

Fitting exponential random graph models via maximum likelihood

David R. Hunter

Department of Statistics

Penn State University

email: dhunter@stat.psu.edu

In this talk I will try to explain...

In this talk I will try to explain...

```
=====
Summary of output
=====
```

```
Formula:   addhealth ~ kstar(1:4) + match("grade", TRUE) + nodecov("sex")
```

```
Newton-Raphson iterations: 5
```

```
MCMC sample of size 5000
```

```
Monte Carlo MLE Results:
```

	theta0	estimate	s.e.	p-value
star1	-5.7302889	-5.7327631	0.011396	<1e-04
star2	0.7152238	0.7150962	0.008898	<1e-04
star3	-0.1594760	-0.1594351	0.009331	<1e-04
star4	0.0084060	0.0056273	0.010524	0.593
dmatch.grade.7	3.0997648	3.0987222	0.009029	<1e-04
dmatch.grade.8	2.7890712	2.7879443	0.011561	<1e-04
dmatch.grade.9	3.1474176	3.1464718	0.009308	<1e-04
dmatch.grade.10	3.3287244	3.3281948	0.009596	<1e-04
dmatch.grade.11	2.9631418	2.9622897	0.009976	<1e-04
dmatch.grade.12	4.4782321	4.4764830	0.009919	<1e-04
nodecov.sex	-0.0002162	-0.0001657	0.008921	0.985

```
Log likelihood:    0.1008128
```

What is an ERG model?

ERG (Exponential Random Graph) \equiv “p-star”.

Given a set of n nodes, let X denote a random graph on those nodes and x denote a particular (fixed) graph on those nodes.

Then

$$P_{\theta}(X = x) = \frac{\exp\{\theta^t s(x)\}}{c(\theta)},$$

where

- θ is an unknown vector of parameters
- $s(x)$ is a known vector of graph statistics on x

With

$$P_{\theta}(X = x) = \frac{\exp\{\theta^t s(x)\}}{c(\theta)},$$

we see that

$$c(\theta) = \sum_{\text{all graphs } y} \exp\{\theta^t s(y)\},$$

where the sum is taken over all possible distinct graphs on the n given nodes.

Even in the simplest case of undirected graphs without self-edges, the number of graphs in the sum is very large.

(This number has roughly $.15n^2$ digits!)

How should we fit the model (i.e., estimate θ)?

Consider the tried and true *maximum likelihood estimation*:

The loglikelihood function $\ell(\theta)$ is $\log P_\theta(X = x^{obs})$, or

$$\begin{aligned}\ell(\theta) &= \log \left(\frac{\exp\{\theta^t s(x^{obs})\}}{c(\theta)} \right) \\ &= \theta^t s(x^{obs}) - \log c(\theta).\end{aligned}$$

We'd like to maximize $\ell(\theta)$, but unfortunately $\log c(\theta)$ is no easier to work with than $c(\theta)$.

Pseudolikelihood: A possible alternative?

Let X_{ij}^c denote the status of all dyads in X except X_{ij} , so that X_{ij}^c is in a sense “everything else”.

If we hold X_{ij}^c fixed at x_{ij}^c , then there are only two possible values of X , depending on whether X_{ij} is 0 or 1.

Stated differently: Conditional on X_{ij}^c , the random behavior of X can essentially be modeled as an (unfair) coin flip.

More precisely,

$$\frac{P(X_{ij} = 1 | X_{ij}^c = x_{ij}^c)}{P(X_{ij} = 0 | X_{ij}^c = x_{ij}^c)} = \frac{\exp\{\theta^t s(x_{ij}^+)\}}{c(\theta)} \frac{c(\theta)}{\exp\{\theta^t s(x_{ij}^-)\}}.$$

From

$$\frac{P(X_{ij} = 1 | X_{ij}^c = x_{ij}^c)}{P(X_{ij} = 0 | X_{ij}^c = x_{ij}^c)} = \exp\{\theta^t [s(x_{ij}^+) - s(x_{ij}^-)]\},$$

we get **logit** $P(X_{ij} = 1 | X_{ij}^c = x_{ij}^c) = \theta^t \Delta(s(x))_{ij}$, where

$$\Delta(s(x))_{ij} = s(x_{ij}^+) - s(x_{ij}^-)$$

is the vector of change statistics.

Remember this form of the logit; it will return in the MLE context.

Maximum Pseudolikelihood

What if the conditional probability $P(X_{ij} = 1 | X_{ij}^c = x_{ij}^c)$ were equal to the marginal probability $P(X_{ij} = 1)$?

Then we would have

$$\text{logit } P(X_{ij} = 1) = \theta^t \Delta(s(x))_{ij}$$

with the X_{ij} independent, so we could estimate θ using logistic regression.

The resulting estimate is called the maximum pseudolikelihood estimate (MPLE).

Unfortunately, little is known about quality of MPL estimates.

Back to MLE!

Remember, $c(\theta)$ is HARD to compute. However:

Fix θ_0 . Consider the quantity

$$\begin{aligned}
 E_{\theta_0} \left(\exp \left\{ (\theta - \theta_0)^t s(X) \right\} \right) &= \sum_{\text{all graphs } y} \exp \left\{ (\theta - \theta_0)^t s(y) \right\} P(X = y) \\
 &= \sum_{\text{all graphs } y} \exp \left\{ (\theta - \theta_0)^t s(y) \right\} \left(\frac{\exp \{ \theta_0^t s(y) \}}{c(\theta_0)} \right) \\
 &= \sum_{\text{all graphs } y} \frac{\exp \left\{ (\theta)^t s(y) \right\}}{c(\theta_0)} \\
 &= \frac{c(\theta)}{c(\theta_0)}.
 \end{aligned}$$

Thus, $c(\theta)/c(\theta_0)$ is a population mean.

Thus, we can *estimate* the population mean (expectation)

$$c(\theta)/c(\theta_0) = E_{\theta_0} \left(\exp \left\{ (\theta - \theta_0) t_s(X) \right\} \right)$$

by the sample mean

$$\frac{1}{M} \sum_{i=1}^M \exp \left\{ (\theta - \theta_0) t_s(X^{(i)}) \right\},$$

where $X^{(1)}, X^{(2)}, \dots, X^{(M)}$ is a random sample of graphs from the distribution defined by the ERGM with parameter θ_0 .

To estimate $c(\theta)/c(\theta_0)$, we need to obtain a sample from the distribution defined by the ERGM with parameter θ_0 .

In principle, this may be accomplished by running a discrete-time Markov chain whose stationary distribution is the distribution we wish to sample from. This is the MCMC (Markov Chain Monte Carlo) idea.

There are two common ways to run such a Markov chain, **Gibbs sampling** and a **Metropolis algorithm**.

Though each can be made more complicated, in its basic form each method works by deciding, at each time step, whether to toggle a randomly selected dyad.

Gibbs sampling

First, select a dyad at random, say (i, j) .

Gibbs sampling sets X_{ij} at the next time step according to the conditional probabilities given X_{ij}^c .

We saw earlier that logit $P(X_{ij} = 1 | X_{ij}^c = x_{ij}^c)$ is given by $\theta_0^t \Delta(s(x))_{ij}$, where $\Delta(s(x))_{ij} = s(x_{ij}^+) - s(x_{ij}^-)$ is the vector of change statistics.

Thus, the Gibbs sampler sets $X_{ij} = 1$ with probability

$$\frac{\exp\{\theta_0^t \Delta(s(x))_{ij}\}}{(1 + \exp\{\theta_0^t \Delta(s(x))_{ij}\})}$$

and it sets $X_{ij} = 0$ otherwise.

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

Metropolis-Hastings

Gibbs sampling is one form of a **Metropolis-Hastings** algorithm. Another is to calculate the ratio

$$\begin{aligned}\pi &= \frac{P(X_{ij} \text{ changes} | X_{ij}^c = x_{ij}^c)}{P(X_{ij} \text{ does not change} | X_{ij}^c = x_{ij}^c)} \\ &= \exp\{\pm\theta_0^t \Delta(s(x))_{ij}\}\end{aligned}$$

and then accept the change of X_{ij} with probability $\min\{1, \pi\}$. (Strictly speaking, this is a **Metropolis** algorithm.)

This is the scheme used by the ERGM program.

Note: Again, the values of $s(x_{ij}^+)$ and $s(x_{ij}^-)$ are not needed; only the difference $\Delta(s(x))_{ij}$ matters.

Change statistics vs. Absolute statistics

As we've seen, the ERGM program only needs to keep track of how $s(X)$ changes at each step of the Markov chain; the actual values of $s(X)$ are not important.

If we start the chain at x^{obs} , keeping a running total of all changes gives $s(X) - s(x^{obs})$.

But we wanted to estimate $\frac{c(\theta)}{c(\theta_0)}$ by

$$\frac{1}{M} \sum_{i=1}^M \exp \left\{ (\theta - \theta_0) s(X^{(i)}) \right\},$$

which appears to require $s(X^{(i)})$.

Fortunately, $s(X^{(i)}) - s(x^{obs})$ is good enough.

Why is $s(X^{(i)}) - s(x^{obs})$ good enough?

Let's revisit the likelihood function:

$$\begin{aligned} P_{\theta}(X = x^{obs}) &= \frac{\exp\{\theta^t s(x^{obs})\}}{\sum_{\text{all graphs } y} \exp\{\theta^t s(y)\}} \\ &= \frac{1}{\sum_{\text{all graphs } y} \exp\{\theta^t [s(y) - s(x^{obs})]\}}. \end{aligned}$$

So the loglikelihood may be rewritten

$$\ell(\theta) = -\log \left(\sum_{\text{all graphs } y} \exp\{\theta^t [s(y) - s(x^{obs})]\} \right).$$

With

$$\ell(\theta) = -\log \left(\sum_{\text{all graphs } y} \exp \left\{ \theta^t [s(y) - s(x^{obs})] \right\} \right),$$

we may estimate $\ell(\theta) - \ell(\theta_0)$ by

$$-\log \left(\frac{1}{M} \sum_{i=1}^M \exp \left\{ (\theta - \theta_0)^t [s(X^{(i)}) - s(x^{obs})] \right\} \right). \quad (1)$$

Thus, ERGM computes and returns a large number of realizations of $s(X) - s(x^{obs})$ from the distribution determined by θ_0 .

This sample is then used to maximize (1) as a function of θ .

How should θ_0 be chosen?

Theoretically, the estimated value of $\ell(\theta) - \ell(\theta_0)$ converges to the true value as the size of the MCMC sample increases, regardless of the value of θ_0 .

However, this convergence can be agonizingly slow if θ_0 is not chosen close to the maximizer of the likelihood.

ERGM allows the user to specify the value of θ_0 , but by default the MPLLE is used.

Another look at the output:

```
=====
Summary of output
=====
```

```
Formula:   addhealth ~ kstar(1:4) + match("grade", TRUE) + nodecov("sex")
```

```
Newton-Raphson iterations: 5
```

```
MCMC sample of size 5000
```

```
Monte Carlo MLE Results:
```

	theta0	estimate	s.e.	p-value
star1	-5.7302889	-5.7327631	0.011396	<1e-04
star2	0.7152238	0.7150962	0.008898	<1e-04
star3	-0.1594760	-0.1594351	0.009331	<1e-04
star4	0.0084060	0.0056273	0.010524	0.593
dmatch.grade.7	3.0997648	3.0987222	0.009029	<1e-04
dmatch.grade.8	2.7890712	2.7879443	0.011561	<1e-04
dmatch.grade.9	3.1474176	3.1464718	0.009308	<1e-04
dmatch.grade.10	3.3287244	3.3281948	0.009596	<1e-04
dmatch.grade.11	2.9631418	2.9622897	0.009976	<1e-04
dmatch.grade.12	4.4782321	4.4764830	0.009919	<1e-04
nodecov.sex	-0.0002162	-0.0001657	0.008921	0.985

```
Log likelihood:   0.1008128
```