Exact Statistics and Semi-Parametric Tests for Small Network Data[1]

George G. Vega Yon, MS Andrew Slaughter, PhD Kayla de la Haye, PhD

Sunbelt 2019, Montreal
June 20, 2019

[1] Contact: vegayon@usc.edu. We thank members of our MURI research team, USC's Center for Applied Network Analysis, Garry Robins, Carter Butts, Johan Koskinen, Noshir Contractor, and attendees of the NASN 2018 conference for their comments.

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 (hpc.usc.edu).

Context: Social abilities and team performance

Two research questions

Context: Social abilities and team performance

Two research questions

How do **social abilities** impact **network structure**?

Context: Social abilities and team performance

Two research questions

How do **social abilities** impact **network structure**?

How does **collective intelligence** affect team (network) **performance**?

Context: Social abilities and team performance

Two research questions

How do **social abilities** impact **network structure**?

How does collective intelligence affect team (network) performance?

We will focus on the first one

To answer this question, we have the following experimental data:

To answer this question, we have the following experimental data:

- 42 mixed-gender teams,

To answer this question, we have the following experimental data:

- 42 mixed-gender teams,
- Which completed 1 hour of group tasks (collective intelligence developed by MIT (Kim et al. 2017))

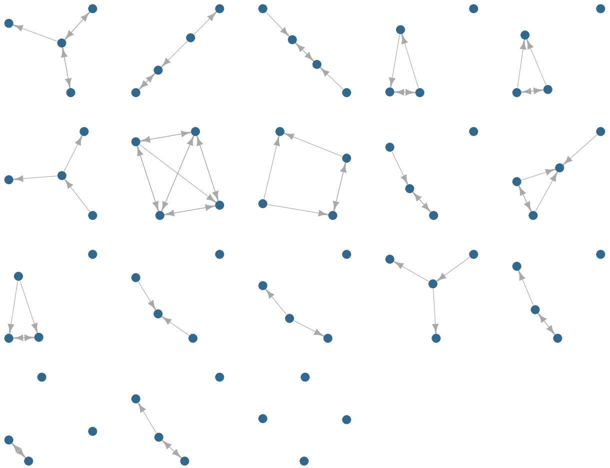
To answer this question, we have the following experimental data:

- 42 mixed-gender teams,
- Which completed 1 hour of group tasks (collective intelligence developed by MIT (Kim et al. 2017))
- Survey capturing information regarding socio-demographics **and**:

To answer this question, we have the following experimental data:

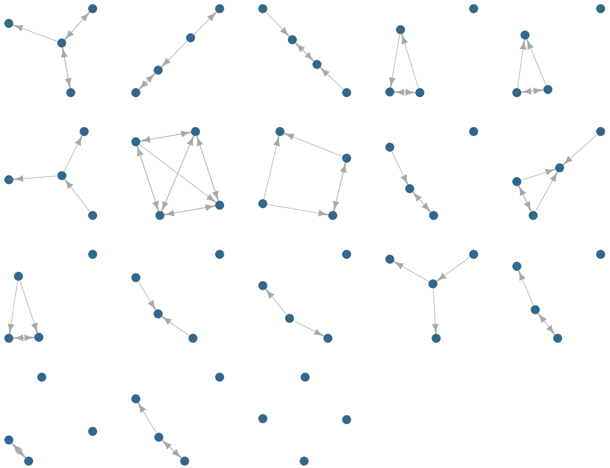
- 42 mixed-gender teams,
- Which completed 1 hour of group tasks (collective intelligence developed by MIT (Kim et al. 2017))
- Survey capturing information regarding socio-demographics **and**:
  - **Social Intelligence** (SI domains): Social Perception (measured by RME), Social Accomodation, Social Gregariousness, and Social Awareness
  - **Social Networks**: Advice Seeking, Leadership, Influence (among others).

Context (cont'd)



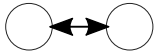
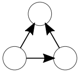
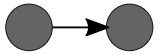
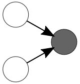
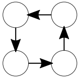
We can do a lot of simple statistics: density, prop of *[blank]*, etc. but...

Context (cont'd)



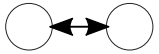
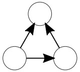
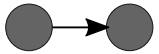
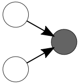
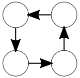
We can do a lot of simple statistics: density, prop of *[blank]*, etc. but... **how can we go beyond that?**

Exponential random graph models

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 $\sum_{i \neq j \neq k \neq l} y_{ij} y_{jk} y_{kl} y_{li}$

ERGMs can do the job, but the only problem is...

Exponential random graph models

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 $\sum_{i \neq j \neq k \neq l} y_{ij} y_{jk} y_{kl} y_{li}$

ERGMs can do the job, but the only problem is... have you tried estimating ERGMs in small networks?

Exponential random graph models for small networks

A lot of

Exponential random graph models for small networks

A lot of

- Playing with the MCMC control parameters to obtain sensible statistics, or

Exponential random graph models for small networks

A lot of

- Playing with the MCMC control parameters to obtain sensible statistics, or
- Sometimes we also go for using a single big (very sparse) graph

Exponential random graph models for small networks

A lot of

- Playing with the MCMC control parameters to obtain sensible statistics, or
- Sometimes we also go for using a single big (very sparse) graph
  - Block diagnoal matrix
  - Constrain the sampling space puting structural zeros (thanks statnet for the `blockdiag(attrname)` constraint!)

Exponential random graph models for small networks

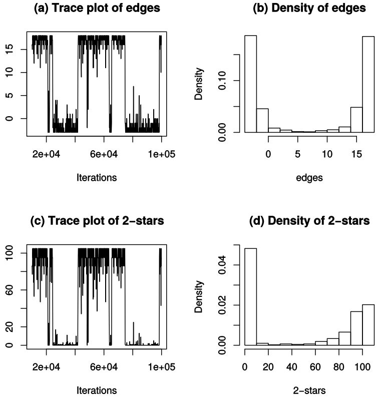
A lot of

- Playing with the MCMC control parameters to obtain sensible statistics, or
- Sometimes we also go for using a single big (very sparse) graph
  - Block diagnoal matrix
  - Constrain the sampling space puting structural zeros (thanks statnet for the `blockdiag(attrname)` constraint!)

This fails too often (smaller networks = higher chance of model degeneracy).

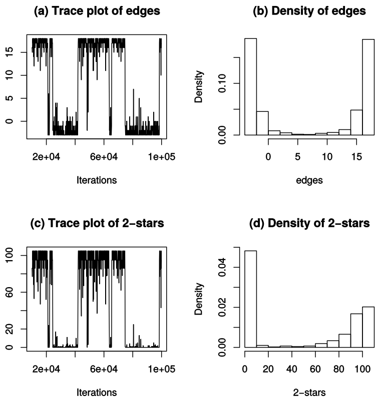
Revising model degeneracy

Following Handcock (2003), the key question is: Where do the sufficient statistics live?



Revising model degeneracy

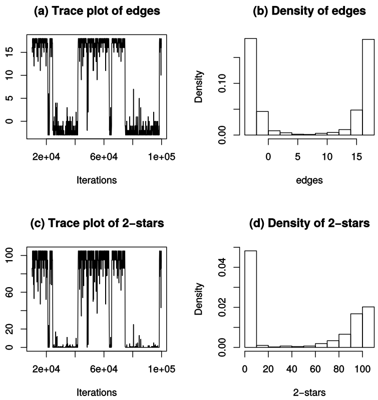
Following Handcock (2003), the key question is: Where do the sufficient statistics live?



- In the interior: **Good**, we (possibly) get nice estimates in both MC-MLE and MLE

Revising model degeneracy

Following Handcock (2003), the key question is: Where do the sufficient statistics live?



- In the interior: **Good**, we (possibly) get nice estimates in both MC-MLE and MLE
- Not in the interior: **We are in trouble**, we mostly get degenerate estimates (more with MC-MLE, but still with MLE)

ERGMs for small networks

- Calculating the likelihood function for a directed graph means (at some point) enumerating $2^{n(n-1)}$ **terms**.

$$
\Pr\left(\mathbf{G} = \mathbf{g} \mid \theta, \mathbf{X}\right) = \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}, \mathbf{X}\right)\right\}}{\sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}}
$$

ERGMs for small networks

- Calculating the likelihood function for a directed graph means (at some point) enumerating $2^{n(n-1)}$ **terms**.

$$\Pr\left(\mathbf{G}=\mathbf{g}\mid\theta,\mathbf{X}\right)=\frac{\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g},\mathbf{X}\right)\right\}}{\sum_{\mathbf{g}'\in\mathcal{G}}\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g}',\mathbf{X}\right)\right\}}$$

- So, if $n=6$, then we have approx 1,000,000,000 terms.

ERGMs for small networks

- Calculating the likelihood function for a directed graph means (at some point) enumerating $2^{n(n-1)}$ **terms**.

$$
\Pr\left(\mathbf{G} = \mathbf{g} \mid \theta, \mathbf{X}\right) = \frac{\exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}, \mathbf{X}\right)\right\}}{\sum_{\mathbf{g}' \in \mathcal{G}} \exp\left\{\theta^{\mathbf{t}} s\left(\mathbf{g}', \mathbf{X}\right)\right\}}
$$

- So, if $n = 6$, then we have approx 1,000,000,000 terms.
- This has lead the field to aim for (very neat) simulation based methods

ERGMs for small networks

- Calculating the likelihood function for a directed graph means (at some point) enumerating $2^{n(n-1)}$ **terms**.

$$\Pr\left(\mathbf{G}=\mathbf{g}\mid\theta,\mathbf{X}\right)=\frac{\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g},\mathbf{X}\right)\right\}}{\sum_{\mathbf{g}'\in\mathcal{G}}\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g}',\mathbf{X}\right)\right\}}$$

- So, if $n=6$, then we have approx 1,000,000,000 terms.
- This has lead the field to aim for (very neat) simulation based methods
- What if our networks have at most that (6 nodes)?

ERGMs for small networks

- Calculating the likelihood function for a directed graph means (at some point) enumerating $2^{n(n-1)}$ **terms**.

$$\Pr\left(\mathbf{G}=\mathbf{g}\mid\theta,\mathbf{X}\right)=\frac{\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g},\mathbf{X}\right)\right\}}{\sum_{\mathbf{g}'\in\mathcal{G}}\exp\left\{\theta^{\mathbf{t}}s\left(\mathbf{g}',\mathbf{X}\right)\right\}}$$

- So, if $n=6$, then we have approx 1,000,000,000 terms.
- This has lead the field to aim for (very neat) simulation based methods
- What if our networks have at most that (6 nodes)?

We can go back to the good-old-fashion MLE!

Keeping $n \leq 6$ we can

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)
- Pooled estimates is a natural approach for small networks

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)
- Pooled estimates is a natural approach for small networks
- Generate random sequences of ERGMs from fitted models fast (in-memory)

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)
- Pooled estimates is a natural approach for small networks
- Generate random sequences of ERGMs from fitted models fast (in-memory)
- etc.

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)
- Pooled estimates is a natural approach for small networks
- Generate random sequences of ERGMs from fitted models fast (in-memory)
- etc.

This and more has been implemented in the `ergmito` (lifecycle experimental) R package (available at https://github.com/muriteams/ergmito)

Keeping $n \leq 6$ we can

- Compute the likelihood function exactly, and hence use ``simple'' optimization to get MLEs

What's more

- Enough flexibility to, e.g., go for pooled estimates (helps with **power** *and* **degeneracy**)
- Pooled estimates is a natural approach for small networks
- Generate random sequences of ERGMs from fitted models fast (in-memory)
- etc.

This and more has been implemented in the `ergmito` (lifecycle experimental) R package (available at https://github.com/muriteams/ergmito)

(built on top of Statnet's amazing `ergm` (Hunter et al. 2008; Handcock et al. 2018) R package)

Sidetrack...

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

e.g.:

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

Sidetrack...

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

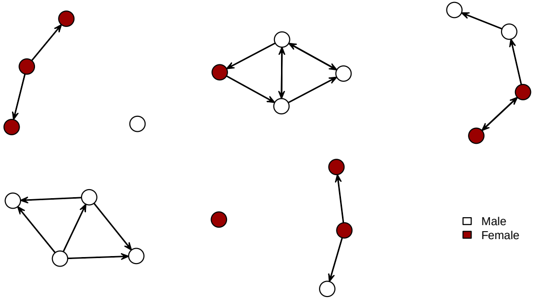
e.g.:

*¡Qué lindo ese perr**ito**! / What a beautiful little dog!*
*¿Me darías 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!

`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
```

How can we fit an ERGMito to this 5 networks?

`ergmito` example (cont'd)

The same as you would do with the `ergm` package

`ergmito` example (cont'd)

The same as you would do with the `ergm` package

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

ergmito example (cont'd)

The same as you would do with the ergm package

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

```r
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
```

`ergmito` example (cont'd)

The same as you would do with the ergm package

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

```r
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
```

By skipping the MCMC part we:

ergmito example (cont'd)

The same as you would do with the ergm package

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

```r
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
```

By skipping the MCMC part we:

- Obtain estimates **faster**, and

ergmito example (cont'd)

The same as you would do with the ergm package

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

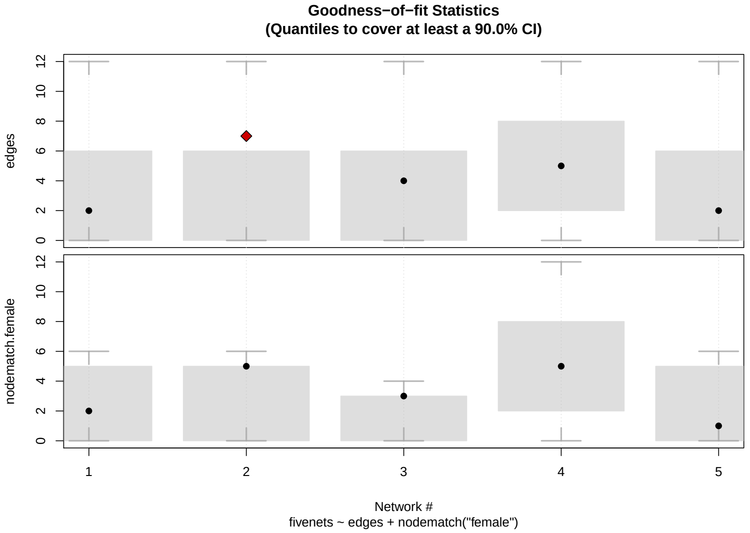
```r
summary(model1) #
```

```
##
## 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
```

By skipping the MCMC part we:

- Obtain estimates **faster**, and
- obtain **more accurate** estimates.

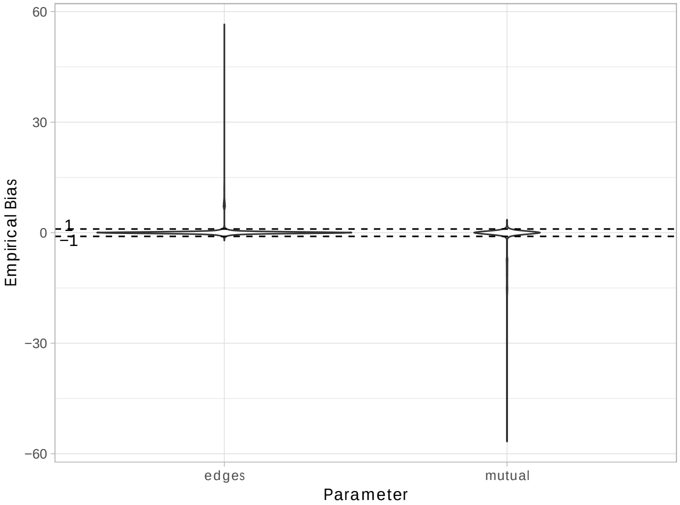
```r
gof1 <- gof_ergmito(model1)
plot(gof1)
```



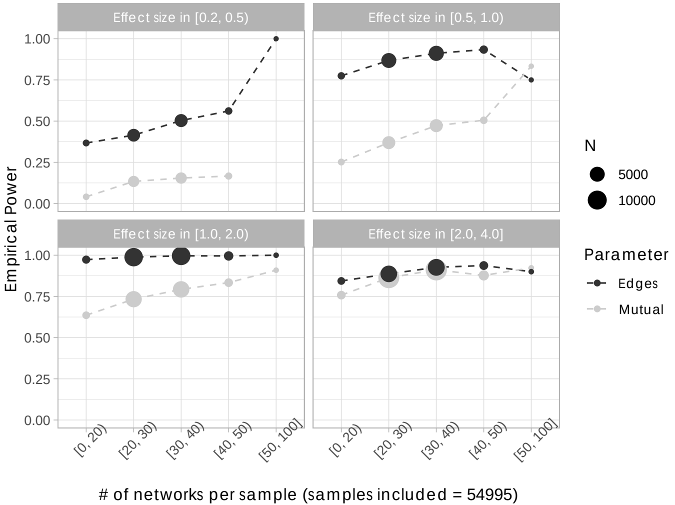
Simulation study

1. Draw parameters for edges and mutual from a uniform(-3, 3).
2. Using those parameters, sampled $n_1 \sim \text{Poisson}(10), n_2 \sim \text{Poisson}(10), n_3 \sim \text{Poisson}(10)$ networks of size 3, 4, and 5 respectively.
3. Estimated the pooled ERGMs using both the MLE and the bootstrap version.

Simulation study: Empirical Bias



Simulation study: Power



Preliminary results

	Advice	Dislike	Influence	Leader	Trust
edges	-0.85^{***}	-2.30^{***}	-0.77^{***}	-0.53^{***}	-0.47^{***}
	(0.17)	(0.20)	(0.13)	(0.14)	(0.14)
ttriple	0.24^{***}		0.21^{**}		0.20^{***}
	(0.06)		(0.08)		(0.06)
nodeicov.RME	0.40^{***}		0.21^{*}	0.42^{***}	0.25^{**}
	(0.09)		(0.09)	(0.11)	(0.09)
nodeocov.Female	0.53^{**}				
	(0.18)				
nodematch.Female		0.56^{*}			
		(0.27)			
nodeicov.SI3Fac1		-0.35^{*}			
		(0.15)			
nodeicov.Female				-0.52^{**}	
				(0.20)	
nodeocov.RME				-0.32^{**}	
				(0.11)	
nodeocov.SI3Fac1					0.31^{***}
					(0.09)
AIC	695.07	381.72	756.84	637.01	776.82
BIC	712.13	394.52	769.92	654.07	794.25
Log Likelihood	-343.54	-187.86	-375.42	-314.50	-384.41
Num. networks	38	38	41	38	41
Convergence	0	0	0	0	0

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

Table 1: Selected models for each one of the studied networks. Results presented here correspond to a forward selection process.

Discussion

- This is not new.

Discussion

- This is not new. What's new is the set of tools to apply it

Discussion

- This is not new. What's new is the set of tools to apply it
- Taking this approach we can improve our estimates (power) and help with degeneracy (all by pooling)

Discussion

- This is not new. What's new is the set of tools to apply it
- Taking this approach we can improve our estimates (power) and help with degeneracy (all by pooling)
- The tool is working (that's what the simulation say)

Discussion

- This is not new. What's new is the set of tools to apply it
- Taking this approach we can improve our estimates (power) and help with degeneracy (all by pooling)
- The tool is working (that's what the simulation say)
- Need to conduct more simulations using nodal attributes

Discussion

- This is not new. What's new is the set of tools to apply it
- Taking this approach we can improve our estimates (power) and help with degeneracy (all by pooling)
- The tool is working (that's what the simulation say)
- Need to conduct more simulations using nodal attributes
- What about goodness-of-fit? Still need to better think about it

Discussion (contd')

The simplicity of the estimation procedure allows us to think of:

Discussion (contd')

The simplicity of the estimation procedure allows us to think of:

- Separable Temporal ERGMitos, a.k.a. TERGMitos

Discussion (contd')

The simplicity of the estimation procedure allows us to think of:

- Separable Temporal ERGMitos, a.k.a. TERGMitos
- Mixture models and Bayesian inference (if you are into that kind of stuff)

Discussion (contd')

The simplicity of the estimation procedure allows us to think of:

- Separable Temporal ERGMitos, a.k.a. TERGMitos
- Mixture models and Bayesian inference (if you are into that kind of stuff)
- More flexible formulas (e.g. interactions between terms)

Discussion (contd')

The simplicity of the estimation procedure allows us to think of:

- Separable Temporal ERGMitos, a.k.a. TERGMitos
- Mixture models and Bayesian inference (if you are into that kind of stuff)
- More flexible formulas (e.g. interactions between terms)
- Better odds ratios (not simply exponentiating the coefficients)

Thanks!

Exact Statistics and Semi-Parametric Tests for Small Network Data

George G. Vega Yon, MS Andrew Slaughter, PhD Kayla de la Haye, PhD

vegayon@usc.edu

https://ggvy.cl

gvegayon gvegayon

Pseudo-bonus track

We still care about the relation between collective intelligence and performance.

Pseudo-bonus track

We still care about the relation between collective intelligence and performance. How can we assess this?

Pseudo-bonus track

We still care about the relation between collective intelligence and performance. How can we assess this?

- GLMs and alike

Pseudo-bonus track

We still care about the relation between collective intelligence and performance. How can we assess this?

- GLMs and alike
- Permutation tests

Pseudo-bonus track

We still care about the relation between collective intelligence and performance. How can we assess this?

- GLMs and alike
- Permutation tests
- Use fitted ERGM to our benefit, e.g. sample from ERGM distribution to do a simulation based test.

Pseudo-bonus track

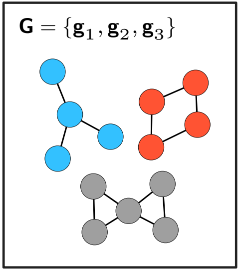
We still care about the relation between collective intelligence and performance. How can we assess this?

- GLMs and alike
- Permutation tests
- Use fitted ERGM to our benefit, e.g. sample from ERGM distribution to do a simulation based test.

Pseudo-bonus track (contd')

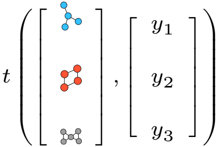
(sorry the heavy notation, but) Suppose that we have a 3 networks of sizes 4, 4, and 5 respectively.

Step 1:
Fit the ERGMito



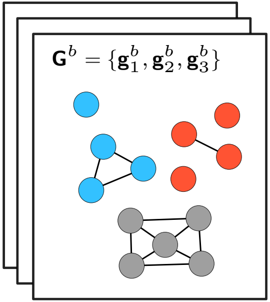
Fit the ERGMito,
This will give us $\mathcal{D}(\hat{\theta}, X_j)$

Step 2:
Calculate $t_0 =$



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

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



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

We can use the distribution of the sequence $\{t_1, \dots, t_B\}$ as null to compare against t_0

Thanks (again)!

Exact Statistics and Semi-Parametric Tests for Small Network Data

George G. Vega Yon, MS Andrew Slaughter, PhD Kayla de la Haye, PhD

vegayon@usc.edu

https://ggvy.cl

gvegayon gvegayon

References

Handcock, Mark S. 2003. ``Assessing Degeneracy in Statistical Models of Social Networks.'' Working Paper No. 39 76 (39): 33--50. https://doi.org/10.1.1.81.5086.

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.

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.

Kim, Young Ji, David Engel, Anita Williams Woolley, Jeffrey Yu-Ting Lin, Naomi McArthur, and Thomas W. Malone. 2017. ``What Makes a Strong Team?: Using Collective Intelligence to Predict Team Performance in League of Legends.'' In Proceedings of the 2017 Acm Conference on Computer Supported Cooperative Work and Social Computing, 2316--29. CSCW '17. New York, NY, USA: ACM. https://doi.org/10.1145/2998181.2998185.