Network Practice Examples 9/16

#Short write-up

Dataset 1: Twitter data

This data consisits of "circles" from Twitter. Twitter data was crawled from public sources. The dataset includes node features, circles, and ego networks. Edges are directed in this dataset. A total number of 81,306 nodes/ 1,768,148 edges were included in the dataset. They did not mention how the ego were selected, but it is assumed they follow every nodes in the corresponding file.

Dataset 2: Enron Email Network

This dataset covers all the email communication within a dataset of around half million emails. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. Nodes of the network are email addresses and if an address i sent at least one email to address j. A total number of 36,692 nodes/ 183,381 edges were included in the dataset.

#Convert the dataset into a network structure

```
#Construct the adjacency from the twitter data
#I only used the data from one ego, since the networks are disconnected for different egos
file1<-read.table(file="./twitter/twitter/100318079.edges")
file1<-file1[order(file1$V2),]
file1$V3<-as.numeric(factor(file1$V2))
file1$V4<-file1$V3[match(file1$V1,file1$V2)]
mat1<-matrix(0L, 221, 221)
for (i in 1:dim(file1)[1]) {
  mat1[file1$V4[i],file1$V3[i]]=1
}
mat1[221,c(1:220)]=1
mat1[c(1:220),221]=1


#Construct the adjacency matrix from the Enron data
file2<-read.table(file="./Enron/email-Enron.txt")
file2$V1<-file2$V1+1
file2$V2<-file2$V2+1
mat2<-matrix(0L,ncol=length(unique(file2$V1)),length(unique(file2$V1)))
for (i in 1:dim(file2)[1]) {
  mat2[file2$V1[i],file2$V2[i]]=1
}
```

#p1 Model

Denote $Pr(Y_{i,j} = y_{i,j}, Y_{j,i} = y_{j,i}) = p(y_{i,j}, y_{j,i})$. Then

$p(0,0) = c_{i,j}$

$p(1,0) = c_{i,j}exp(\mu_{i,j})$

$p(0,1) = c_{i,j}exp(\mu_{j,i})$

$p(1,1) = c_{i,j}exp(\mu_{i,j} + \mu_{j,i} + \gamma)$

where $\mu_{i,j} = \alpha_i + \beta_j + \mu$; $\mu_{j,i} = \alpha_j + \beta_i + \mu$; γ is the reciprocal effect; and $c_{i,j} = \frac{1}{1+exp(\mu_{i,j})+exp(\mu_{j,i})+exp(\mu_{i,j}+\mu_{j,i}+\gamma)}$

Question: What is the llk for p1 model?

#Fit a p1 model using ergm package

```
# The original adjacency matrix is too large, it took forever to run
# Therefore, I subset samples from the original matrix and make it into a smaller one

library("ergm")
```

```
## Loading required package: network
```

```
## network: Classes for Relational Data
## Version 1.15 created on 2019-04-01.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.
```

```
##
## ergm: version 3.10.4, created on 2019-06-10
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Carter T. Butts, University of California -- Irvine
##                     Steven M. Goodreau, University of Washington
##                     Pavel N. Krivitsky, University of Wollongong
##                     Martina Morris, University of Washington
##                     with contributions from
##                     Li Wang
##                     Kirk Li, University of Washington
##                     Skye Bender-deMoll, University of Washington
##                     Chad Klumb
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").
```

```
## NOTE: Versions before 3.6.1 had a bug in the implementation of the
## bd() constriant which distorted the sampled distribution somewhat.
## In addition, Sampson's Monks datasets had mislabeled vertices. See
## the NEWS and the documentation for more details.
```

```
## NOTE: Some common term arguments pertaining to vertex attribute
## and level selection have changed in 3.10.0. See terms help for
## more details. Use 'options(ergm.term=list(version="3.9.4"))' to
## use old behavior.
```

```
#Twitter data
index1<-sample(c(1:ncol(mat1)),20)
mat1_sub<-mat1[index1,index1]
ergm(mat1_sub~edges+sender + receiver + mutual)
```

```
## Observed statistic(s) sender5, sender7, sender17, receiver4, and receiver19 are at their smallest at
```

```
## 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 0.9416695933239.
## The log-likelihood improved by 2.091.
## Iteration 2 of at most 20:
## Optimizing with step length 1.
## The log-likelihood improved by 0.1003.
## Step length converged once. Increasing MCMC sample size.
## Iteration 3 of at most 20:
## Optimizing with step length 1.
## The log-likelihood improved by 0.04711.
## Step length converged twice. Stopping.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Using 20 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 1
## This model was fit using MCMC.  To examine model diagnostics and
## check for degeneracy, use the mcmc.diagnostics() function.
## MCMC sample of size 4096 based on:
##      edges     sender2     sender3     sender4     sender5     sender6
##   -2.10967    -1.58056    -1.27846    -1.82870        -Inf    -0.84118
##    sender7     sender8     sender9    sender10    sender11    sender12
##       -Inf    -0.53054    -1.96758    -0.44229    -1.35701    -1.38774
##   sender13    sender14    sender15    sender16    sender17    sender18
##   -2.99515    -3.34248    -2.73512    -0.71216        -Inf    -0.94011
##   sender19    sender20   receiver2   receiver3   receiver4   receiver5
##   -2.81241    -2.34789     1.66503     1.20664        -Inf     2.61244
##  receiver6   receiver7   receiver8   receiver9  receiver10  receiver11
##    1.59141     4.43354    -1.51981     0.24622     0.26942    -0.17474
## receiver12  receiver13  receiver14  receiver15  receiver16  receiver17
##    1.99220     4.06257     0.08898     1.52312     0.88703     0.61137
## receiver18  receiver19  receiver20      mutual
##    0.61273        -Inf    -0.28727     4.09436
##
## Monte Carlo MLE Coefficients:
##      edges     sender2     sender3     sender4     sender5     sender6
##    -2.0985     -1.6230     -1.3243     -1.8818        -Inf     -0.8770
##    sender7     sender8     sender9    sender10    sender11    sender12
##       -Inf     -0.5535     -2.0540     -0.4774     -1.3941     -1.4045
##   sender13    sender14    sender15    sender16    sender17    sender18
##    -2.9492     -3.4465     -2.7390     -0.7996        -Inf     -0.9836
##   sender19    sender20   receiver2   receiver3   receiver4   receiver5
##    -2.8879     -2.3965      1.7492      1.1790        -Inf      2.6487
##  receiver6   receiver7   receiver8   receiver9  receiver10  receiver11
##     1.6551      4.4372     -1.4756      0.2621      0.3020     -0.1570
## receiver12  receiver13  receiver14  receiver15  receiver16  receiver17
##     1.9864      4.0593      0.1221      1.5529      0.8829      0.6099
## receiver18  receiver19  receiver20      mutual
##     0.5913        -Inf     -0.3603      4.1098
```

```r
#Enron data
index2<-sample(c(1:ncol(mat2)),20)
mat2_sub<-mat2[index2,index2]
ergm(mat2_sub~edges+sender + receiver + mutual)
```

```
## Warning in ergm(mat2_sub ~ edges + sender + receiver + mutual): Network is
## empty and no target stats are specified.
```

```
## Observed statistic(s) edges, sender2, sender3, sender4, sender5, sender6, sender7, sender8, sender9,
## All terms are either offsets or extreme values. No optimization is performed.
```

```
##
## MLE Coefficients:
##  [1]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
## [12]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
## [23]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
## [34]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
```