

What drives social networks?

A gentle introduction to exponential random graph models (with a focus on **small networks**)

George G Vega Yon



Department of Preventive Medicine

LAERUG
June 10, 2019

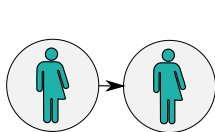
Social networks



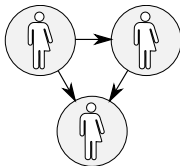
Figure 1: Friendship network of a UK university faculty. Source: **igraphdata** R package (Csardi, 2015). Figure drawn using the R package **netplot** (yours truly, <https://github.com/usccana/netplot>)

What drives **social** networks?

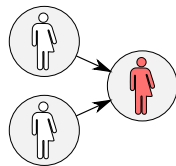
Why are you and I are *[blank]* ? (friends, collaborators, etc.)



Homophily



Transitive Triad



Popularity

Let's build a model for this!

Exponential Family Random Graph Models (ERGMs)



We need to build a probability function for ...

$\#edges, \#homophilic\ ties, \dots$

$\theta_1 \times \#edges + \theta_2 \times \#homophilic\ ties + \dots$

$\exp \{ \theta_1 \times \#edges + \theta_2 \times \#homophilic\ ties + \dots \}$

$$\frac{\exp \{ \theta_1 \times \#edges + \theta_2 \times \#homophilic\ ties + \dots \}}{\sum \exp \{ \dots \}}$$

You got yourself an ERGM!

ERGMs... the *lingua franca* of SNA

A vector of
model parameters

A vector of
sufficient statistics

$$\Pr(\mathbf{Y} = \mathbf{y} \mid \theta, \mathbf{X}) = \frac{\exp\{\theta^t s(\mathbf{y}, \mathbf{X})\}}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\{\theta^t s(\mathbf{y}', \mathbf{X})\}}, \quad \forall \mathbf{y} \in \mathcal{Y}$$

Observed data

The normalizing
constant

All possible
networks

There is one problem with this model ...

$$\Pr(\mathbf{Y} = \mathbf{y} \mid \theta, \mathbf{X}) = \frac{\exp\{\theta^t \mathbf{s}(\mathbf{y}, \mathbf{X})\}}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\{\theta^t \mathbf{s}(\mathbf{y}', \mathbf{X})\}}, \quad \forall \mathbf{y} \in \mathcal{Y}$$

A vector of model parameters A vector of sufficient statistics

Observed data The normalizing constant All possible networks

because of \mathcal{Y} , the **normalizing constant** is
a summation of $2^{n(n-1)}$ terms 🤯!

To solve this, instead of directly computing this function, estimation is done by approximating ratios of likelihood functions instead (TL;DR we use simulations).



Let's get going

We will use the famous Monk data from Sampson (1969)

```
library(ergm)
data(samplk, package="ergm")
```

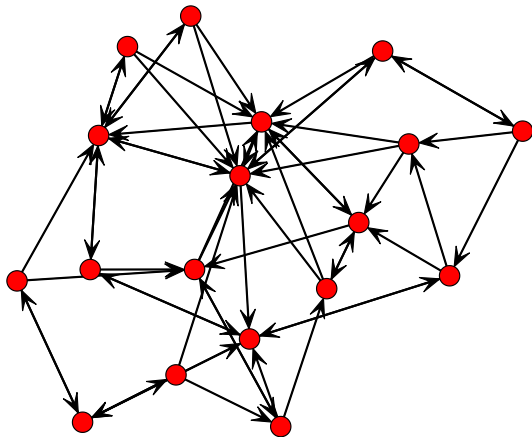
This is an object of class network

```
samplk1
```

```
## Network attributes:
##   vertices = 18
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 55
##     missing edges= 0
##     non-missing edges= 55
##
## Vertex attribute names:
##   cloisterville group vertex.names
##
## No edge attributes
```

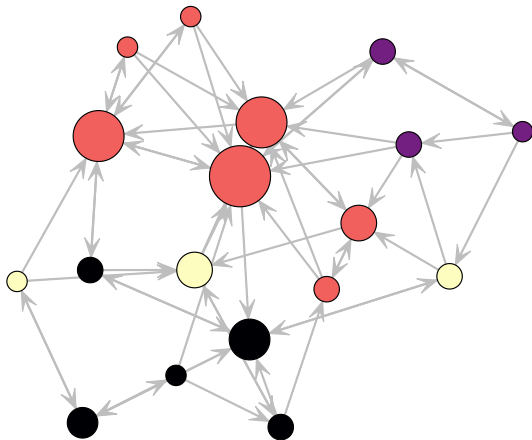


```
library(sna) # Tools for SNA  
set.seed(1) # Graph layout is usually random-driven  
gplot(samplk1)
```



Let's add some color and other features

```
set.seed(1)
cols <- viridisLite::magma(4)[as.factor((samplk1 %v% "group"))]
gplot(samplk1, vertex.cex = degree(samplk1)/4, vertex.col = cols, edge.col = "gray")
```



A simple ergm model

- Suppose we want to test whether homophily on *group* (individuals of the same group tend to connect with each other) and transitive triads (the friend of my friend) are driving the structure:

```
summary(samplk1 ~ edges + nodematch("group") + ttriad)
```

```
##          edges nodematch.group          ttriple  
##             55             30             24
```

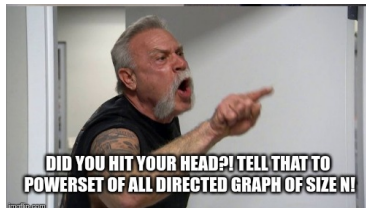
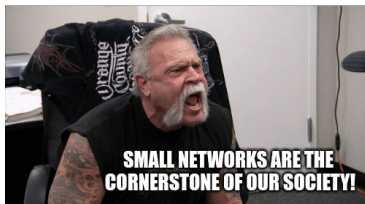
- To estimate this model we do:

```
ans <- ergm(  
  samplk1 ~ edges + nodematch("group") + ttriad,  
  control = control.ergm(seed = 112)  
)
```


```
summary(ans)
```

```
##
## =====
## Summary of model fit
## =====
##
## Formula:    samplk1 ~ edges + nodematch("group") + ttriad
##
## Iterations: 2 out of 20
##
## Monte Carlo MLE Results:
##           Estimate Std. Error MCMC % z value Pr(>|z|)
## edges      -1.7738     0.3049      0  -5.819   <1e-04 ***
## nodematch.group  1.9730     0.3906      0   5.052   <1e-04 ***
## ttriple     -0.2984     0.1954      0  -1.527    0.127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 424.2  on 306  degrees of freedom
## Residual Deviance: 255.8  on 303  degrees of freedom
##
## AIC: 261.8    BIC: 272.9    (Smaller is better.)
```

Now its time for small networks!



ERGMs for Small Networks

- ▶ In the case of small networks (e.g. at most 6 nodes), the calculation of **normalizing constant** becomes computationally feasible.
- ▶ This allows direct calculation of the likelihood, **avoiding the need for simulations** and allowing us to obtain Maximum Likelihood Estimates using *standard* optimization techniques.
- ▶ In addition, most of the time samples of small networks include multiple of them, e.g.: Families, Small teams (like our data), Ego-nets, etc.
- ▶ This makes pooled ERGM estimates a natural way of modeling the data.
- ▶ This and more can be found in the **ergmito** R package (/muriteams/ergmito)

Sidetrack...

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

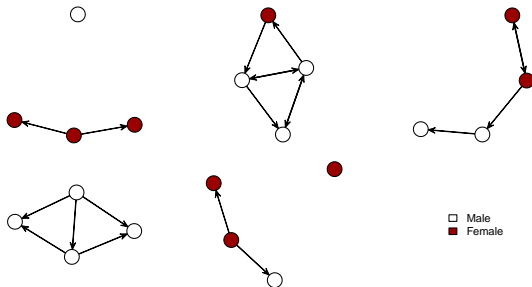
¡Qué lindo ese perrito! / What a beautiful little dog!

¿Me darías una tacita de azúcar? / Would you give me a small cup of sugar?

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

ergmito example

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



```
# 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
```

How can we fit an ERGMito to this 5 networks?

ergmito example (cont'd)

The same as you would do with the ergm package:

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



```
##
## ERGMito estimates
##
## formula:  fivenets ~ edges + nodematch("female")
##
##              Estimate Std. Error  z value    Pr(>|z|)
## edges          -1.704748   0.5435573 -3.136280 0.001711055
## nodematch.female  1.586965   0.6430475  2.467882 0.013591530
```

Some features of this (lifecycle experimental) R package

- ▶ Built on top of **statnet**'s `ergm` R package.
- ▶ Allows estimating ERGMs for small networks (less than 7 and perhaps 6) via MLE.
- ▶ Implements pooled ERGM models.
- ▶ Includes a simulation function for efficiently drawing samples of small networks, and by **efficiently** we mean **fast**.

And much more!



Thanks!



George G. Vega Yon

vegayon@usc.edu

<https://ggvy.cl>

gvegayon gvegayon

Appendix

Structures




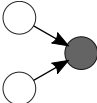

Representation	Description
	Mutual Ties (Reciprocity) $\sum_{i \neq j} y_{ij} y_{ji}$
	Transitive Triad (Balance) $\sum_{i \neq j \neq k} y_{ij} y_{jk} y_{ik}$
	Homophily $\sum_{i \neq j} y_{ij} \mathbf{1}(x_i = x_j)$
	Covariate Effect for Incoming Ties $\sum_{i \neq j} y_{ij} x_j$
	Four Cycle $\sum_{i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

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.

References I

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2018. Rmarkdown: Dynamic Documents for R. <https://rmarkdown.rstudio.com>.

Csardi, Gabor. 2015. Igraphdata: A Collection of Network Data Sets for the 'Igraph' Package. <https://CRAN.R-project.org/package=igraphdata>.

Handcock, Mark S., David R. Hunter, Carter T. Butts, Steven M. Goodreau, Pavel N. Krivitsky, and Martina Morris. 2018. Ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks. The Statnet Project (<http://www.statnet.org>). <https://CRAN.R-project.org/package=ergm>.

Handcock, Mark, Peng Wang, Garry Robins, Tom Snijders, and Philippa Pattison. 2006. "Recent developments in exponential random graph (p^*) models for social networks." Social Networks 29 (2): 192–215. <https://doi.org/10.1016/j.socnet.2006.08.003>.

Hunter, David R., Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. 2008. "Ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks." Journal of Statistical Software 24 (3): 1–29.

References II

R Core Team. 2018. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

Sampson, Samuel F. 1969. "A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships."

Vega Yon, George. 2018. ergmito: Exponential Random Graph Models for Small Networks. <https://github.com/muriteams/ergmito>.

Vega Yon, George, and de la HayeKayla. n.d. "Exponential Random Graph models for Little Networks."

Wasserman, Stanley, and Philippa Pattison. 1996. "Logit models and logistic regressions for social networks: I. An introduction to Markov graphs andp." Psychometrika 61 (3): 401–25. <https://doi.org/10.1007/BF02294547>.

Xie, Yihui. 2018. Knitr: A General-Purpose Package for Dynamic Report Generation in R. <https://yihui.name/knitr/>.