Big Problems for Small Networks: Statistical Analysis of Small Networks and Team Performance

George G Vega Yon Kayla de la Haye

Department of Preventive Medicine

SONIC Speaker Series
Northwestern University, IL
March 20, 2019

Contents

Acknowledgements

This material is based upon work support by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-15-1-0577

Computation for the work described in this paper was supported by the University of Southern California's Center for High-Performance Computing (https://hpcc.usc.edu).

We thank members of our MURI research team, USC's Center for Applied Network Analysis, Andrew Slaughter, and attendees of the NASN 2018 conference for their comments.

Network Science of Teams
a Multidisciplinary University Research Initiative
UC SANTA BARBARA
USC University of Southern California
Massachusetts Institute of Technology
NORTHWESTERN UNIVERSITY

Part I: Exponential Random Graph Models for Small Networks

Exponential Random Graph Models (ERGMs)

Figure 1: Friendship network of a UK university faculty. Source: **igraphdata** R package (Gabor Csardi, 2015). Figure drawn using the R package **netplot** ((https://github.com/usccana/netplot)

Exponential Random Graph Models (ERGMs)

Figure 1: Friendship network of a UK university faculty. Source: **igraphdata** R package (Gabor Csardi, 2015). Figure drawn using the R package **netplot** ((https://github.com/usccana/netplot)

How can we explain what we see here?

ERGMs... the *lingua franca* of SNA

- Seeks to answer the question: What local social structures gave origin to a given observed graph?

ERGMs... the *lingua franca* of SNA

- Seeks to answer the question: What local social structures gave origin to a given observed graph?
- The model is centered around a vector of **sufficient statistics** $s\left(\right)$, and is operationalized as:

$$\Pr\left(\mathbf{G} = \mathbf{g} \mid \theta, \mathbf{X}\right) = \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}, \mathbf{X}\right)\right\}}{\kappa\left(\theta, \mathbf{X}\right)}, \quad \forall \mathbf{g} \in \mathcal{G} \tag{1}$$

Where $\kappa\left(\theta, \mathbf{X}\right)$ is the normalizing constant and equals $\sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}$. Figure 2 shows some examples of values in $s\left(\right)$.

ERGMs... the *lingua franca* of SNA

- Seeks to answer the question: What local social structures gave origin to a given observed graph?
- The model is centered around a vector of **sufficient statistics** $s()$, and is operationalized as:

$$
\Pr\left(\mathbf{G} = \mathbf{g} \mid \theta, \mathbf{X}\right) = \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}, \mathbf{X}\right)\right\}}{\kappa\left(\theta, \mathbf{X}\right)}, \quad \forall \mathbf{g} \in \mathcal{G} \tag{1}
$$

  Where $\kappa\left(\theta, \mathbf{X}\right)$ is the normalizing constant and equals $\sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}$. Figure 2 shows some examples of values in $s()$.

- In the case of directed networks, $\mathcal{G}$ has $2^{n(n-1)}$, terms.

ERGMs... the *lingua franca* of SNA

- Seeks to answer the question: What local social structures gave origin to a given observed graph?
- The model is centered around a vector of **sufficient statistics** $s()$, and is operationalized as:

$$\Pr(\mathbf{G} = \mathbf{g} \mid \theta, \mathbf{X}) = \frac{\exp\{\theta^{\mathbf{t}} s(\mathbf{g}, \mathbf{X})\}}{\kappa(\theta, \mathbf{X})}, \quad \forall \mathbf{g} \in \mathcal{G} \tag{1}$$

  Where $\kappa(\theta, \mathbf{X})$ is the normalizing constant and equals $\sum_{\mathbf{g}' \in \mathcal{G}} \exp\{\theta^{\mathbf{t}} s(\mathbf{g}', \mathbf{X})\}$. Figure 2 shows some examples of values in $s()$.

- In the case of directed networks, $\mathcal{G}$ has $2^{n(n-1)}$, terms.
- See Wasserman, Pattison, Robins, Snijders, Handcock and others.

Structures

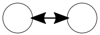
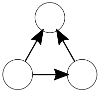
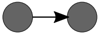
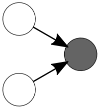
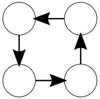
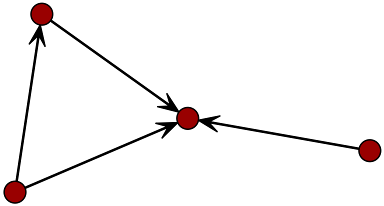
Representation	Description
	Mutual Ties (Reciprocity) $\sum_{i \neq j} y_{ij} y_{ji}$
	Transitive Triad (Balance) $\sum_{i \neq j \neq k} y_{ij} y_{jk} y_{ik}$
	Homophily $\sum_{i \neq j} y_{ij} \mathbf{1}\left(x_i = x_j\right)$
	Covariate Effect for Incoming Ties $\sum_{i \neq j} y_{ij} x_j$
	Four Cycle $\sum_{i \neq j \neq k \neq l} y_{ij} y_{jk} y_{kl} y_{li}$

Figure 2: Besides of the common edge count statistic (number of ties in a graph), ERGMs allow measuring other more complex structures that can be captured as sufficient statistics.

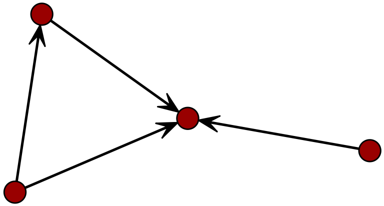
Example of model

In this network



Example of model

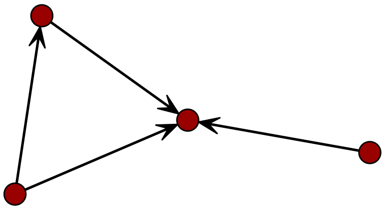
In this network



We see 4 **edges**, 1 **transitive triad** and **no mutual ties**.

Example of model

In this network



We see 4 **edges**, 1 **transitive triad** and **no mutual ties**.

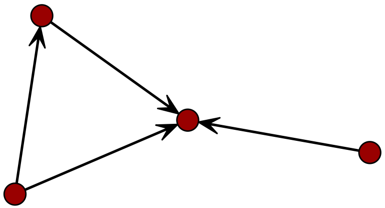
The probability function of this model would be

$$\Pr\left(\mathbf{G}=\mathbf{g}\mid\theta\right)=\frac{\exp\left\{4\theta_{edges}+\theta_{ttriads}+0\theta_{mutual}\right\}}{\sum_{\mathbf{g}'\in\mathcal{G}}\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g}'\right)\right\}}$$

with $\theta=\left[\theta_{edges}\quad\theta_{ttriads}\quad\theta_{mutual}\right]^{\mathbf{t}}$

Example of model

In this network



We see 4 **edges**, 1 **transitive triad** and **no mutual ties**.

The probability function of this model would be

$$\Pr\left(\mathbf{G}=\mathbf{g}\mid\theta\right)=\frac{\exp\left\{4\theta_{edges}+\theta_{ttriads}+0\theta_{mutual}\right\}}{\sum_{\mathbf{g}'\in\mathcal{G}}\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g}'\right)\right\}}$$

with $\theta=\left[\theta_{edges}\quad\theta_{ttriads}\quad\theta_{mutual}\right]^{\mathbf{t}}$

This model has **MLE parameter estimates** of -0.19 (low density), 0.27 (high chance of ttriads), and -9.75 (low chance of mutuality) for the parameters `edges`, `ttriads`, and `mutual` respectively.

Estimation of ERGMs

- Calculating of the normalizing constant in (1), $\kappa = \sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}$, makes ERGMs difficult to estimate.

Estimation of ERGMs

- Calculating of the normalizing constant in (1), $\kappa = \sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}$, makes ERGMs difficult to estimate.
- For this reason, statistical methods have focused on avoiding the direct calculation of κ; most modern methods for estimating ERGMs rely on MCMC.

Estimation of ERGMs (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$
2. While the algorithm doesn't converge, do:
   a. Generate a large sample of graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \,\middle|\, \hat{\theta}_0, \mathbf{X}\right)$ using MCMC
   b. Use the simulated sample of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.
   c. next iteration

Estimation of ERGMs (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$

2. While the algorithm doesn't converge, do:

  a. Generate a large sample of graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \;\middle|\; \hat{\theta}_0, \mathbf{X}\right)$ using MCMC

  b. Use the simulated sample of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.

  c. next iteration

Once it coverges, the last value of $\hat{\theta}_0$ is the MCMC-MLE estimates.

Estimation of ERGMs (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$
2. While the algorithm doesn't converge, do:
   a. Generate a large sample of graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \,\middle|\, \hat{\theta}_0, \mathbf{X}\right)$ using MCMC
   b. Use the simulated sample of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.
   c. next iteration

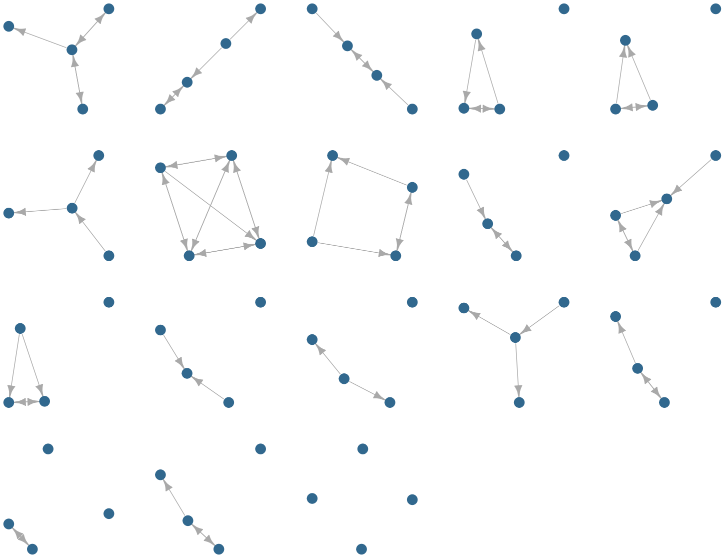
   Once it coverges, the last value of $\hat{\theta}_0$ is the MCMC-MLE estimates.The variance approximation is another story.

- While significant advances have been made in the area, simulation based models can suffer from **model degeneracy**.

- While significant advances have been made in the area, simulation based models can suffer from **model degeneracy**.
- Model degeneracy is particularly problematic with small networks.

Figure 3: Model generacy. Figure

ERGMs for Small Networks



ERGMs for Small Networks (cont'd)

- In the case of small networks (e.g. at most 6 nodes), the calculation of κ becomes computationally feasible.

ERGMs for Small Networks (cont'd)

- In the case of small networks (e.g. at most 6 nodes), the calculation of κ becomes computationally feasible.
- This allows direct calculation of (1), **avoiding the need for simulations** and allowing us to obtain Maximum Likelihood Estimates using *standard* optimizations techniques.

ERGMs for Small Networks (cont'd)

- In the case of small networks (e.g. at most 6 nodes), the calculation of κ becomes computationally feasible.
- This allows direct calculation of (1), **avoiding the need for simulations** and allowing us to obtain Maximum Likelihood Estimates using *standard* optimizations techniques.
- More importantly, in the case that a common data generating process can be assumed, a pooled version of the ERGMs can be estimated.

$$\Pr\left(\mathbf{G}_1 = \mathbf{g}_1, \dots, \mathbf{G}_p = \mathbf{g}_p \mid \theta, \mathbf{X}_1, \dots, \mathbf{X}_p\right) = \prod_p \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}_p, \mathbf{X}_p\right)\right\}}{\kappa_p\left(\theta, \mathbf{X}_p\right)}$$

ERGMs for Small Networks (cont'd)

- In the case of small networks (e.g. at most 6 nodes), the calculation of κ becomes computationally feasible.
- This allows direct calculation of (1), **avoiding the need for simulations** and allowing us to obtain Maximum Likelihood Estimates using *standard* optimizations techniques.
- More importantly, in the case that a common data generating process can be assumed, a pooled version of the ERGMs can be estimated.

$$\Pr\left(\mathbf{G}_1 = \mathbf{g}_1, \dots, \mathbf{G}_p = \mathbf{g}_p \mid \theta, \mathbf{X}_1, \dots, \mathbf{X}_p\right) = \prod_p \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}_p, \mathbf{X}_p\right)\right\}}{\kappa_p\left(\theta, \mathbf{X}_p\right)}$$

How different is this from the "normal" way to fit ERGMs?

Estimation of ERGMs (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$

2. While the algorithm doesn't converge, do:

a. Simulate B graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \,\middle|\, \hat{\theta}_0, \mathbf{X}\right)$ using MCMC

b. Use the simulated sequence of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.

c. next iteration

Estimation of ERGMitos (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$
2. While the algorithm doesn't converge, do:
   a. Simulate B graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \,\middle|\, \hat{\theta}_0, \mathbf{X}\right)$ using MCMC
   b. Use the simulated sequence of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.
   c. next iteration

By skiping the MCMC part we:

1. are able to get MLE estiamates directly,
2. avoiding the degeneracy problem latent in MCMC, and
3. obtain more accurate estimates faster.

Estimation of ERGMs (cont'd)

Description of the MCMC-MLE algorithm (one of the approaches)

1. Make an initial guess of the model parameters using MPL (maximum pseudo likelihood), $\hat{\theta}_0$
2. While the algorithm doesn't converge, do:
   a. Simulate B graphs from $\Pr\left(\mathbf{G} = \mathbf{g} \;\middle|\; \hat{\theta}_0, \mathbf{X}\right)$ using MCMC
   b. Use the simulated sequence of graphs to approximate the likelihood function. And with this approximation update the parameter θ_0 using a Newton-Raphson step.
   c. next iteration

By skiping the MCMC part we:

1. are able to get MLE estiamates directly,
2. avoiding the degeneracy problem latent in MCMC, and
3. obtain more accurate estimates faster.

We have implemented this and more in the `ergmito` R package

An important pause...

ito, ita: From the latin -*īttus*. suffix in spanish used to denote small or affection.
e.g.:

*¡Que lindo ese perr***ito***! / What a beautiful little dog!*
*¿Me darias una tac***ita*** de azúcar? / Would you give me a small cup of sugar?*

An important pause...

ito, ita: From the latin -*īttus*. suffix in spanish used to denote small or affection. e.g.:

*¡Que lindo ese perr**ito**! / What a beautiful little dog!*
*¿Me darias una tac**ita** de azúcar? / Would you give me a small cup of sugar?*

Special thanks to George Barnett who proposed the name during the 2018 NASN!

Features of `ergmito`

This (lifecycle experimental) R package has the following features

Features of `ergmito`

This (lifecycle experimental) R package has the following features

- Built on top of **statnet**'s `ergm` R package.

[1] A directed graph of size 6 has a support set with $2^{6\times(6-1)} = 1,073,741,824$ elements.

Features of `ergmito`

This (lifecycle experimental) R package has the following features

- Built on top of **statnet**'s `ergm` R package.
- Allows estimating ERGMs for small networks (less than 7 and perhaps 6)[1] via MLE.

[1] A directed graph of size 6 has a support set with $2^{6\times(6-1)} = 1,073,741,824$ elements.

Features of `ergmito`

This (lifecycle experimental) R package has the following features

- Built on top of **statnet**'s `ergm` R package.
- Allows estimating ERGMs for small networks (less than 7 and perhaps 6)[1] via MLE.
- Implements pooled ERGM models.

[1] A directed graph of size 6 has a support set with $2^{6\times(6-1)} = 1,073,741,824$ elements.

Features of `ergmito`

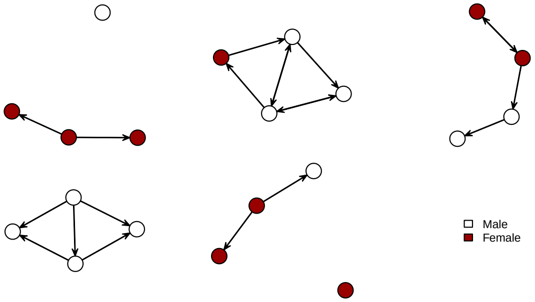
This (lifecycle experimental) R package has the following features

- Built on top of **statnet**'s `ergm` R package.
- Allows estimating ERGMs for small networks (less than 7 and perhaps 6)[1] via MLE.
- Implements pooled ERGM models.
- In the same spirit of the exhaustive enumeration, includes a simulation function for small networks sampling from the true distribution.

[1] A directed graph of size 6 has a support set with $2^{6\times(6-1)} = 1,073,741,824$ elements.

`ergmito` example

```r
library(ergmito)
data(fivenets, package = "ergmito")
```



```r
# Looking at one of the five networks
fivenets[[1]]
```

```
##  Network attributes:
##   vertices = 4
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 2
##     missing edges= 0
##     non-missing edges= 2
##
##  Vertex attribute names:
##     female name
##
## No edge attributes
```

```r
# Looking at one of the five networks
fivenets[[1]]
```

```
##  Network attributes:
##   vertices = 4
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 2
##     missing edges= 0
##     non-missing edges= 2
##
##  Vertex attribute names:
##     female name
##
## No edge attributes
```

So how can we fit this model?

ergmito example (cont'd)

The same as you would do with the ergm package:

```r
(model1 <- ergmito(fivenets ~ edges + nodematch("female")))
```

```
##
## ERGMito estimates
##
##  Coefficients:
##            edges  nodematch.female
##           -1.705             1.587
```

	Model 1
edges	-1.70^{**}
	(0.54)
nodematch.female	1.59^{*}
	(0.64)
AIC	73.34
BIC	77.53
Log Likelihood	-34.67
# Networks	5

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 1: Statistical models

Simulation Study

We conducted a simulation study to explore the properties of MLE for small networks (a.k.a. ERGMito). To generate each sample of teams:

Simulation Study

We conducted a simulation study to explore the properties of MLE for small networks (a.k.a. ERGMito). To generate each sample of teams:

1. Draw the **population parameters** from a piecewise Uniform with values in $[-4, -.1] \cup [.1, 4]$

Simulation Study

We conducted a simulation study to explore the properties of MLE for small networks (a.k.a. ERGMito). To generate each sample of teams:

1. Draw the **population parameters** from a piecewise Uniform with values in $[-4, -.1] \cup [.1, 4]$
2. We will draw **groups of sizes 3 to 5**. The number of networks per group size are drawn from a Poisson distribution with parameter 10 (hence, an expected size of 30 networks per sample).

Simulation Study

We conducted a simulation study to explore the properties of MLE for small networks (a.k.a. ERGMito). To generate each sample of teams:

1. Draw the **population parameters** from a piecewise Uniform with values in $[-4, -.1] \cup [.1, 4]$
2. We will draw **groups of sizes 3 to 5**. The number of networks per group size are drawn from a Poisson distribution with parameter 10 (hence, an expected size of 30 networks per sample).
3. Use the **drawn parameters** and **group sizes** to **generate random graphs** using an ERGM data generating process.

Simulation Study

We conducted a simulation study to explore the properties of MLE for small networks (a.k.a. ERGMito). To generate each sample of teams:

1. Draw the **population parameters** from a piecewise Uniform with values in $[-4, -.1] \cup [.1, 4]$
2. We will draw **groups of sizes 3 to 5**. The number of networks per group size are drawn from a Poisson distribution with parameter 10 (hence, an expected size of 30 networks per sample).
3. Use the **drawn parameters** and **group sizes** to **generate random graphs** using an ERGM data generating process.

We simulated 100,000 samples, each one composed of an average of 30 networks.

Simulation Study (cont'd)

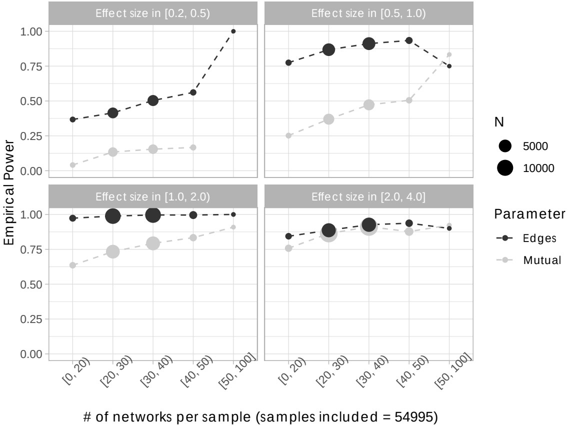


Figure 4: Empirical power of Pooled-ERGM estimates at various levels of effect size. As expected, power increases significantly with sample size (# of networks per sample). Interestingly, the discovery rate of an effect size within $[1, 2)$ is very high even with a sample size of 20-30 networks. More extreme points have higher volatility due to small number of samples included.

Part II: Association between group structure and team performance

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:
  - Sample size is problematic: How costly is getting enough teams to run get a desired level of power?

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:
  - Sample size is problematic: How costly is getting enough teams to run get a desired level of power?
- Permutation tests:

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:
  - Sample size is problematic: How costly is getting enough teams to run get a desired level of power?
- Permutation tests:
  - Common approach: sample from graphs with the same degree sequence–the observed sequence of in/out degree

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:
  - Sample size is problematic: How costly is getting enough teams to run get a desired level of power?
- Permutation tests:
  - Common approach: sample from graphs with the same degree sequence–the observed sequence of in/out degree
  - This is oversimplifying/constraining

Testing effects of social network structure on group performance

Two common approaches: Generalized Linear Models (GLMs), or Mantel-like tests (a.k.a. permutation tests). Both have limitations:

- GLMs:
  - Sample size is problematic: How costly is getting enough teams to run get a desired level of power?
- Permutation tests:
  - Common approach: sample from graphs with the same degree sequence–the observed sequence of in/out degree
  - This is oversimplifying/constraining
  - And worse, in a network of size 4, how many different networks can be observed **holding the degree sequence fixed**?

An idea

- Lack of power:

An idea

- Lack of power:What if we just simulate them?

An idea

- Lack of power:What if we just simulate them?OK, but aren't we doing this with permutation tests?. . .
- Sure, but what about ERGMs?

An idea

- Lack of power:What if we just simulate them?OK, but aren't we doing this with permutation tests?...
- Sure, but what about ERGMs?Ultimately these models describe a distribution of graphs that have <u>on average</u> the same set of network statistics

An idea

- Lack of power:What if we just simulate them?OK, but aren't we doing this with permutation tests?...
- Sure, but what about ERGMs?Ultimately these models describe a distribution of graphs that have <u>on average</u> the same set of network statistics
- This overcomes the problem observed in permutation tests.

A semiparametric test

Notation

- $\mathbf{G} = \{\mathbf{g}_j\}$ is a sequence of J graphs that share a common data-generating-process, e.g. teams formed in a lab.
- Each network has node-level attributes $x \in \mathcal{X}$.
- A group(graph) level outcome variable, such as team performance, Y.
- Under the null, network structure and group performance are not associated, this is $Y \perp \mathbf{G}$.

Algorithm

1. Estimate an ERGM (estimates can come from a single graph or pooled estimates). We denote the data-generating-process of this model as $\mathcal{D} : \Theta \times \mathcal{X} \mapsto \mathcal{G}$.

2. Calculate the value $s_0 = s(\mathbf{G}, Y)$.

3. Now, for $b \in \{1, \dots, B\}$ do:

  3.1 For each group j in $\{1, \dots, J\}$, draw a new network $\mathbf{g}_j^b \sim \mathcal{D}(\hat{\theta}, X_j)$, this new sequence is denoted $\mathbf{G}^b$

  3.2 Using $\mathbf{G}^b$ and Y, calculate $s_b = s(\mathbf{G}^b, Y)$

  3.3 Next b.

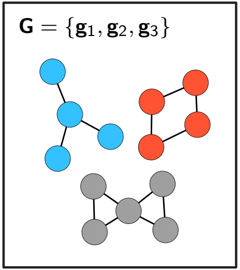
This will generate a null distribution for the statistic s, which we can use to compare against the observed statistic, s_0.

Note An important distinction to make is that structures that gave origin to the graph need not to be relevant for the team's performance per se.

Illustrated example

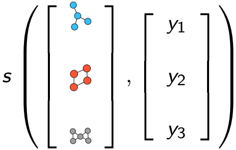
Suppose that we have a 3 networks of sizes 4, 4, and 5 respectively. The

Step 1:
Fit the ERGMito



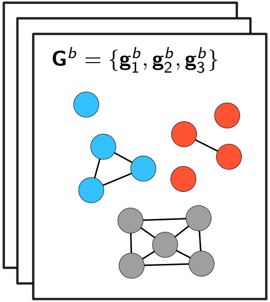
Fit the ERGMito,
This will give us $\mathcal{D}(\hat{\theta}, X_j)$

Step 2:
Calculate $s_0 =$



Throughout the simulations the only part that changes is the networks, not Y

Step 3:
For $b \in 1, \ldots, B$ do



3.1) For $j \in \{1, 2, 3\}$ draw a new network from $\mathcal{D}$
3.2) Use the new sample to calculate $s_b = s(\mathbf{G}^b, Y)$

We can use the distribution of the sequence $\{s_1, \ldots, s_B\}$ as null to compare against s_0

Part III: An empirical example

Small teams performance

Data

Results

Concluding remarks

Exponential Random Graph Models for Small Networks:

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model
3. Can be extended to other types of ERGMs... our next target: TERGMs (Separable Exponential Random Graph Models)

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model
3. Can be extended to other types of ERGMs. . . our next target: TERGMs (Separable Exponential Random Graph Models)

Test for Association between graph level outcomes and graph structures:

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model
3. Can be extended to other types of ERGMs... our next target: TERGMs (Separable Exponential Random Graph Models)

Test for Association between graph level outcomes and graph structures:

1. Still need to run simulation studies

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model
3. Can be extended to other types of ERGMs... our next target: TERGMs (Separable Exponential Random Graph Models)

Test for Association between graph level outcomes and graph structures:

1. Still need to run simulation studies
2. Key question 1: What happens when the graph structure is in both the ERGM and the stat

Concluding remarks

Exponential Random Graph Models for Small Networks:

1. Not a new thing,what's new is the tool to do so in a smooth way
2. Still work to do (on the development side of things): Goodness of fit tests, better algorithms for drawing random graphs, bayesian model
3. Can be extended to other types of ERGMs... our next target: TERGMs (Separable Exponential Random Graph Models)

Test for Association between graph level outcomes and graph structures:

1. Still need to run simulation studies
2. Key question 1: What happens when the graph structure is in both the ERGM and the stat
3. What about other properties such as type I error?

Thanks!

George G. Vega Yon

Let's chat!

vegayon@usc.edu

https://ggvy.cl

@gvegayon

@gvegayon