GDAT622\_Invest\_2

James Stanfield

6/6/2020

readMat(here("Data/facebook100/Howard90.mat")) -> how\_data  
how\_data$A -> how\_mat

ergm(how\_mat ~ edges) -> how1 # AIC: 3831040

## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient  
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Stopping at the initial estimate.

## Evaluating log-likelihood at the estimate.

summary(how1)

##   
## ==========================  
## Summary of model fit  
## ==========================  
##   
## Formula: how\_mat ~ edges  
##   
## Iterations: 7 out of 20   
##   
## Monte Carlo MLE Results:  
## Estimate Std. Error MCMC % z value Pr(>|z|)  
## edges -3.66270 0.00158 0 -2315 <1e-04  
##   
## Null Deviance: 22699408 on 16374162 degrees of freedom  
## Residual Deviance: 3831038 on 16374161 degrees of freedom  
##   
## AIC: 3831040 BIC: 3831055 (Smaller is better.)

So starting with edges gives us an AIC of 3831040 to start

ergm(how\_mat ~ edges + istar(2)) -> how2 # AIC: 3824915

## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:

## Optimizing with step length 0.854329496338699.

## The log-likelihood improved by 3.909.

## Iteration 2 of at most 20:

## Optimizing with step length 0.0469337619755651.

## The log-likelihood improved by 2.67.

## Iteration 3 of at most 20:

## Optimizing with step length 0.0158017100410617.

## The log-likelihood improved by 1.853.

## Iteration 4 of at most 20:

## Optimizing with step length 0.015296336670728.

## The log-likelihood improved by 1.616.

## Iteration 5 of at most 20:

## Optimizing with step length 0.0106950312707619.

## The log-likelihood improved by 2.852.

## Iteration 6 of at most 20:

## Optimizing with step length 0.00905664277414525.

## The log-likelihood improved by 3.354.

## Iteration 7 of at most 20:

## Optimizing with step length 0.00601210494199512.

## The log-likelihood improved by 2.236.

## Iteration 8 of at most 20:

## Optimizing with step length 0.00894558807188058.

## The log-likelihood improved by 1.322.

## Iteration 9 of at most 20:

## Optimizing with step length 0.00901490753773012.

## The log-likelihood improved by 1.422.

## Iteration 10 of at most 20:

## Optimizing with step length 0.00526012462716561.

## The log-likelihood improved by 1.982.

## Iteration 11 of at most 20:

## Optimizing with step length 0.0104160450140413.

## The log-likelihood improved by 2.065.

## Iteration 12 of at most 20:

## Optimizing with step length 0.0090499672616652.

## The log-likelihood improved by 1.1.

## Iteration 13 of at most 20:

## Optimizing with step length 0.0127741586708145.

## The log-likelihood improved by 1.113.

## Iteration 14 of at most 20:

## Optimizing with step length 0.0166925538363576.

## The log-likelihood improved by 1.557.

## Iteration 15 of at most 20:

## Optimizing with step length 0.0160984099767045.

## The log-likelihood improved by 2.571.

## Iteration 16 of at most 20:

## Optimizing with step length 0.00536062406876109.

## The log-likelihood improved by 2.093.

## Iteration 17 of at most 20:

## Optimizing with step length 0.0104187829267522.

## The log-likelihood improved by 1.

## Iteration 18 of at most 20:

## Optimizing with step length 0.0324436609680547.

## The log-likelihood improved by 1.123.

## Iteration 19 of at most 20:

## Optimizing with step length 0.0127960130738905.

## The log-likelihood improved by 2.14.

## Iteration 20 of at most 20:

## Optimizing with step length 0.00834847469686936.

## The log-likelihood improved by 1.296.

## MCMLE estimation did not converge after 20 iterations. The estimated coefficients may not be accurate. Estimation may be resumed by passing the coefficients as initial values; see 'init' under ?control.ergm for details.

## Finished MCMLE.

## Evaluating log-likelihood at the estimate. <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient  
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient  
## Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 .  
## This model was fit using MCMC. To examine model diagnostics and  
## check for degeneracy, use the mcmc.diagnostics() function.

summary(how2)

##   
## ==========================  
## Summary of model fit  
## ==========================  
##   
## Formula: how\_mat ~ edges + istar(2)  
##   
## Iterations: 20 out of 20   
##   
## Monte Carlo MLE Results:  
## Estimate Std. Error MCMC % z value Pr(>|z|)  
## edges -4.31e+00 2.18e-03 100 -1977 <1e-04  
## istar2 5.83e-03 2.11e-06 85 2764 <1e-04  
##   
## Null Deviance: 22699408 on 16374162 degrees of freedom  
## Residual Deviance: 3824888 on 16374160 degrees of freedom  
##   
## AIC: 3824892 BIC: 3824922 (Smaller is better.)

AIC dropped.

ergm(how\_mat ~ edges + istar(2) + mutual) -> how3 # AIC: 3658574

## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Warning in ergm.mple(nw, fd, m, MPLEtype = MPLEtype, init = init, control =  
## control, : glm.fit: fitted probabilities numerically 0 or 1 occurred

## Finished MPLE.

## Starting Monte Carlo maximum likelihood estimation (MCMLE):

## Iteration 1 of at most 20:

## Optimizing with step length 1.

## The log-likelihood improved by < 0.0001.

## Step length converged once. Increasing MCMC sample size.

## Iteration 2 of at most 20:

## Optimizing with step length 1.

## Warning in ergm.MCMCse.lognormal(theta = theta, init = init, statsmatrix =  
## statsmatrix0, : Approximate Hessian matrix is singular. Standard errors due  
## to MCMC approximation of the likelihood cannot be evaluated. This is likely  
## due to insufficient MCMC sample size or highly correlated model terms.

## The log-likelihood improved by < 0.0001.

## Step length converged twice. Stopping.

## Finished MCMLE.

## Evaluating log-likelihood at the estimate. <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient  
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient  
## Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 .  
## This model was fit using MCMC. To examine model diagnostics and  
## check for degeneracy, use the mcmc.diagnostics() function.

summary(how3)

##   
## ==========================  
## Summary of model fit  
## ==========================  
##   
## Formula: how\_mat ~ edges + istar(2) + mutual  
##   
## Iterations: 2 out of 20   
##   
## Monte Carlo MLE Results:  
## Estimate Std. Error MCMC % z value Pr(>|z|)  
## edges -36.26277 NA NA NA NA  
## istar2 0.00296 NA NA NA NA  
## mutual 68.03721 NA NA NA NA  
##   
## Null Deviance: 22699408 on 16374162 degrees of freedom  
## Residual Deviance: 3658569 on 16374159 degrees of freedom  
##   
## AIC: 3658575 BIC: 3658618 (Smaller is better.)

That’s another AIC drop.

After the last ergm took so long to run, I’m not inclined to add any other predictors. Looking at what we have, both number of ties (edges) and repriprocity of ties (mutual) were considered significant in how2 model. Unfortunately, the p-value of mutual is given as NA above. Lets’s look at the values for our predictors.

The istar values is incredibly small compared to edges and mutual, so we can probably ignore that term with very little trouble.

The mutual term is the largest as well as positive, meaning that the model predicts that connections are very likely to be returned.

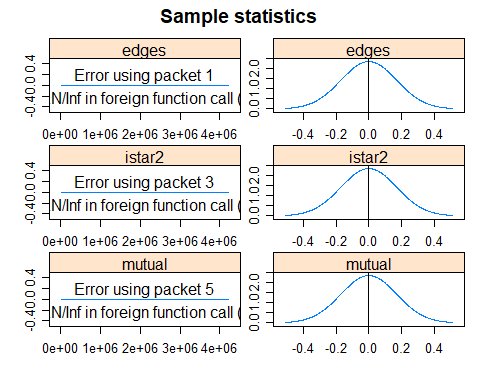
Let’s do some more diagnostics.

mcmc.diagnostics(how3)

## Sample statistics summary:  
##   
## Iterations = 16384:4209664  
## Thinning interval = 1024   
## Number of chains = 1   
## Sample size per chain = 4096   
##   
## 1. Empirical mean and standard deviation for each variable,  
## plus standard error of the mean:  
##   
## Mean SD Naive SE Time-series SE  
## edges 0 0 0 0  
## istar2 0 0 0 0  
## mutual 0 0 0 0  
##   
## 2. Quantiles for each variable:  
##   
## 2.5% 25% 50% 75% 97.5%  
## edges 0 0 0 0 0  
## istar2 0 0 0 0 0  
## mutual 0 0 0 0 0  
##   
##   
## Sample statistics cross-correlations:

## Warning in cor(as.matrix(x)): the standard deviation is zero

## edges istar2 mutual  
## edges 1 NA NA  
## istar2 NA 1 NA  
## mutual NA NA 1  
##   
## Sample statistics auto-correlation:  
## Chain 1   
## edges istar2 mutual  
## Lag 0 NaN NaN NaN  
## Lag 1024 NaN NaN NaN  
## Lag 2048 NaN NaN NaN  
## Lag 3072 NaN NaN NaN  
## Lag 4096 NaN NaN NaN  
## Lag 5120 NaN NaN NaN  
##   
## Sample statistics burn-in diagnostic (Geweke):  
## Error in svd(X) : a dimension is zero



##   
## MCMC diagnostics shown here are from the last round of simulation, prior to computation of final parameter estimates. Because the final estimates are refinements of those used for this simulation run, these diagnostics may understate model performance. To directly assess the performance of the final model on in-model statistics, please use the GOF command: gof(ergmFitObject, GOF=~model).

well diagnostics pretty much just says that it can’t compute properly, so I won’t be accepting anything from the output as accurate.