Fitting ERGMs using \texttt{statnet} in R

David R. Hunter

dhunter@stat.psu.edu

Department of Statistics
The Pennsylvania State University
February 17, 2005
After starting R...

```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:**

```r
> 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:**

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

Pseudolikelihood Results: (Note: Standard errors are suspect.)
The difficulty of maximum likelihood

Goal:

Maximize \[ L(\theta) = \frac{\exp\{\theta^t g(y_{\text{obs}})\}}{\sum_{\text{all } x} \exp\{\theta^t g(x)\}} \]

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 \(\theta\)) to be sure our sample mean is unbiased.
Generating a sample of graphs using \textit{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.

\textit{statnet} allows the user to select from (currently) 10 different types of proposals:

- \texttt{conddeg}
- \texttt{conddegdistswitch}
- \texttt{condoutdeg}
- \texttt{node}
- \texttt{tnt}
- \texttt{conddegdist}
- \texttt{condindeg}
- \texttt{constantedges}
- \texttt{nodeedges}
- \texttt{toggle}
Examples of *statnet* Proposal types

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

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

**ergm**(my.nw ~ match("Sex") + kstar(1:2) + triangle,
    proposaltype="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.**
Users may add network statistics to \texttt{statnet}.

For example, when \texttt{ergm} encounters the \texttt{degree} model term as in
\begin{verbatim}
  ergm(my.nw  ~  match("Sex") + degree(1:5))
\end{verbatim}
it knows exactly what to do as long as the following two functions are in place:

\textbf{In R:}
\begin{verbatim}
  InitErgm.degree<-function(g, model, d, drop=TRUE, ...)
  {
      ...
  }
\end{verbatim}

\textbf{In C:}
\begin{verbatim}
  void d_degree (int ntoggles, Vertex *heads, Vertex *tails,
                  struct OptionInput *inp, Gptr g)
  {
      ...
  }
\end{verbatim}
Goodness of fit intuition

If this ERGM fits the observed network $y_{obs}$ well, then the randomly generated graphs should resemble $y_{obs}$. 

$\exp\{\theta^t g(y)\} \xrightarrow{\text{MLE}} \hat{\theta} \xrightarrow{\text{Fitted ERGM}} \exp\{\hat{\theta}^t g(y)\} \\
\uparrow \\
y_{obs} \\
\downarrow \\
\text{Randomly generated networks } Y_1, Y_2, \ldots$
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:

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