INFX 576: Problem Set 6 - Estimating Networks[*]

Jay Chauhan

Due: Thursday, February 23, 2017

Collaborators: Avanti Chande, Gosuddin Siddiqi

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset6.Rmd` file from Canvas. You will also need the data contained in `problemset6_data.Rdata` and the additional R library `degreenet`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps6.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```r
# Load standard libraries
library(statnet)
library(degreenet)
load("problemset6_data.Rdata")
```

Problem 1: Perception and Recall of Social Relationships

Pick you favorite social network dataset, this can be data we have encountered in class, data you have collected as part of your own research, or data that was used in one of the readings for the course. Write a short response (3-4 paragraphs) discussing how issues of informant accuracy may or may not affect this data. Be sure to specifically discuss how possible error might be addressed.

- The data that I have taken to be studied with the scope of informant accuracy is the sampson dataset which we have used in many of out assignments through the course. The sampson dataset is a network of monks and their ties with other monks. Here i have taken into consideration the Like network of the monks, where they were asked to nominate the top 3 to 4 monks whom they like, at three different points in time.

- The informant accuracy can be affected in a number of ways such as: 1) There can be a much complex model in the minds of monks when asked the question of who do the like rather than just a simple dyadic relation. This might lead some inaccuracy to trickle into their responses where they nominate the monks based on some criterion. 2) Also since the data which is collected has the monks to report their ties with

[*] Problems originally written by C.T. Butts (2009)

other monks at 3 different points in time, their ties might lead to false representations. This is due to the possibility that a monk formed a tie with some other monk only recently but since there is a tie, it might lead to the interpretation that the dyad spanned across all the points in time. Even if the This is another source where the reported ties are based on the memories of the monks, and can't be validated.

• The possible time aggregated errors might get resolved if the edged are asked to be weighted by all the monks based on the duration of their ties with the other monks. • Another possible solution to the problem is that the experiment can be performed on groups rather than individual level, where the accuracy would increase since the group structure would help in revealing both the real ties as well as improve memory reated inaccuracies.

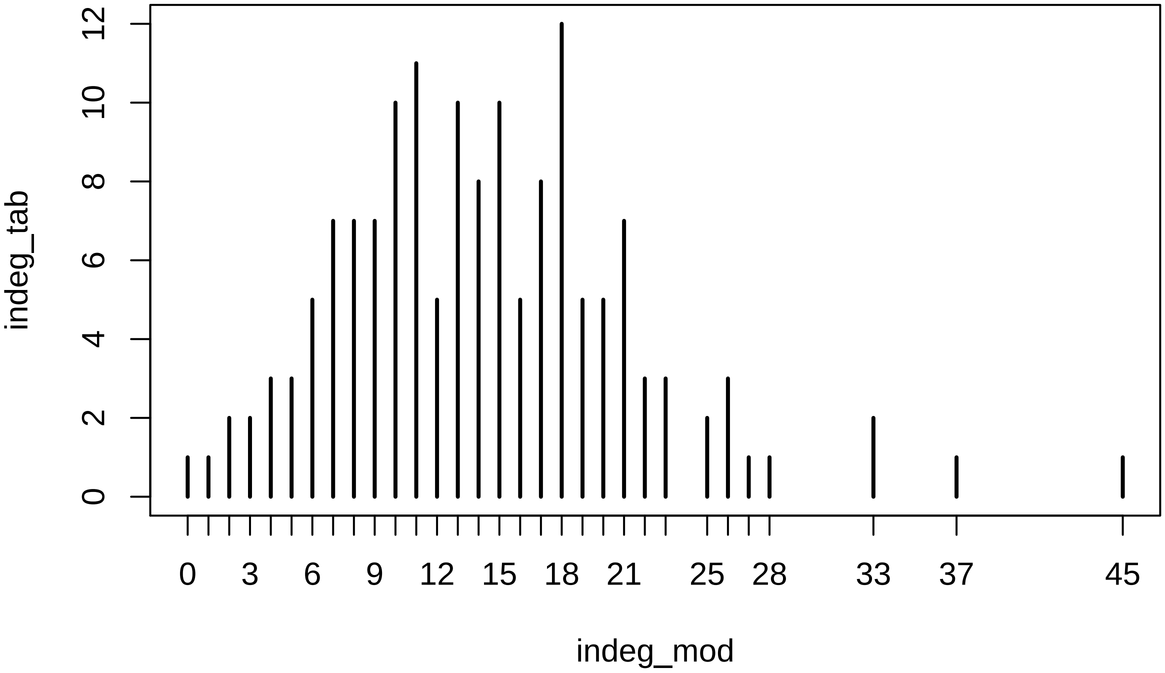
Problem 2: Modeling Degree Distributions

In the data for this problem set you will find a dataset named `EnronMailUSC1`. This object is the time-aggregated network of emails among 151 employees of Enron Corporation, as prepared by researchers at USC.

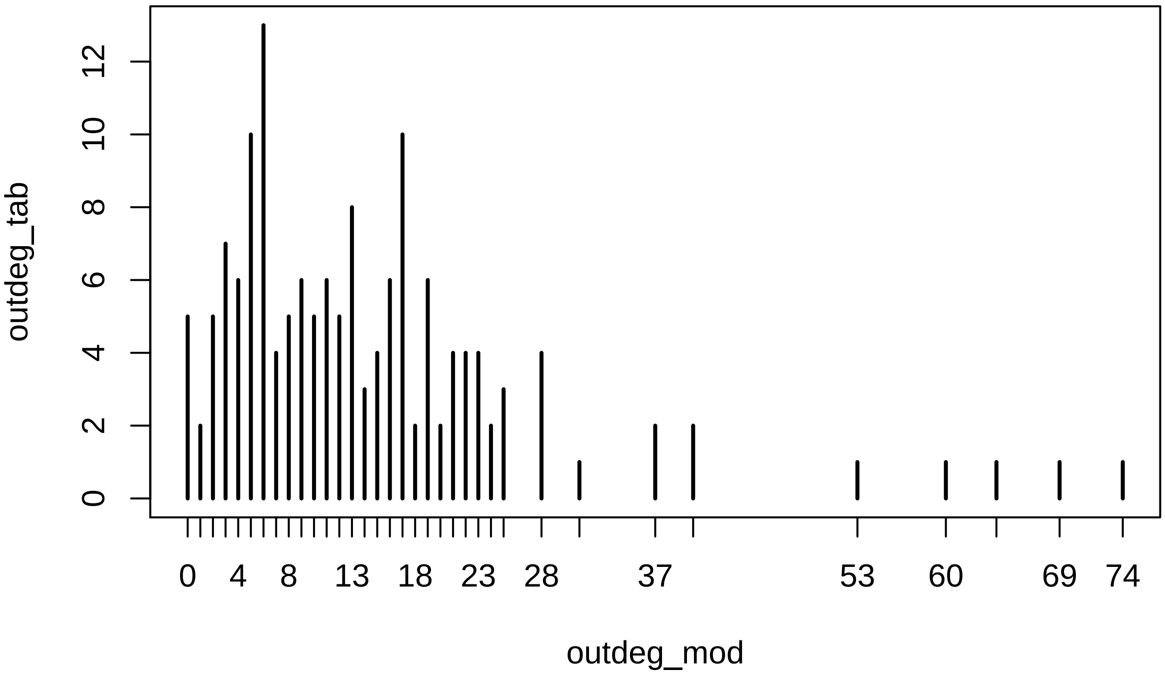
(a) Degree Distribtuions

Begin your investigation by plotting histograms of the indegree, outdegree, and total degree for the Enron email data. Interpret the patterns you see. Do any suggest (or rule out) specific functional form and/or partner formation processes?

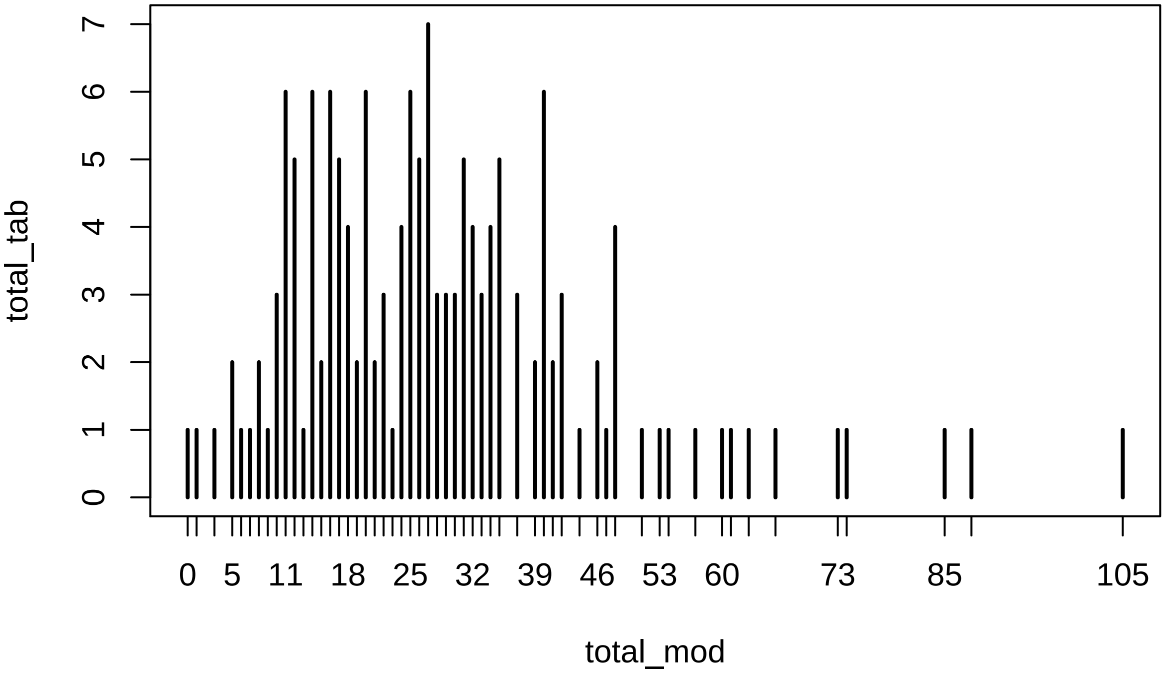
```r
#plot indegree of EnronMailUSC1 network
indeg_mod<-degree(EnronMailUSC1,cmode="indegree")
indeg_tab<-table(indeg_mod)
plot(indeg_tab)
```



```r
#plot outdegree of EnronMailUSC1 network
outdeg_mod<-degree(EnronMailUSC1,cmode="outdegree")
outdeg_tab<-table(outdeg_mod)
plot(outdeg_tab)
```



```r
#plot total degree of EnronMailUSC1 network
total_mod<-degree(EnronMailUSC1)
total_tab<-table(total_mod)
plot(total_tab)
```



From the above histograms we can see that the indegree distribution seems to be similar to a Poisson model, the outdegree distribution seems to be a Negative binomial model and the total degree distribution also seems to be a Poisson model. Observing the graphs we might be able to rule out the Geometric Yule distribution, and also the Yule/Warring distribution since the distribution of these degrees don't resemble these two models.

(b) Degree Distribution Models

Using the degreenet package, fit models to the indegree, outdegree, and total degree distributions for the Enron dataset. Which model provides the best fit in each case in terms of AICC and BIC? In addition to goodness-of-fit information, show the parameters of the best-fitting model.

```
#fit the candidate models to indegree distribution
fit.indeg.war<-awarmle(indeg_mod)
fit.indeg.yule<-ayulemle(indeg_mod)
fit.indeg.geo<-ageomle(indeg_mod)
fit.indeg.nb<-anbmle(indeg_mod)
fit.indeg.poi<-apoimle(indeg_mod)
```

```
## Warning in optim(par = guess, fn = llpoi, hessian = hessian, control = list(fnscale = -10), : one-di
## use "Brent" or optimize() directly
```

```
fit.indeg.gy<-agymle(indeg_mod,guess=c(10,6000))
fit.indeg.nby<-anbymle(indeg_mod,guess=c(5,50,0.3))

fittab.indeg<-rbind(
  llpoiall(v=fit.indeg.poi$theta,x=indeg_mod),
  llgeoall(v=fit.indeg.geo$theta,x=indeg_mod),
  llnball(v=fit.indeg.nb$theta,x=indeg_mod),
  llyuleall(v=fit.indeg.yule$theta,x=indeg_mod),
  llgyall(v=fit.indeg.gy$theta,x=indeg_mod),
  llnbyall(v=fit.indeg.nby$theta,x=indeg_mod),
  llwarall(v=fit.indeg.war$theta,x=indeg_mod)
)
rownames(fittab.indeg)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")

#print the tabulated values of AICC and BIC across all the models
fittab.indeg
```

```
##             np   log-lik     AICC      BIC
## Poisson      1 -596.9712 1195.969 1198.960
## Geometric    3 -542.5341 1091.232 1100.120
## NegBinom     3 -502.2345 1010.632 1019.521
## Yule         3 -653.1647 1312.493 1321.381
## GeoYule      4 -653.1803 1314.635 1326.430
## NegBinYule   5 -502.9002 1016.214 1030.887
## Waring       3 -556.9045 1119.972 1128.861
```

```
#examine the paramters for negative bionomial
fit.indeg.nb
```

```
## $theta
## expected stop   prob 1 stop
##    18.1762351     0.2668257
```

- Here we can see that based on the observations of AICC and BIC, the negative bionamial model fits the best for the indegree distribution of the Enron dataset.

```
#fit the candidate models to outdegree distribution
fit.outdeg.war<-awarmle(outdeg_mod)
fit.outdeg.yule<-ayulemle(outdeg_mod)
fit.outdeg.geo<-ageomle(outdeg_mod)
fit.outdeg.nb<-anbmle(outdeg_mod)
fit.outdeg.poi<-apoimle(outdeg_mod)
```

```
## Warning in optim(par = guess, fn = llpoi, hessian = hessian, control = list(fnscale = -10), : one-di
## use "Brent" or optimize() directly
```

```
fit.outdeg.gy<-agymle(outdeg_mod,guess=c(10,6000))
fit.outdeg.nby<-anbymle(outdeg_mod,guess=c(5,500,0.3))

fittab.outdeg<-rbind(
  llpoiall(v=fit.outdeg.poi$theta,x=outdeg_mod),
  llgeoall(v=fit.outdeg.geo$theta,x=outdeg_mod),
  llnball(v=fit.outdeg.nb$theta,x=outdeg_mod),
  llyuleall(v=fit.outdeg.yule$theta,x=outdeg_mod),
  llgyall(v=fit.outdeg.gy$theta,x=outdeg_mod),
  llnbyall(v=fit.outdeg.nby$theta,x=outdeg_mod),
  llwarall(v=fit.outdeg.war$theta,x=outdeg_mod)
)
rownames(fittab.outdeg)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")

#print the tabulated values of AICC and BIC across all the models
fittab.outdeg
```

```
##            np   log-lik     AICC      BIC
## Poisson     1 -999.1225 2000.272 2003.262
## Geometric   3 -550.5567 1107.277 1116.165
## NegBinom    3 -547.9978 1102.159 1111.047
## Yule        3 -625.1602 1256.484 1265.372
## GeoYule     4 -625.1598 1258.594 1270.389
## NegBinYule  5 -625.1602 1260.734 1275.407
## Waring      3 -558.3307 1122.825 1131.713
```

```
#examine the paramters for negative bionomial
fit.outdeg.nb
```

```
## $theta
## expected stop   prob 1 stop
##    15.3759296     0.1077503
##
## $asycov
##               expected stop   prob 1 stop
## expected stop   0.362731013 -0.0014139341
## prob 1 stop    -0.001413934  0.0001836278
##
## $se
## expected stop   prob 1 stop
##    0.60227154    0.01355093
##
## $asycor
##               expected stop prob 1 stop
## expected stop     1.0000000  -0.1732477
## prob 1 stop      -0.1732477   1.0000000
##
## $npar
## gamma mean gamma s.d.
##   13.71917   10.65855
##
## $value
## [1] -526.0423
```

- Here we can see that based on the observations of AICC and BIC, the negative bionamial model fits the best for the otdegree distribution of the Enron dataset.

```r
#fit the candidate models to total degree distribution
fit.total.war<-awarmle(total_mod)
fit.total.yule<-ayulemle(total_mod)
fit.total.geo<-ageomle(total_mod)
fit.total.nb<-anbmle(total_mod)
fit.total.poi<-apoimle(total_mod)
```

```
## Warning in optim(par = guess, fn = llpoi, hessian = hessian, control = list(fnscale = -10), : one-di
## use "Brent" or optimize() directly
```

```r
fit.total.gy<-agymle(total_mod,guess=c(10,6000))
fit.total.nby<-anbymle(total_mod,guess=c(5,50,0.3))

fittab.total<-rbind(
  llpoiall(v=fit.total.poi$theta,x=total_mod),
  llgeoall(v=fit.total.geo$theta,x=total_mod),
  llnball(v=fit.total.nb$theta,x=total_mod),
  llyuleall(v=fit.total.yule$theta,x=total_mod),
  llgyall(v=fit.total.gy$theta,x=total_mod),
  llnbyall(v=fit.total.nby$theta,x=total_mod),
  llwarall(v=fit.total.war$theta,x=total_mod)
)
rownames(fittab.total)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")

#print the tabulated values of AICC and BIC across all the models
fittab.total
```

```
##            np    log-lik     AICC      BIC
## Poisson     1 -1096.2222 2194.471 2197.462
## Geometric   3  -654.0145 1314.192 1323.081
## NegBinom    3  -622.9454 1252.054 1260.943
## Yule        3  -789.3682 1584.900 1593.788
## GeoYule     4  -789.3980 1587.070 1598.865
## NegBinYule  5  -622.0948 1254.603 1269.276
## Waring      3  -668.4949 1343.153 1352.042
```

```r
#examine the paramters for negative bionomial
fit.total.nb
```

```
## $theta
## expected stop   prob 1 stop
##    30.7407862     0.1004351
##
## $asycov
##               expected stop   prob 1 stop
## expected stop  0.1760495995 -0.0001380963
## prob 1 stop   -0.0001380963  0.0001492933
##
```

```
## $se
## expected stop   prob 1 stop
##    0.41958265    0.01221856
##
## $asycor
##               expected stop prob 1 stop
## expected stop     1.0000000  -0.0269367
## prob 1 stop      -0.0269367   1.0000000
##
## $npar
## gamma mean gamma s.d.
##   27.65333   15.73792
##
## $value
## [1] -616.9315
```

- Here we can see that based on the observations of AICC and BIC, the negative bionamial model fits the best for the total degree distribution of the Enron dataset.