Inference and Simulation within the general ERGM framework

Mark S. Handcock

handcock@ucla.edu

Statistical Analysis of Networks
November 3, 2024

Obtaining samples via MCMC

MCMC Idea:

Simulate a discrete-time Markov chain whose stationary distribution is the distribution we want to sample from.

Obtaining samples via MCMC

MCMC Idea:

Simulate a discrete-time Markov chain whose stationary distribution is the distribution we want to sample from.

We'll discuss two common ways to run such a Markov chain:

- Gibbs sampling
- A Metropolis algorithm

Gibbs sampling

- First, select a pair of nodes at random, say (i, j).

Gibbs sampling

- First, select a pair of nodes at random, say (i, j).
- Decide whether to set $Y_{ij} = 0$ or $Y_{ij} = 1$ at the next time step according to the conditional distribution of Y_{ij} given the rest of the network (Y_{ij}^c).

Gibbs sampling

- First, select a pair of nodes at random, say (i, j).
- Decide whether to set $Y_{ij} = 0$ or $Y_{ij} = 1$ at the next time step according to the conditional distribution of Y_{ij} given the rest of the network (Y_{ij}^c).
- Based on an earlier calculation, we obtain

$$P_{\eta_0}(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c) = \frac{\exp\{\eta_0 \cdot \Delta(g(y))_{ij}\}}{(1 + \exp\{\eta_0 \cdot \Delta(g(y))_{ij}\})}.$$

Note: To run the MCMC, the values of $g(y_{ij}^+)$ *and* $g(y_{ij}^-)$ *are not needed; only the* difference $\Delta(g(y))_{ij}$ *matters.*

Metropolis algorithm

- First, select a pair of nodes at random, say (i, j).

Metropolis algorithm

- First, select a pair of nodes at random, say (i,j).
- Calculate the ratio

$$\begin{aligned}\pi &= \frac{P(Y_{ij} \text{ changes} | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} \text{ does not change} | Y_{ij}^c = y_{ij}^c)} \\ &= \exp\{\pm\eta_0 \cdot \Delta(g(y))_{ij}\}\end{aligned}$$

Metropolis algorithm

- First, select a pair of nodes at random, say (i, j).
- Calculate the ratio

$$\begin{aligned}\pi &= \frac{P(Y_{ij} \text{ changes} | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} \text{ does not change} | Y_{ij}^c = y_{ij}^c)} \\ &= \exp\{\pm\eta_0 \cdot \Delta(g(y))_{ij}\}\end{aligned}$$

- Accept the change of Y_{ij} with probability $\min\{1, \pi\}$.

Metropolis algorithm

- First, select a pair of nodes at random, say (i, j).
- Calculate the ratio

$$\begin{aligned} \pi &= \frac{P(Y_{ij} \text{ changes} | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} \text{ does not change} | Y_{ij}^c = y_{ij}^c)} \\ &= \exp\{\pm\eta_0 \cdot \Delta(g(y))_{ij}\} \end{aligned}$$

- Accept the change of Y_{ij} with probability $\min\{1, \pi\}$.
- This scheme generally has better properties than Gibbs sampling.

Note: To run the MCMC, the values of $g(y_{ij}^+)$ *and* $g(y_{ij}^-)$ *are not needed; only the* difference $\Delta(g(y))_{ij}$ *matters.*

Maximum Likelihood Estimation

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta{\cdot}g(y)\}}{\kappa(\eta)}$$

Maximum Likelihood Estimation

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta \cdot g(y)\}}{\kappa(\eta)}$$

- It follows that $\kappa(\eta)$ is a normalizing "constant":

$$\kappa(\eta) = \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\eta \cdot g(z)\}.$$

Maximum Likelihood Estimation

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta \cdot g(y)\}}{\kappa(\eta)}, \text{ where } g(y^{\text{obs}}) = 0$$

- It follows that $\kappa(\eta)$ is a normalizing "constant":

$$\kappa(\eta) = \sum_{\substack{\text{all possible}\\ \text{graphs } z}} \exp\{\eta \cdot g(z)\}.$$

- Replacing $g(y)$ by $g(y) - g(y^{\text{obs}})$ leaves $P_\eta(Y = y)$ unchanged; thus, we "recenter" $g(y)$ so that $g(y^{\text{obs}}) = 0$.

Why it is difficult to find the MLE

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta \cdot g(y)\}}{\kappa(\eta)}, \text{ where } g(y^{\text{obs}}) = 0$$

Why it is difficult to find the MLE

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta \cdot g(y)\}}{\kappa(\eta)}, \text{ where } g(y^{\text{obs}}) = 0$$

- The loglikelihood function is

$$\ell(\eta) = -\log \kappa(\eta) = \log \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\eta \cdot s(z)\}.$$

Why it is difficult to find the MLE

The model:

$$P_\eta(Y = y) = \frac{\exp\{\eta \cdot g(y)\}}{\kappa(\eta)}, \text{ where } g(y^{\text{obs}}) = 0$$

- The loglikelihood function is

$$\ell(\eta) = -\log\kappa(\eta) = \log \sum_{\substack{\text{all possible} \\ \text{graphs } z}} \exp\{\eta \cdot s(z)\}.$$

- Merely evaluating (let alone maximizing) $\ell(\eta)$ is somewhat computationally burdensome...

How burdensome?

For this undirected, 34-node graph, computing $\kappa(\eta)$ directly requires summation of

How burdensome?

For this undirected, 34-node graph, computing $\kappa(\eta)$ directly requires summation of

$$7{,}547{,}924{,}849{,}643{,}082{,}704{,}483{,}109{,}161{,}976{,}537{,}781{,}833{,}842{,}440{,}832{,}880{,}856{,}752{,}412{,}600{,}491{,}248{,}324{,}784{,}297{,}704{,}172{,}253{,}450{,}355{,}317{,}535{,}082{,}936{,}750{,}061{,}527{,}689{,}799{,}541{,}169{,}259{,}849{,}585{,}265{,}122{,}868{,}502{,}865{,}392{,}087{,}298{,}790{,}653{,}952$$

terms.

MLE Revisited

- Remember, $\kappa(\eta)$ is *really* hard to compute.

MLE Revisited

- Remember, $\kappa(\eta)$ is *really* hard to compute.
- However, suppose we fix η_0. A bit of algebra shows that

$$\mathbf{E}_{\eta_0}\left[\exp\left\{(\eta_0 - \eta)\cdot g(Y)\right\}\right] = \frac{\kappa(\eta_0)}{\kappa(\eta)}.$$

MLE Revisited

- Remember, $\kappa(\eta)$ is *really* hard to compute.
- However, suppose we fix η_0. A bit of algebra shows that

$$\mathbf{E}_{\eta_0}\left[\exp\left\{(\eta_0 - \eta)\cdot g(Y)\right\}\right] = \frac{\kappa(\eta_0)}{\kappa(\eta)}.$$

- Thus, $\kappa(\eta_0)/\kappa(\eta)$ is the expectation of a function of a random network, where the random behavior is governed by the known constant η_0.

Law of Large Numbers to the Rescue!

The LOLN suggests that we approximate an unknown population mean by a sample mean.

Law of Large Numbers to the Rescue!

The LOLN suggests that we approximate an unknown population mean by a sample mean.

Thus,

$$
\begin{aligned}
\kappa(\eta_0)/\kappa(\eta) &= \mathbf{E}_{\eta_0}\left(\exp\left\{(\eta_0 - \eta)\cdot g(Y)\right\}\right) \\
&\approx \frac{1}{m}\sum_{i=1}^{m} \exp\left\{(\eta_0 - \eta)\cdot s(Y_i)\right\},
\end{aligned}
$$

where $Y_1, Y_2, \ldots, Y_m$ is a random sample of networks from the distribution defined by the ERGM with parameter η_0.

An approximate loglikelihood

- Using the LOLN approximation, we find that the difference in loglikelihoods is

$$\ell(\eta) - \ell(\eta_0) \quad = \quad \log \frac{\kappa(\eta_0)}{\kappa(\eta)}$$

An approximate loglikelihood

- Using the LOLN approximation, we find that the difference in loglikelihoods is

$$\begin{aligned} \ell(\eta) - \ell(\eta_0) &= \log \frac{\kappa(\eta_0)}{\kappa(\eta)} \\ &= \log \mathbf{E}_{\eta_0} \left(\exp\left\{ (\eta_0 - \eta) \cdot g(Y) \right\} \right) \end{aligned}$$

An approximate loglikelihood

- Using the LOLN approximation, we find that the difference in loglikelihoods is

$$\begin{aligned}
\ell(\eta) - \ell(\eta_0) &= \log \frac{\kappa(\eta_0)}{\kappa(\eta)} \\
&= \log \mathbf{E}_{\eta_0} \left(\exp\left\{(\eta_0 - \eta)\cdot g(Y)\right\}\right) \\
&\approx \log \frac{1}{M} \sum_{i=1}^{M} \exp\left\{(\eta_0 - \eta)\cdot s(Y^{(i)})\right\}.
\end{aligned}$$

An approximate loglikelihood

- Using the LOLN approximation, we find that the difference in loglikelihoods is

$$
\begin{aligned}
\ell(\eta) - \ell(\eta_0) &= \log \frac{\kappa(\eta_0)}{\kappa(\eta)} \\
&= \log \mathbf{E}_{\eta_0} \left(\exp\left\{ (\eta_0 - \eta) \cdot g(Y) \right\} \right) \\
&\approx \log \frac{1}{M} \sum_{i=1}^{M} \exp\left\{ (\eta_0 - \eta) \cdot s(Y^{(i)}) \right\}.
\end{aligned}
$$

- Given a random sample of networks from P_{η_0}, we can thus approximate (and subsequently maximize) the loglikelihood shifted by a constant.

How should η_0 be chosen?

- Theoretically, the estimated value of $\ell(\eta) - \ell(\eta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of η_0.

How should η_0 be chosen?

- Theoretically, the estimated value of $\ell(\eta) - \ell(\eta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of η_0.
- However, this convergence can be agonizingly slow, especially if η_0 is not chosen close to the maximizer of the likelihood.

How should η_0 be chosen?

- Theoretically, the estimated value of $\ell(\eta) - \ell(\eta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of η_0.
- However, this convergence can be agonizingly slow, especially if η_0 is not chosen close to the maximizer of the likelihood.
- A choice that sometimes works is the MPLE (maximum pseudolikelihood estimate)

Conditional log-odds of an edge

Notation: For a network x and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i, j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

Conditional log-odds of an edge

Notation: For a network x and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i, j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

Conditional on $Y_{ij}^c = y_{ij}^c$, Y has only two possible states, depending on whether $Y_{ij} = 0$ or $Y_{ij} = 1$.

Conditional log-odds of an edge

Notation: For a network x and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i, j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

Conditional on $Y_{ij}^c = y_{ij}^c$, Y has only two possible states, depending on whether $Y_{ij} = 0$ or $Y_{ij} = 1$.
Let's calculate the ratio of the two respective probabilities:

Conditional log-odds of an edge

Notation: For a network x and a pair (i,j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i,j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

$$\frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \frac{\exp\{\eta \cdot g(y_{ij}^+)\}}{\exp\{\eta \cdot g(y_{ij}^-)\}}$$

Conditional log-odds of an edge

Notation: For a network x and a pair (i,j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i,j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

$$\frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \exp\{\eta \cdot [g(y_{ij}^+) - g(y_{ij}^-)]\}$$

Conditional log-odds of an edge

Notation: For a network x and a pair (i, j) of nodes,

- $y_{ij} = 0$ or 1, depending on whether there is an edge
- y_{ij}^c denotes the status of all pairs in x other than (i, j)
- y_{ij}^+ denotes the same network as x but with $y_{ij} = 1$
- y_{ij}^- denotes the same network as x but with $y_{ij} = 0$

$$\log \frac{P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)}{P(Y_{ij} = 0 | Y_{ij}^c = y_{ij}^c)} = \eta \cdot [g(y_{ij}^+) - g(y_{ij}^-)]$$

Conditional log-odds of an edge

Notation: For a network x and a pair (i, j) of nodes,

- $\Delta(g(y))_{ij}$ denotes the vector of change statistics,

$$\Delta(g(y))_{ij} = g(y_{ij}^{+}) - g(y_{ij}^{-}).$$

So $\Delta(g(y))_{ij}$ is the conditional log-odds of edge (i, j).

$$\log \frac{P(Y_{ij} = 1 | Y_{ij}^{c} = y_{ij}^{c})}{P(Y_{ij} = 0 | Y_{ij}^{c} = y_{ij}^{c})} \quad = \quad \eta \cdot \Delta(g(y))_{ij}$$

Maximum Pseudolikelihood: Alternative to MLE?

- What if we approximate the marginal $P(Y_{ij} = 1)$ by the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$?

Maximum Pseudolikelihood: Alternative to MLE?

- What if we approximate the marginal $P(Y_{ij} = 1)$ by the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$?
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \eta \cdot \Delta(g(y^{\mathrm{obs}}))_{ij},$$

so we obtain $\hat{\eta}$ using simple logistic regression.

Maximum Pseudolikelihood: Alternative to MLE?

- What if we approximate the marginal $P(Y_{ij} = 1)$ by the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$?
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \eta \cdot \Delta(g(y^{\text{obs}}))_{ij},$$

so we obtain $\hat{\eta}$ using simple logistic regression.

- The log-pseudolikelihood function is then

$$\ell\mathrm{PL}(\eta) = \sum \log[P(Y_{ij} = y_{ij} | Y_{ij}^c = y_{ij}^c)]$$

Maximum Pseudolikelihood: Alternative to MLE?

- What if we approximate the marginal $P(Y_{ij} = 1)$ by the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$?
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \eta \cdot \Delta(g(y^{\mathrm{obs}}))_{ij},$$

so we obtain $\hat{\eta}$ using simple logistic regression.

- The log-pseudolikelihood function is then

$$\ell \mathrm{PL}(\eta) = \sum \log[P(Y_{ij} = y_{ij} | Y_{ij}^c = y_{ij}^c)]$$

- Result: The maximum pseudolikelihood estimate is then the value that maximizes $\ell \mathrm{PL}(\eta)$ as a function of η..

Maximum Pseudolikelihood: Alternative to MLE?

- What if we approximate the marginal $P(Y_{ij} = 1)$ by the conditional $P(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)$?
- Then the Y_{ij} are independent with

$$\log \frac{P(Y_{ij} = 1)}{P(Y_{ij} = 0)} = \eta \cdot \Delta(g(y^{\mathrm{obs}}))_{ij},$$

  so we obtain $\hat{\eta}$ using simple logistic regression.
- The log-pseudolikelihood function is then

$$\ell\mathrm{PL}(\eta) = \sum \log[P(Y_{ij} = y_{ij} | Y_{ij}^c = y_{ij}^c)]$$

- Result: The maximum pseudolikelihood estimate is then the value that maximizes $\ell\mathrm{PL}(\eta)$ as a function of η..
- Unfortunately, little is known about the quality of MPL estimates.

Implementation

- This has been implemented in the `ergm` package, available on CRAN.
- See `http://www.statnet.org` for more general code.
- Many different versions and ideas.
- See Volume 24 of the *Journal of Statistical Software*

MCMC diagnostics

- The `ergm` package uses a Metroplis-Hastings algorithm to sample from the space of graphs given parameter values.
- It then uses the Geyer-Thompson device to estimate the MLE based on the sample of graphs.
- How do we know the MCMC algorithm has approximately converged, so that the graph sample can be used to get an accurate approximation of the MLE?
- We can use standard numerical and graphical disgnostics on graph statistics of the sampled graphs (i.e., statistics of the sampled graphs).

mcmc.diagnostics

- The `mcmc.diagnostics` command computes many diagnostics from an `ergm` fit object.
- By default it produces *trace plots* and *density plots* of the statistics.
- In fact, an `ergm` output object contains the matrix of statistics from the MCMC run as component `$sample`. This matrix is actually an object of class `mcmc` and can be used directly in the `coda` package to assess MCMC convergence.
- Hence all MCMC diagnostic methods available in coda are available directly by calling `coda`.

Example mcmc.diagnostics

```
data(florentine)
# Fit a simple model
gest <- ergm(flomarriage ~ edges + kstar(2))
summary(gest)

Call: ergm(formula = flomarriage ~ edges + kstar(2))

Monte Carlo Maximum Likelihood Results:

        Estimate Std. Error MCMC % z value Pr(>|z|)
edges  -1.541591   0.822422      0  -1.874   0.0609 .
kstar2 -0.008306   0.167888      0  -0.049   0.9605
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 166.4  on 120  degrees of freedom
 Residual Deviance: 108.1  on 118  degrees of freedom

AIC: 112.1  BIC: 117.7  (Smaller is better. MC Std. Err. = 0.006
```

Example mcmc.diagnostics

```
mcmc.diagnostics(gest)

Sample statistics summary:

Iterations = 14336:262144
Thinning interval = 1024
Number of chains = 1
Sample size per chain = 243

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean     SD Naive SE Time-series SE
edges  -0.4815  4.076   0.2615         0.2615
kstar2 -1.5350 19.967   1.2809         1.2809
```

Example mcmc.diagnostics

```
Sample statistics cross-correlations:
          edges    kstar2
edges  1.0000000 0.9540829
kstar2 0.9540829 1.0000000

Sample statistics auto-correlation:
Chain 1
                edges       kstar2
Lag 0      1.00000000  1.000000000
Lag 1024 -0.01853626  0.008273854
Lag 2048 -0.07054771 -0.052840888
Lag 3072  0.03625916 -0.009676151
Lag 4096  0.02364600 -0.007372892
Lag 5120  0.02513591  0.026592066
```

mcmc.diagnostics