Exponential Random Graph Models for Multimodal Data

Lorien Jasny[1] and Örjan Bodin

[1]Q-Step Centre, Exeter University
l.jasny@exeter.ac.uk

Exponential Random Graph Models!

Baseline Models

Baseline models only take us so far

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given "rejection" of the null hypothesis, no clear path forward

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given "rejection" of the null hypothesis, no clear path forward

Solution: Parametric models

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given “rejection” of the null hypothesis, no clear path forward

Solution: Parametric models

- Identify candidate structural mechanisms

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given "rejection" of the null hypothesis, no clear path forward

Solution: Parametric models

- Identify candidate structural mechanisms
- Parameterize using graph statistics

Baseline Models

Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given "rejection" of the null hypothesis, no clear path forward

Solution: Parametric models

- Identify candidate structural mechanisms
- Parameterize using graph statistics
- Fit models to data
    - Compare alternatives
    - Interpret parameter estimates
    - Assess adequacy

Baseline Models

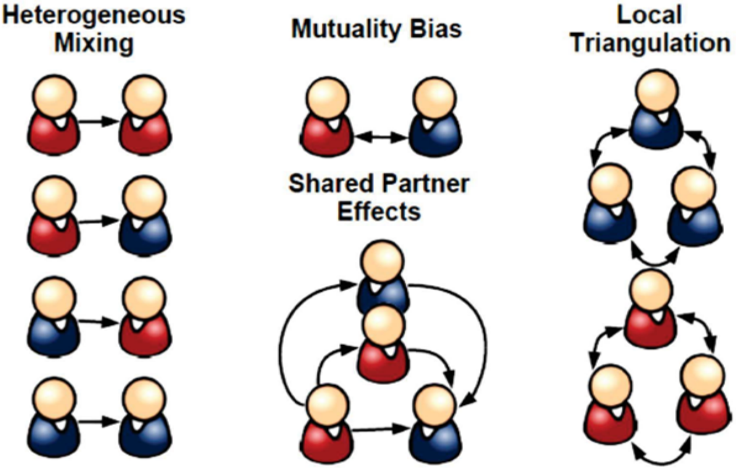
Baseline models only take us so far

- Few statistics lend themselves to conditioning
- Difficult to look at multiple biases at once
- Hard to account for sampling, error, etc.
- given "rejection" of the null hypothesis, no clear path forward

Solution: Parametric models

- Identify candidate structural mechanisms
- Parameterize using graph statistics
- Fit models to data
  - Compare alternatives
  - Interpret parameter estimates
  - Assess adequacy
- Can apply/extend for prediction, etc.

Sample Mechanisms



Evaluating Competing Explanations

Edges	Mixing	Mutuals	GWESP	LocalTri	AIC	Rank
1	0	0	0	0	1777.684	15
1	1	0	0	0	1565.073	14
1	0	1	0	0	1516.578	13
1	0	0	1	0	1227.656	2
1	0	0	0	1	1478.532	12
1	1	1	0	0	1428.158	11
1	1	0	1	0	1279.456	6
1	1	0	0	1	1416.441	10
1	0	1	1	0	1234.932	3
1	0	1	0	1	1348.794	9
1	0	0	1	1	1290.241	7
1	1	1	1	0	1216.762	1
1	1	1	0	1	1339.640	8
1	1	0	1	1	1238.285	5
1	0	1	1	1	1236.924	4

Logistic Network Regression

- A classic starting point:

Logistic Network Regression

- A classic starting point:
  - why not treat edges as independent, with log-odds as a linear function of covariates?

Logistic Network Regression

- A classic starting point:
  - why not treat edges as independent, with log-odds as a linear function of covariates?
  - Special case of standard logistic regression

Logistic Network Regression

- A classic starting point:
  - why not treat edges as independent, with log-odds as a linear function of covariates?
  - Special case of standard logistic regression
  - Dependent variable is a network adjacency matrix
- Model form: $log(\frac{P(Y_{ij}=1)}{P(Y_{ij}=0)} = \theta_1 X_{ij1} + \theta_2 X_{ij2} + ... + \theta_m X_{ijm} = \theta^T X_{ij})$
- Where Y_{ij} is the value of the edge from i to j on the dependent relation, X_{ijk} is the value of the kth predictor for the (i, j) ordered pair, and $\theta_1 ... \theta_m$ are coefficients

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)
  - Cannot handle exotic support constraints

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)
  - Cannot handle exotic support constraints
- A more general framework: discrete exponential families

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)
  - Cannot handle exotic support constraints
- A more general framework: discrete exponential families
  - Very general way of representing discrete distributions

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)
  - Cannot handle exotic support constraints
- A more general framework: discrete exponential families
  - Very general way of representing discrete distributions
  - Turns up frequently in statistics, physics, etc.

Beyond the Logistic Case

- The logistic model can be quite powerful, but still very limiting
  - No way to model conditional dependence among edges (clustering, reciprocity)
  - Cannot handle exotic support constraints
- A more general framework: discrete exponential families
  - Very general way of representing discrete distributions
  - Turns up frequently in statistics, physics, etc.
  - ERGM is more like a language of models than a specific book

Exponential Random Graph Models

$$P(Y = y|t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

Exponential Random Graph Models

$$P(Y = y|t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

$\uparrow$

Probability that a random graph drawn from Y is the realized graph y

Exponential Random Graph Models

Given sufficient statistics t, the parameters θ, the countable support Y, and the covariates X

$$P(Y = y|t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

Probability that a random graph drawn from Y is the realized graph y

Exponential Random Graph Models

Given sufficient statistics t, the parameters θ, the countable support Y, and the covariates X

The empirical realization of covariates, statistics, and parameters

$$P(Y = y|t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

Probability that a random graph drawn from Y is the realized graph y

Exponential Random Graph Models

Given sufficient statistics t, the parameters θ, the countable support Y, and the covariates X

The empirical realization of covariates, statistics, and parameters

$$P(Y = y|t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

Probability that a random graph drawn from Y is the realized graph y

Normalizing factor counting over every other graph in the support

Exponential Random Graph Models

Given sufficient statistics t, the parameters θ, the countable support Y, and the covariates X

The empirical realization of covariates, statistics, and parameters

$$P(Y = y | t, \theta, Y, X) = \frac{exp(\theta^T t(y,X))}{\sum y' \in Y exp(\theta^T t(y',X))} I_Y(y)$$

Probability that a random graph drawn from Y is the realized graph y

Normalizing factor counting over every other graph in the support

An indicator that Y is in the support

Conditional Log-Odds of an Edge

$$\frac{P(Y = y_{ij}^{+} | t, \theta, Y, X)}{P(Y = y_{ij}^{-} | t, \theta, Y, X)}$$

Conditional Log-Odds of an Edge

$$\frac{P(Y{=}y_{ij}^{+}|t,\theta,Y,X)}{P(Y{=}y_{ij}^{-}|t,\theta,Y,X)} = \frac{exp(\theta^T t(y_{ij}^{+},X))}{\sum y' \in Y exp(\theta^T t(y',X))} * \frac{\sum y' \in Y exp(\theta^T t(y',X))}{exp(\theta^T t(y_{ij}^{-},X))}$$

Conditional Log-Odds of an Edge

$$\frac{P(Y = y_{ij}^{+} | t, \theta, Y, X)}{P(Y = y_{ij}^{-} | t, \theta, Y, X)} = \frac{exp(\theta^T t(y_{ij}^{+}, X))}{\sum y' \in Y exp(\theta^T t(y', X))} * \frac{\sum y' \in Y exp(\theta^T t(y', X))}{exp(\theta^T t(y_{ij}^{-}, X))}$$

$$\frac{exp(\theta^T t(y_{ij}^{+}, X))}{exp(\theta^T t(y_{ij}^{-}, X))} = exp(\theta^T (t(y_{ij}^{+}, X) - t(y_{ij}^{-}, X)))$$

Conditional Log-Odds of an Edge

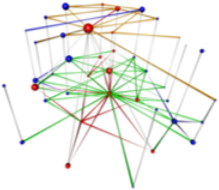
$$\frac{P(Y=y_{ij}^{+}|t,\theta,Y,X)}{P(Y=y_{ij}^{-}|t,\theta,Y,X)} = \frac{exp(\theta^{T}t(y_{ij}^{+},X))}{\sum y'\in Y exp(\theta^{T}t(y',X))} * \frac{\sum y'\in Y exp(\theta^{T}t(y',X))}{exp(\theta^{T}t(y_{ij}^{-},X))}$$

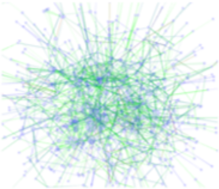
$$\frac{exp(\theta^{T}t(y_{ij}^{+},X))}{exp(\theta^{T}t(y_{ij}^{-},X))} = exp(\theta^{T}(t(y_{ij}^{+},X) - t(y_{ij}^{-},X)))$$

= P  |the rest of the graph / P  |the rest of the graph $= exp(\theta^{T} * \Delta\text{change score})$

ERG Fitting using MPNet

melnet.org.au/pnet





MPNET FOR MULTILEVEL NETWORKS

In addition to most of functions implemented under PNet, MPNet is also designed for:

- ERGMs for two-mode and two-level networks
- Autologistic Actor Attribute models (ALAAMs)

DOWNLOAD MPNET (32-BIT)

PNET FOR ONE-MODE NETWORKS

PNet is for the simulation and estimation of ERGMs for one-mode networks.

DOWNLOAD PNET GUI (32-BIT)

DOWNLOAD PNET DLL (32-BIT)

DOWNLOAD PNET GUI (64-BIT)

DOWNLOAD PNET DLL (64-BIT)

XPNET FOR BIVARIATE ANALYSIS

PNet is for the simulation and estimation of ERGMs for two one-mode networks.

DOWNLOAD XPNET GUI (32-BIT)

DOWNLOAD XPNET DLL (32-BIT)

DOWNLOAD XPNET GUI (64-BIT)

DOWNLOAD XPNET DLL (64-BIT)

ERG Fitting using `ergm`

- Dedicated statnet package for fitting, simulating models in ERG form

ERG Fitting using `ergm`

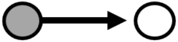
- Dedicated statnet package for fitting, simulating models in ERG form
- Basic call structure: `ergm(y~term1(arg)+term2(arg))`
  - y is a network
  - term1, term2, etc are the "sufficient statistics", or terms written in the `ergm` package
  - see "ergm-terms"

ERG Fitting using ergm

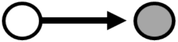
- Dedicated statnet package for fitting, simulating models in ERG form
- Basic call structure: `ergm(y~term1(arg)+term2(arg))`
    - y is a network
    - term1, term2, etc are the "sufficient statistics", or terms written in the `ergm` package
    - see "ergm-terms"
- Output: ergm object
    - Summary, print and other methods can be used to examine it
    - Simulate command can also be used to take draws from the fitted model

Dyadic independent terms

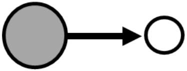
Edge – the baseline probability of a tie



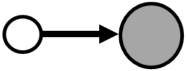
Outdegree (Sender) for an attribute



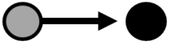
Indegree (Receiver) for an attribute



Outdegree (Sender) for a valued parameter

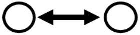


Outdegree (Sender) for a valued parameter

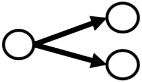


Mixing terms

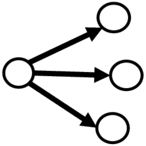
Dyadic dependent terms



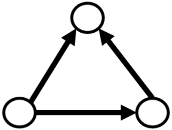
Reciprocity



Out 2-star (popularity)

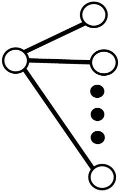


Out 3-star (more popularity)

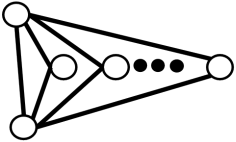


Transitive Triad

Higher Order Terms



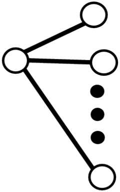
Geometrically Weighted Stars (altkstar or gwdegree)

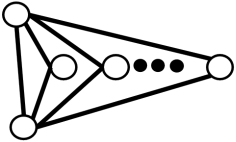


Geometrically Weighted Edgewise Shared Partners (gwesp)

- Diminishing returns makes sense (every three-star has 3 two-stars)

Higher Order Terms





- Diminishing returns makes sense (every three-star has 3 two-stars)
- Makes fitting the MCMC much easier - we'll see why next...

Interpreting Coefficients

```
Formula:   samplk3 ~ edges + mutual

Iterations:  2 out of 20

Monte Carlo MLE Results:
       Estimate Std. Error MCMC % p-value
edges   -2.1505     0.2181      0  <1e-04 ***
mutual   2.2879     0.4782      0  <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 424.2  on 306  degrees of freedom
 Residual Deviance: 267.9  on 304  degrees of freedom

AIC: 271.9    BIC: 279.3    (Smaller is better.)
```

- The log-odds of an unreciprocated edge is -2.15

Interpreting Coefficients

```
Formula:   samplk3 ~ edges + mutual

Iterations:  2 out of 20

Monte Carlo MLE Results:
       Estimate Std. Error MCMC % p-value
edges   -2.1505     0.2181      0  <1e-04 ***
mutual   2.2879     0.4782      0  <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 424.2  on 306  degrees of freedom
 Residual Deviance: 267.9  on 304  degrees of freedom

AIC: 271.9    BIC: 279.3    (Smaller is better.)
```

- The log-odds of an unreciprocated edge is -2.15
- The probability of an unreciprocated edge is $\frac{exp(-2.15)}{1+exp(-2.15)} = 0.10$

Interpreting Coefficients

```
Formula:   samplk3 ~ edges + mutual

Iterations:  2 out of 20

Monte Carlo MLE Results:
       Estimate Std. Error MCMC % p-value
edges   -2.1505     0.2181      0  <1e-04 ***
mutual   2.2879     0.4782      0  <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 424.2  on 306  degrees of freedom
 Residual Deviance: 267.9  on 304  degrees of freedom

AIC: 271.9    BIC: 279.3    (Smaller is better.)
```

- The log-odds of an reciprocated edge is -2.15+2.29=.14

Interpreting Coefficients

```
Formula:   samplk3 ~ edges + mutual

Iterations:  2 out of 20

Monte Carlo MLE Results:
       Estimate Std. Error MCMC % p-value
edges   -2.1505     0.2181      0  <1e-04 ***
mutual   2.2879     0.4782      0  <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 424.2  on 306  degrees of freedom
 Residual Deviance: 267.9  on 304  degrees of freedom

AIC: 271.9    BIC: 279.3    (Smaller is better.)
```

- The log-odds of an reciprocated edge is -2.15+2.29=.14
- The probability of an reciprocated edge is $\frac{exp(.14)}{1+exp(.14)} = 0.53$

Model Fit and Model Assessment

- We've seen how to construct and fit nontrivial ERGs
  - Started with dyadic independent terms
  - Added basic dependence terms
  - Fit the whole thing via MLE
- Now we turn to degeneracy checking and model assessment
  - Looking under the hood to make sure that the engine is still running - and occasionally, getting out to turn the crank
  - Checking the results to make sure that the model makes sense

The role of Simulation in ERG Research

- Simulation is central to ERG modeling
  - Even simple models too complex to get analytical solutions - need to use simulation to study model behaviour, make predictions
  - ERG computations too difficult to perform directly (that support term in the denominator) - simulation used purely for computational purposes
- Implication: we need to know something about ERG simulation to use tools effectively

Mc, MC, and MCMC in one slide

- Markov chain
    - Stochastic process such that $P(X_i|X_{i-1}, X_{i-2}, ...) = P(X_i|X_{i-1})$

Mc, MC, and MCMC in one slide

- Markov chain
  - Stochastic process such that $P(X_i|X_{i-1}, X_{i-2}, ...) = P(X_i|X_{i-1})$
  - meaning that only the previous state (i-1) matters in predicting the current state (i)

Mc, MC, and MCMC in one slide

- Markov chain
  - Stochastic process such that $P(X_i|X_{i-1}, X_{i-2}, ...) = P(X_i|X_{i-1})$
  - meaning that only the previous state (i-1) matters in predicting the current state (i)
- Monte Carlo procedure
  - Any procedure which uses randomization to perform computation, having a fixed execution time and uncertain output

Mc, MC, and MCMC in one slide

- Markov chain
  - Stochastic process such that $P(X_i|X_{i-1}, X_{i-2}, ...) = P(X_i|X_{i-1})$
  - meaning that only the previous state (i-1) matters in predicting the current state (i)
- Monte Carlo procedure
  - Any procedure which uses randomization to perform computation, having a fixed execution time and uncertain output
- Markov chain Monte Carlo (MCMC)
  - Family of procedures using Markov chains to perform computations and/or simulate target distributions

ERG MCMC

- When we need to simulate ERGs, we turn to MCMC
  - Every ‘step’ in the Markov chain is changing one edge from on (1) to off (0) or vice versa
  - Then, the probability of the next step given the current state of the chain is the change score we saw before
  - General procedure: start with a ‘seed’ graph (random or data)
    - Early “burn-in” draws contaminated by an initial state - discard
    - need to ensure that sample is large enough to have good properties
    - both aspects sloppily called “convergence” – the chain has “converged” when approximation is adequate
  - mostly automated, but important to use diagnostics to verify behavior

What happens when you run `ergm`

What happens when you run `ergm`

- Little gnomes make an initial guess at θ using the MPLE

What happens when you run `ergm`

- Little gnomes make an initial guess at θ using the MPLE
- More gnomes simulate $y_1, ...y_n$ based on initial guess

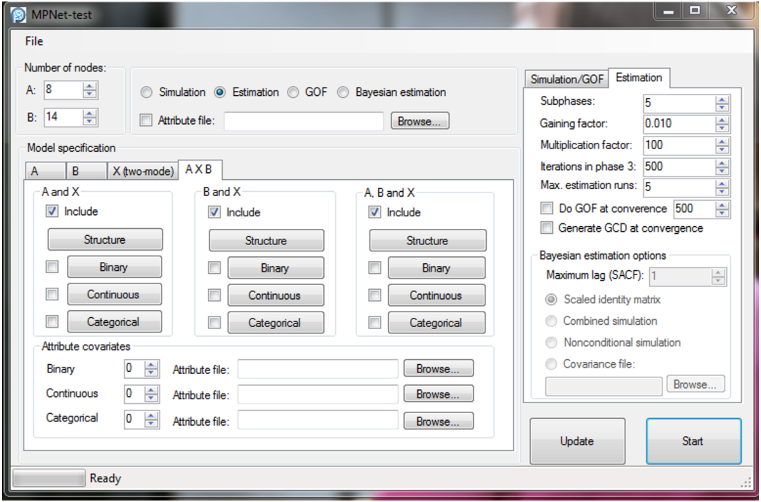
What happens when you run `ergm`

- Little gnomes make an initial guess at θ using the MPLE
- More gnomes simulate $y_1, ...y_n$ based on initial guess
- This simulated sample is used to find θ using MLE

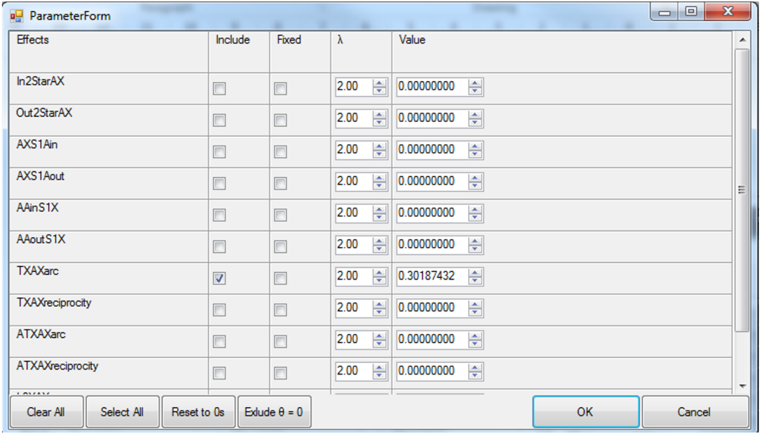
What happens when you run `ergm`

- Little gnomes make an initial guess at θ using the MPLE
- More gnomes simulate $y_1, ...y_n$ based on initial guess
- This simulated sample is used to find θ using MLE
- Possibly, the previous two steps are iterated a few times for good measure (since initial estimate may be off)

MPNet


MPNet-test
File
Number of nodes:
A: 8
B: 14
Simulation
Estimation
GOF
Bayesian estimation
Attribute file:
Browse...
Model specification
A
B
X (two-mode)
A X B
A and X
Include
Structure
Binary
Continuous
Categorical
B and X
Include
Structure
Binary
Continuous
Categorical
A, B and X
Include
Structure
Binary
Continuous
Categorical
Attribute covariates
Binary 0 Attribute file: Browse...
Continuous 0 Attribute file: Browse...
Categorical 0 Attribute file: Browse...
Simulation/GOF
Estimation
Subphases: 5
Gaining factor: 0.010
Multiplication factor: 100
Iterations in phase 3: 500
Max. estimation runs: 5
Do GOF at converence 500
Generate GCD at convergence
Bayesian estimation options
Maximum lag (SACF): 1
Scaled identity matrix
Combined simulation
Nonconditional simulation
Covariance file:
Browse...
Update
Start
Ready

MPNet

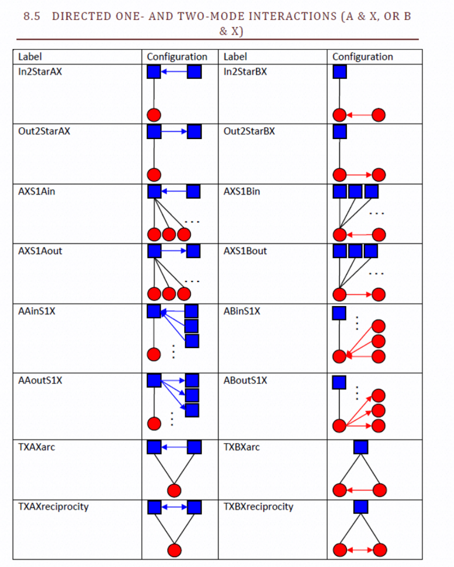


ParameterForm

Effects	Include	Fixed	λ	Value
In2StarAX	☐	☐	2.00	0.00000000
Out2StarAX	☐	☐	2.00	0.00000000
AXS1Ain	☐	☐	2.00	0.00000000
AXS1Aout	☐	☐	2.00	0.00000000
AAinS1X	☐	☐	2.00	0.00000000
AAoutS1X	☐	☐	2.00	0.00000000
TXAXarc	☑	☐	2.00	0.30187432
TXAXreciprocity	☐	☐	2.00	0.00000000
ATXAXarc	☐	☐	2.00	0.00000000
ATXAXreciprocity	☐	☐	2.00	0.00000000

Clear All | Select All | Reset to 0s | Exlude θ = 0 | OK | Cancel

MPNet



8.5 DIRECTED ONE- AND TWO-MODE INTERACTIONS (A & X, OR B & X)

Label	Configuration	Label	Configuration
In2StarAX		In2StarBX	
Out2StarAX		Out2StarBX	
AXS1Ain		AXS1Bin	
AXS1Aout		AXS1Bout	
AAinS1X		ABinS1X	
AAoutS1X		ABoutS1X	
TXAXarc		TXBXarc	
TXAXreciprocity		TXBXreciprocity	

MPNet

test_est.txt - Notepad

File Edit Format View Help

```
Estimation
Observed graph statistics:
7.00    0.00    17.00   0.00    31.00   14.00   20.00
Phase1 simulation:
Mean statistics:
6.18    0.36    17.00   0.75    30.07   6.61    8.89
Phase2 estimation
Subphase 1 started with a = 0.01000000
Parameters after subphase 1:
-2.27184577     -1.77998679     -2.48459229     -1.18421208     -1.43884069     0.32161640      0.35314236
Subphase 2 started with a = 0.01000000
Parameters after subphase 2:
-2.28114267     -2.32135390     -2.48696203     -1.52643670     -1.46997728     0.32843981      0.35907221
Subphase 3 started with a = 0.00500000
Parameters after subphase 3:
-2.23908720     -2.69578457     -2.52115983     -1.65594540     -1.48169729     0.32303999      0.38200674
Subphase 4 started with a = 0.00250000
Parameters after subphase 4:
-2.21692182     -2.76169643     -2.49492044     -1.81960339     -1.46099548     0.31118205      0.35855059
Subphase 5 started with a = 0.00125000
Parameters after subphase 5:
-2.22078614     -2.86965183     -2.46927732     -1.93482677     -1.44855507     0.30187432      0.34677568
Phase3 simulation
Number of steps: 500
nNumber of iteration in steps: 830
Mean statistics:
6.76200000      0.03400000      16.22400000     0.12200000      29.03800000     8.57400000      12.63800000
Estimation results
NOTE: t-statistics = (observation - sample mean)/standard error
NOTE: SACF (sample autocorrelation)
Effects Lambda  Parameter       Stderr  t-ratio SACF
ArcA    2.00000000      -2.22078614     0.61828668      0.09943151      0.13383289      *
ReciprocityA    2.00000000      -2.86965183     5.59585862      -0.18760780     -0.03533776
ArcB    2.00000000      -2.46927732     0.35332583      0.20265985      0.16422591
ReciprocityB    2.00000000      -1.93482677     2.87976338      -0.34769831     0.02511791
XEdge   2.00000000      -1.44855507     0.35015265      0.33729191      0.08125887
TXAXarc 2.00000000      0.30187432      0.36867397      0.94511082      0.12844882
TXBXarc 2.00000000      0.34677568      0.28248019      0.96031660      0.12234763
```

Data Prep

- A, B and X networks as separate .txt files
- Attribute files separated for Binary, Continuous, and Categorical attributes

A Puzzle

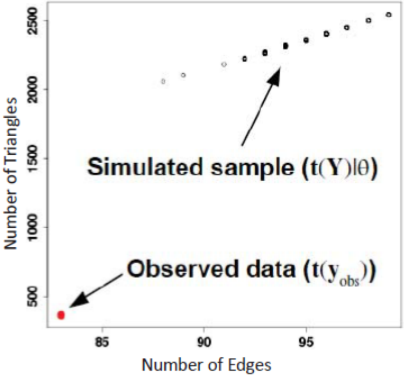
Lots of interest early on in a very (at first glance) simple model:

```
ergm(net~edges+triangle)
```

But some puzzling results when we simulated from the model

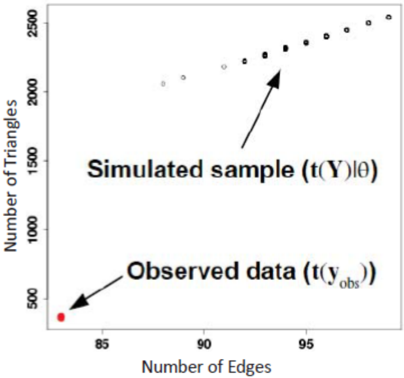
A Puzzle

- The simulated networks look nothing like the observed data
- Even when the correct coefficients are not simulated (was done on an example with 7 nodes) the networks simulated from that model show the same result (Ke Li, 2015)



Model Degeneracy

- Almost all the graphs are the same (usually complete/empty)
- The probability of a given statistic pushes the MCMC to always/never add edges



More Broadly

- Simulation can fail in several (essentially four) ways
  - Insufficient burn-in - starting point still affects results
  - Insufficient post-burn samples - sample hasn't converged
  - degeneracy
  - Sample does not cover observed graph - you couldn't generate your given graph from any combination of sufficient statistics

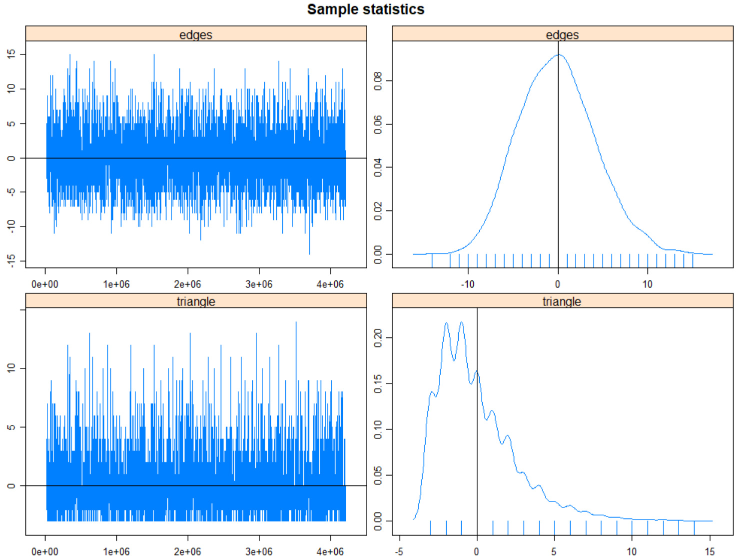
Assessing Simulation Quality

- No foolproof method, but several heuristics
- in `ergm`, primary tool is `mcmc.diagnostics`
- calculates various diagnostics on MCMC output
- Can also directly plot statistics (from the MCMC) vs observed values

What if things go wrong?

- Different MCMC controls are set using the sequence `control=control.ergm(`*terms*`)`
- For burn-in issues, increase MCMC.burnin parameter
- For post-burn convergence, increase MCMC.samplesize
- If none of these work, may need to change the model

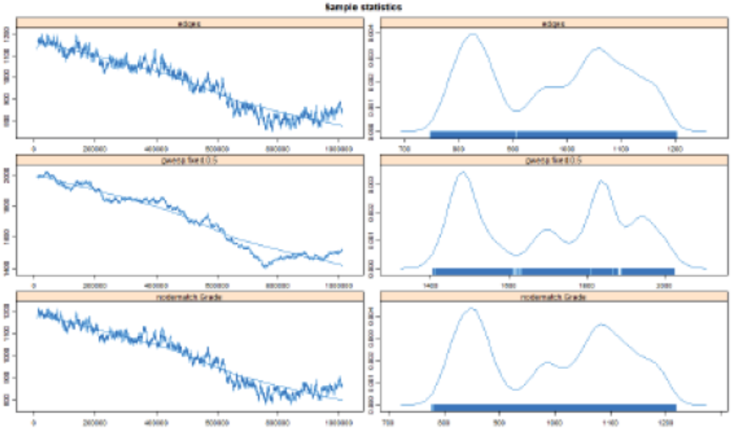
Diagnostics



Social Network Analysis

LJasny

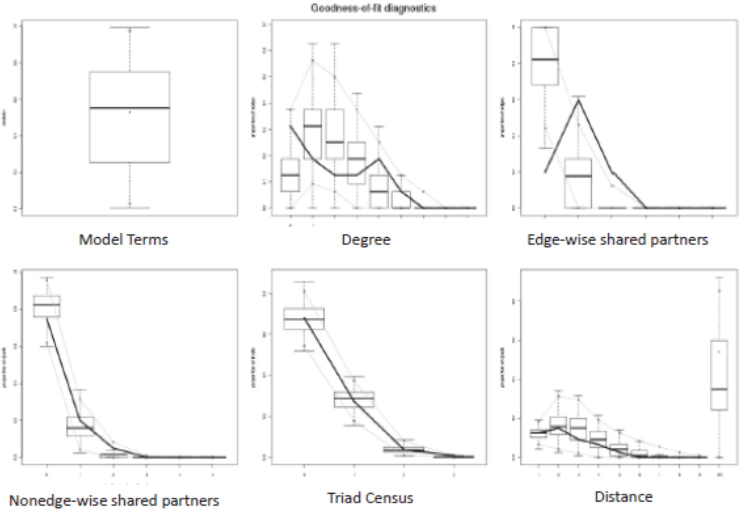
Diagnostics



Assessing Adequacy

- How does one assess model adequacy? Simulation!
    - Simulate draws from fitted model
    - Compare observed graph to simulated graphs on measures of interest
    - Verify that observed properties are well-covered by simulated ones (e.g. not in 5% tails)
- What properties should be considered?
    - This is application-specific - no single uniform answer
    - Start with “in-model” statistics - ERG must get means right, but should still verify non-pathological distributions (remember the triangles)
    - “out-of-model” statistics can be common low-level properties (e.g. degree, triad census) or theoretically motivated quantiles

Example - a model only with edges



Goodness of Fit

Configuration	*Observed*	*Mean*	*StdDev*	*t-ratio*	
EdgeA	22	22.39	3.98	-0.09	
Star2A	71	65.84	24.58	0.21	
Star3A	62	59.14	36.77	0.07	
Star4A	30	36.16	35.31	-0.17	
Star5A	8	15.68	24.14	-0.31	
TriangleA	7	5.33	3.65	0.45	
Cycle4A	16	10.78	9.47	0.55	
IsolatesA	2	0.32	0.59	2.84	#
IsolateEdgesA	0	0.02	0.15	-0.15	
ASA	46.56	43.62	12.82	0.22	
ATA	17.75	13.64	8.13	0.50	
A2PA	57.12	56.06	17.34	0.06	
AETA	33.62	24.39	19.15	0.48	
stddev_degreeA	2.51	2.22	0.33	0.87	
skew_degreeA	1.06	1.30	0.15	-1.60	
clusteringA	0.29	0.22	0.09	0.77	
Mahalanobis distance = 193					

What if model is inadequate?

- Option 1: add terms
  - Which features are poorly captured? Is there a term which would add in such effects?

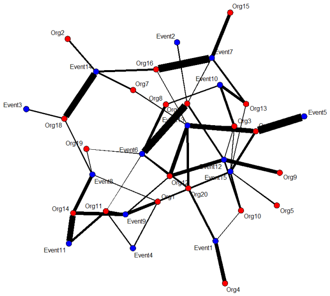
What if model is inadequate?

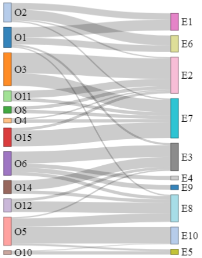
- Option 1: add terms
  - Which features are poorly captured? Is there a term which would add in such effects?
- Option 2: switch terms

What if model is inadequate?

- Option 1: add terms
  - Which features are poorly captured? Is there a term which would add in such effects?
- Option 2: switch terms
- Option 3: do nothing
  - Is the type of inadequacy a problem for your specific question? Can it be tolerated in this case? How good is the overall fit?

Bipartite Data

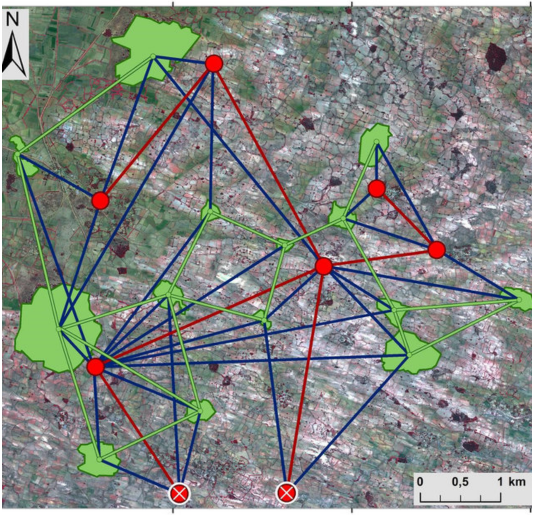




Bipartite Data

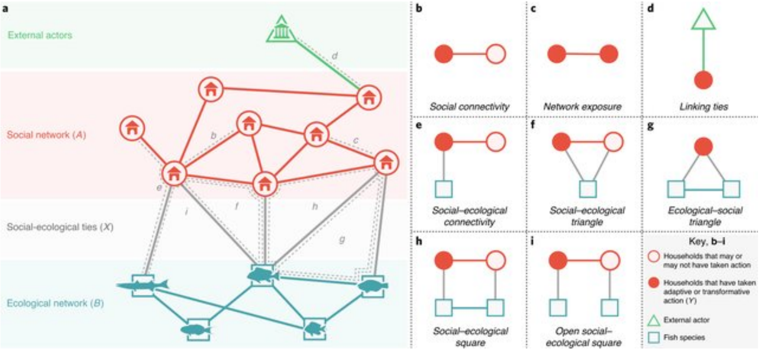
- Many terms already written
- Already in MPNet - just use the X tab
- Simulated networks will not have within-mode ties

Multi-level Data



Bodin, Örjan, and Maria Tengö. "Disentangling intangible social–ecological systems." Global Environmental Change 22.2 (2012): 430-439.

Multi-level Data



Barnes, Michele L., et al. "Social determinants of adaptive and transformative responses to climate change." Nature Climate Change 10.9 (2020): 823-828.

Multi-level Data

- Tons of confusion over the term ‘multi-level’

Multi-level Data

- Tons of confusion over the term ‘multi-level’
- New functionality in Statnet to write these terms

Multi-level Data

- Tons of confusion over the term ‘multi-level’
- New functionality in Statnet to write these terms
- Essentially, treats social and ecological parts as an attribute and runs a normal ERGM

Multi-level Data

- Tons of confusion over the term ‘multi-level’
- New functionality in Statnet to write these terms
- Essentially, treats social and ecological parts as an attribute and runs a normal ERGM
- We’ll see an example using **F** and **Sum**

Multi-level Data

- Tons of confusion over the term ‘multi-level’
- New functionality in Statnet to write these terms
- Essentially, treats social and ecological parts as an attribute and runs a normal ERGM
- We’ll see an example using **F** and **Sum**
- ...but it’s complicated

One last trick

- Say a term is theoretically very important

One last trick

- Say a term is theoretically very important
- But the term hasn't been written

One last trick

- Say a term is theoretically very important
- But the term hasn't been written
  - One solution - write your own term (ergm-terms package)

One last trick

- Say a term is theoretically very important
- But the term hasn't been written
  - One solution - write your own term (ergm-terms package)
- There is a term, but you can't get it to fit

One last trick

- Say a term is theoretically very important
- But the term hasn't been written
  - One solution - write your own term (ergm-terms package)
- There is a term, but you can't get it to fit
  - Simulate a model with lower order parameters (and not your term of interest)

One last trick

- Say a term is theoretically very important
- But the term hasn't been written
    - One solution - write your own term (ergm-terms package)
- There is a term, but you can't get it to fit
    - Simulate a model with lower order parameters (and not your term of interest)
    - Use the goodness-of-fit method to see how extreme your parameter of interest is in your empirical data compared to a sample/simulation from this model