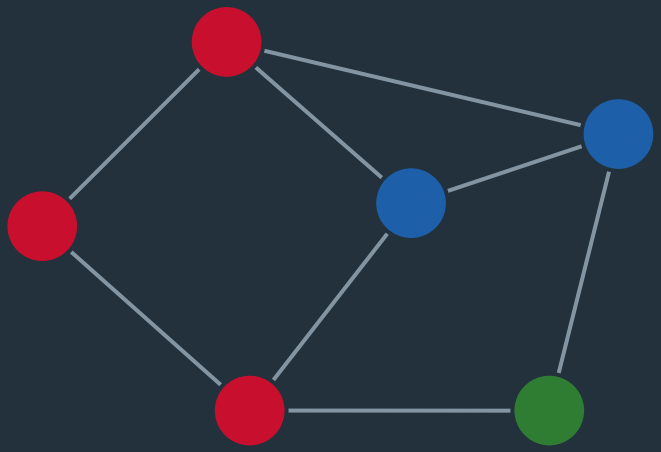
ERGM III: simulation, degeneracy and missing data



Course B, Week 7

Chapter 22 · a lecture and a lab

About 90 minutes each

Instructors are free to use, adapt and share these slides.

Today: a lecture, a break and a lab

0:00	1	**Simulating from a model: what it implies**	22 min
0:22	2	**Degeneracy: why gwesp and not triangle**	16 min
0:38	3	**Missing data and the options beyond the workflow**	14 min
0:52		Break	10 min
1:02	4	**Lab: simulate and change a coefficient**	40 min
1:42	5	**Lab: Faux Mesa High, a raw triangle, missing dyads**	45 min

Weeks 5 and 6 gave a model that is useful and incomplete

Week 5: reading a coefficient

The change in the conditional log-odds of one tie, given the rest of the network. The model is built in stages and the MCMC chain is checked after every fit.

Week 6: a verdict on fit

Convergence is not a verdict. The final advice model reproduces density and reciprocity, and misses the out-degree spread and the pairs at distance 2.

Week 7: what remains

What networks does the model produce? Does it carry over to an undirected network? What breaks, and what do holes in the data do?

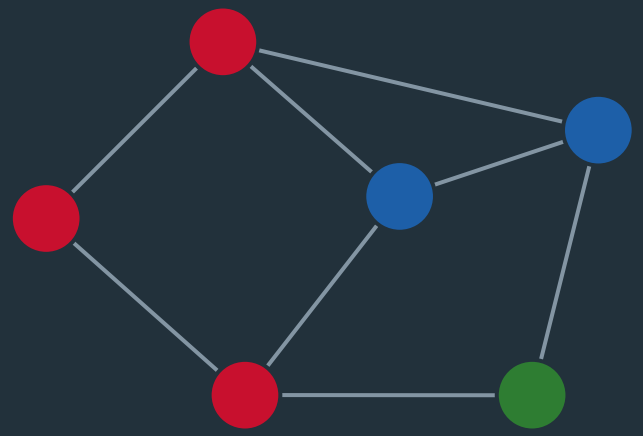
```
. quietly nwergm lazega_adv, edges mutual ///
>         nodematch(office practice) ///
>         nodeofactor(status) nodeifactor(status) ///
>         gwesp(.5) seed(12345)
```

The final advice model of Chapter 21 is the model we simulate from.

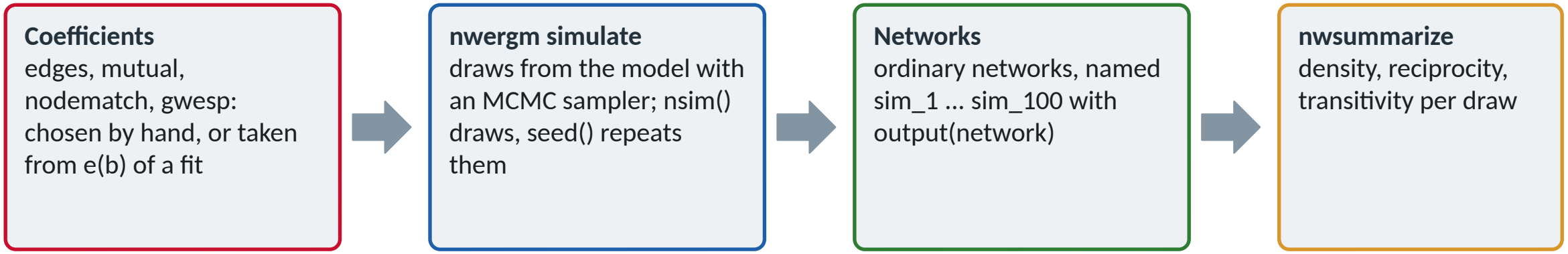
1

Simulating from a model: what it implies

22 min · Chapter 22, Section 22.2: nwergm simulate, coef(), simsummary



Simulation draws networks from a fitted ERGM



- With coefficients chosen by hand, the truth is known: fit the model to a draw and see what comes back.
- With the coefficients of a fit, the draws show what the fitted model implies, and where the data depart from it.
- Successive draws of the sampler are not independent; burnin(), interval() and nosequential control that.

Hand-chosen coefficients: 100 nodes, four groups, four terms

```
. clear
. nwclear
. set obs 100
. generate grp = ceil(_n / 25)
. nwergm simulate 100, edges mutual nodematch(grp) ///
>         gwesp(.5) directed theta(-4 1.5 1.2 .6) ///
>         seed(12345) generate(art)
. nwsummarize art, detail
   Arcs: 574   Density: .058
   Reciprocity: .193   Transitivity: .153
```

Term	Coefficient
edges	-4
mutual	1.5
nodematch(grp)	1.2
gwesp(.5)	0.6

theta() takes one value per term, in the order nwergm processes them: edges and mutual, then node attributes, then structural terms.

The draw is an ordinary network; nothing in it says where it came from.

Fit the same model to the draw, and the coefficients come back

```
. nwergm art, edges mutual nodematch(grp) ///
>         gwesp(.5) seed(12345)

MCMLE converged after 6 iteration(s).
```

Term	Chosen	Estimate	Std. err.
edges	-4	-4.106	.071
mutual	1.5	1.516	.165
nodematch_grp	1.2	1.180	.082
gwesp_.5	0.6	.558	.054

4 of 4

estimates within two standard errors of the coefficient that generated the network

One fit to one draw is a demonstration, not a test of the estimator.

What recovery says, and what it does not

It says

from this one network, generated by known coefficients, the fit comes back close to them: -4.11, 1.52, 1.18 and 0.56 for -4, 1.5, 1.2 and 0.6.

It does not say

that the estimator recovers the coefficients reliably; that needs many draws (Ch. 22, Ex. 3). Nor that the coefficients fitted to a real network are the true ones: recovery works because the model generated the network.

Left out is pushed in

A model that leaves out a tendency pushes its effect into the terms that are included.

Many draws, many fits

Centered on the true values? As spread as the standard errors say? That is Ex. 3, for home.

From the model fitted to the firm: its coefficients are the dials

```
. nwwebuse lazega, nwclear
. quietly nwergm lazega_adv, edges mutual ///
>     nodematch(office practice) ///
>     nodeofactor(status) nodeifactor(status) ///
>     gwesp(.5) seed(12345)
. matrix b = e(b)
. matlist b', twidth(20)
```

```
                edges |  -4.662469
               mutual |   1.031625
    nodematch_office |   1.016895
  nodematch_practice |   .7571778
nodeofactor_status_2 |   .8242996
nodeifactor_status_2 |   .4019506
            gwesp_.5 |   .5597618
```

No nodes, no terms

After a fit, nwergm simulate takes both from the fit, and the coefficients from coef(b); without coef() it uses e(b), the same matrix.

Why save b

A copy of b with one element changed runs the model with that coefficient changed and everything else at its fitted value.

The draws match density and reciprocity, not clustering

```
program simsummary
    quietly nwsummarize `1'_*, detail save(simstats)
    preserve
    use simstats, clear
    summarize reciprocity density transitivity
    restore
end
```

```
. nwergm simulate, coef(b) nsim(100) ///
>         output(network) seed(1)
. simsummary sim
```

	Observed	100 draws, mean	Range
Reciprocity	.274	.272	.226 to .310
Density	.179	.180	.170 to .194
Transitivity	**.329**	**.264**	**.243 to .293**

Not one draw reaches the firm's .329. The model understates its clustering, even below the .272 that the degrees alone give.

Setting mutual to zero lowers the density, too

```
. matrix b0 = b
. matrix b0[1, colnumb(b0, "mutual")] = 0
. nwergm simulate, coef(b0) nsim(100) ///
>         output(network) generate(nomut) seed(1)
. simsummary nomut
```

Mean of 100 draws	Fitted	Mutual = 0
Reciprocity	.272	.136
Density	.180	**.144**
Transitivity	.264	**.223**

Not a dial

The edges coefficient never changed. A positive mutual coefficient raises the log-odds of every arc that would complete a reverse arc, so it also raises the number of arcs. Setting a coefficient to zero removes the term's contribution; it does not remove the pattern.

Raise the edges coefficient to isolate one tendency

```
. matrix b0[1, colnumb(b0, "edges")] = -4.45
. nwergm simulate, coef(b0) nsim(100) ///
>         output(network) generate(nomut2) seed(1)
. simsummary nomut2
```

Mutual = 0, edges = -4.45: density .175, reciprocity .155, not zero. Office, practice and gwesp raise both arcs of a pair, and independent arcs at this density are reciprocal about .10 of the time.

```
. matrix b1 = b
. matrix b1[1, colnumb(b1, "gwesp_.5")] = 0
. nwergm simulate, coef(b1) nsim(100) ///
>         output(network) generate(nogw) seed(1)
. matrix b1[1, colnumb(b1, "edges")] = -3.52
```

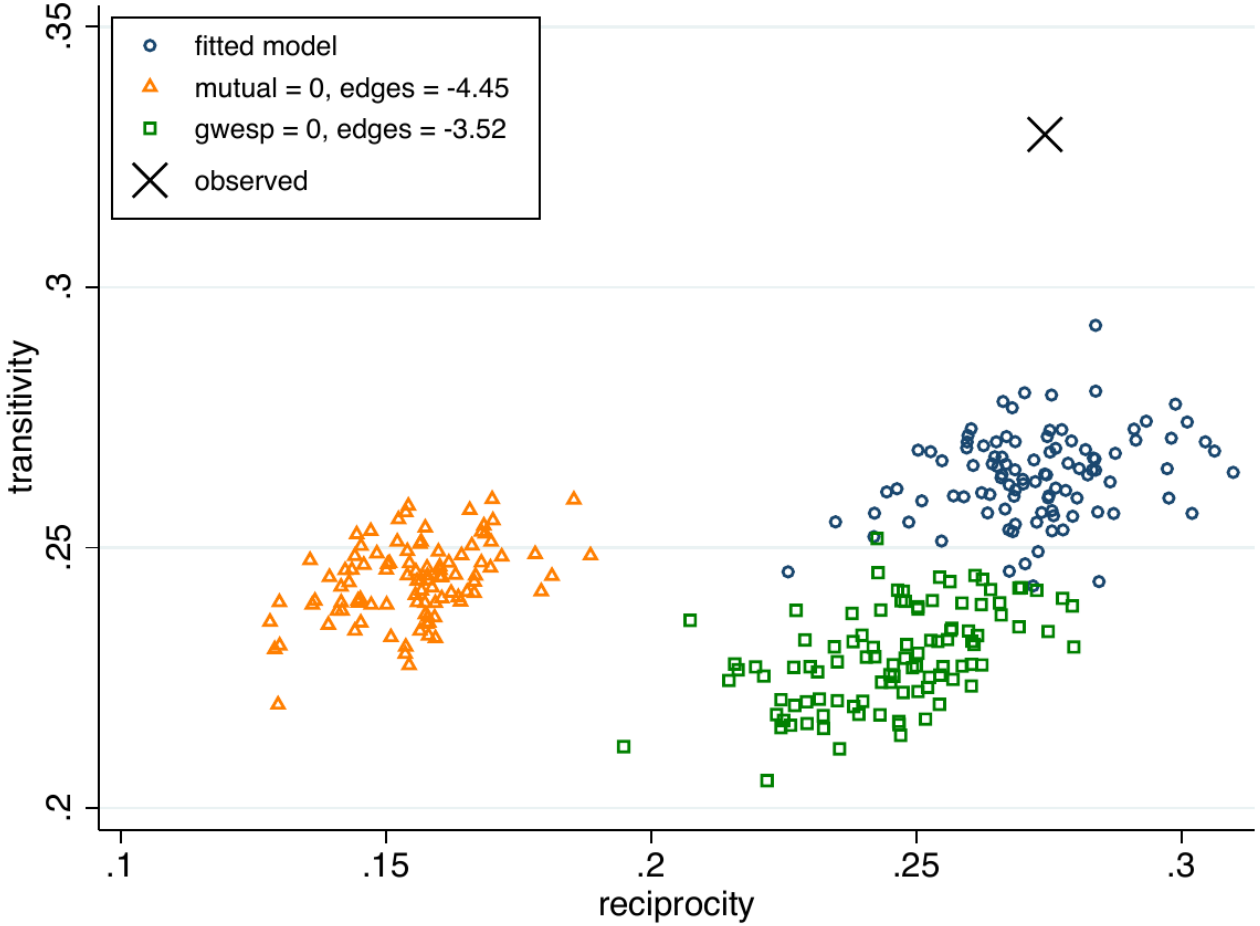
Gwesp = 0 at the fitted edges: density .061, two thirds of the ties gone. With edges = -3.52 the density is back at .179, and transitivity is .229, not the .264 of the fitted model.

Print the density next to every comparison

Statistics of simulated advice networks	Reciprocity	Density	Transitivity
Observed network	.274	.179	.329
Fitted model	.272	.180	.264
Mutual = 0, edges -4.66 (fitted)	.136	**.144**	.223
Mutual = 0, edges -4.45	.155	.175	.244
Gwesp = 0, edges -4.66 (fitted)	.107	**.061**	.085
Gwesp = 0, edges -3.52	.246	.179	.229

To separate a tendency from the number of ties, raise the edges coefficient until the densities match (rows 4 and 6), and print the density next to any comparison of two simulated settings.

Each model forms its own cloud, and none reaches the firm

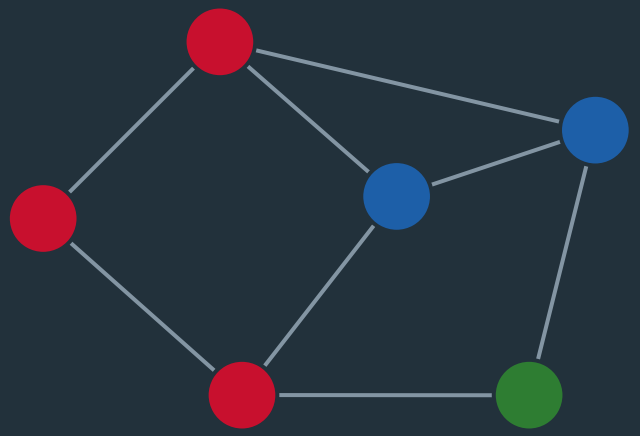


- Without the mutual term the cloud moves left, to less reciprocity, and also down.
- Without gwesp it moves down, to less closure, and a little left.
- None of the clouds reaches the firm, which has more closure than any draw.
- Finding the matching edges coefficient takes trial and error: a change of 0.2 moves the density by about 0.03.

2

Degeneracy: why gwesp and not triangle

16 min · Chapter 22, Sections 22.3 and 22.4: Faux Mesa High, MPLE and MCMLE, triangle, gwesp



Faux Mesa High: homophily on grade and race alone

```
. nwwebuse mesa, nwclear
. nwergm mesa, edges nodematch(grade race) ///
>         seed(12345)

Estimation:            =  MPLE

           mesa | Coefficient  Std. err.
----------------+--------------------
          edges |  -6.225411   .1737316
nodematch_grade |    2.82789   .1773647
 nodematch_race |   .4266748   .1427659
```

205

students, 203 friendships, density .0097

Same grade: odds about 17 times. Same race: about 1.5 times. A model without a closure term cannot tell homophily from closure.

The model is dyad-independent, so the pseudolikelihood is the likelihood. Race is a string and enters as it is.

Add closure, and both homophily coefficients shrink

```
. nwergm mesa, edges nodematch(grade race) ///
>         gwesp(.25) seed(12345)

MCMLE converged after 7 iteration(s).

. estat mcmcdiag
      Statistic |   Autocorr       ESS   Geweke z
          edges |     0.732     179.4     -0.58
nodematch_grade |     0.770     163.2     -0.75
 nodematch_race |     0.705     187.3     -0.55
      gwesp_.25 |     0.750     162.2     -0.79
```

Coefficient	Dyads	+ gwesp	Odds ratio
edges	-6.225	-6.415	
nodematch_grade	2.828	1.997	17 to 7
nodematch_race	.427	.279	1.5 to 1.3
gwesp_.25		1.448	

Homophily and closure compete for the same ties. Students of one grade share many friends, so with closure in the model the grade coefficient is smaller. Diagnostics acceptable: ESS 162 to 187, |Geweke z| < 0.8.

A raw triangle count: the estimation stops

```
. capture noisily nwergm mesa, edges ///
>         nodematch(grade race) triangle seed(12345)
Warning: MCMLE iter 1 - statistic(s) at column(s) 1 2 3 4
not varying this iteration - this may indicate an extreme point
or dead-end configuration.
Unconstrained MCMC sampling did not mix at all.
Optimization cannot continue.
```

The simulated networks sit in a corner where nothing changes. Each closed triad makes the next tie more attractive by the same amount, and the probability piles up near the empty or the complete network.

Why capture noisily

The command fails with return code 498. Under capture noisily the message prints and the do-file continues, so a script with a degenerate model does not stop at the first error.

Gwesp counts each further shared partner less: the weights

Shared partners of a tie	1	2	3	4	5
triangle: added by the k-th partner	1	1	1	1	1
gwesp(.5): added by the k-th partner	1	.39	.15	.06	.02
triangle: total for k partners	1	2	3	4	5
gwesp(.5): total for k partners	1	1.39	1.55	1.61	1.63

Triangle: no ceiling

A tie that closes k triangles adds k times the coefficient to its log-odds. The reward grows with every closed triad.

Gwesp: a ceiling of e^0.5 = 1.65

Each further partner adds 1 - e^-0.5 = 0.39 times the one before. The decay (here 0.5) is a modeling choice.

Degeneracy depends on the network and the model

Network	Model	Result
Faux Mesa High	edges, nodematch(grade race), triangle	**stops (498)**
Lazega advice	edges, ctriple	**stops (498)**
Lazega advice	final model + ctriple	**converges**
Lazega advice	final model + gwesp(.5) + ctriple	**converges**

A raw count that works on one network is no guarantee for the next. The safe default remains the geometrically weighted term.

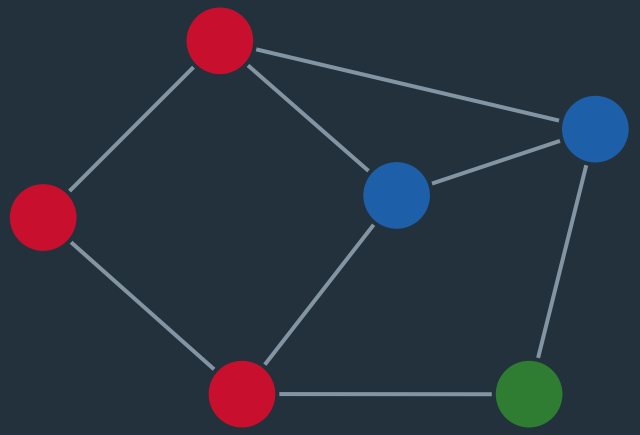
Free decay: a curved family

gwespfree() estimates the decay. Limits: one curved term per model; estimate(cd) and mainmethod(sa) are refused; after MCMLE the likelihood is not evaluated, so lrtest and AIC are unavailable.

3

Missing data and the options beyond the workflow

14 min · Chapter 22, Sections 22.5 and 22.6: missing dyads, nwtomata, constraints and estimation options



A missing dyad is not a non-tie

	1	2	3	4
1	0	.	.	1
2	.	0	1	0
3	.	1	0	0
4	1	0	0	0

Two pairs of students whose friendship nobody recorded.

Coded as 0

Treats the pair as evidence of absence: the pair counts as two students who are not friends.

Left missing

Out of the likelihood. The fit rests on the dyads that were observed, and says nothing about the others.

A dyad nobody reported is not evidence that the tie is absent.

Three holes in Faux Mesa High: found and left out

```
. nwtomata mesa, mat(A)
. mata: A[1,2] = A[2,1] = A[1,3] = A[3,1] = A[10,20] = A[20,10] = .
. mata: st_matrix("M", A)
. nwset, mat(M) undirected name(mesa_na)
. nwergm mesa_na, edges nodematch(grade race) seed(12345)
Note: network mesa_na has 3 missing dyad(s); they are left out of the
      likelihood, as in ergm.
(output omitted)
. display "informative dyads: " e(N) "   missing dyads: " e(nmissing)
informative dyads: 20907   missing dyads: 3
```

20,907

informative dyads: 20,910 minus the three missing ones

The estimates stay close to those of the complete network. Edges -6.2255 against -6.2254, grade homophily 2.8281 against 2.8279.

What nwergm does with holes, and where it stops

Situation	What happens
Missing dyad in the network	left out of the likelihood; found automatically; e(nmissing) counts them
Tie dependence, MCMLE	a second, constrained sample lets only the missing dyads change
method(mple)	missing dyads dropped from the pseudolikelihood
Missing value in a covariate	an error (return code 416): drop or impute first
freedyads(), blockdiag(), fixdensity	cannot be combined with missing dyads
constraints(observed)	refused; use obsconstraints(observed) or obsconstraints(fixallbut(...))
AIC, BIC, likelihood-ratio test	comparable only between fits with the same missing dyads

Valid if the dyads are missing at random. If the most central actors did not respond, it does not hold.

Beyond the main workflow: terms, constraints, options

Option	Effect
`estimate(mple), estimate(cd)`	pseudolikelihood; contrastive divergence, fast and approximate
`mainmethod(sa)`	stochastic approximation instead of MCMLE
`init()`	starting values in the order of the table
`mplecheck(on), mpletype(penalized)`	warn when separation makes the MPLE infinite; Firth penalty
`mplecovariance(godambe)`	sandwich covariance for an MPLE with tie dependence
`mcmcinterval(), mcmcburnin(), mcmcsamplesize(), mcmleeffectivesize()`	the chain, as in Chapter 21

constraints()

edges, degrees, bd(), blocks(), blockdiag(), fixedas(), fixallbut(): restrict the networks the model compares with.

structural()

Further terms of R's ergm in R's own notation: twopath, isolates, cycle, altkstar and more.

What to take away from the lecture

Simulate to see the model

The draws match the firm's density and reciprocity, not its clustering (.264 against .329). A coefficient moves more than its own statistic: match the density.

Gwesp, not triangle

Homophily and closure compete for the same ties. A raw triangle count can make the model degenerate; geometrically weighted terms avoid it.

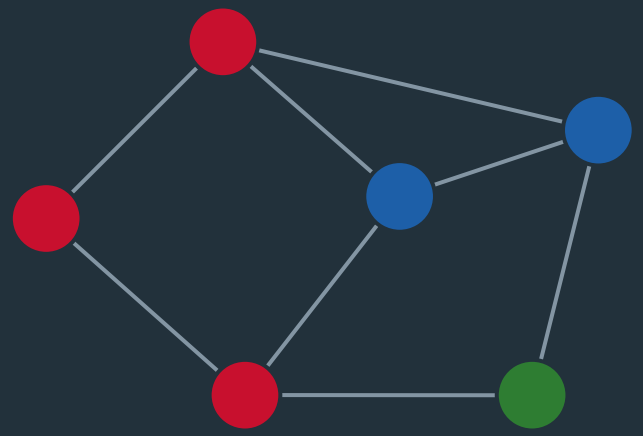
A hole is not a zero

Leave a missing dyad missing; nwergm leaves it out of the likelihood. That assumes the dyads are missing at random.

The next model starts from the same question in time: what changes between waves (Week 8).

Break

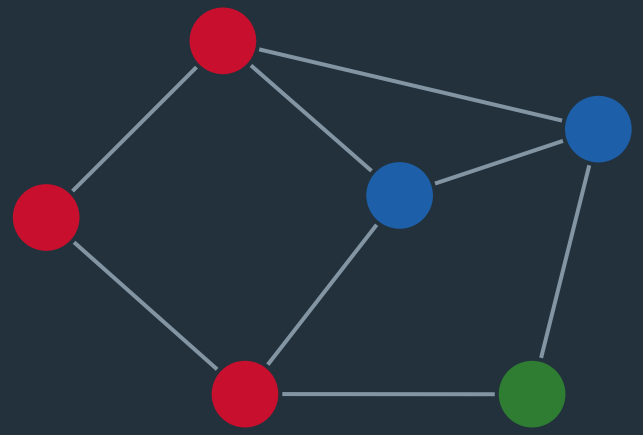
10 minutes



4

Lab: simulate and change a coefficient

40 min · Run along: nwergm simulate, coef(), simsummary; Ch. 22, Ex. 4



Run along 1: draw a network from coefficients chosen by hand

```
. clear
. nwclear
. set obs 100
. generate grp = ceil(_n / 25)
. nwergm simulate 100, edges mutual nodematch(grp) ///
>         gwesp(.5) directed theta(-4 1.5 1.2 .6) ///
>         seed(12345) generate(art)
. nwsummarize art, detail
. nwergm art, edges mutual nodematch(grp) ///
>         gwesp(.5) seed(12345)
```

Predict first. What do you expect for the four estimates?

Checkpoint	Value
arcs in art	574
density, reciprocity	.058, .193
transitivity	.153
edges	-4.105729
mutual	1.515528
nodematch_grp	1.180105
gwesp_.5	.558259

Run along 2: the firm's model, with one coefficient changed

```
. nwwebuse lazega, nwclear
. quietly nwergm lazega_adv, edges mutual ///
>     nodematch(office practice) nodeofactor(status) ///
>     nodeifactor(status) gwesp(.5) seed(12345)
. matrix b = e(b)
. nwergm simulate, coef(b) nsim(100) ///
>         output(network) seed(1)
. simsummary sim
. matrix b0 = b
. matrix b0[1, colnumb(b0, "mutual")] = 0
. nwergm simulate, coef(b0) nsim(100) output(network) ///
>         generate(nomut) seed(1)
. simsummary nomut
```

Mean of draws	Recip.	Dens.	Trans.
fitted (sim)	.272	.180	.264
mutual = 0	.136	.144	.223

Predict first. What happens to the density when mutual is set to zero?

The program simsummary of block 1 must be defined first: capture program drop simsummary, then the program.

Hands-on (20 min)

Ch. 22, Ex. 4

Simulate 100 networks from the final advice model with the gwesp coefficient set to 0.3, between the fitted 0.56 and the 0 of the table. Use a copy of b, raise the edges coefficient until the mean density of the draws is near 0.18, and report the edges coefficient you chose and the mean reciprocity, density and transitivity of the draws (simsummary). Put the result next to the rows of the table: how does transitivity change as the gwesp coefficient falls from 0.56 to 0.3 and to 0 at about the same density?

Hint matrix b2 = b, then matrix b2[1, colnumb(b2, "gwesp_.5")] = .3 and the same for "edges". Between -4.66 (fitted, gwesp 0.56) and -3.52 (gwesp 0), try a value, and use nsim(30) while searching to save time.

Ch. 22, Ex. 4: the answer

gwesp coef.	edges coef.	Reciprocity	Density	Transitivity
0.56 (fitted)	-4.66	.272	.180	.264
0.3	-4.14	.262	.180	.250
0 (matched)	-3.52	.246	.179	.229

Transitivity falls with the gwesp coefficient at a matched density, and the observed .329 stays out of reach in every row.

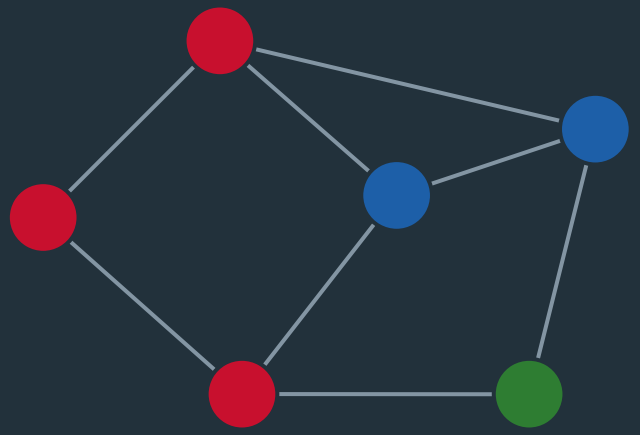
The search

Density at nsim(30), gwesp 0.3: edges -4.3: .154 · -4.2: .171 · -4.1: .187 · -4.0: .204. The hint's starting point, a third of the way (-4.28), gives .15.

5

Lab: Faux Mesa High, a raw triangle, missing dyads

45 min · Run along: nwergm on mesa, triangle, nwtomata; Ch. 22, Ex. 1, 2 and 6



Run along 3: homophily, closure, a raw triangle

```
. nwwebuse mesa, nwclear
. nwergm mesa, edges nodematch(grade race) seed(12345)
. nwergm mesa, edges nodematch(grade race) ///
>         gwesp(.25) seed(12345)
. estat mcmcdiag
. capture noisily nwergm mesa, edges ///
>         nodematch(grade race) triangle seed(12345)
```

Predict first. What happens to the grade coefficient when gwesp(.25) is added?

Checkpoint	Value
edges, dyads only	-6.225411
nodematch_grade	2.82789
edges, with gwesp	-6.414667
nodematch_grade	1.997106
gwesp_.25	1.448323
MCMLE iterations	7
triangle model	stops, rc 498

Hands-on (10 min)

Ch. 22, Ex. 1

Faux Mesa High (mesa) is undirected, and nwergm mesa, edges mutual is refused. Explain why a mutual term, which counts reciprocated pairs, has no meaning in an undirected network, while triangle and nodematch() still do.

Hint Start from the dyad census of Chapter 6: in a directed network a pair is null, asymmetric or mutual. Which of these states can a pair take in an undirected network, and what would a count of mutual pairs then equal?

Ch. 22, Ex. 2

Explain why, with a positive coefficient, a raw triangle count can put nearly all of the model's probability on almost empty or almost complete networks, and why gwesp does not.

Hint Under triangle, a tie that closes k triangles adds k times the coefficient to its log-odds; ask what happens to k, and to the next tie's log-odds, as ties are added one by one. Under gwesp(.5), each further shared partner of a tie adds only 1 - e^-0.5 = 0.39 times the weight of the one before; ask what that sum approaches.

Ch. 22, Ex. 1 and 2: the answers

Ex. 1: no direction, no mutual

An undirected pair is null or connected; the asymmetric state does not exist. The count of mutual pairs would equal the count of edges. nwergm refuses it with a message and return code 198. A triangle (three connected nodes) and nodematch() (a tie between like nodes) need no direction.

Ex. 2: a ceiling for gwesp

Under triangle the reward k times the coefficient grows with every closed triad; the next tie is always more attractive and the probability runs to an extreme. Under gwesp(.5) the weights are 1, .39, .15, .06 …: the total stays below e^0.5 = 1.65.

```
. nwergm mesa, edges mutual
option mutual requires a directed network; mesa is undirected.
```

Hands-on (15 min)

Ch. 22, Ex. 6

Set ten dyads of Faux Mesa High to missing, those between student 1 and students 2 to 11: after nwtomata mesa, mat(A), type mata: A[1,2..11] = J(1,10,.) and mata: A[2..11,1] = J(10,1,.), then build mesa_na as in the lecture. Fit edges nodematch(grade race) gwesp(.25) with seed(12345) to mesa and to mesa_na. How many informative dyads does the second fit use, and how much do the four coefficients change compared with the complete network?

Hint e(N) and e(nmissing) give the counts; the complete network has 205 x 204 / 2 = 20,910 dyads. Store both fits and compare their coefficients with estimates table.

Ch. 22, Ex. 6: ten holes, small changes, wider errors

	Complete	Ten dyads missing	Std. err., complete	Std. err., missing
edges	-6.4147	-6.3938	.1545	.1651
nodematch_grade	1.9971	1.9775	.1649	.1752
nodematch_race	.2789	.2731	.1200	.1225
gwesp_.25	1.4483	1.4522	.1082	.1213

20,900

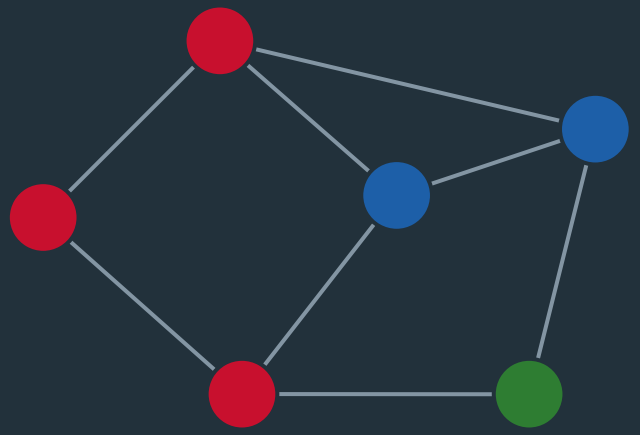
informative dyads in the second fit (e(N));
e(nmissing) = 10

Coefficients move in the second decimal, standard errors grow. The fit assumes the dyads are missing at random, and its AIC cannot be compared with the complete network's.

6

Optional: hidden groups and hidden positions

10 to 15 min, not in the timetable · Chapter 23: nwsbm and nwlsm; Ex. 2 and 3



Two models, one idea: ties depend on a hidden attribute

Block model: nwsbm

Each lawyer is in one of Q blocks; the chance of a tie depends on the blocks of the two lawyers. The number of blocks comes from the integrated classification likelihood (ICL).

Latent space: nwlsm

Each lawyer has a position in a plane; the chance of a tie falls with the distance between the two. Closure comes from geometry, and the fit is a Bayesian MCMC run: check the chains.

What both need you to watch

How much structure to allow. Block numbers and axes are arbitrary. No term stands for reciprocity or closure: for named configurations, use the ERGM.

```
. nwwebuse lazega, nwclear
. nwsbm lazega_adv, maxblocks(6) seed(12345) generate(block) replace
```

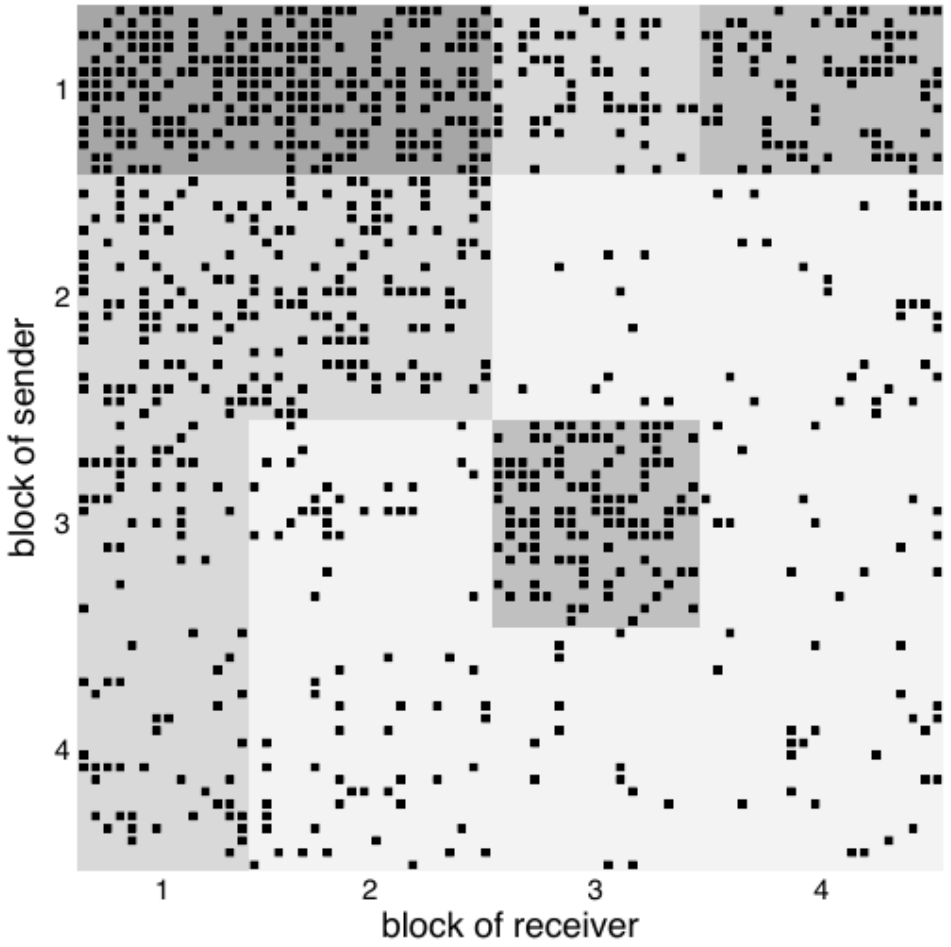
The same questions as Chapters 9 and 10, with a likelihood behind the answers: how many groups, who is in them, and does a picture reproduce the pattern?

Four blocks in the advice network

Block	Lawyers	Partners	Office
1	14	12	12 of 14 Boston
2	20	12	all Boston
3	17	12	all Hartford
4	20	0	16 of 20 Boston

ICL: four blocks, but three to five are close (-2161.6, -2158.5, -2165.2). Report the whole table.

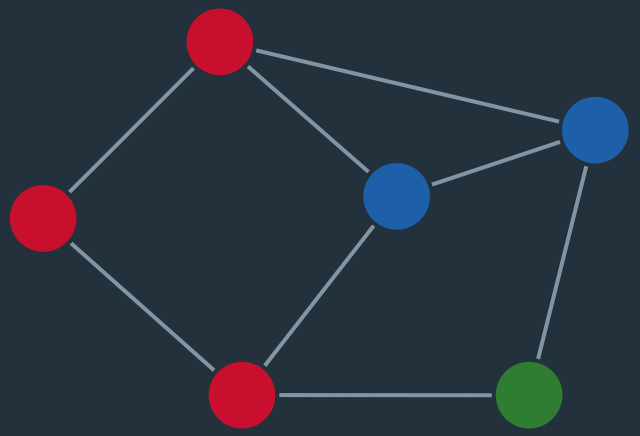
```
. nwsbmplot, type(matrix) sort shade
```



7

Optional: an attribute as the outcome (ALAAM)

10 to 15 min, not in the timetable · Chapter 24, Section 24.4: nwalaam; Ex. 4 and 5



ALAAM: the ERGM turned around

The outcome is random

A binary attribute of the actors (smokes, does not) on a network that is held fixed. The ERGM does the reverse: the network is random, the attributes are fixed.

Terms count configurations

density: actors with the outcome. contagion: ties between two actors who both have it. sender, receiver: ties sent and received by actors with it. ego(x): the actor's own covariate.

Read it as an odds ratio

A coefficient is a change in the log odds that an actor has the outcome, given all the others. exp(contagion) is the odds ratio for one more tie to an actor with the outcome.

```
. generate byte smoker1 = (smoke1 > 1)
. nwalaam smoker1 glasgow1, effects(density contagion sender receiver) seed(12345)
```

Pseudolikelihood is fast but can understate the uncertainty: add the option mple to see it.

Glasgow smoking: contagion in one wave

Fit	contagion	std. err.	reading
pseudolikelihood (mple)	.453	.199	standard error likely too small
stochastic approximation	.743	.286	odds ratio 2.10
+ ego(alcohol), ego(sport)	.410	.354	alcohol 1.84 (.70)
method(ee)	.816	.298	close; EE is biased

An association, not influence

One wave cannot separate friends influencing each other from girls choosing friends who smoke. That takes waves: Week 9.

A check outside the model

estat gof: alteroutstar2 is 22 in the data and 13.4 in simulated outcomes. Useful and incomplete, as in Chapter 21.

To read after the session

Week 8: Chapter 14, and Chapter 25 through the objective function

Left over: Ch. 22 Ex. 3 and Ex. 5; optional Ch. 23 with Ex. 2 and 3, optional Ch. 24, Section 24.4 with Ex. 4 and 5

Reread: Chapter 22, What can go wrong

Do-files and teaching material: nwcommands.org/book

Instructors are free to use, adapt and share these slides.

