STATNET WEB
THE EASY WAY TO LEARN (OR TEACH) STATISTICAL MODELING OF NETWORK DATA WITH ERGMS

SUNBELT
June 26, 2018

Martina Morris, Ph.D.
Skye Bender-deMoll
What this workshop will cover

- Basic intro to statistical modeling for networks
  - Descriptive methods/EDA
  - Exponential-family Random Graph Models (ERGMs)
    - A bit of the underlying theory
    - Cross sectional network modeling

- Software: statnetWeb
  - A browser-based GUI for the statnet packages
Intro to statnet software family

Core statnet packages
For descriptive and statistical network analysis
- network, sna, ergm (cross-sectional nets)
- networkDynamic, tsna, tergm (temporal nets)

statnetWeb
Rshiny app for statnet with user-friendly GUI for descriptive and statistical analysis
Currently limited to cross-sectional methods
All of our software is

- Open source
- Published on CRAN
- Supported by online training materials

- The functionality is continuing to be improved and extended with new methods/packages

- And others are writing packages that extend our software
  - E.g., Xergm, Bergm, Hergm
Motivation

Statistical methods for network analysis are new-ish

- **Older stuff:** permutation and CUG tests
  - These are relatively easy to learn and use

- **Modern methods:** Exponential-family Random Graph Models
  - These are more complex
  - And the software can be intimidating

**statnetWeb makes it easier to get started**
Network basics (quick review, then we’ll get started)

- **Node**: the entity of interest
  - often, nodes represent people; also called actors or vertices

- **Link**: the relationship of interest
  - also called a tie, an edge, or a line

- **Network**: a set of actors and the relations among them
  - Also called a graph
Types of nodes

- Individual units
  - Humans
  - Animals
  - Airports
  - Computers
  - Genes

- Collectivities
  - Countries, cities
  - Families
  - Species
  - Organs, Sensory systems

In social networks, a focal node is called “ego”, and the nodes linked to this focal node are “alters”
Types of links (these are just examples)

- **Social**
  - Affective (like/dislike, trust/do not trust)
  - Kinship / social role (mother of, brother of, boss of)
  - Exchange (advice seeking, sexual intercourse, trade)
  - Cognitive (knows/does not know)
  - Affiliation (belongs to, is a member of)

- **Physical**
  - Road
  - Flight path
  - Wire / Wireless

- **Regulatory** (as in gene expression)
Link properties

- Directed (e.g., likes)
  - Mutual
  - Asymmetric
  - Null
  - A directed graph is also called a di-graph
  - A directed edge is also called an arc

- Undirected (e.g., talks with)

- Binary (0,1 on or off only)

- Signed and/or Valued (... -2, -1, 0, 1, 2 ...)

Nodes are now classified as senders and/or receivers.
Any collection of nodes and links can be defined as a configuration.
Types of networks

- Simplest form: 1-mode, undirected, binary ties, single relation

- 2-mode (aka Bipartite)
  - Two different types of nodes
  - Ties only allowed between groups
    
    Examples: Online network groups and persons (an Affiliation network)
    Heterosexual sex network

- Multiplex
  - More than one type of link possible
    
    Example: Neighbor and running partner
Representing network data

- **Sociomatrix**
  - aka adjacency matrix
  - simple but inefficient for large sparse networks (order $n^2$)

- **Edgelist**

- **Graph**
Intro to statnetWeb

**statnetWeb** is a GUI for the statnet suite of R packages

- Runs in a web browser
- Or in a pop-up window from R
Getting started: R and Rstudio

- Install R
  https://cran.r-project.org/

  R is the software that will run the statnetWeb package

- Install Rstudio
  https://www.rstudio.com/products/rstudio/download/

  Rstudio is an interface that makes it easier to work with R
Getting started: Installing statnetWeb

- First: Launch Rstudio

We’ll type commands here
Getting started: Installing statnetWeb

Then type the following commands at the > prompt

- `install.packages("devtools")`
- `devtools::install_github('statnet/statnetWeb')`
- `library(statnetWeb)`
- `run_sw()`
You should get a pop-up window
Data: for loading your network data

- Can upload your own
- Or use one of the built-in datasets (we’ll do this)
  - Load the faux.mesa.high network
Descriptive Statistics

- This tab in statnetWeb can be used to explore your data

![Network Plot]

**Tables**

<table>
<thead>
<tr>
<th>Network Plot</th>
<th>Attributes</th>
<th>Degree Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] Grade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 8 9 10 11 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62 40 42 25 24 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1] Race</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Black Hisp NatAm Other White</td>
<td>6 109 68 4 18</td>
<td></td>
</tr>
<tr>
<td>[1] Sex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99 106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1] Missing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Levels of measurement in networks

As we look at ways of describing network data, keep in mind the different levels of measurement

- **Node level**: attributes of each individual node
  - Examples: age, sex, infection state, degree

- **Dyad level**: attributes of pairs or edges
  - Examples: type of relationship, duration

- **Component level**: subgraph attributes and distributions
  - Examples: size, density, degree and geodesic distributions ...

- **Network level**: overall structural attributes and distributions
  - Examples: density, degree, geodesic distribution ...
Nodal Attributes

- Nodes can have attributes like age, race, sex, etc.

- Explore:
  - Click on a node in the network plot to see its name and attributes
  - Double click to highlight a node’s neighbors
  - Color-code or size nodes with menu options on the right
  - Sort or search attributes in the interactive table
  - What can you say about the structure of the network after editing the plot?
Measuring degree

- **Node level:** The number of edges adjacent to a node
  - Every node has a degree: \( \text{deg}(i) \)
  - For di-graphs, in- and out-degrees: \( \text{ideg}(i) \) and \( \text{odeg}(i) \)
    - Indegree: the number of arcs that terminate at \( n_i \)
    - Outdegree: the number of arcs that originate from \( n_i \)

- **Network level:** The degree distribution
  - Well-known parametric degree distributions
    - Uniform, Binomial, Poisson, Power-law
  - An empirical degree distribution may or may not resemble any of these
Degree distribution

- The degree distribution is a basic structural property
- To view it in statnetWeb:
Connectivity measures: Geodesic

- Nodes are **reachable** if there is a path between them.

- A **geodesic** is the shortest path between two nodes
  - Two nodes have an infinite geodesic distance if they are unreachable
Geodesic distribution

- The geodesic distribution is another basic structural property of a network.
- To view it in statnetWeb:

The last bar represents the node pairs with infinite geodesic distance.
Description vs. Inference

- So far we have been using descriptive statistics to explore our data

- Next, we might want to compare these statistics to what we would expect by chance
  - What do we mean “by chance”?
  - Can we use inferential statistics to draw more general conclusions?

- Need to define a reference distribution to act as the “null model”

- Two common null model distributions
  - Conditional Uniform Graph (CUG): Same density as the observed net
  - Bernoulli Random Graph (BRG): Same tie probability as the observed net
Suppose kids have a tendency to become friends with their friends’ friends
   - And this is the only generative process occurring.

Presumably, this would mean that you would observe more triangles than expected by chance in the graph.
   - How would you test this in an empirical dataset?
A basic statistical test approach

- Begin by counting the # triangles in your network
  - Say this is “T”, your test statistic

- Then determine the probability of observing T or more triangles in this network ...

- And see if it is less than 5%

But how would you determine that probability?
What is this probability distribution?

Start with a network this size (size = # nodes)

- Enumerate all possible networks for a fixed number of nodes,
- Count the number of triangles in each network, and
- Construct the frequency distribution of the counts.

This is a “permutation test”

for 4 nodes: # of dyads is $4\times3/2 = 6$
# of possible networks = $2^6 = 64$

for 10 nodes: # of dyads is $10\times9/2 = 45$
# of possible networks = $2^{45} \approx 35$ trillion

for 20 nodes: # of dyads is $20\times19/2 = 190$
# of possible networks = $2^{190} \approx 10^{57}$
A conditional null probability distribution

Condition on \textit{both the size and density} of the network

This is the Conditional Uniform Graph test (CUG)

- Enumerate all possible networks for a fixed number of nodes and links,
- Count the number of triangles in each network, and
- Construct the frequency distribution of the counts.

Better (in terms of reducing the sample space)

but still a lot of graphs...

we use a sample of about 50 of these for our CUG tests
Another conditional null probability distribution

Condition on the probability of a link
The Bernoulli Random Graph model (BRG)

- Simulate networks by randomly selecting a dyad, and using a coin flip to update the tie status
- Count the number of triangles after each 1000 updates
- Construct the frequency distribution of the counts.

This is a stochastic approach, in comparison to the CUG.
- It doesn’t enumerate the whole space, it just randomly wanders around it

we use a sample of 50 for our BRG tests
Example of a BRG simulation

faux.mesa.high network

Simple random graph with the same tie prob
Using null models

- Select a summary measure for the observed data
- Compare it to the distribution simulated from a null model

In statnetWeb:
- We can conduct CUG and BRG tests for network summary measures
- We can plot overlays on degree and geodesic distributions
Conditional uniform graph tests

Compare the number of isolates in faux.mesa.high to what we would expect by chance
In the degree distribution, add overlays for each null model

Mean and 95% confidence intervals from 50 random draws are plotted
Let’s look at triangles

- Are there more triangles than expected in faux.mesa.high?

- In statnetWeb, go to the Conditional Uniform Graph Tests

  - Choose the triangle term and run 100 simulations to see how our network compares to random graphs
    - “CUG” and “BRG” versions
Limitations of these simple tests

- Why are there so many more triangles?
- What do you see when color-coding the nodes by their attributes?
Friend of a friend, or birds of a feather?

(At least) two theories about the process that generates triangles:

1. **Homophily**: People tend to chose friends who are like them, in grade, race, etc. (“birds of a feather”), triad closure is a by-product.

2. **Transitivity**: People who have friends in common tend to become friends (“friend of a friend”), closure is the key process.

So, for three actors in the same grade,

A cycle-closing tie may form because of transitivity but also because of homophily.
Transitivity and homophily are confounded

But not completely. Any tie may be classified by whether it is:

<table>
<thead>
<tr>
<th>Triangle forming:</th>
<th>Within Grade:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>No</td>
<td>Transitivity</td>
</tr>
</tbody>
</table>

The cells show which processes can influence that tie.

This suggests we should be able to disentangle the two processes statistically, by looking at the relative frequency in each cell.
We want to model the probability of a tie as a function of:

- Nodal attributes (that influence degree and mixing)
- The propensity for certain “configurations” (like triangles)

The tie-status of dyads may be dependent

- Nodal attribute effects do not induce dyad dependence (homophily)
- But triad closure effects do

So we model the joint distribution directly
Exponential Random Graph Model (ERGM)

Probability of observing a graph (set of relationships) \( y \) on a fixed set of nodes:

\[
P(Y = y \mid \theta) = \frac{\exp(\theta' g(y))}{k(\theta)}
\]

where:
- \( g(y) \) = vector of network statistics
- \( \theta \) = vector of model parameters
- \( k(\theta) \) = numerator summed over all possible networks on node set \( y \)

- Exponential family model
- Very general and flexible
Probability of observing a graph (set of relationships) $y$ on a fixed set of nodes:

$$P(Y = y | \theta) = \frac{\exp(\theta' g(y))}{k(\theta)}$$

If you’re not familiar with this kind of compact vector notation, the numerator is just:

$$\exp(\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_3 x_3)$$

Kind of like a linear model, but a bit different (watch out for this later)
The conditional probability of a tie

\[
P(Y = y | \theta) = \frac{\exp(\theta' g(y))}{k(\theta)}
\]

can be re-expressed as

\[
\text{logit}(P(Y_{ij} = 1 \mid \text{rest of the graph})) = \log \left( \frac{P(Y_{ij} = 1 \mid \text{rest of the graph})}{P(Y_{ij} = 0 \mid \text{rest of the graph})} \right)
\]

\[
= \theta' \partial(g(y))
\]

The “change statistic”

the change in \( g(y) \) when \( Y_{ij} \)

is toggled from 0 to 1
ERGM specification: $\theta'g(y)$

The $g(y)$ terms in the model represent “network statistics”

- These are counts of network configurations, for example:
  1. Edges: $\sum y_{ij}$
  2. Within-group ties: $\sum y_{ij}I(i \in C, j \in C)$
  3. 2-stars: $\sum y_{ij}y_{ik}$
  4. 3-cycles: $\sum y_{ij}y_{ik}y_{jk}$

- A key distinction in the types of terms:
  - Dyad independent (1 & 2 are examples)
  - Dyad dependent (3 & 4 are examples)

statnetWeb Workshop (Sunbelt 2018)
ERGM specification: $\theta' g(y)$

Model specification involves:

1. Choosing the set of network statistics $g(y)$
   - From minimal: # of edges
   - To saturated: one term for every dyad in the network
   - statnetWeb allows you to choose from the list of terms and retrieve documentation for each one

2. Choosing “homogeneity constraints” on $\theta$. For example, with edges:
   - all homogeneous
   - group specific (e.g., sex or age specific)
   - dyad specific
Ok, back to statnetWeb
Flomarriage: Bernoulli Model

- Load the flomarriage network
  - Network of marriage ties between families in Renaissance Florence

- On the Fit Model page, look up the documentation on the edges term
Flomarriage: Bernoulli Model

- Add edges to the `ergm` formula
- Fit the model

Assumes homogeneous edge probability (the BRG)
- Every tie is equally likely
- Not a very interesting model
Flomarriage: Bernoulli Model

- How to interpret coefficients? The log-odds of any tie existing is

\[
-1.609 \times \text{change in # ties}
\]

= \(-1.609 \times 1\)

- Corresponding probability:

\[
\frac{\exp(-1.609)}{1 + \exp(-1.609)}
\]

= 0.1667
The “triangle” term is one measure of clustering

Read the documentation for the triangle term

Fit the model edges + triangle

- Hint: you can just add the triangle term if edges is already in your formula
- Then click Fit Model

Triangle is a dyad dependent term, so the estimation algorithm changes to MCMC

Monte Carlo MLE Results:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>MCMC %</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>-1.6770</td>
<td>0.3508</td>
<td>0</td>
<td>&lt;1e-04  ***</td>
</tr>
<tr>
<td>triangle</td>
<td>0.1568</td>
<td>0.5854</td>
<td>0</td>
<td>0.789</td>
</tr>
</tbody>
</table>

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Now how to interpret results?

- Conditional log-odds of two actors having a tie:
  \((-1.68 \times \text{change in the # of ties}) + (0.16 \times \text{change in # of triangles})\)

  \textit{how many triangles can one tie change?}

- For a tie that will create zero triangles \(-1.68 + 0 = -1.68\)
- One triangle \(-1.68 + 0.16 = -1.52\)
- Two triangles \((-1.68 \times 1) + (0.16 \times 2) = -1.36\)

Note, not significant
Flomarriage: Nodal covariates

flomarriage sized by wealth

- What do you notice?
- We can test whether edge probabilities are a function of wealth
- This is a quantitative nodal attribute, so we use the ergm term “nodecov”
Flomarriage: Nodal covariates

- Reset the `ergm` formula and fit the following model:

  ![ERGM terms](image)

- There is a significant positive wealth effect on the odds of a tie.

- What does the positive coefficient mean?
  - Not that there is homophily by wealth
  - Just that wealthy nodes have more ties
  - Note that the wealth effect operates on both nodes in a dyad.
Flomarriage: Nodal covariates

- **The conditional log-odds of a tie between two actors is:**
  
  \[-2.59 \times \text{change in # ties} + 0.01 \times \text{wealth of node 1} + 0.01 \times \text{wealth of node 2}\]

  - **For a tie between two nodes with minimum wealth**
    \[-2.59 + 0.01 \times (3 + 3) = -2.53\]
  
  - **For a tie between two nodes with maximum wealth**
    \[-2.59 + 0.01 \times (146 + 146) = 0.33\]
  
  - **For a tie between nodes with maximum and minimum wealth**
    \[-2.59 + 0.01 \times (146 + 3) = -1.1\]

- **Note:** To specify homophily on wealth, you would use the ergm-term `absdiff`
Model Degeneracy

- Models with dyad dependent terms can behave differently than we expect
  - They look simple
  - But they represent effects that cascade through a network via a chain of dependence (this is the “watch out” from earlier)

- Homogeneous triangle and k-star terms turn out to be some of the worst offenders
Model Degeneracy

Definition

- When a model places almost all probability on a small number of uninteresting graphs

Most common “uninteresting” graphs:

- Complete (all links exist)
- Empty

Model degeneracy = misspecification
Model Degeneracy

- Switch back to the faux.mesa.high network
- Fit a model where the formula is edges + triangle
  - What happens?

```
Error: Number of edges in a simulated network exceeds that in the observed by a factor of more than 20. This is a strong indicator of model degeneracy or a very poor starting parameter configuration. If you are reasonably certain that neither of these is the case, increase the MCMLE.density.guard control.ergm() parameter.
```

- Trying to fit this model, the algorithm heads off into networks that are much more dense than the observed network.

- What does this mean? That this model would not have produced this network, for any combination of parameter estimates for the two terms
  - i.e., this is a model misspecification problem
Degeneracy Plot (for the 2 star model)

- Only the white area has networks with some interesting variation.
- The dark areas are complete graphs, or empty graphs (+/- 1 or 2 edges).
- This model does not produce many useful networks.

Figure 3: Cumulative Degeneracy Probabilities for graphs with 7 actors. From Mark Handcock’s 2003 tech report:
Solution: better network statistics

- **Old statistic:** # of triangles in the graph

  \[ g(y) = \sum y_{ij} y_{jk} y_{ki} \]

  *Here, every additional 3-cycle has the same impact, \( \theta \)*

- **New statistic:** GWESP

  \[ g(y) = e^\alpha \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} sp_i \]

  - **geometrically weighted edge-wise shared partners**
  - Sets declining marginal returns for each additional 3-cycle involving the same edge
  - The parameter that specifies the rate of decline in marginal returns is \( \alpha \)
  - The smaller the \( \alpha \), the more rapid the decline
Solution: better network statistics

\[ gwesp = e^\alpha \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^i\} sp_i \]

\( sp_i = \# \) of edges with \( i \) shared partners

This configuration contains:
- 1 edge with 3 shared partners
- 6 edges with 1 shared partner
- 9 edges involved in the three triangles (one of these is counted 3 times)

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>GWESP(( \alpha ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>= 7              this is the number of edges involved in at least one triangle</td>
</tr>
<tr>
<td>0.5</td>
<td>= 7.55</td>
</tr>
<tr>
<td>1</td>
<td>= 8.03           this is headed towards 9, the number of edges in the 3 triangles</td>
</tr>
</tbody>
</table>
Solution: better network statistics

\[ gwesp = e^\alpha \sum_{i=1}^{n-2} \left\{ 1 - (1 - e^{-\alpha})^i \right\} sp_i \]

\( sp_i = \# \) of edges with \( i \) shared partners

A count of each edge in each triangle (i.e. \( \# \) of triangles \( \times 3 \))

A count of edges in at least one triangle (only the first triangle counts)
Fit the model with edges and gwesp

- Try edges + gwesp(0.25, fixed = TRUE)
MCMC Diagnostics

- Go to the MCMC Diagnostics tab
  - MCMC Diagnostics tell us if the estimation algorithm is mixing well

- But is it a good fit to the observed data? We need to check the goodness-of-fit diagnostics for that
Testing goodness of fit

- Traditional GOF stats can be used
  - AIC, BIC in the model summary

- We also take another approach
  - We are interested in how well we fit aggregate properties of the network structure that we did not include as model terms
  - This helps to identify what the model gets wrong
  - We use 3 “higher order” statistics:
    - Degree distribution
    - Shared partner distribution (non-parametric) (local clustering)
    - Geodesic distance distribution (global clustering)
DATA

MODEL

ESTIMATED COEFFICIENTS

SIMULATED DATA
(draws from the prob. dist.)

HIGHER ORDER GRAPH STATISTICS OF DATA

HIGHER ORDER GRAPH STATISTICS OF SIMULATED DATA

GOODNESS OF FIT OF MODEL TO DATA

statnetWeb Workshop (Sunbelt 2018)
Goodness of Fit

- Run the default set of GOF terms for this model

```
faux.mesa.high ~ edges + gwesp(0.25, fixed = TRUE)
```
And the eyeball test...

Even though our model isn’t degenerate, we haven’t fit the structure of the observed data - specifically, the levels of clustering.

So, back to our original question:
How much of the clustering is due to homophily, and how much to transitivity?
Test this by comparing four models

<table>
<thead>
<tr>
<th>Model</th>
<th>Network Statistics g(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edges</td>
<td># of edges</td>
</tr>
<tr>
<td>Edges + Attributes (homophily)</td>
<td># of edges</td>
</tr>
<tr>
<td></td>
<td># of edges for each race, sex, grade</td>
</tr>
<tr>
<td></td>
<td># of edges that are within-race, within-grade, within-sex</td>
</tr>
<tr>
<td>Edges + GWESP (transitivity)</td>
<td># of edges</td>
</tr>
<tr>
<td></td>
<td>weighted shared partners</td>
</tr>
<tr>
<td>Edges + Attributes + GWESP (both)</td>
<td># of edges</td>
</tr>
<tr>
<td></td>
<td># of edges for each race, sex, grade</td>
</tr>
<tr>
<td></td>
<td># of edges that are within-race, within-grade, within-sex</td>
</tr>
<tr>
<td></td>
<td>weighted shared partners</td>
</tr>
</tbody>
</table>
Fitting and saving models

- **statnetWeb allows you to save up to five models at a time**

1. edges
   - Fit model, save model, reset formula
2. edges + nodefactor("Grade") + nodefactor("Race") + nodefactor("Sex") + nodematch("Grade", diff = TRUE) + nodematch("Race", diff = FALSE) + nodematch("Sex", diff = FALSE)
   - Fit model, save model, reset formula
3. edges + gwesp(0.25, fixed = TRUE)
   - Fit model, save model, reset formula
4. edges + nodefactor("Grade") + nodefactor("Race") + nodefactor("Sex") + nodematch("Grade", diff = TRUE) + nodematch("Race", diff = FALSE) + nodematch("Sex", diff = FALSE) + gwesp(0.25, fixed = TRUE)
   - Fit model, save model
# Model Comparison

<table>
<thead>
<tr>
<th></th>
<th>Model1</th>
<th>Model2</th>
<th>Model3</th>
<th>Model4</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>-4.63***</td>
<td>-8.491***</td>
<td>-5.58***</td>
<td>-9.056***</td>
</tr>
<tr>
<td>nodefactor.Grade.8</td>
<td>NA</td>
<td>1.562*</td>
<td>NA</td>
<td>1.413*</td>
</tr>
<tr>
<td>nodefactor.Grade.9</td>
<td>NA</td>
<td>2.533***</td>
<td>NA</td>
<td>2.184***</td>
</tr>
<tr>
<td>nodefactor.Grade.10</td>
<td>NA</td>
<td>2.942***</td>
<td>NA</td>
<td>2.520***</td>
</tr>
<tr>
<td>nodefactor.Grade.11</td>
<td>NA</td>
<td>2.660***</td>
<td>NA</td>
<td>2.260***</td>
</tr>
<tr>
<td>nodefactor.Grade.12</td>
<td>NA</td>
<td>3.470***</td>
<td>NA</td>
<td>2.901***</td>
</tr>
<tr>
<td>nodefactor.Race.Hisp</td>
<td>NA</td>
<td>-1.571***</td>
<td>NA</td>
<td>-1.015***</td>
</tr>
<tr>
<td>nodefactor.Race.NatAm</td>
<td>NA</td>
<td>-1.103***</td>
<td>NA</td>
<td>-0.758***</td>
</tr>
<tr>
<td>nodefactor.Race.Other</td>
<td>NA</td>
<td>-2.916**</td>
<td>NA</td>
<td>-2.013*</td>
</tr>
<tr>
<td>nodefactor.Race.White</td>
<td>NA</td>
<td>-0.809**</td>
<td>NA</td>
<td>-0.499*</td>
</tr>
<tr>
<td>nodefactor.Sex.M</td>
<td>NA</td>
<td>-0.335***</td>
<td>NA</td>
<td>-0.163*</td>
</tr>
<tr>
<td>nodematch.Grade.7</td>
<td>NA</td>
<td>7.441***</td>
<td>NA</td>
<td>5.974***</td>
</tr>
<tr>
<td>nodematch.Grade.8</td>
<td>NA</td>
<td>4.330***</td>
<td>NA</td>
<td>3.254***</td>
</tr>
<tr>
<td>nodematch.Grade.9</td>
<td>NA</td>
<td>2.060**</td>
<td>NA</td>
<td>1.645***</td>
</tr>
<tr>
<td>nodematch.Grade.10</td>
<td>NA</td>
<td>1.234*</td>
<td>NA</td>
<td>1.036.</td>
</tr>
<tr>
<td>nodematch.Grade.11</td>
<td>NA</td>
<td>2.525**</td>
<td>NA</td>
<td>1.912***</td>
</tr>
<tr>
<td>nodematch.Grade.12</td>
<td>NA</td>
<td>1.358.</td>
<td>NA</td>
<td>1.057.</td>
</tr>
<tr>
<td>nodematch.Race</td>
<td>NA</td>
<td>0.832**</td>
<td>NA</td>
<td>0.734***</td>
</tr>
<tr>
<td>nodematch.Sex</td>
<td>NA</td>
<td>0.638**</td>
<td>NA</td>
<td>0.543***</td>
</tr>
<tr>
<td>gwesp.fixed.0.25</td>
<td>NA</td>
<td>NA</td>
<td>1.86***</td>
<td>1.388***</td>
</tr>
<tr>
<td>AIC</td>
<td>2288</td>
<td>1809</td>
<td>1999</td>
<td>1648</td>
</tr>
<tr>
<td>BIC</td>
<td>2296</td>
<td>1960</td>
<td>2015</td>
<td>1807</td>
</tr>
</tbody>
</table>
Goodness of fit measure 1: degree distribution

Data:

Model: Bernoulli
(i.e. edges only)
Goodness of fit measure 2: ESP distribution
(local clustering)

Data:

Model: Bernoulli
(i.e. edges only)

This edge has an ESP value of 3
Goodness of fit measure 3: geodesic distribution (global clustering)

Data:

Model: Bernoulli (i.e. edges only)

A/B have geodesic 2
A/C have geodesic $\infty$
Goodness of fit measures assembled

Summary: Not a good fit to any of the aggregate structural properties observed
All 4 models:

Model: Edges
AIC: 2288

Model: Edges + Attributes
AIC: 1809

Model: Edges + GWESP
AIC: 1999

Model: Edges + Attributes + GWESP
AIC: 1648
Attributes + GWESP Model as network size increases

\begin{align*}
\mathbf{n}: \quad & 71 \\
\quad & 159 \\
\quad & 291 \\
\quad & 1011 \\
\end{align*}

degree \quad \text{edgewise shared partner} \quad \text{geodesic}
While there is variation across the schools, there are also clear and systematic patterns that are shared.

And the structural effects are generally stronger than the nodal attribute effects.
Findings

- Both transitivity and homophily play a role in clustering friendships
  - Homophily alone would generate the distribution of path lengths
  - A simple parametric form of transitivity captures local clustering
  - 25% of the transitivity effect is a by-product of homophily
  - Grade mixing is typically stronger than race mixing
    - but also less robust to the transitivity confound

- All 4 of these models begin to fail on the largest schools
  - There is more clustering than these models predict
  - Suggests additional sources of heterogeneity
  - Or perhaps an endogeneous “fissioning” of groups
Simulations

Choose one of the models that you have saved and run 100 simulations with the default control settings

- Choose the model on the Simulations page next to “ergm formula”
- Do you see autocorrelation in the simulation statistics?

Increase the MCMC interval to 10,000 and re-run the simulations to see how this changes
Common statistics in ergms

An undirected network of 10 nodes, including nodal attribute “color”, with values 1=black, 2=red, 3=green.

<table>
<thead>
<tr>
<th>Term</th>
<th>Formula</th>
<th>Unit</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edges</td>
<td># of edges</td>
<td>edges</td>
<td></td>
</tr>
<tr>
<td>degree(0)</td>
<td># of nodes of degree 0</td>
<td>nodes</td>
<td></td>
</tr>
<tr>
<td>degree(2:5)</td>
<td># of nodes of degrees 2, 3, 4, 5 each</td>
<td>nodes</td>
<td></td>
</tr>
<tr>
<td>concurrent</td>
<td># of nodes of at least degree 2</td>
<td>nodes</td>
<td></td>
</tr>
</tbody>
</table>
Common statistics in ergms

undirected network of 10 nodes, including nodal attribute “col”, with values 1=black, 2=red, 3=green

<table>
<thead>
<tr>
<th>Term</th>
<th>Formula</th>
<th>Unit</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodemix(“col”, base=1)</td>
<td># of edges between nodes of each color combo</td>
<td>edges</td>
<td></td>
</tr>
<tr>
<td>nodefactor(“col”)</td>
<td>Sum of degrees for nodes of each color</td>
<td>nodes/edges*</td>
<td></td>
</tr>
<tr>
<td>nodefactor(“col”, base=2)</td>
<td>Sum of degrees for nodes of each color uses group 2 (red) as the omitted baseline category</td>
<td>nodes/edges*</td>
<td></td>
</tr>
<tr>
<td>nodematch(“col”)</td>
<td># of edges between nodes of same color</td>
<td>edges</td>
<td></td>
</tr>
<tr>
<td>nodematch(“col”, diff=T)</td>
<td># of edges between nodes of same color, for each color</td>
<td>edges</td>
<td></td>
</tr>
</tbody>
</table>
Common statistics in ergms

An undirected network of 10 nodes, including nodal attribute “color”, with values 1=black, 2=red, 3=green.

<table>
<thead>
<tr>
<th>Term</th>
<th>Formula</th>
<th>Unit</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>~triangle</td>
<td># of triangles (beware!)</td>
<td>triangles</td>
<td></td>
</tr>
<tr>
<td>~gwesp(0)</td>
<td># of edges in at least one triangle</td>
<td>edges</td>
<td></td>
</tr>
<tr>
<td>~gwesp(∞)</td>
<td># of edges in triangles total (=3 * # triangles)</td>
<td>triangles</td>
<td></td>
</tr>
</tbody>
</table>
Selected References


*Journal of Statistical Software* (v42) 2008 – Eight papers on ERGMs and statnet