NME 2024

Network Modeling for Epidemics

NETWORK PARAMETERIZATION

Martina Morris, Ph.D.
Steven M. Goodreau, Ph.D.
Samuel M. Jenness, Ph.D.

UNIVERSITY OF WASHINGTON 1861

EMORY

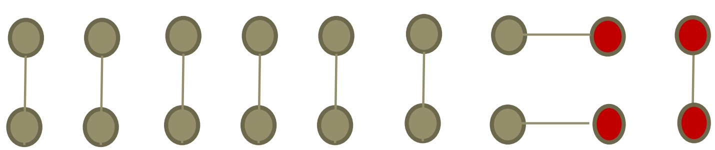
Supported by the US National Institutes of Health

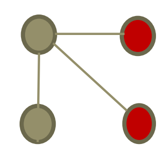
Module 6 Outline

- Model specification and parameterization issues
  - Balance
  - Model specification and degrees of freedom
- Practice with selecting terms and calculating target statistics

Balance

- The idea that the **number** of contacts group A has with group B must equal the **number** that group B has with group A
- Does **not** necessarily mean that the **proportion** of group A's contacts that are with group B equals the **proportion** of group B's contacts that are with group A
- For example, in Seattle the proportion of Black persons' ties that are with White persons is much higher that the proportion of White persons' ties that are with Black persons. Why?





Balance: network models

- E.g. if you are building a purely heterosexual model
  - In the real world, in any population:

    # of relationships/acts that females have with males =

    # of relationships/acts males have with females

  - But this may not be exactly true in **egocentric** data
    - (Random) sampling error
    - Bias (sex ratio of sample does not equal empirical sex ratio, female sex workers are under-sampled)
    - Misreporting (e.g. females may under-report)

  - Nevertheless, one needs to be explicit about balance in the target statistics

Balance in STERGMs

- Occurs initially in the construction of the target stats, and must involve explicitly thinking about data sources. E.g.:
  - Number of ties (pop size * mean degree with other group) must always balance between two contacting groups
  - Imagine a purely heterosexual population (and sample) that both have a 1:1 sex ratio
  - Equal pop sizes implies that mean degree must be equal
  - Males report mean degree of 0.74, females report 0.68
  - You must choose whether to use 0.68, 0.74, 0.71, or something else when calculating target stat

Balance in STERGMs

- Note: the statnet package `ergm.ego` exists to handle much of this for you.
- We are not teaching it here, as it's worth making sure you understand the issues
- But could be useful for you in the future

Balance in STERGMs

- Note: once estimation is done, and simulation begins then balance will happen automatically forever, even when we introduce vital dynamics
- This is because the target stats have been converted into parameters based in log-odds
- This is true no matter the nature of complexity of the nodal dynamics

Balance in STERGMs

- Quick quiz:
  - Purely heterosexual population
  - Females have mean sex partner degree of 0.8
  - Males must have mean sex partner degree of:

    A. 0.8

    B. 0.4

    C. 0.89

    D. Not enough information

Balance in STERGMs

- Quick quiz:
  - Purely heterosexual population
  - Females have mean sex partner degree of 0.8
  - There are 200 females and 180 males
  - Males must have mean sex partner degree of:

    A. 0.8

    B. 0.4

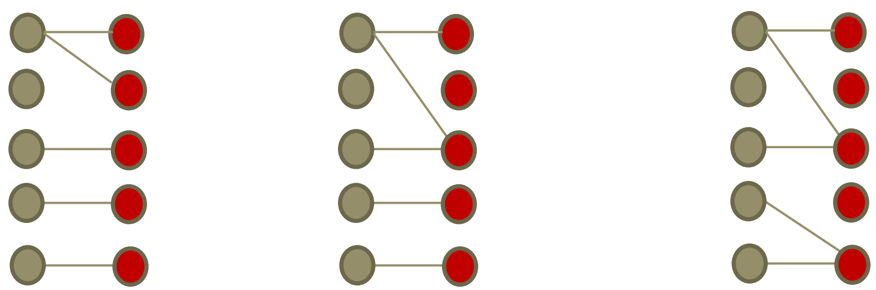
    C. 0.89

    D. Not enough information

$$= \frac{200 \times 0.8}{180}$$

Balance

- Balance only applies to numbers of **ties**.
- It can be easy to mistakenly over-apply the concept of balance.
- For instance, imagine a model that considers relational concurrency in heterosexual relationships.
  - Assume equal sex ratio
  - Assume 20% of men report having concurrent partnerships
  - What does that tell us about the % of women having concurrent partnerships?
  - Nothing!



Degrees of freedom

- You can only use as many terms/target stats as you have degrees of freedom
- Can be tricky to identify
- E.g. heterosexual degree distributions
  - You are estimating a model on a network containing 250 females and 250 males
  - You have already included an edges term with target stat 165
  - You have included a constraint that nobody can have more than 3 edges at one time
  - How many more sex–specific degree terms/target stats can you add before your model is fully specified?

Degrees of freedom

Deg	M	F
0	M_0	F_0
1	M_1	F_1
2	M_2	F_2
3	M_3	F_3

Additional constraints:

- 250 males total
- 250 females total
- 165 edges total

- Given that nobody can have degree >3, there are 8 cells that can be filled in.

Degrees of freedom

Deg	M	F
0	M_0	F_0
1	M_1	F_1
2	M_2	F_2
3	M_3	F_3
Total N	250	250
Total Pships	165	165

Additional constraints:

- 250 males total
- 250 females total
- 165 edges total

- Given that nobody can have degree >3, there are 8 cells that can be filled in.

Degrees of freedom

Deg	M	F
0	M_0	F_0
1	M_1	F_1
2	M_2	F_2
3	M_3	F_3
Total N	250	250
Total Pships	165	165

Deg	M	F
0	120	92
1	104	153
2	17	3
3	9	2
Total N	250	250
Total Pships	165	165

- Given that nobody can have degree >3, there are 8 cells that can be filled in.
  - $M_0 + M_1 + M_2 + M_3 = 250$ $\quad$ $M_0 + 104 + 17 + 9 = 250$
  - $F_0 + F_1 + F_2 + F_3 = 250$ $\quad$ $F_0 + 153 + 3 + 2 = 250$
  - $M_1 + 2M_2 + 3M_3 = 165$ $\quad$ $M_1 + 2(17) + 3(9) = 165$
  - $F_1 + 2F_2 + 3F_3 = 165$ $\quad$ $F_1 + 2(3) + 3(2) = 165$

- So users can specify **at most** 2 male degree terms and 2 female degree terms

Practice

- You have a sample of 20 heterosexuals
- They live in two communities
- You have extracted their partnerships on the day of the interview
- You want to simulate an artificial population of size 2,000
- You want to include in your model mixing by community as well as sex-specific degree distributions
- You notice that nobody has more than two ongoing ties
- Relationships average 60 time steps
- How do you set up your network? What model terms and target stats will you specify?

Egocentric data

Ongoing partnerships by sex and community of ego and alters

Ego	Partner 1	Partner 2
F1	M1	
F1		
F1	M1	M1
F1	M1	
F2	M2	
F2	M1	
F2	M2	
F2		
F2		
F2	M1	

Ego	Partner 1	Partner 2
M1	F1	F1
M1		
M1		
M1	F2	
M2	F2	
M2		
M2	F2	F1
M2	F1	
M2	F2	F2
M2		

Set up network

- Note: you got lucky!
  - Sample has same # of males and females, and same community breakdown for each
  - Just need to scale up to 2,000

```
library(EpiModel)
mynet <- network_initialize(2000)

sex <- c(rep(1, 1000), rep(2, 1000))
mynet <- set_vertex_attribute(mynet, 'group', sex)

cmty <- c(rep(1,400), rep(2,600), rep(1,400), rep(2,600))
mynet <- set_vertex_attribute(mynet, 'cmty', cmty)

table(get_vertex_attribute(mynet, "group"),
      get_vertex_attribute(mynet, "cmty"))
```

Establish terms and target stats

- Term for overall relational effect
  - `~edges`
  - Have to reconcile that male mean deg = 0.9 and female mean deg = 0.8, and sex ratio in sample is equal
  - Could:
    1. assume a different sex ratio in population
    2. assume males are over-reporting (or sample is biased towards more active males)
    3. assume females are under-reporting (or sample is biased towards less active females)
  - We'll assume some of 2&3
  - Target stat = 850 = (2000 * 0.85 / 2)

Establish terms and target stats

- Mean degree by community
  - Mean deg for community 1 = 7/8 = 0.875
  - Mean deg for community 2 = 10/12 = 0.833
  - Worth modeling this difference?
  - Could put in a nodefactor term into the ergm and see whether it is significant
  - Foreshadowing: it's not, so we'll just ignore

- Mixing by community:
  - Proportion of ties that are within community = 12/17 = 0.706
  - Term: `~nodematch('cmty', diff= FALSE)`
  - Target stat = 0.706*850 = 600

Establish terms and target stats

- Let's first add a constraint that nobody has >2 partnerships at a time
  - term = `degrange(from=3)`
  - target stat = 0
- Then add degree terms = `~degree(1, by='group')`
- Why only 1 term per sex?
  - See earlier slides in this session
- Target stats gets very tricky, since the mean degree was not the same by sex
  - How to adjust degree distribution for each sex to match the new degree distribution?
  - You **must** make assumptions
  - Observed degree dist =

Deg	F	M
0	0.30	0.40
1	0.60	0.30
2	0.10	0.30
Mean deg	0.80	0.90

Let's assume that all movement is between 1&2

Deg	F	M
0	0.30	0.40
1	0.55	0.35
2	0.15	0.25
Mean deg	0.85	0.85

  - target stats = c(550, 350) = c(0.55*1000, 0.35*1000)

Estimating and diagnosing

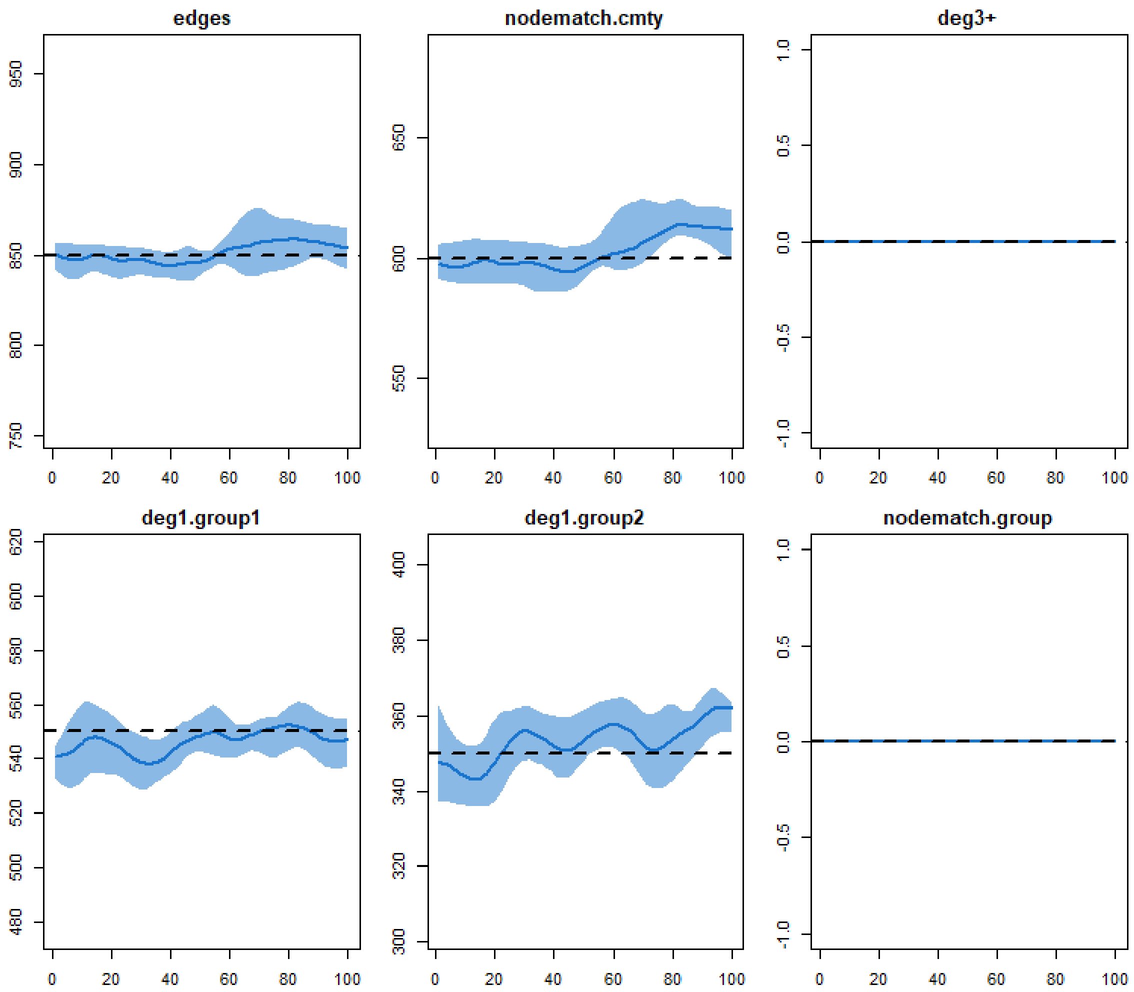
```
formation <- ~edges+nodematch('cmty', diff = FALSE) + degrange(from = 3) +
                  degree(1, by = 'group') + nodematch('group', diff = FALSE)

target.stats <- c(850, 600, 0, 550, 350, 0)

myfit <- netest(mynet,
            formation=formation,
            target.stats = target.stats,
            coef.diss = dissolution_coefs(~offset(edges), 60))

mydx <- netdx(myfit, nsims=10, nsteps=100)
mydx
get_nwstats(mydx)
plot(mydx)
```

Estimating and diagnosing



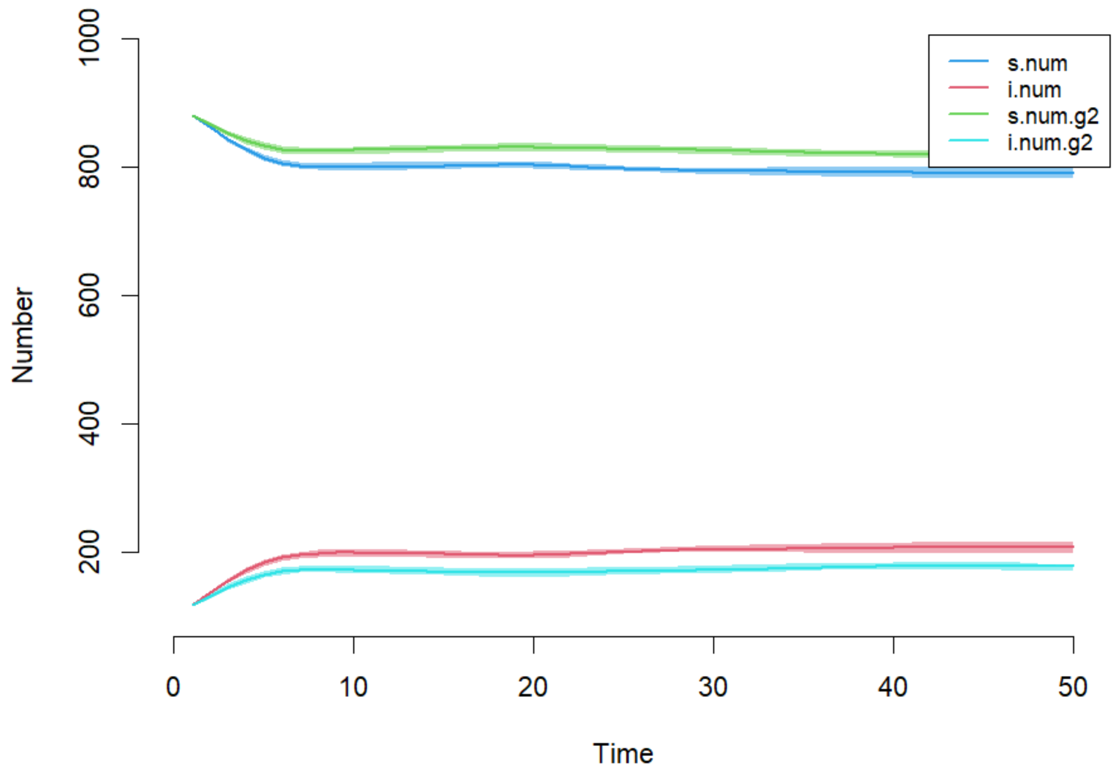
Disease simulation

- Let's do a disease simulation on top of it just for fun!

```
mycontrol <- control.net("SIS", nsteps = 50, nsims = 5,
               nwstats.formula = ~edges + nodematch('cmty') +
               degree(0:5, by = 'group'), verbose = TRUE)
myinit <- init.net(i.num = 100, i.num.g2 = 100)
myparam <-param.net(inf.prob = 0.6, inf.prob.g2 = 0.6,
                    act.rate = 1.8,
                    rec.rate = 0.1, rec.rate.g2 = 0.1)

mySIS <- netsim(myfit, param = myparam, control = mycontrol,
                init = myinit)
plot(mySIS)
```

Disease simulation



Examining target stats

```
get_nwstats(mySIS)
plot(mySIS, type = "formation", sim.lines = TRUE)
```

Examining target stats