

Fitting ERGMs using `statnet` in R

David R. Hunter

`dhunter@stat.psu.edu`

Department of Statistics

The Pennsylvania State University

February 17, 2005

Loading statnet

After starting R...

```
> library(statnet)
Statistical Modeling of Network and Graph Data
Version 0.6-8 created on January 1, 2004.
copyright (c) 2003, Mark S. Handcock, University of Washington
                David R. Hunter, Penn State University
                Carter T. Butts, University of California-Irvine
                Martina Morris, University of Washington
Type help(package="statnet") to get started.
```

The R interface allows command-line control of ERGM-fitting routines.

Analogy: Fitting a linear model vs. fitting an ERGM

Linear model:

```
> summary( lm(weight ~ height + sex))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-111.6065	31.9985	-3.488	0.000582
height	3.6199	0.5417	6.682	1.73e-10
sex	14.8654	4.7929	3.102	0.002164

ERGM:

```
> summary( ergm(my.nw ~ match("Sex") + kstar(1:2) + triangle,  
               MPLEonly = TRUE))
```

Pseudolikelihood Results: (Note: Standard errors are suspect.)

	estimate	s.e.	p-value
nodematch.Sex	0.572253	0.15553	0.000234
kstar1	-2.715075	0.08822	< 1e-04
kstar2	0.001115	0.02946	0.969800
triangle	2.887518	0.14842	< 1e-04

The difficulty of maximum likelihood

Goal:

$$\text{Maximize } L(\boldsymbol{\theta}) = \frac{\exp\{\boldsymbol{\theta}^t \mathbf{g}(\mathbf{y}_{\text{obs}})\}}{\sum_{\text{all } \mathbf{x}} \exp\{\boldsymbol{\theta}^t \mathbf{g}(\mathbf{x})\}}$$

Problem: Sum in denominator has WAY too many terms.

Solution: After multiplying by $1/N$, denominator is a mean; thus, we can approximate it using a sample mean.

However, we'll need a representative sample of graphs (from a given ERGM with a fixed parameter $\boldsymbol{\theta}$) to be sure our sample mean is unbiased.

Generating a sample of graphs using `statnet`

Use Markov chain Monte Carlo — specifically, Metropolis-Hastings.

In Metropolis-Hastings, selection of a proposal distribution can be very influential on the mixing properties of the Markov chain.

`statnet` allows the user to select from (currently) 10 different types of proposals:

`conddeg`

`conddegdist`

`conddegdistswitch`

`condindeg`

`condoutdeg`

`constantedges`

`node`

`nodeedges`

`tnt`

`toggle`

Examples of `statnet` Proposal types

```
ergm(my.nw ~ match("Sex") + kstar(1:2) + triangle,  
     propoposaltype="toggle")
```

Toggle type: Select a dyad at random; propose toggling it.

```
ergm(my.nw ~ match("Sex") + kstar(1:2) + triangle,  
     propoposaltype="tnt")
```

TNT (tie/no tie) type: Select with equal probability a dyad with a tie or a dyad without a tie; propose toggling it.

TNT usually exhibits better mixing than toggle for sparse networks.

Modular design of `statnet`

Users may add network statistics to `statnet`.

For example, when `ergm` encounters the degree model term as in

```
ergm(my.nw ~ match("Sex") + degree(1:5))
```

it knows exactly what to do as long as the following two functions are in place:

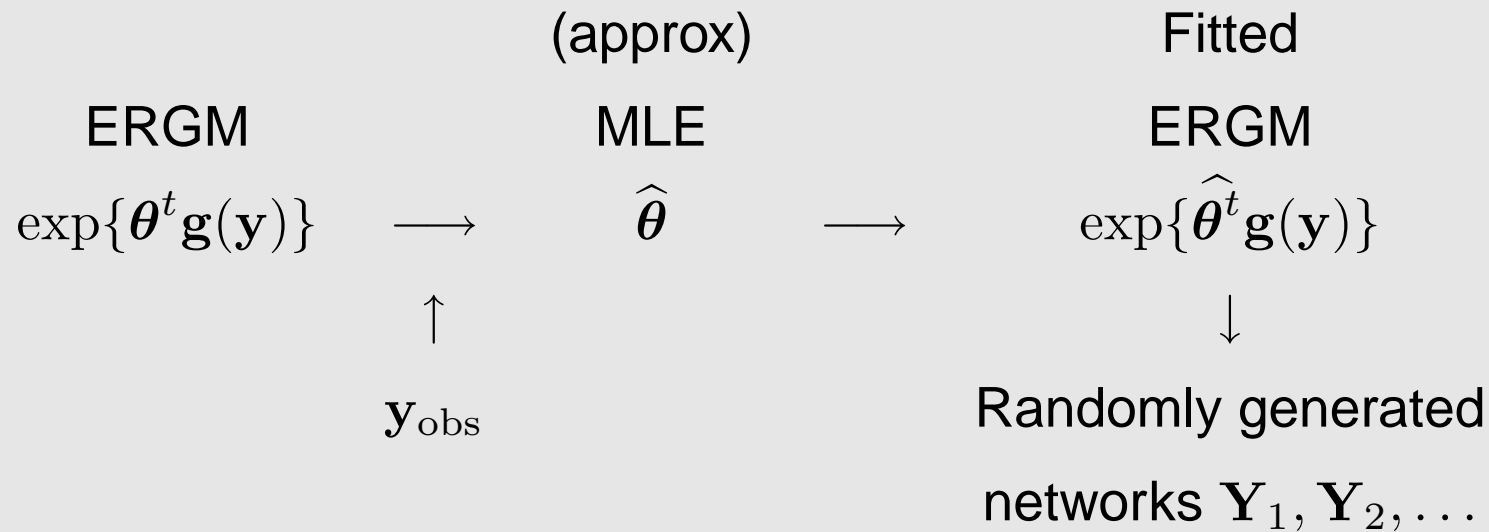
In R:

```
InitErgm.degree<-function(g, model, d, drop=TRUE, ...)  
{  
  ...
```

In C:

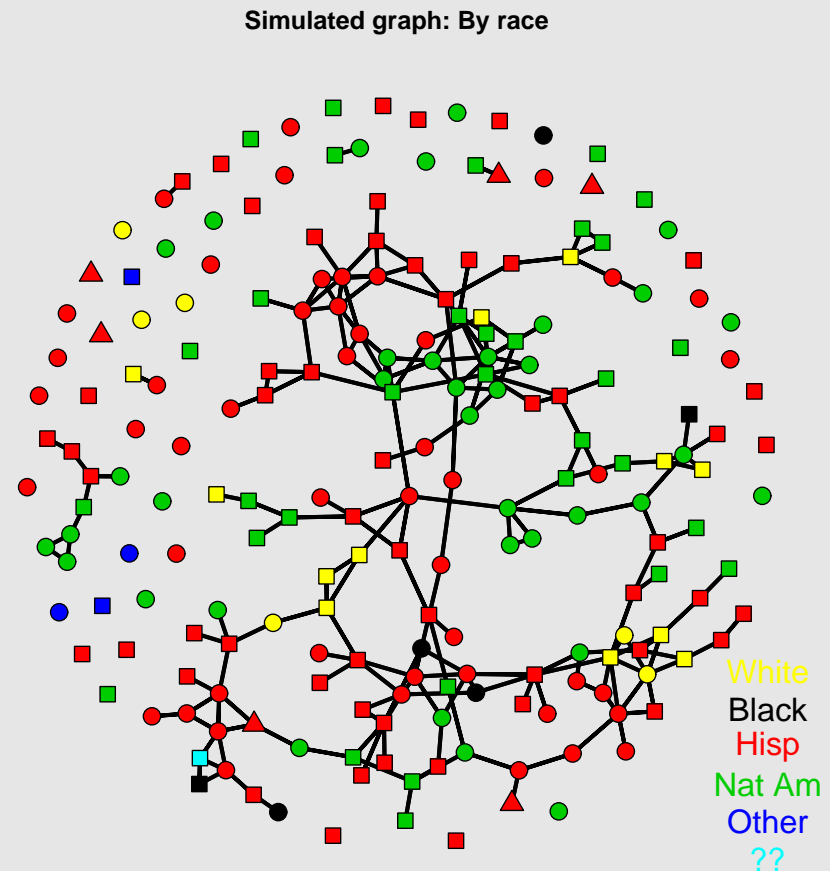
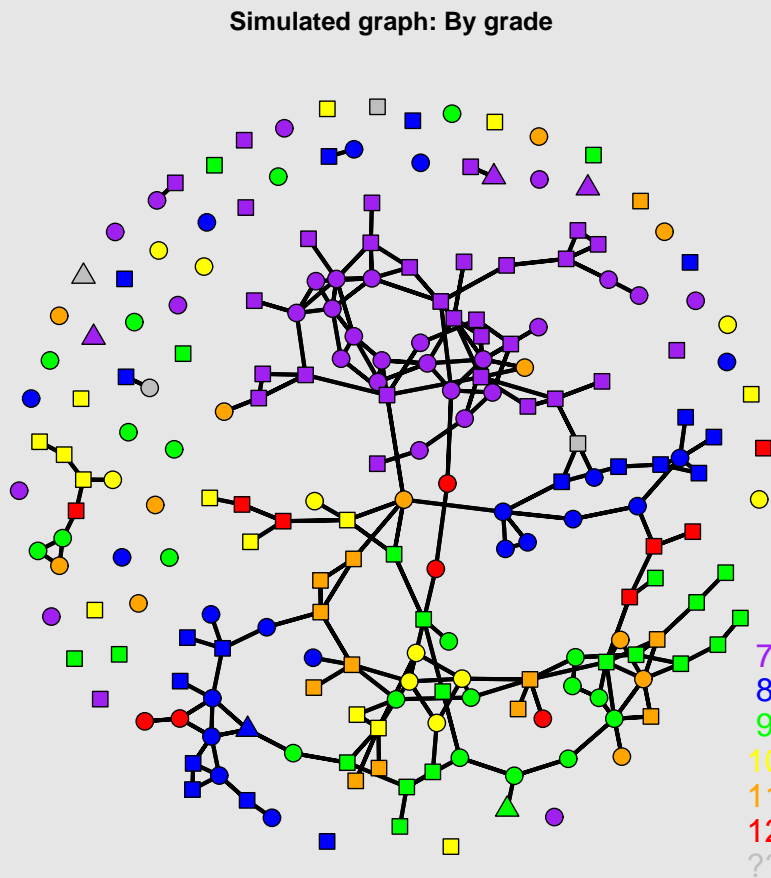
```
void d_degree (int ntoggles, Vertex *heads, Vertex *tails,  
              struct OptionInput *inp, Gptr g)  
{  
  ...
```

Goodness of fit intuition



If this ERGM fits the observed network \mathbf{y}_{obs} well, then the randomly generated graphs should resemble \mathbf{y}_{obs} .

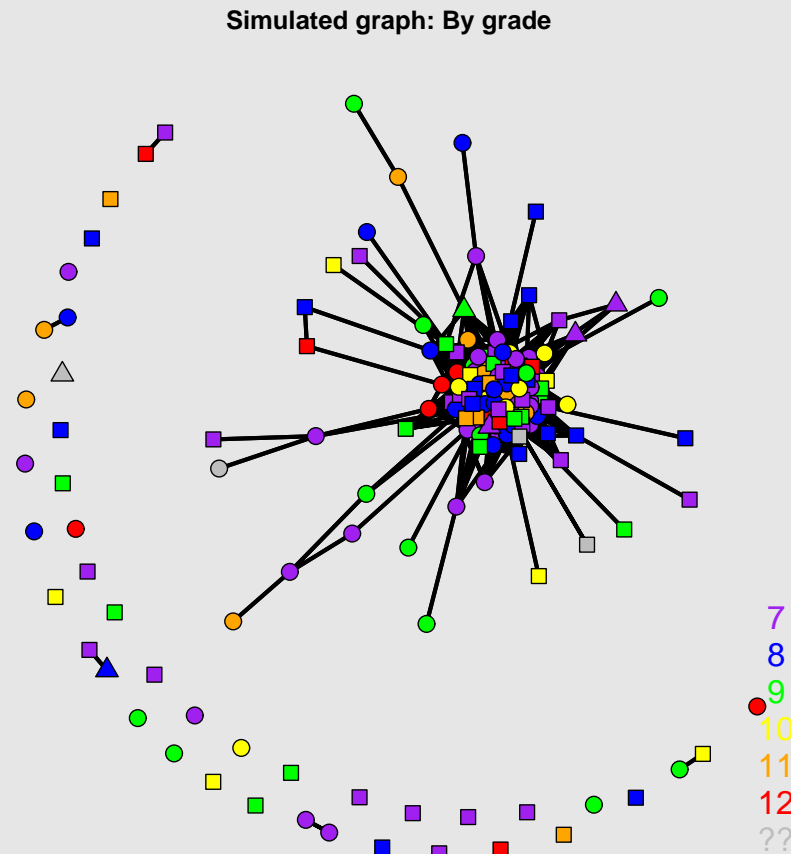
Example network: High school mutual friendships



This network was randomly generated using a simple `statnet` command and a model that fits the original AddHealth network well:

```
newgraph = rergm(goodfit, burnin=500000)
```

Example network: Poor fitting model



This network was randomly generated as follows:

```
mplefit = ergm(my.nw ~ match("Sex") + kstar(1:2) + triangle,  
              MPLOnly=TRUE)  
newgraph2 = rergm(mplefit, burnin=80000)
```

Conclusion

- `statnet` is a flexible, modifiable ERGM-fitting package.
- It is designed to work within the popular, cross-platform, command-line-driven R environment. This allows the user to take advantage of the myriad additional packages that have been written for the R environment (e.g., `sna`, `network`).
- It is written to be very fast and it has numerous features, such as the multiple proposal types, to help speed convergence in particular cases.
- It has built-in goodness-of-fit testing functionality.