INFX 576: Problem Set 7 - Exploring Networks*

Harkar Talwar

Due: Friday, May 18, 2018

Collaborators: Prateek Tripathi, Aakash Agrawal

Instructions:

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

1. Download the `problemset7.Rmd` file from Canvas. You will also need the data contained in `problemset7_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_ps7.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("problemset7_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.

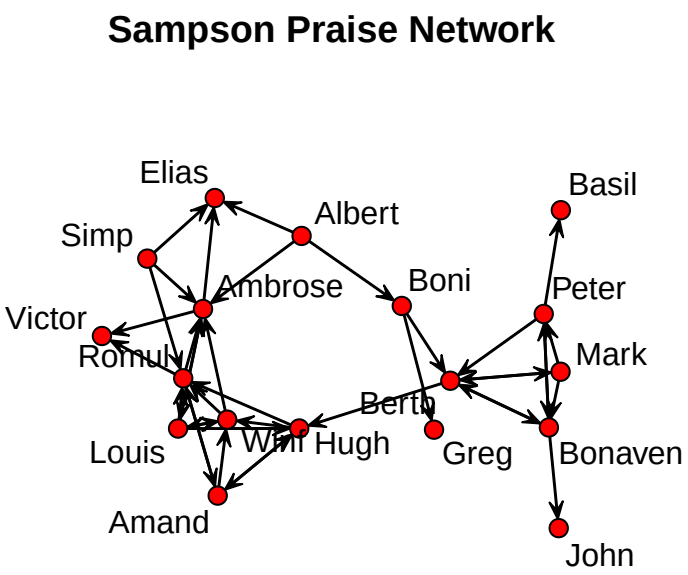
- The social network data that I've picked for this discussion is Sampson's Monastery Data (Sampson, 1969). The networks in this dataset are based on self-reported relations among monks in a New England monastery. The data represents four types of relational networks - esteem, liking, influence and praise, as well as the corresponding negative relations - disesteem, disliking, negative influence, and blame. Some example relational networks are shown below:

```
load("sampson.Rdata")
```

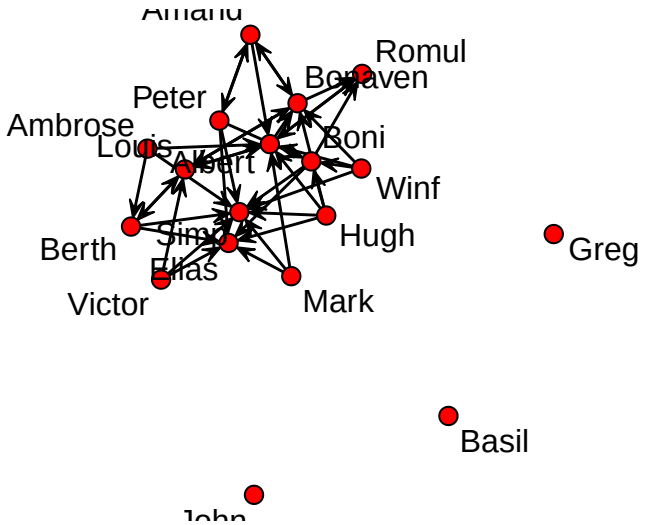
*Problems originally written by C.T. Butts (2009)

Plots of two of the Sampson Networks

```
gplot(sampson$Praise, label = sampson$Praise %v% "vertex.names",
      main = "Sampson Praise Network")
```




```
gplot(sampson$Blame, label = sampson$Blame %v% "vertex.names",
      main = "Sampson Blame Network")
```



Concerns of Informant Accuracy and possible resolutions based on Bernard et al. (1984) and Kirkland et al. (2018):

- Informant accuracy may decrease as the time elapsed since the last interaction increases. In the study, the monks were asked about their social relations at different points of time. However, it is likely that respondents would not be able to accurately estimate their relations over longer periods of time, as the memory of distant positive or negative events fades away. A way to address these errors in future studies could be to employ 'Record Assisted Recall', where the respondent keeps a record of their interactions with others in the network.
- Different monks may have different cognitive thresholds based on which they decide if a particular kind of relation exists or not. Like, dislike, praise, blame etcetera could be seen as conceptual or abstract variables, whose interpretation may vary among the different respondents. This can lead to inaccurate estimates of the network structure. A way to address this would be to ask solid practical questions that minimise distortion and ambiguity.
- Extending from the previous point, a more fundamental concern is that self reported data for the monks only represents human cognition about an external reality and that there should be a way to cross validate the responses. In this regard, cumulative reports for each respondent can yield more accurate results, than reports purely from an ego-centric perspective, which may be exaggerations/understatements. These reports can be further supplemented by employing non-obtrusive observers who are not part of the network, and are not likely biased in their observations.

References

- Sampson, S. (1969). Crisis in a cloister. Unpublished doctoral dissertation, Cornell University
- Bernard et al. (1984). The Problem of Informant Accuracy: The Validity of Retrospective Data. Annual Review of Anthropology 1984 13:1, 495-517

- Kirkland et al. (2018). Social Network Analysis and Estimating Size of Hard-to-Count Subpopulations.

Problem 2: Modeling Degree Distributions

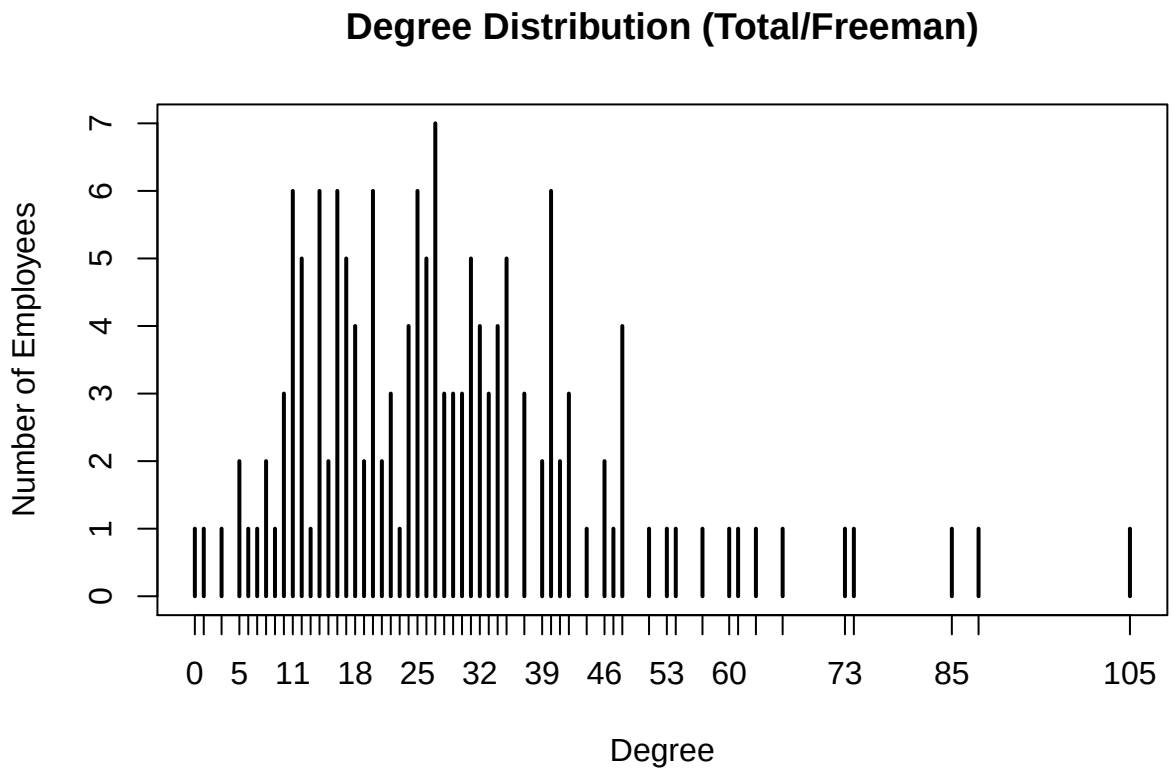
In the data for this problem set you will find a dataset named `EnronMailUSC1`. This object is the time-aggregated network of emails among 151 employees of Enron Corporation, as prepared by researchers at USC.

(a) Degree Distribtuions

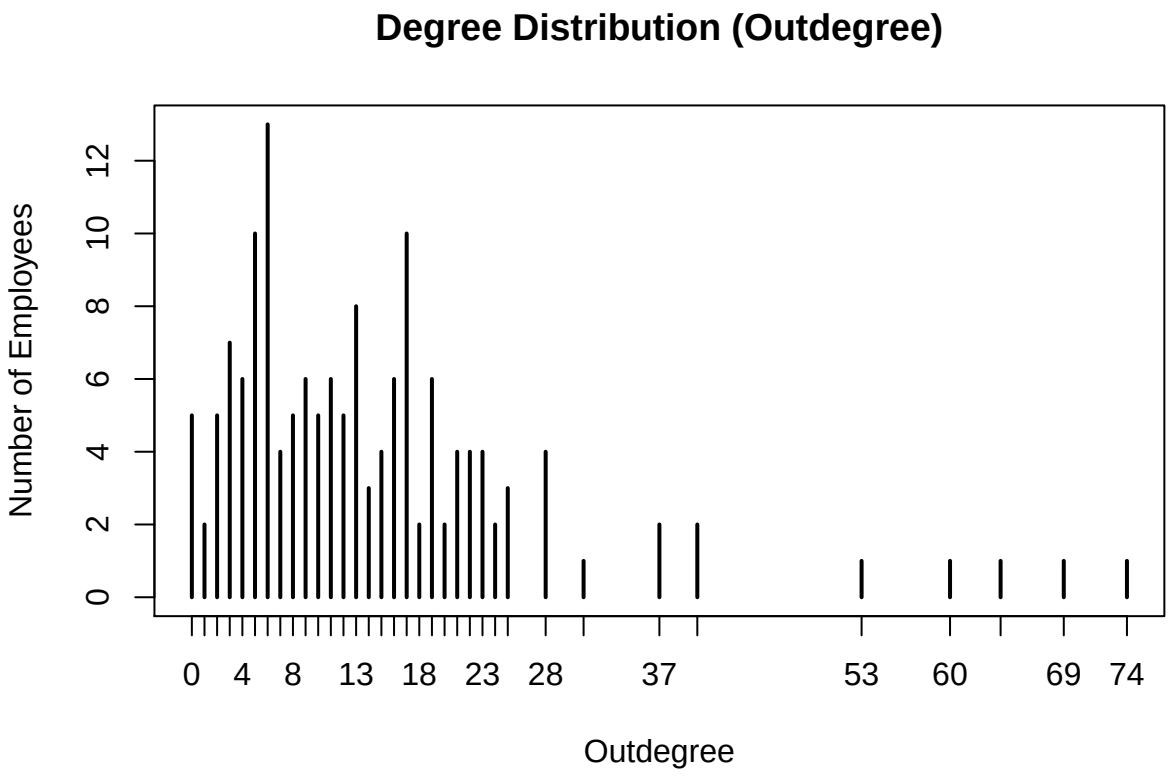
Begin your investigation by plotting histograms of the indegree, outdegree, and total degree for the Enron email data. Interpret the patterns you see. Do any suggest (or rule out) specific functional form and/or partner formation processes?

```
enron = EnronMailUSC1
d.enron = degree(enron)  # compute total degree
ind.enron = degree(enron, cmode = "indegree")  # compute indegree
outd.enron = degree(enron, cmode = "outdegree") # compute outdegree
```

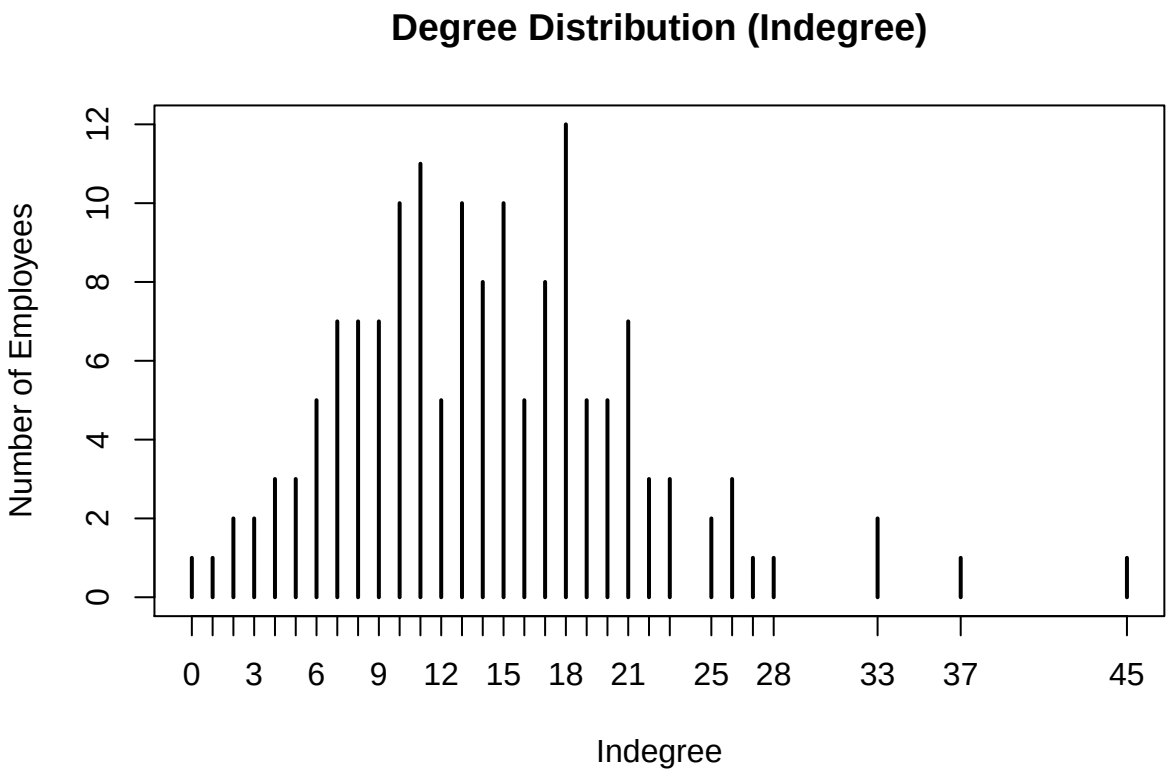
```
# Plot histograms for degree distribution
plot(table(d.enron), xlab = "Degree", ylab = "Number of Employees",
     main = "Degree Distribution (Total/Freeman)")
```




```
plot(table(outd.enron), xlab = "Outdegree", ylab = "Number of Employees",
     main = "Degree Distribution (Outdegree)")
```




```
plot(table(ind.enron), xlab = "Indegree", ylab = "Number of Employees",
     main = "Degree Distribution (Indegree)")
```



Observations

- From the histograms depicting degree distribution, we observe that email interactions between the studied Enron employees are not evenly distributed.
- There are a small number of employees that interacted with a large number of co-workers via email. This leads us to believe that the degree distribution could be modeled with a Binomial model. A Poisson model would likely not be relevant here since N is not sufficiently large.
- Further, since all of these plots have a lightly decaying tail, it seems unlikely that a Power Law distribution would model the behavior appropriately, i.e. although the number of higher degree employees is relatively small, the contrast is not as extreme as one would expect for a Power Law 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.

```
# Function to fit various distribution models to the passed degree
# distribution
fit.deg.dist.models = function(net.degree) {
  fit.enron.war = awarmle(net.degree)  # Waring Model
  fit.enron.yule = ayulemle(net.degree) # Yule Model
  fit.enron.geo = ageomle(net.degree)  # Geometric Model
  fit.enron.nb = anbmle(net.degree)  # Negative Binomial Model
  fit.enron.poi = apoimle(net.degree)  # Poisson Model
  fit.enron.gy = agymle(net.degree,guess=c(10,6000)) # Geometric Yule Model
```

```
  # Negative Binomial Yule Model
  fit.enron.nby = anbymle(outd.enron,guess=c(5,300,0.2))

  # Estimate the Log-likelihood for each
  model.fit = rbind(
  llpoiall(v=fit.enron.poi$theta,x=net.degree),
  llgeoall(v=fit.enron.geo$theta,x=net.degree),
  llnball(v=fit.enron.nb$theta,x=net.degree),
  llyuleall(v=fit.enron.yule$theta,x=net.degree),
  llgyall(v=fit.enron.gy$theta,x=net.degree),
  llnbyall(v=fit.enron.nby$theta,x=net.degree),
  llwarall(v=fit.enron.war$theta,x=net.degree)
  )
  rownames(model.fit)<-c("Poisson","Geometric",
                         "NegBinom","Yule","GeoYule",
                         "NegBinYule","Waring")
  model.fit
}
```

Fit Degree Distribution Models to Total Degree

```
fit.deg.dist.models(d.enron)
```

```
##            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  -784.7541 1579.922 1594.595
## Waring      3  -668.4949 1343.153 1352.042
```

It can be seen above that the Negative Binomial model provides the best fit for Total Degree distribution. It has the lowest values of the AICC and BIC.

Parameters of Best Fitting Model

```
anbmle(d.enron)
```

```
## $theta
## expected stop   prob 1 stop
##    30.7407884     0.1004351
##
## $asycov
##                expected stop   prob 1 stop
## expected stop  0.3522025032 -0.0002314724
## prob 1 stop   -0.0002314724  0.0001493371
##
## $se
## expected stop   prob 1 stop
##    0.59346651    0.01222035
##
## $asycor
##                expected stop prob 1 stop
```

```
## expected stop     1.0000000  -0.0319168
## prob 1 stop      -0.0319168   1.0000000
##
## $npar
## gamma mean gamma s.d.
##   27.65334   15.73792
##
## $value
## [1] -616.9315
```

Fit Degree Distribution Models to Indegree

```
fit.deg.dist.models(ind.enron)
```

```
##            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 -651.1975 1312.809 1327.481
## Waring      3 -556.9045 1119.972 1128.861
```

For Indegree as well, the Negative Binomial model gives the lowest values of AICC and BIC and is thus the best fit.

Parameters of Best Fitting Model

```
anbmle(ind.enron)
```

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

Fit Degree Distribution Models to Outdegree

```
fit.deg.dist.models(outd.enron)
```

```
##            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
```

Lastly, the Outdegree distribution is also fit best by a Negative Binomial model, with the lowest values of AICC and BIC

Parameters of Best Fitting Model

```
anbmle(outd.enron)
```

```
## $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
```