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

George G Vega Yon[1]   Kayla de la Haye

Department of Preventive Medicine

SONIC Speaker
March 12, 2019

[1] Contact: vegayon@usc.edu. We thank members of our MURI research team, USC's Center for Applied

Contents

Exponential Random Graph Models for Small Networks

Testing

Funding Acknowledgement

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).

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

Exponential Random Graph Models for Small Networks

Exponential Random Graph Models

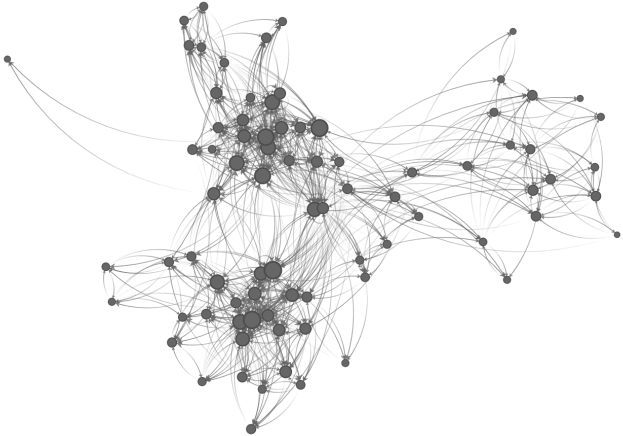


Figure 1: Friendship network of a UK university faculty. Source: `igraphdata` R package (Gabor Csardi, 2015)

How can we explain what we see here?

ERGMs

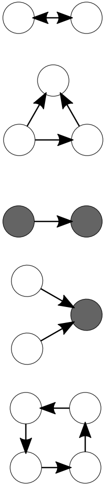
- The *lingua franca* of social network analysis.
- Seeks to answer the question: *What local social structures gave origin to a given observed graph?*
- The model is centered around a vector of **sufficient statistics** $s\left(\right)$, and is operationalized as:

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

  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 $s\left(\right)$.

- Overall, an ERGM identifies the set of parameters θ that maximize the likelihood of observing a given graph $\mathbf{g}$ over the entire set of possible networks, $\mathcal{G}$,
- In the case of directed networks, $\mathcal{G}$ has $2^{n(n-1)}$, terms.
- See Wasserman, Pattison, Robins, Snijders, Handcock and others.

Structures



Mutual Ties
$\sum_{i \neq j} y_{ij} y_{ji}$

Transitive Triad
$\sum_{i \neq j \neq k} y_{ij} y_{jk} y_{ik}$

Homophily
$\sum_{i \neq j} y_{ij} \mathbf{1} (x_i = x_j)$

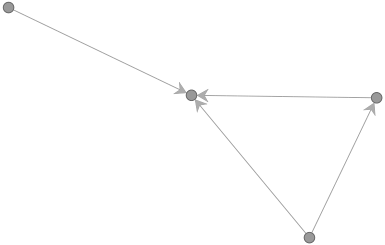
Covariate Effect for Incoming Ties
$\sum_{i \neq j} y_{ij} x_j$

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.

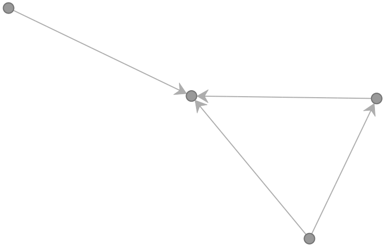
Example of model

In this network



Example of model

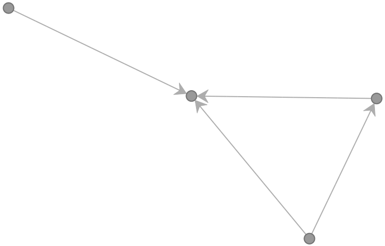
In this network



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**.

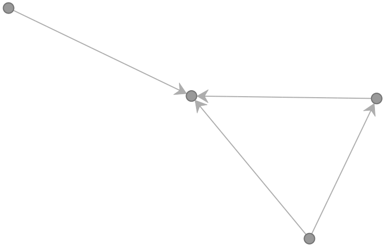
The probability function of this model would be

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

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

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

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), $\sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}$, 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.
- 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.

ERGMs for Small Networks

- 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.

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

- We have implemented this and more in the `ergmito` (lifecycle experimental) R package (https://github.com/muriteams/ergmito)

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)[2] 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.

[2] A directed graph of size 6 has 1,073,741,824

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)

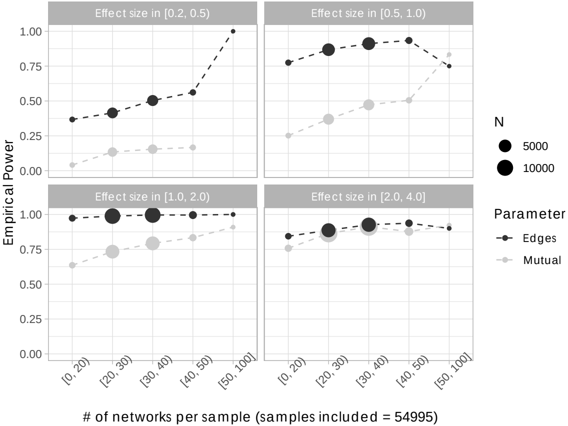


Figure 3: Empirical power of Pooled-ERGM estimates at various levels of effect size. As expected, power increases significantly with sample size (# of networks per sample). Interestingly, the discovery rate of an effect size within $[1, 2)$ is very high even with a sample size of 20-30 networks. More extreme points have higher volatility due to small number of samples included.

Testing

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 often lack power: reaching higher levels of discovery rates implies working with a larger number of teams, which can be impractical.
- Permutation tests oversimplify the network model: typically degree sequence characterizes the family of networks Proposed solution: a semiparametric test version of a permutation tests that aims to overcome both problems using Exponential Random Graph Models.

A semiparametric test

Preamble

- $\mathbf{G} = \{\mathbf{g}_j\}$ 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.

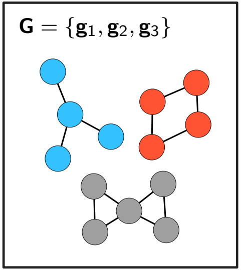
   This will generate a null distribution for the statistic s, which we can use to compare against the observed statistic, s_0.

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 $\mathcal{D}(\hat{\theta}, X_j)$

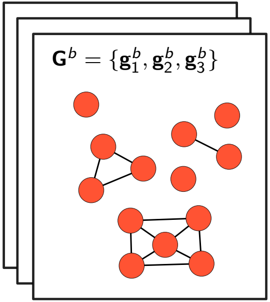
Step 2:
Calculate $s_0 =$

$$s\left(\begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}\right)$$

This is the observed statistic.

We will generate a null distribution where $\mathcal{G} \perp Y$.

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


$\mathbf{G}^b = \{\mathbf{g}_1^b, \mathbf{g}_2^b, \mathbf{g}_3^b\}$

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

The generated sequence of statistics $\{s_1, \dots, s_B\}$