((\theta - \theta_0)^t T(a))] \\ &\approx -\log \left(\frac{1}{L} \sum_{i=1}^L \exp[(\theta - \theta_0)^t T(a^{(i)})] \right), \end{aligned}$$

where $a^{(1)} \dots a^{(L)}$ are a random sample of ERGM networks with parameter vector θ_0 . By differentiating on both sides, we obtain an approximate score function which can be iteratively approximately optimised using Newton-Raphson or Fisher scoring:

$$s(\theta) \approx -\frac{\partial}{\partial \theta} \log \left(\frac{1}{L} \sum_{i=1}^L \exp[(\theta - \theta_0)^t T(a^{(i)})] \right).$$

To simulate the L networks, we use the Metropolis Hastings algorithm provided by Snijders. We start with a graph $a^{(0)} \in \mathcal{A}_n$ —the set of all graphs on n nodes— and proceed as follows for $k \in \{0 \dots L-1\}$:

Algorithm 1: Metropolis Hastings for MCMCMLE

Given a_{ij} at the k^{th} iteration

1. Choose an edge (i, j) at random from $a^{(k)}$ where $i \neq j$
 2. Calculate $\pi = \frac{P(a_{ij} \text{ changes } |a_{ij}^c)}{P(a_{ij} \text{ does not change } |a_{ij}^c)}$
 3. Fix $\delta = \min\{1, \pi\}$ and draw u from $\text{Bin}(1, \delta)$.
 4. If $u = 0$, set $a^{(k+1)} = a^{(k)}$. If $u = 1$, set $a_{pq}^{(k+1)} = a_{pq}^{(k)}$ for $(p, q) \neq (i, j)$ and $a_{pq}^{(k+1)} = 1 - a_{pq}^{(k)}$ for $(p, q) = (i, j)$.
-

The Exchange Algorithm

The exchange algorithm (Murray et al. [2012]) falls within the class of *auxiliary variable approaches*, which consist in introducing an auxiliary variable u with a well-chosen conditional density $f(u|\theta, x)$ so that the intractable normalizing function cancels out in the Metropolis-Hastings acceptance probability. In the case of the exchange algorithm, this conditional density is $\frac{h(u|\theta)}{Z(\theta)}$.

The exchange algorithm augments the state from θ to (θ, θ', u) and targets the augmented joint density

$$\begin{aligned} \pi(\theta, \theta', u|x) &\propto p(\theta) L(\theta|x) q(\theta'|\theta) L(\theta'|u) \\ &= p(\theta) \frac{h(x|\theta)}{Z(\theta)} q(\theta'|\theta) \frac{h(u|\theta')}{Z(\theta')} \end{aligned}$$

. with a symmetric swapping proposal between the states $\{\theta', \theta\}$ and $\{\theta, \theta'\}$. Therefore the acceptance probability is

$$\alpha = \min \left\{ 1, \frac{\pi(\theta', \theta, u|x)}{\pi(\theta, \theta', u|x)} \right\} = \min \left\{ 1, \frac{p(\theta')h(x|\theta')h(u|\theta)q(\theta|\theta')}{p(\theta)h(x|\theta)h(u|\theta')q(\theta'|\theta)} \right\}$$

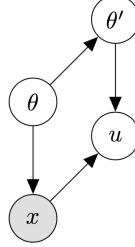


Figure 3: The augmented model

Algorithm 2: Exchange algorithm

Given $\theta_n \in \Theta$ at the n^{th} iteration

1. Propose $\theta' \sim q(\cdot|\theta_n)$
 2. Generate the auxiliary variable $u \sim \frac{h(\cdot|\theta')}{Z(\theta')}$
 3. Accept θ_{n+1} with probability $\alpha = \min \left\{ 1, \frac{p(\theta')h(x|\theta')h(u|\theta_n)q(\theta_n|\theta')}{p(\theta_n)h(x|\theta_n)h(u|\theta')q(\theta'|\theta_n)} \right\}$
 Reject otherwise
-

The Double Metropolis-Hastings algorithm

The double Metropolis-Hastings algorithm uses a Metropolis-Hastings sampler to generate θ draws (outer sampler), and another one to generate the auxiliary variables (inner sampler). Compared to the exchange algorithm, the auxiliary variable u is now obtained after m -MH updates starting from x , under θ' . Besides, since the inner sampler update $T_{\theta'}^m(u|x)$ satisfies the detailed balance condition

$$h(x|\theta')T_{\theta'}^m(u|x) = h(u|\theta')T_{\theta'}^m(x|u)$$

the acceptance probability of the DMH algorithm is

$$\alpha = \min \left\{ 1, \frac{p(\theta')T_{\theta'}^m(x|u)h(u|\theta_n)q(\theta_n|\theta')}{p(\theta_n)h(x|\theta_n)T_{\theta'}^m(u|x)q(\theta'|\theta_n)} \right\}$$

Algorithm 3: Double Metropolis-Hastings algorithm

Given $\theta_n \in \Theta$ at the n^{th} iteration

1. Propose $\theta' \sim q(\cdot|\theta_n)$
 2. Generate the auxiliary variable using m MH-updates $u \sim T_{\theta'}^m(\cdot|x)$
 3. Accept θ_{n+1} with probability $\alpha = \min \left\{ 1, \frac{p(\theta')T_{\theta'}^m(x|u)h(u|\theta_n)q(\theta_n|\theta')}{p(\theta_n)h(x|\theta_n)T_{\theta'}^m(u|x)q(\theta'|\theta_n)} \right\}$
 Reject otherwise
-

3 Results

3.1 EM algorithm for noisy observations

Dataset

In this section, we propose to validate the model's estimates using a synthetic dataset for which the ground truth network and the noisy observations are generated via a predefined probabilistic model. We simulate a network with $n = 100$ vertices, and we set $\rho = 0.1$. The noisy observations are then simulated for $k = 5$ days, with true positive rate $\alpha = 0.6$ and false positive rate $\beta = 0.009$. Our stopping criterion is met when the absolute value of the difference of all parameter values after an iteration is less than $\epsilon = 0.001$.

Results

In this setting, the stopping criterion is met after 11 iterations. The figure below, shows the comparison of the ground truth network with the inferred network (obtained by thresholding the posterior probabilities at $t = 0.5$), where the size of the nodes is proportional to their degree.

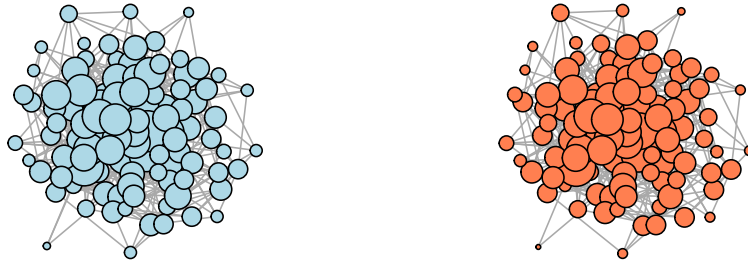


Figure 4: (left) Ground truth underlying network (right) Inferred underlying network

The following figure displays the convergence of the parameter estimates after 13 iterations, when the criterion is met with a running time of 22ms.

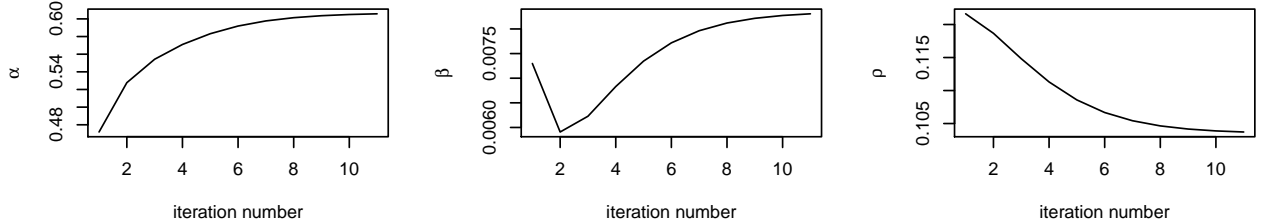


Figure 5: Convergence of the parameter estimates

We then assess the performance of the algorithm by computing several binary classification metrics involving the number of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN).

Metric	Formula	Result (n=100,k=5)
Precision	$TP/(TP+FP)$	1
Recall	$TP/(TP+FN)$	0.94
Accuracy	$(TP+TN)/(TP+TN+FP+FN)$	0.994
F-measure	$2*((Precision*Recall)/(Precision+Recall))$	0.969

The figure below shows the influence of the number of repeated observations k and the number of nodes n on the performances of the algorithm. We observe how the performance metrics improve by increasing the

number of repeated observations k , with values close to 1 after 5 days. Similarly, for fixed $k = 5$, an increase in the size of the network yields better estimates of the underlying network.

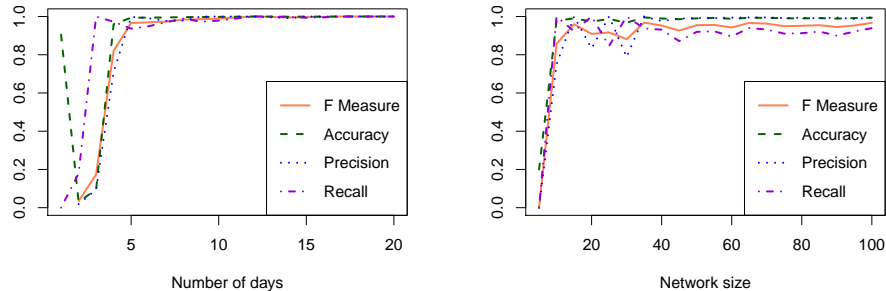


Figure 6: Performance metrics versus the number of repeated observations k

3.2 Exchange algorithm

Dataset

We have used the Florentine business dataset from the Statnet Package for the Statistical Modeling of Social Networks, a data set of marriage and business ties among 16 Renaissance Florentine families. An ERGM model was fitted with sufficient statistics being the number of edges (with coefficient θ_1), 2-stars (θ_2), 3-stars (θ_3) and triangles (θ_4).

Results

The exchange algorithm was run with 10000 iterations and a random walk proposal with a single parameter $\sigma = 0.15$. We chose uniform priors in the interval $[-5, 5]$. The running time was 11 minutes.

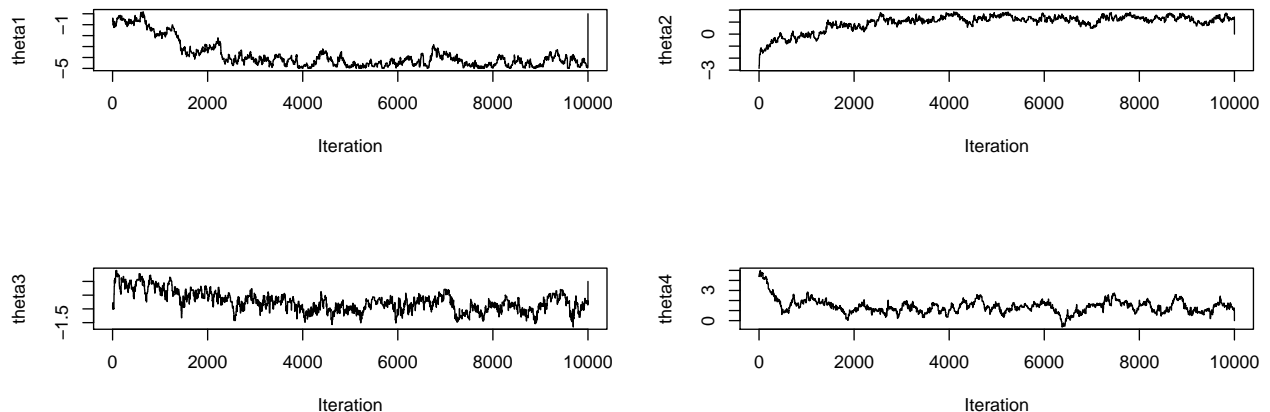


Figure 7: Convergence of the parameter estimates for the exchange algorithm

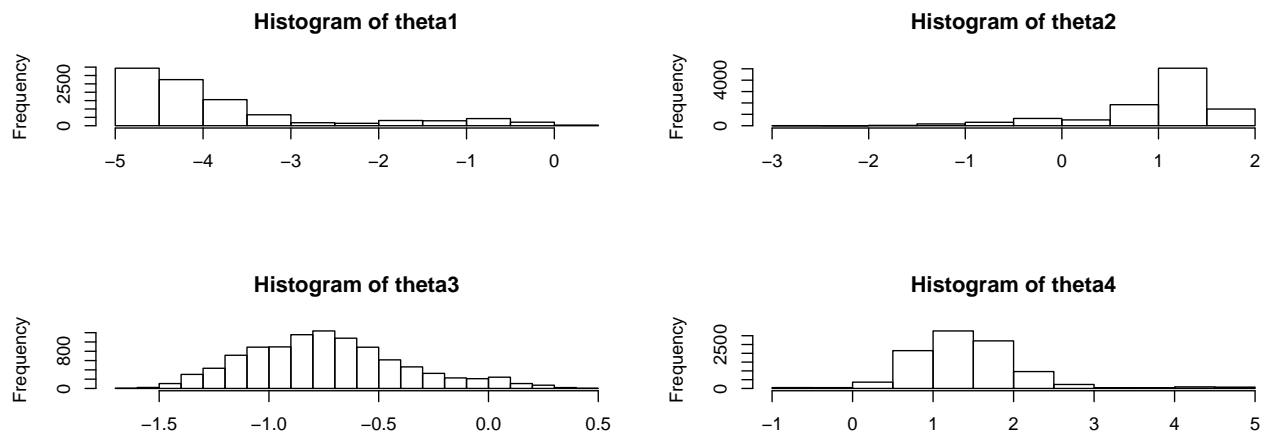


Figure 8: Histograms of the Exchanged MCMC samples for the parameters of interest

References

- Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.
- Richard G Everitt. Bayesian parameter estimation for latent markov random fields and social networks. *Journal of Computational and graphical Statistics*, 21(4):940–960, 2012.
- Richard G Everitt, Dennis Prangle, Philip Maybank, and Mark Bell. Marginal sequential monte carlo for doubly intractable models. *arXiv preprint arXiv:1710.04382*, 2017.
- David Hunter. Statistical models for networks. *Statistics Student Seminar*, 2006.
- Ick Hoon Jin, Ying Yuan, and Faming Liang. Bayesian analysis for exponential random graph models using the adaptive exchange sampler. *Statistics and its interface*, 6(4):559, 2013.
- Julian Korber. Bayesian analysis of network data. model selection and evaluation of the exponential random graph model. 2018.
- Johan H Koskinen, Garry L Robins, and Philippa E Pattison. Analysing exponential random graph (p-star) models with missing data using bayesian data augmentation. *Statistical Methodology*, 7(3):366–384, 2010.
- Martina Morris, Mark S Handcock, and David R Hunter. Specification of exponential-family random graph models: terms and computational aspects. *Journal of statistical software*, 24(4):1548, 2008.
- Iain Murray, Zoubin Ghahramani, and David MacKay. Mcmc for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848*, 2012.
- MEJ Newman. Network structure from rich but noisy data. *Nature Physics*, 14(6):542, 2018.
- Jaewoo Park and Murali Haran. Bayesian inference in the presence of intractable normalizing functions. *Journal of the American Statistical Association*, (just-accepted), 2018.
- Christian S Schmid and Bruce A Desmarais. Exponential random graph models with big networks: Maximum pseudolikelihood estimation and the parametric bootstrap. In *Big Data (Big Data)*, 2017 IEEE International Conference on, pages 116–121. IEEE, 2017.
- Tom AB Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40, 2002.
- David Strauss and Michael Ikeda. Pseudolikelihood estimation for social networks. *Journal of the American statistical association*, 85(409):204–212, 1990.