INFX 576: Problem Set 6 - Estimating Networks*

Suchitra Sundararaman

Due: Thursday, February 23, 2017

Collaborators:

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.

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

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.

One of the social network dataset that interests me is the emon(Interorganizational Search and Rescue Networks) networks. This network reflects the reported frequency of organizational interaction during the search and rescue effort, where frequency is on a 1-4 level. There can be some accuracy issues with the informant reporting as discussed below: There is no fixed definition for the scale of frequency of interaction and this may result in different informants reporting different values according to their perception. There can a be possibility of an informant missing to report an interaction. Since informants know that their interactions are being monitored, this may change their inherent behavior and may skew the data. Informants generally tend to over report interactions in the event of them being monitored.

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

The possible error in data can be addressed by using a better data collection technique. The K-replication balanced arc sampling design, provides multiple observations on a single tie and informant and also maintains linear complexity in network size. This is because, if there is a need of k observations on each tie, the informant supplies only KN observations.

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.

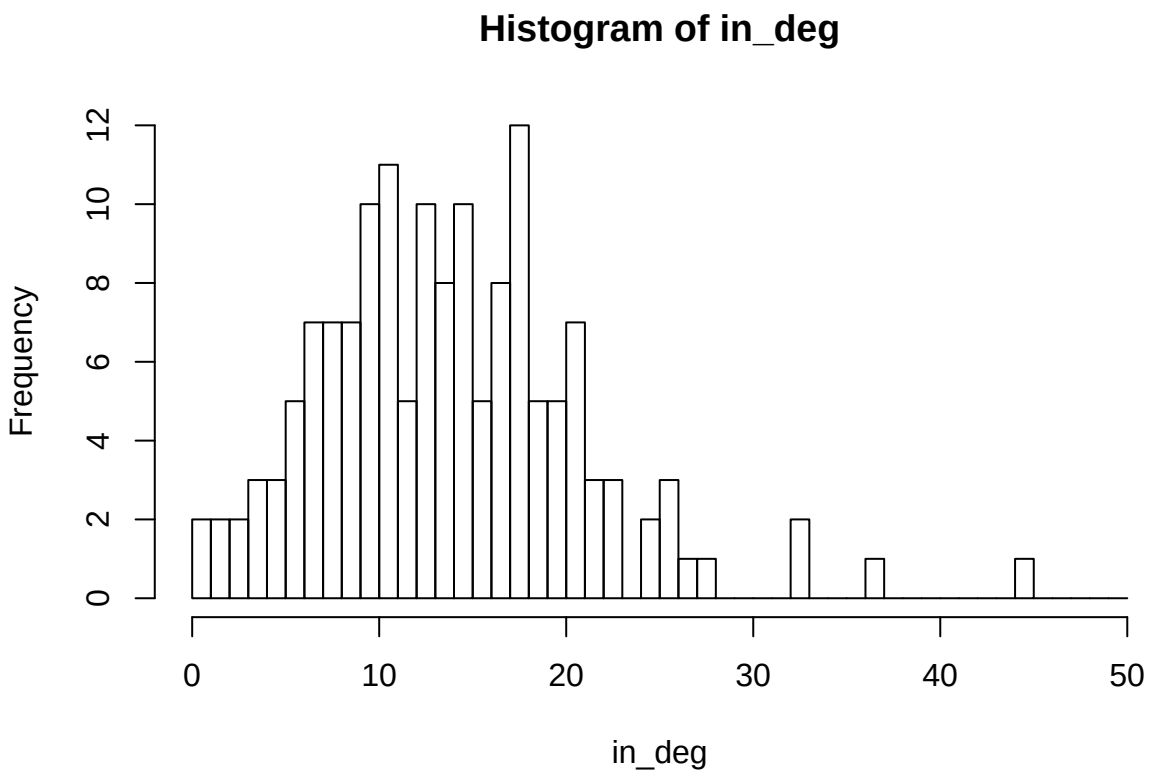
```
EnronMailUSC1
```

```
##  Network attributes:
##   vertices = 151
##   directed = TRUE
##   hyper = FALSE
##   loops = TRUE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 2235
##     missing edges= 0
##     non-missing edges= 2235
##
##  Vertex attribute names:
##     email employeeID firstName fullName lastName
##
##  Edge attribute names not shown
```

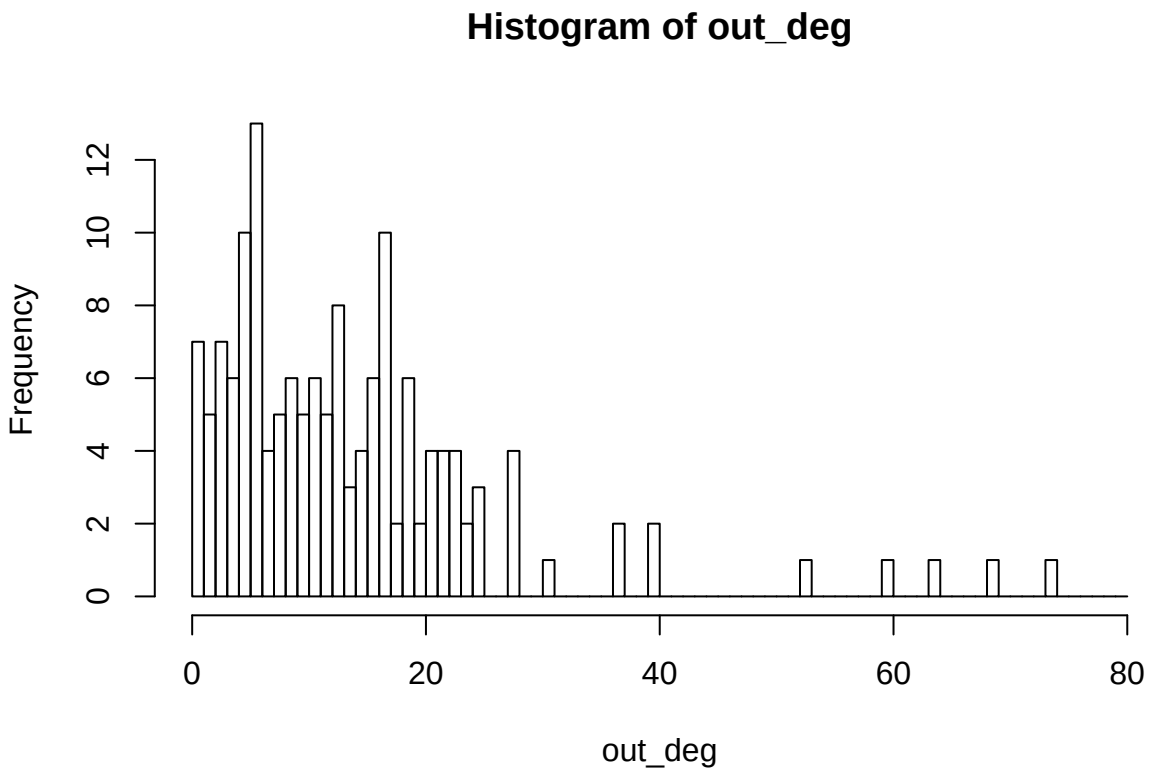
(a) Degree Distribtuions

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?

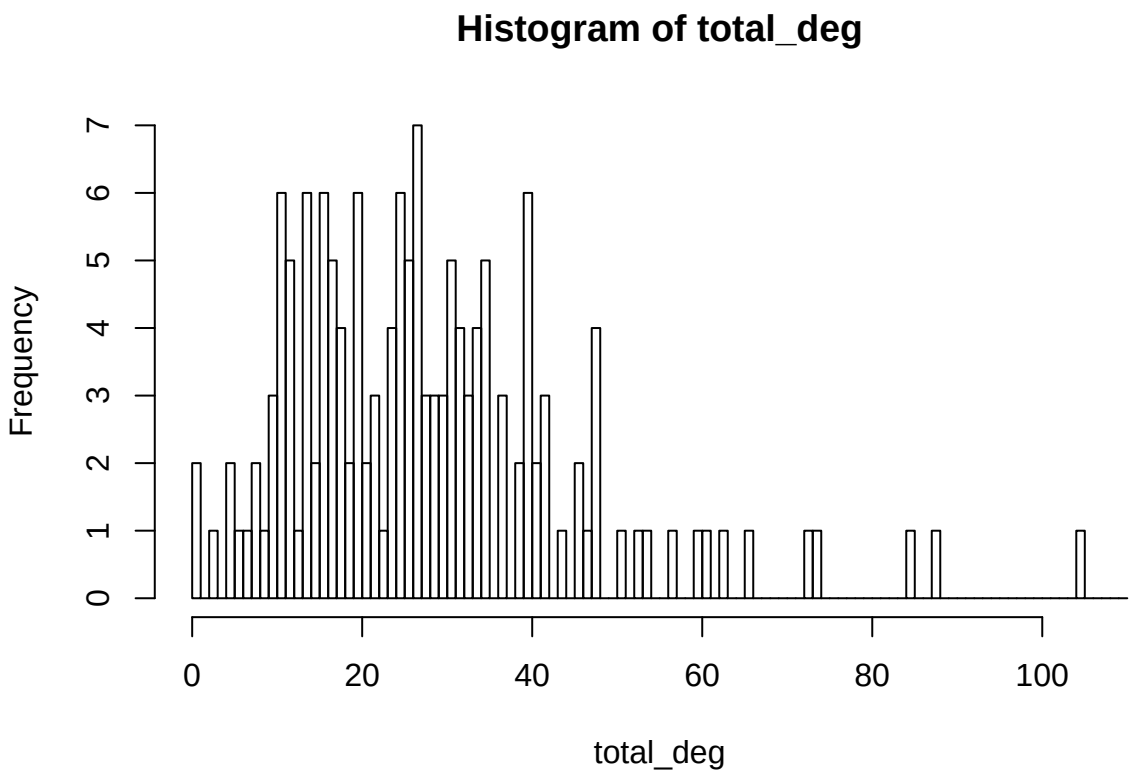
```
in_deg <- degree(EnronMailUSC1,gmode="digraph", cmode = "indegree")
out_deg <- degree(EnronMailUSC1,gmode="digraph", cmode = "outdegree")
total_deg <- in_deg + out_deg

hist(in_deg, breaks= seq(0,50,1))
```




```
hist(out_deg, breaks= seq(0,80,1))
```




```
hist(total_deg, breaks= seq(0,110,1))
```



The distribution for indegree, outdegree and total degree seem to be similar to certain distributions as mentioned below:

Indegree: The distribution of the indegree ties seem to represent a poisson distribution. The distribution is near-normal and has a light tail.

Outdegree: The distribution of the Outdegree ties seem to represent a Yule/Waring distribution. This implies that the probability of sending a mail to someone will depend on the number of outdegrees that person has. This is representative of preferential attachment.

Total Degree: The distribution of the total degree seem to represent a negative binomial distribution.

(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.

Indegree

```
fit.in.war<-awarmle(in_deg)
fit.in.yule<-ayulemle(in_deg)
fit.in.geo<-ageomle(in_deg)
fit.in.nb<-anbmle(in_deg)
fit.in.poi<-apoimle(in_deg)
```

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

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


# Compare the AICC and BIC for the various models
fittab<-rbind(
  llpoiall(v=fit.in.poi$theta,x=in_deg),
  llgeoall(v=fit.in.geo$theta,x=in_deg),
  llnball(v=fit.in.nb$theta,x=in_deg),
  llyuleall(v=fit.in.yule$theta,x=in_deg),
  llgyall(v=fit.in.gy$theta,x=in_deg),
  llnbyall(v=fit.in.nby$theta,x=in_deg),
  llwarall(v=fit.in.war$theta,x=in_deg)
)
rownames(fittab)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")
fittab
```

```
##            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
```

```
fit.in.nb
```

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

Outdegree

```
fit.out.war<-awarmle(out_deg)
fit.out.yule<-ayulemle(out_deg)
fit.out.geo<-ageomle(out_deg)
fit.out.nb<-anbmle(out_deg)
fit.out.poi<-apoimle(out_deg)
```

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

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

# Compare the AICC and BIC for the various models
fittab<-rbind(
  llpoiall(v=fit.out.poi$theta,x=out_deg),
  llgeoall(v=fit.out.geo$theta,x=out_deg),
  llnball(v=fit.out.nb$theta,x=out_deg),
  llyuleall(v=fit.out.yule$theta,x=out_deg),
```

```
  llgyall(v=fit.out.gy$theta,x=out_deg),
  llnbyall(v=fit.out.nby$theta,x=out_deg),
  llwarall(v=fit.out.war$theta,x=out_deg)
)
rownames(fittab)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")
fittab
```

```
##            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
```

```
fit.out.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
```

Total degree

```
fit.tot.war<-awarmle(total_deg)
fit.tot.yule<-ayulemle(total_deg)
fit.tot.geo<-ageomle(total_deg)
fit.tot.nb<-anbmle(total_deg)
fit.tot.poi<-apoimle(total_deg)
```

```
## Warning in optim(par = guess, fn = llpoi, hessian = hessian, control = list(fnscale = -10), : one-di
```

```
## use "Brent" or optimize() directly
```

```
fit.tot.gy<-agyml(total_deg,guess=c(10,6000))
fit.tot.nby<-anbyml(total_deg,guess=c(5,500,0.3))

# Compare the AICC and BIC for the various models
fittab<-rbind(
  llpoiall(v=fit.tot.poi$theta,x=total_deg),
  llgeoall(v=fit.tot.geo$theta,x=total_deg),
  llnball(v=fit.tot.nb$theta,x=total_deg),
  llyuleall(v=fit.tot.yule$theta,x=total_deg),
  llgyall(v=fit.tot.gy$theta,x=total_deg),
  llnbyall(v=fit.tot.nby$theta,x=total_deg),
  llwarall(v=fit.tot.war$theta,x=total_deg)
)
rownames(fittab)<-c("Poisson","Geometric","NegBinom","Yule","GeoYule",
    "NegBinYule","Waring")
fittab
```

```
##            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  -789.3683 1589.150 1603.823
## Waring      3  -668.4949 1343.153 1352.042
```

```
fit.tot.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
```

In all the three cases, the negative binomial model seems to be the best fit for the indegree, outdegree and

the total degree distributions.