NME 2021

Network Modeling for Epidemics

1

ERGMs: Next steps

Improving model specifications (esp. for triads)

Estimating from egocentrically sampled data

So what happened?

- Everything was going so well, and then:

Summary statistics:

```
edges triangle
  203       62
```

Fit Model | Save Current Model (0/5) | Clear All Models

Current Model Summary | Current Model Fit Report | Model Comparison

```
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.
```

- To understand why, we need to take a step back

Why did the estimation fail?

- MCMC has been key to statistical estimation of complex (i.e., realistic and interesting) models for dependent data
  - And to the emergence of the field of “data science”

- In most cases, it works really well
  - And there is lots of mathematical theory proving it has good convergence properties (see the appendix to the previous session)

- … but, it can run into trouble
  - especially if the model you’re trying to fit is not a good one for the observed network

Dependency cascades

- Models with dyad dependent terms can behave differently than we expect
  - They look simple, almost like logistic regression
  - 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 for creating cascades
- Leads to something called “model degeneracy”

Model Degeneracy

- Technical 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 is a sign of misspecification**

  The model you specified would almost never produce the network you observed

Model Degeneracy

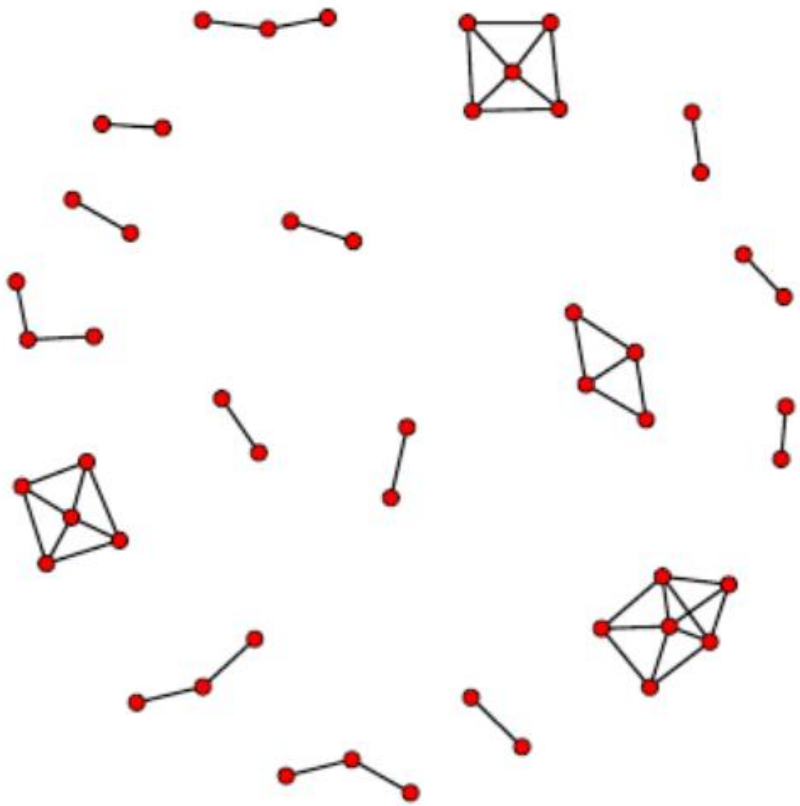
- What does this error message mean?

```
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 co
ntrol.ergm() parameter.
```

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

- Let's see why that is

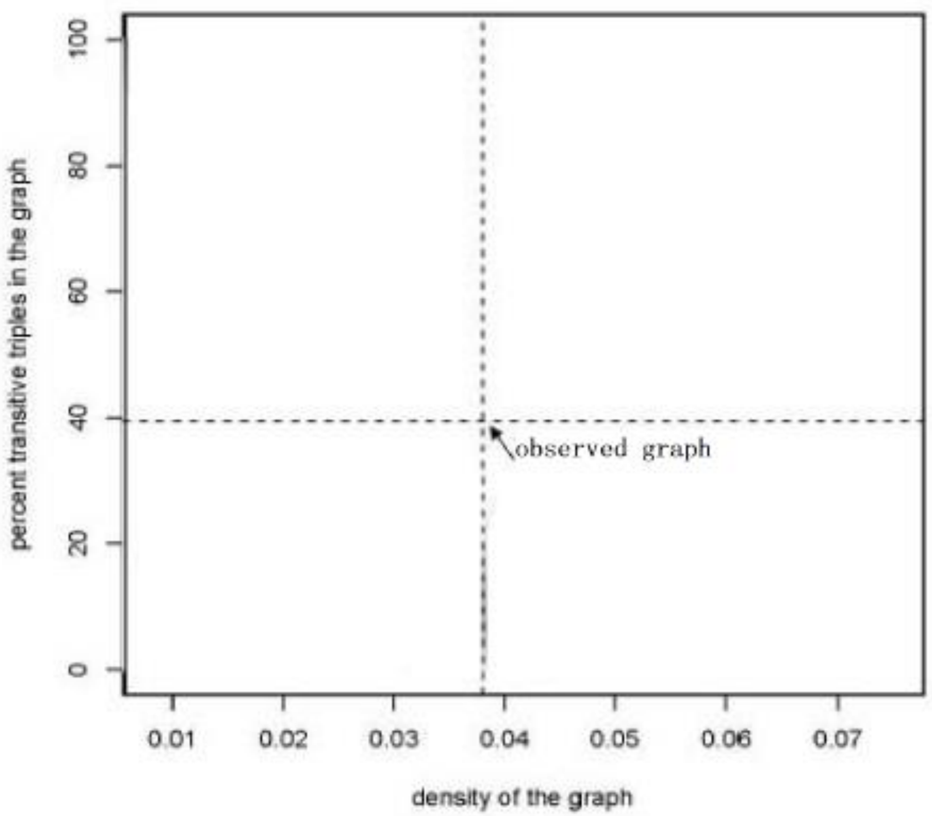
Let’s take a simple example



- This network seems to have lots of triangles
  - 50 nodes
  - 4% density
  - 40% clustering
    - Fraction of all 2stars with the triangle completed

- So it would be natural to fit
  - edges + triangle model

Our network statistics



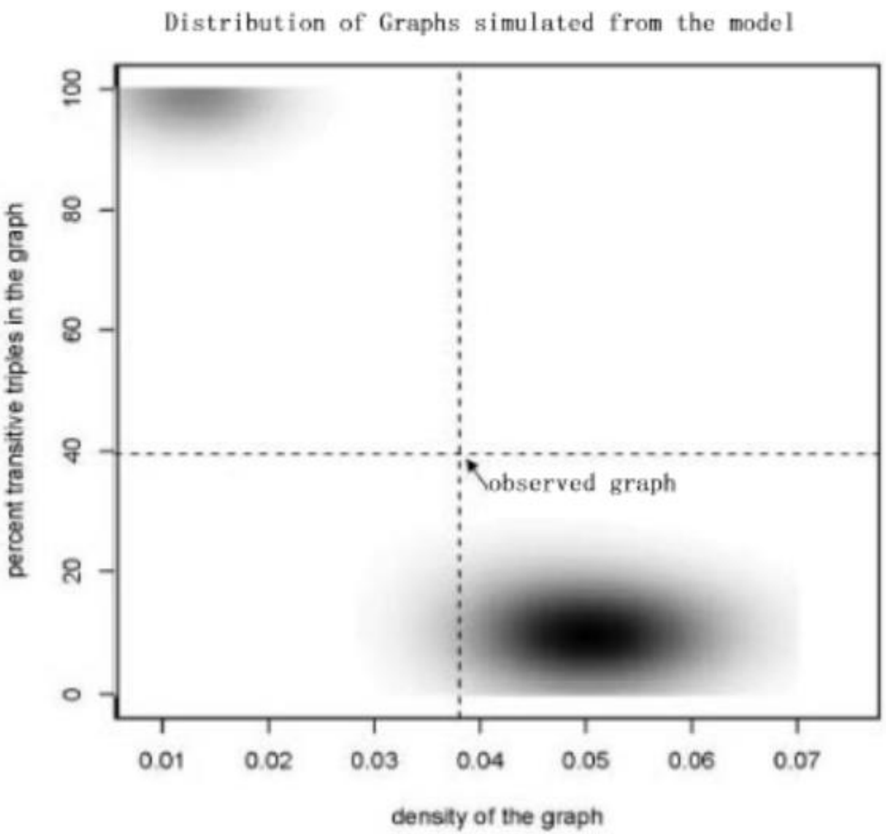
- We can represent our model statistics as a 2D plot

  And our observed graph in this plane

- Statistical theory guarantees that at the MLEs for θ:

  E(netstats) = Observed

At the MLE, this is what the model produces



- The theory is not wrong
- Indeed, the means of the netstats are correct
- But this model produces a *bimodal* distribution to get those means
- It would never produce the observed graph

The problem is the model

- The theory is fine, and the algorithm is fine

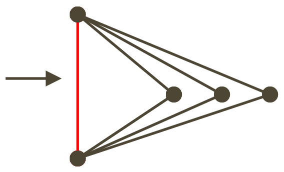
- The problem is the model

  The simple edges + triangle model would not produce our observed graph

- This is what model misspecification looks like with dependent data

Solution: replace the triangle term

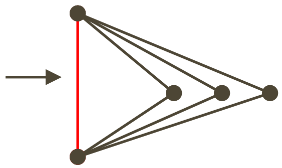
- Old statistic: $t(x) = \sum y_{ij} y_{jk} y_{ik}$



  - $t(x)$ = # of triangles in the graph
    - Here $t(x)$ = 3 if the red edge is toggled on
  - With this term every additional 3-cycle has the same impact, θ
    - So the odds of the red edge above are 3 times higher than an edge that creates only 1 triangle.
    - And an edge that creates 10 triangles has 10x higher odds
  - This is what creates the cascade (and doesn't seem reasonable)

Solution: a better term for triads

- New statistic: $gwesp = e^{\alpha}\sum_{i=1}^{n-2}\{1-(1-e^{-\alpha})^{i}\}sp_i$



  - *gwesp* = a *weighted* sum of the triangles created by each edge
  - Where the weights decline for each additional triangle created
    - For each additional “shared partner” of an edge (like the red edge here)
    - This sets declining marginal returns, with a smooth decay function
  - The decay function we use involves a geometric weighting
    - Hence the name: geometrically weighted edge-wise shared partners
    - a.k.a. GWESP

Details in the Appendix

… to StatnetWeb

Add a gwesp term to the faux.mesa.high model

And conduct model assessments

We will compare four models

Model	Network Statistics g(y)
Edges	# of edges
Edges + GWESP (transitivity)	# of edges weighted shared partners
Edges + Attributes (homophily)	# of edges # of edges for each race, sex, grade # of edges that are within-race, within-grade, within-sex
Edges + Attributes + GWESP (both)	# of edges # of edges for each race, sex, grade # of edges that are within-race, within-grade, within-sex weighted shared partners

These fits can take a while

- So we won't do this interactively now
  - We'll just show the results

- But you can implement these on your own when you have some time

Sequence of fitting and saving models

1. edges

   Fit model, save model

2. + gwesp(0.25, fixed = T)

   Fit model, save model, reset formula

3. + edges + nodefactor("Grade") + nodefactor("Race") + nodefactor("Sex") + nodematch("Grade", diff = T) + nodematch("Race", diff = F) + nodematch("Sex", diff = F)

   Fit model, save model

4. + gwesp(0.25, fixed = TRUE)

   Fit model, save model

Model Comparison

Current Model Summary | Current Model Fit Report | Model Comparison

	Model1	Model2	Model3	Model4
edges	-4.63***	-5.58***	-8.491***	-8.997***
gwesp.fixed.0.25	NA	1.86***	NA	1.377***
nodefactor.Grade.8	NA	NA	1.562*	1.393*
nodefactor.Grade.9	NA	NA	2.533***	2.168***
nodefactor.Grade.10	NA	NA	2.942***	2.445***
nodefactor.Grade.11	NA	NA	2.660***	2.236***
nodefactor.Grade.12	NA	NA	3.470***	2.857***
nodefactor.Race.Hisp	NA	NA	-1.571***	-1.017***
nodefactor.Race.NatAm	NA	NA	-1.103***	-0.765***
nodefactor.Race.Other	NA	NA	-2.916**	-1.930.
nodefactor.Race.White	NA	NA	-0.809**	-0.474*
nodefactor.Sex.M	NA	NA	-0.335***	-0.156*
nodematch.Grade.7	NA	NA	7.441***	5.912***
nodematch.Grade.8	NA	NA	4.330***	3.265***
nodematch.Grade.9	NA	NA	2.060***	1.601**
nodematch.Grade.10	NA	NA	1.234*	1.144*
nodematch.Grade.11	NA	NA	2.525***	1.910***
nodematch.Grade.12	NA	NA	1.358.	1.040.
nodematch.Race	NA	NA	0.832***	0.761***
nodematch.Sex	NA	NA	0.638***	0.536***
AIC	2288	2000	1809	1648
BIC	2296	2015	1960	1807

- Note how the gwesp estimate changes from model 2 to 4
  - About 25% smaller
  - That’s the impact of controlling for attribute effects, including homophily
- Homophily estimates change also, once you control for transitivity

GOF comparison for all 4 models:

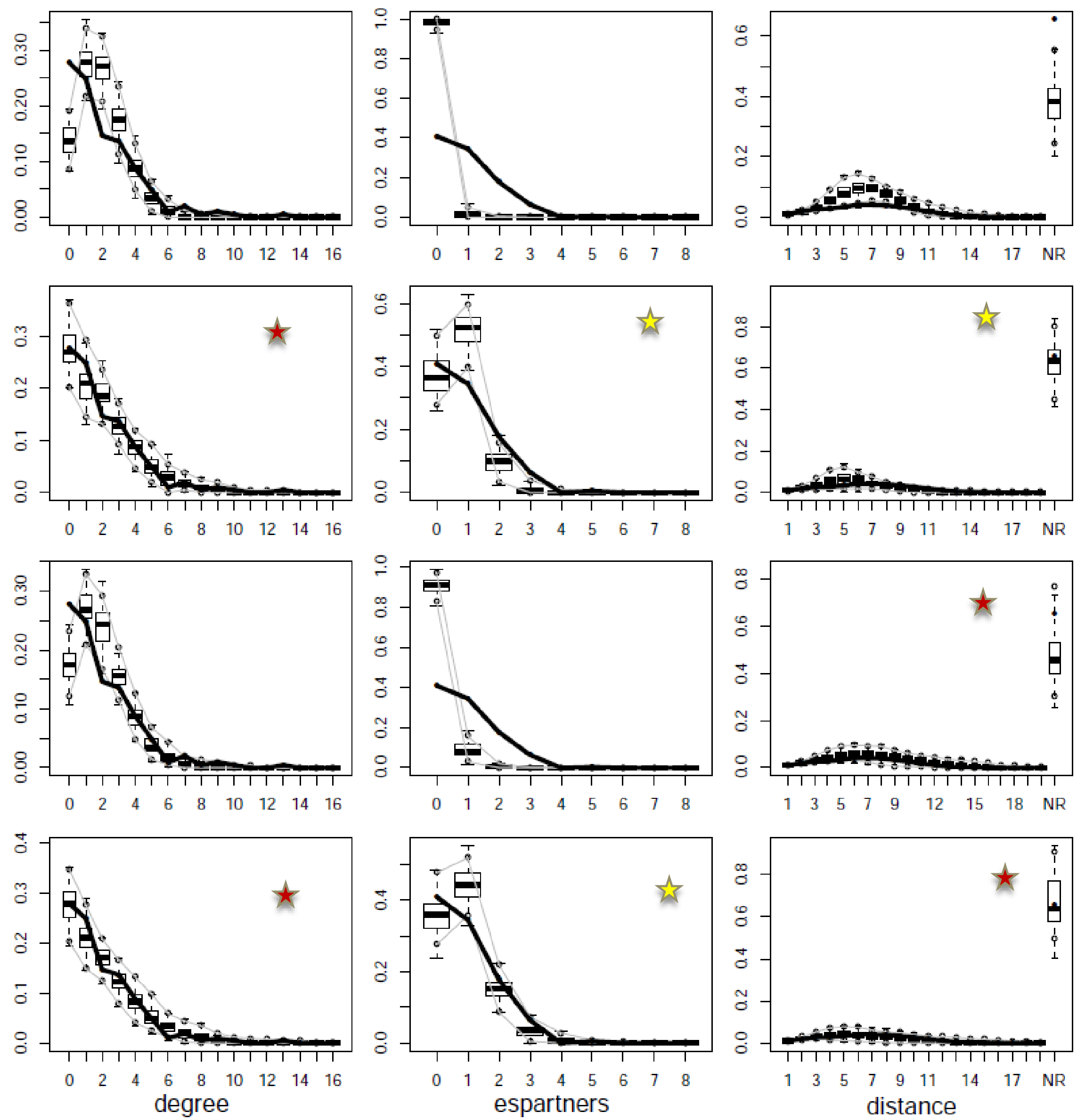
This will take some time to run

1. Edges
AIC: 2288

2. Edges + GWESP
AIC: 1999

3. Edges + Attributes
AIC: 1809

4. Edges + Attributes + GWESP
AIC: 1648



Summary

- Both transitivity and homophily play a role in clustering these friendships
  - Homophily reproduces the geodesic distribution
  - Transitivity (Triadic closure)
    - Reproduces the large number of isolates (degree)
    - Captures the local clustering (ESP) reasonably well, but not the global clustering (geodesics)
  - Both have strong independent effects, but also some correlation
    - ~25% of the transitivity effect is a by-product of homophily (and vice versa)

- The GOF suggests the ESP distribution is still not well fit
  - You could tinker some more, if this was a real research question
  - But we'll move on…

Simulating networks from the model

- A fitted model describes a probability distribution across all networks of this size
  - The model assigns a probability to every possible network
  - The model terms and the estimated coefficients make some networks more likely than others

- You can simulate networks from this distribution
  - Using the same MCMC algorithm that was used for estimation

- And the simulated networks will be centered on the network statistics in the original observed network
  - This is why these models are really useful for network epidemiology

Simulations

- On your own time:

- 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 the autocorrelation

Network Data (redux)

Leveraging the principle of sufficiency

to estimate ERGMs from egocentric samples

What is “sufficiency” ?

- A principle in statistical theory
- That defines what you need to observe in data
- In order to estimate the parameters in your model
  - The data “sufficient” for estimation

Example: from simple linear regression

- The OLS regression coefficient is related to the data as:

$$\hat{\beta} = \frac{Cov(X,Y)}{Var(X)}$$

- I only need to observe these two summary statistics
  - $Cov(X,Y)$ and $Var(X)$
- In order to estimate β
- They are "sufficient"
  - I don't need to have the original data from the individual observations
  - Just these two aggregate summary values

This is *very* helpful for network models

- Because it reduces the burden of data collection

Network data: Three main types (review)

- Network census
  - Data on every node and every link

Often infeasible in practice

- Adaptively sampled networks
  - Link tracing designs (e.g., snowball or RDS)

Challenging to collect, and the statistical methods for analysis are very limited

- Egocentrically sampled networks
  - Enroll population sample (“egos”)
  - Ask them the usual questions about themselves
  - Ask them non-identifying information about their partners (“alters”)
    - Timing (start and end of partnership)
    - Alter characteristics (sex, age, race, etc.)
    - Relational characteristics (type, cohabitation, etc.)
    - Pair-specific behaviors (act frequency, condom use, etc.)
  - Optional: ask about alter-alter ties
  - Optional: ask about perceptions of alters’ alters more generally

Feasible, statistically supported and general

What can we observe in egocentric data

- Aggregate network statistics for:
  - Degree
    - Mean degree, which sets density
    - Degree distributions
  - Nodal attribute heterogeneity
    - Heterogeneity in degree
    - Mixing by nodal attributes
  - Triads
    - Only if the alter-alter matrix data are collected
  - Timing
    - Start/End, Duration of active and completed partnerships

Much of the global structure of a network is set by these local properties

- We can use what we observe to estimate the ERGM coefficients

Egocentric data in ERGMs

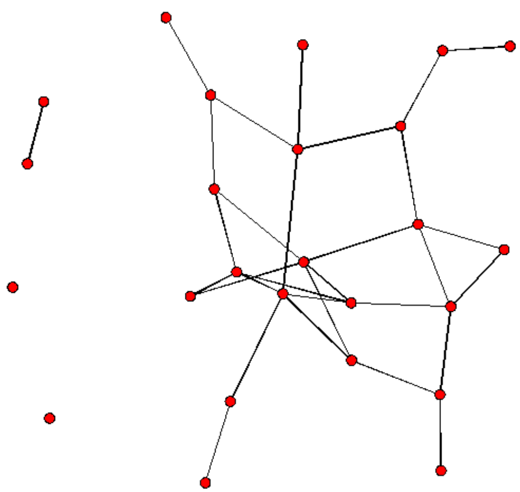
- These can be handled in the software quite easily.

- Recall with faux.mesa.high above, we fit the ergm by providing:
  - A model formula
  - A complete network containing:
    - nodes with their attributes
    - the relations among those nodes

- But alternatively, one can pass:
  - A model formula
  - An set of nodes with their attributes
  - The sufficient statistics for the terms in the model formula
    - Calculated from the observed data, and scaled if desired
    - These are called “target stats” in ergm

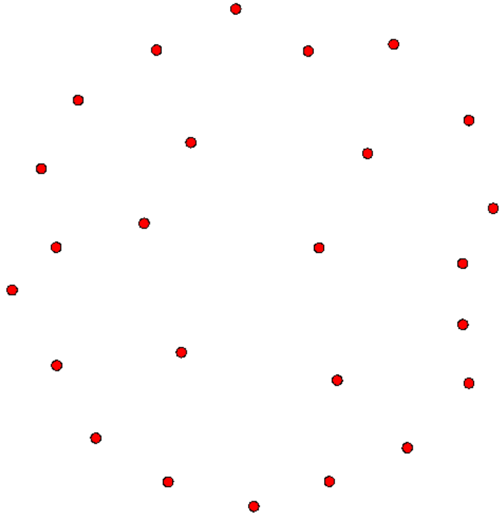
Network statistics in ERGMs

Option 1: network census



```
net~edges+triangle
(ergm automatically
 calculates suff. stats
 from the network data)
```

Option 2: pass nodeset and targets



```
net~edges+triangle
target.stats = c(40, 7)
```

We'll be using this extensively this week

- EpiModel is designed to work with both
  - Complete network data (census)
  - Egocentric data with target stat specifications

- You'll get lots of practice during the labs with target stats

- And we will be reviewing published examples
  - Based on egocentric data
  - That address key issues in HIV prevention and care

Egocentric data for temporal ERGMs

- The same principles apply to estimating temporal ERGMs
  - TERGMS -- For dynamic networks
  - Specify the dynamics of link formation and dissolution

- This requires collecting data on the duration of ties
  - You'll learn more about this in the next session (on STERGMs)
  - And this is the foundation for dynamic, stochastic network-based epidemic simulations

- This is what makes the EpiModel framework so powerful
  - Simple data collection requirements (egocentric samples)
  - Robust statistical methodology for estimation and inference (ergms/tergms)
  - Simulations rooted in empirical network data (that reproduce observed stats)

Lunch!

And after lunch

Temporal ERGMs

- Representing network structure
- And partnership dynamics over time

Selected References

Handcock MS. (2003) Assessing Degeneracy in Statistical Models of Social Networks. CSSS working paper 39. https://www.csss.washington.edu/research/working-papers/39

Hunter DR. Curved Exponential Family Models for Social Networks. (2007) Social networks. 29(2):216-30. doi: 10.1016/j.socnet.2006.08.005. PubMed PMID: PMC2031865.

Hunter DR, Handcock MS. Inference in Curved Exponential Family Models for Networks. (2006) Journal of Computational and Graphical Statistics. 15(3):565-83. doi: 10.1198/106186006X133069.

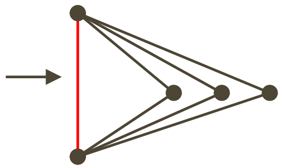
Krivitsky, P. N. and M. Morris (2017). "Inference for social network models from egocentrically sampled data, with application to understanding persistent racial disparities in HIV prevalence in the US." Annals of Applied Statistics 11(1): 427-455.

Appendices

1. The calculation formula for GWESP, and some intuition

2. Technical details of egocentric estimation

1. GWESP calculation



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

sp_i = # of edges with i shared partners

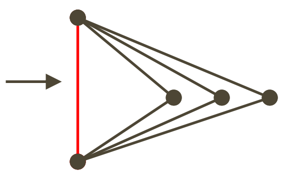
This configuration contains:

- 1 edge with 3 shared partners
- 6 edges with 1 shared partner

α	GWESP(α)
0	$e^{0}\left[\left(1-\left(1-e^{-0}\right)^{1}\right)\times 6\right] + e^{0}\left[\left(1-\left(1-e^{-0}\right)^{3}\right)\times 1\right] = 7$
0.5	$e^{0.5}\left[\left(1-\left(1-e^{-0.5}\right)^{1}\right)\times 6\right] + e^{0.5}\left[\left(1-\left(1-e^{-0.5}\right)^{3}\right)\times 1\right] = 7.55$
1	$e^{1}[(1-(1-e^{-1})^{1})\times 6] + e^{1}[(1-(1-e^{-1})^{3})\times 1] = 8.03$

The # of edges with 1+ shared partners

GWESP: a bit of intuition

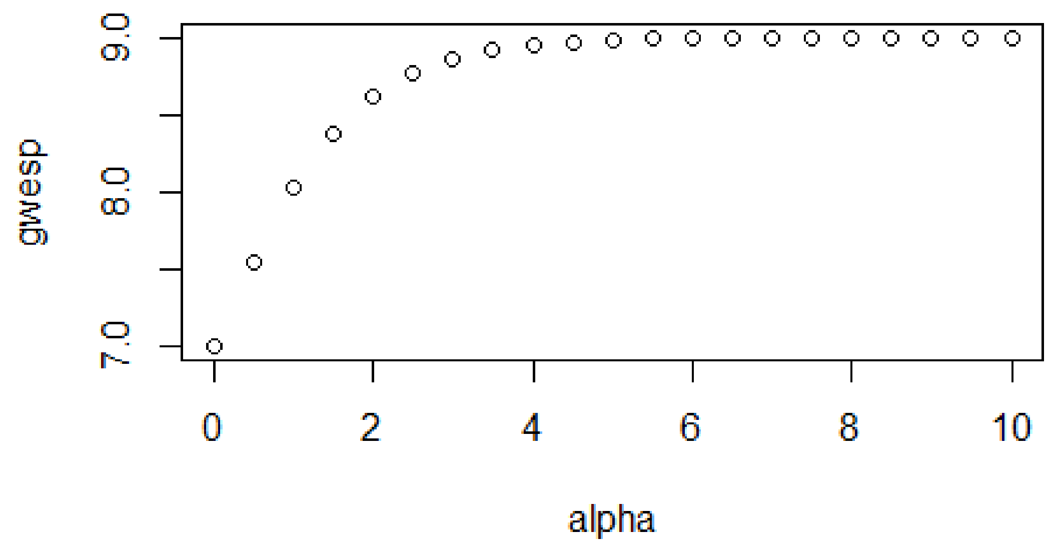


$$gwesp = e^{\alpha} \sum_{i=1}^{n-2} \{1 - (1 - e^{-\alpha})^{i}\} sp_i$$

sp_i = # of edges with i shared partners

Count of edges in each triangle (i.e. # of triangles *x* 3)

Count of edges in at least one triangle (because only an edge's first triangle counts)



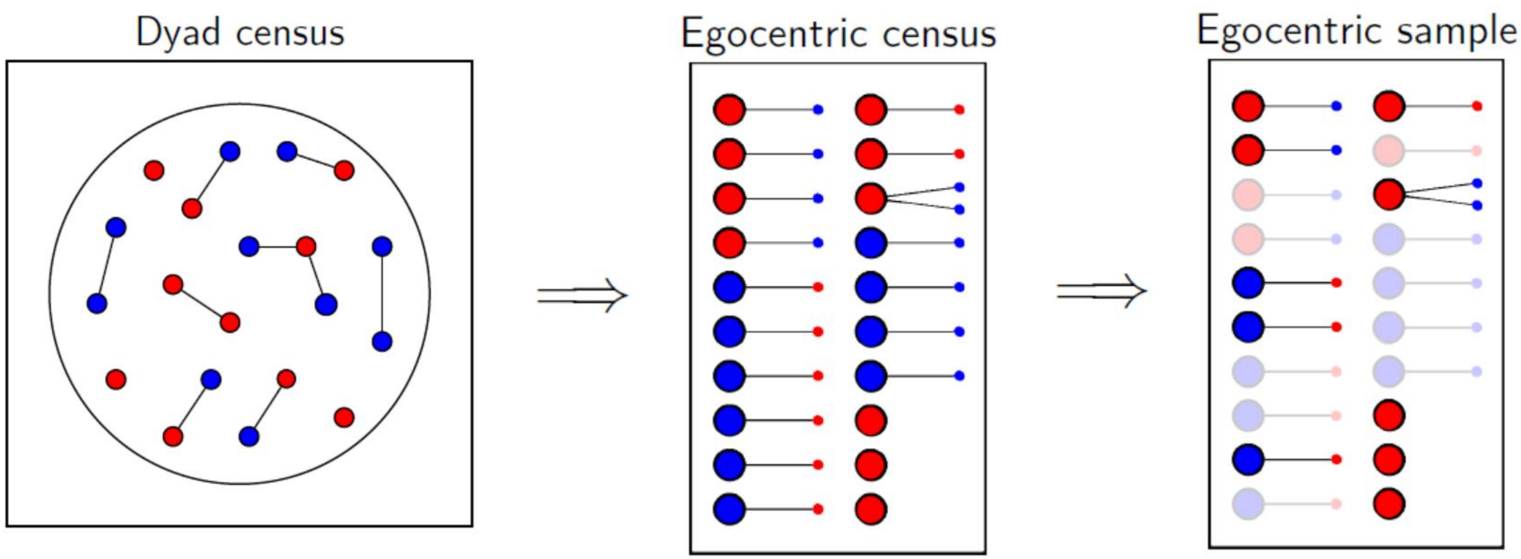
2. Technical details of egocentric estimation

Why does this work? (in a nutshell)

- MLEs for exponential families
  - ERGMs are based in exponential family theory
  - One of the properties of MLEs for exponential families is that
    E(sufficient stats under the model) = observed sufficient stats.
  - Any graph with the same observed sufficient stats has the same probability under the model
    So we don't need to observe the specific complete network
  - We just iterate our way (using MCMC) to finding the coefficients that satisfy
    E(sufficient stats under the model) = observed sufficient stats.

- Statistical inference for sampled data
  - The sufficient stats are like any other sample statistic (e.g., a sample mean)
  - There is a sampling distribution for these statistics
  - Which allows the standard errors to be estimated

How to think about an egocentric sample



Observe the complete network

Observe all egos +
Reported info on alters

Sample egos +
Reported info on alters

Inference from an egocentric sample

Ref: Krivitsky & Morris 2017

- A two-step, finite population framework for inference
  - Step 1: inference on the network statistics $g(y)$
    - We observe $g_s(y)$, the sample network statistics
    - The target of inference is $g(y)$, the population level statistics
    - Relies on a scaling assumption, to define what is size-invariant (see next slide)
    - Can use survey weights, this is a design-based estimator
  - Step 2: inference on the coefficients θ
    - Similar to traditional ERGM inference
    - Relies on the statistical principle of sufficiency, that $g(y)$ is sufficient for estimating θ
      - Intuitively: all networks with the same sufficient statistics have the same probability under the model
    - But this is now a PMLE (Binder, 1983), and the variances are adjusted for step 1 estimates.

Intuition: Scaling up $g_s(y)$ to $g(y)$

- What is the natural size invariant parameterization?
  - Consider, $g(y) = \sum y_{ij}$, the edges term
    - There are 9 ties in our set of 20 nodes on the previous slide

Mean degree	Density *p(tie)*
$\frac{2T}{N} = \frac{2*9}{20} \approx 1$	$\frac{T}{\binom{N}{2}} = \frac{2T}{N(N-1)} = \frac{2*9}{20*19} \approx 0.05$

    - If you double the set to 40 nodes, how many ties would you expect?

$18 = \frac{9*40}{20}$ This preserves the mean degree, but density is now $\frac{2*18}{40*39} \approx 0.02$

$39 = \binom{40}{2} * 0.05$ This preserves the density, but mean degree is now $\frac{2*39}{40} \approx 2$

- It is often natural to preserve the mean degree in social networks
  - Note: Mean degree = Density dependence; P(tie) = Frequency dependence
  - (Krivitsky, Handcock and Morris 2011)

Mean Degree Scaling Adjustment

- This is easy to accomplish with ERGM
  - Include an offset in the model for $-\log(N_{obs})$ to get a per capita scaling
  - Transform the per capita estimates to any desired population size by adding $\log(N_*)$

- Can show that
  - Adjusting the edges term by the offset automatically scales all dyad independent terms
  - Empirically, it also scales degree terms properly
  - Empirically, it does not scale other dyad-dependent terms properly
    - This is not an issue in most egocentrically sampled networks, b/c we don't observe those statistics
    - Other scalings have been proposed for these terms (Krivitsky & Kolaczyk 2015)

Temporal changes in network size and composition

These, too, are easily handled by TERGMs

- Network size changes are handled by dynamic offsets
  - At each time step, add the offset $N_{sim}(t)$ back to the per capita estimate
- Network composition changes require no special treatment
  - ERGMs coefficients are (log) odds ratios
  - Odds ratios are margin independent
  - So the odds-ratio is a natural composition-invariant scaling
  - This is a general solution to the "two-sex problem" in open cohort dynamic modeling

The PMLEs have good statistical properties

- Bias
  - Estimates for unweighted data display no systematic bias
  - For weighted data, bias can be controlled by using larger network size during estimation. (see Krivitsky & Morris 2017 for more information)
- Variance
  - Estimated standard errors appear to be slightly conservative

Egocentric estimation for ERGMs

- There is a also a specific package for estimating ERGMs from egocentrically sampled data

  - ergm.ego
    - Automates calculation of the target stats
    - Handles survey weighting
    - Provides other utilities for egocentric EDA

  - Available on CRAN
    - But is currently being refactored with a new API
    - And is not yet integrated with EpiModel

- In the (near) future, this will be integrated with EpiModel…

Additional references for Appendix

Krivitsky, P. N., M. S. Handcock and M. Morris (2011). "Adjusting for Network Size and Composition Effects in Exponential-Family Random Graph Models." Statistical Methodology **8(4): 319–339.**

Krivitsky, P. N. and M. S. Handcock (2014). "A separable model for dynamic networks." Journal of the Royal Statistical Society, Series B **76(1): 29-46.**

Krivitsky, P. N. and E. D. Kolaczyk (2015). "On the Question of Effective Sample Size in Network Modeling: An Asymptotic Inquiry." Statistical Science **30(2): 184-198.**