NME

Network Modeling for Epidemics

1

Model specification

For dyad dependent models

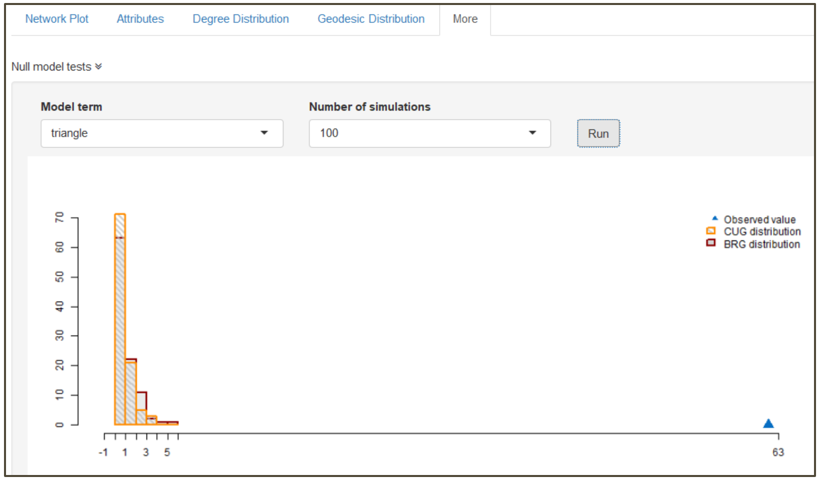
These models behave differently

- They're more like a complex system
  - And while the terms might look like they represent simple local configurations
    - E.g., Triangles and stars
  - They actually imply processes that cascade through the whole network

- Our intuition about them is often wrong
  - And that can lead to trouble

Simple example in statnetWeb

- Let's revisit the faux.mesa.high network
  - Recall that the CUG test showed there were many more triangles than expected for this level of tie density
  - How might you test this in an ERGM?

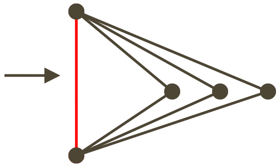


Triangle term

- The triangle term: $\mathrm{t}(\mathrm{x}) = \sum y_{ij} y_{jk} y_{ik}$

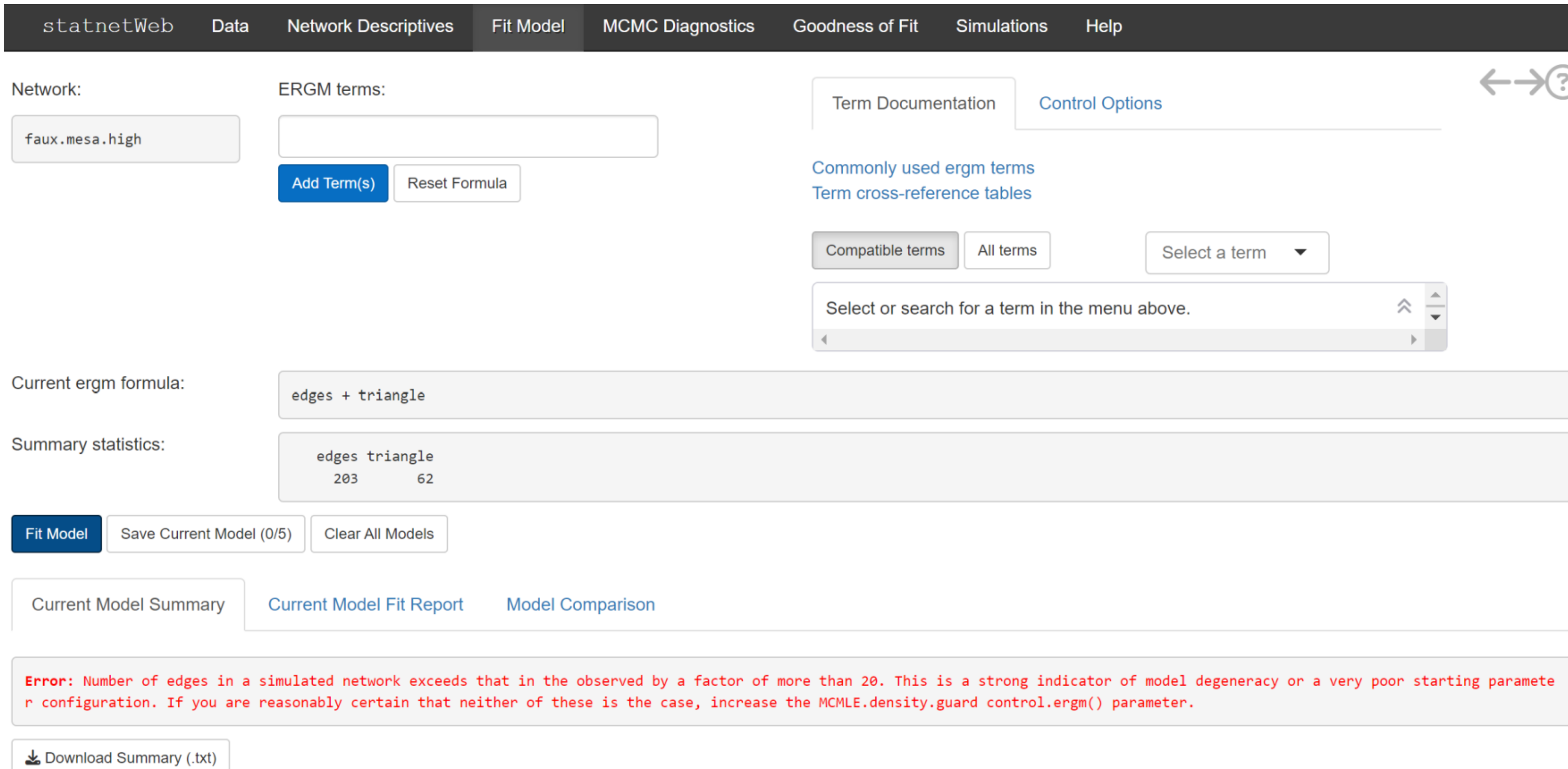
  $\mathrm{t}(\mathrm{x})$ = # of triangles in the graph

  - Here $\mathrm{t}(\mathrm{x})$ = 3 if the red edge is toggled on



- This is one of the classic Markov Graph terms
  - From the Frank and Strauss (1986) paper (see Module 2 appendix)

Fit model: edges + triangle



What happened?

- The process triggered a built-in error detector
- And that automatically stopped the run
- Note the error message:

```
Number of edges in a simulated network
exceeds that in the observed by a factor of
more than 20.
```

- The MCMC estimation chain was producing networks with WAY too many edges

To really see what's happening

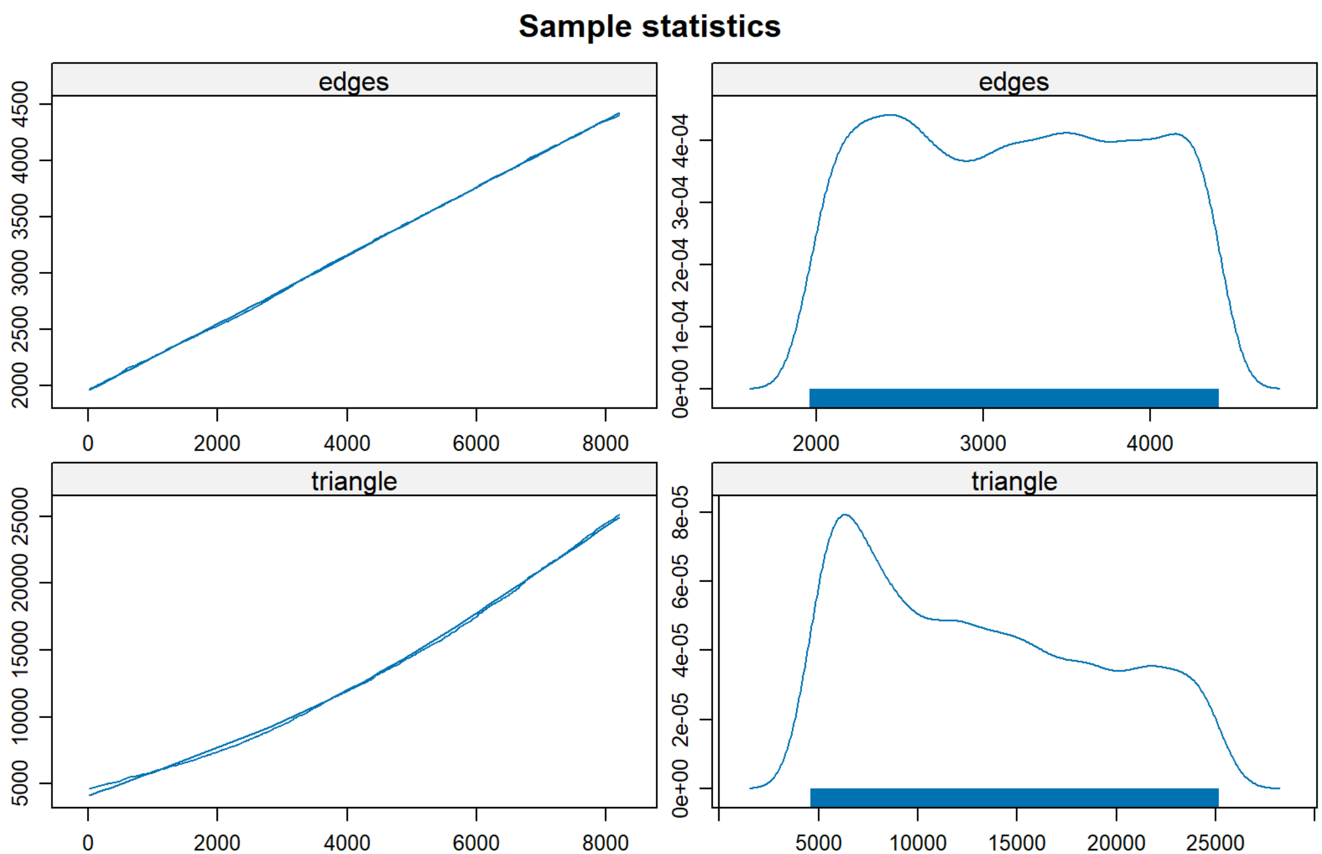
- We need some advanced ergm options only available from the command line
- We will set some MCMC control parameters
  - To track each single toggle
  - And stop before triggering the built in error detector

If you want to try this yourself:

```
library(ergm)
data("faux.mesa.high")
summary(faux.mesa.high ~ edges + triangle)
fit <- ergm(faux.mesa.high ~ edges + triangle,
                      control=snctrl(MCMC.interval=1, MCMLE.maxit=15,

MCMLE.effectiveSize=NULL))
mcmc.diagnostics(fit)
```

The MCMC dx plots …



This is really bad

And it doesn't look like it's going to get better with a longer run

Why is this happening?

- Because this is a poorly specified model
  - It would never produce the network we observed
  - So the MCMC algorithm can’t find ANY coefficients that work
  - And the ergm package automatically puts it out of its misery

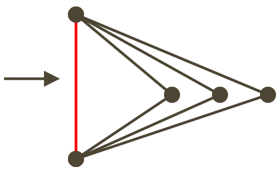
- There’s nothing wrong with
  - the theory
  - the algorithm
  - the data

- It’s just a bad model

Intuition: Why is this a bad model?

Because triad formation doesn't actually work like this

The triangle term: $\mathrm{t(x)} = \sum y_{ij} y_{jk} y_{ik}$



- With this term every additional triangle has the same impact, θ
  - So the odds of the red edge above are 3 times higher than an edge that creates only 1 triangle.
  - And an edge that creates 10 triangles has 10x higher odds

- This creates a cascading runaway process
  - Edges are most likely when they create huge clusters of triangles
  - And that's not what we see in our network

This is called “Model degeneracy”

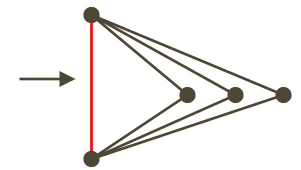
- The model would not produce the observed network
  - Instead it places all probability on networks that are nearly empty, or nearly complete
  - On average, this gives the right value for the netstats, but you would never get the observed network from this model

- And this is what model misspecification looks like with dependent data:
  - You typically won’t even get a fit to converge
  - So there’s no fit object to diagnose
  - The classic diagnostic is the MCMC algorithm heading off into graphs with much higher density than observed

- See the appendix on Model Degeneracy for more details

The solution: Better specification

- New statistic: $gwesp = e^{\alpha}\sum_{i=1}^{n-2}\{1-(1-e^{-\alpha})^{i}\}sp_i$



  - *$gwesp$* = a *weighted* sum of the triangles created by each edge
  - Where the weights decline for each additional triangle created
    - For each additional “shared partner” of an edge (like the red edge here)
    - This sets declining marginal returns, with a smooth decay function
  - The decay function we use involves a geometric weighting
    - Hence the name: geometrically weighted edge-wise shared partners
    - a.k.a. GWESP

Details in the Appendix

Practical advice

- Stay away from the canonical Markov graph terms
  - Unless you are working with very small networks

- The ergm package includes both the Markov graph terms and more stable alternatives

To represent	**Markov graph ergm term**	**More stable alternatives**
Ties on a node	kstar	degree(n) (non-parametric) gwdegree (parametric)
Triads	triangle	esp(n) (non-parametric) gwesp (parametric)