Prediction with lasso

Michael YANG

12/6/2018

1.

library(igraph)

size=matrix(1,20)  
density=matrix(1,20)  
clustercoef=matrix(1,20)  
for( i in 1:20){  
 dat=read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/adoption/sample',i,'.csv',sep = ""))  
 s=nrow(dat)  
 size[i]=s  
 edge=as.matrix(read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/adoption/cluster',i,'\_edge.csv',sep = ""),header = FALSE))  
 graph=graph\_from\_adjacency\_matrix(edge)  
 d=edge\_density(graph,loops = FALSE)  
 density[i]=d  
 ccoef=transitivity(graph)  
 clustercoef[i]=ccoef  
}   
result=cbind(size,density,clustercoef)  
colnames(result)=c("SIZE","DENSITY","CLUSTERCOEF")  
result

## SIZE DENSITY CLUSTERCOEF  
## [1,] 293 0.01538174 0.2671433  
## [2,] 72 0.14671362 0.5576833  
## [3,] 105 0.04450549 0.4917197  
## [4,] 52 0.07315234 0.2357860  
## [5,] 130 0.15980918 0.4423363  
## [6,] 82 0.04275821 0.2869565  
## [7,] 114 0.02204627 0.1127013  
## [8,] 179 0.03295462 0.3620633  
## [9,] 231 0.01366460 0.1756026  
## [10,] 107 0.04937401 0.3594675  
## [11,] 153 0.03164775 0.3568005  
## [12,] 60 0.09830508 0.5767226  
## [13,] 57 0.05137845 0.1855670  
## [14,] 150 0.02219239 0.4136292  
## [15,] 261 0.01388152 0.1014339  
## [16,] 131 0.01972989 0.0740038  
## [17,] 128 0.02977362 0.1775218  
## [18,] 51 0.05098039 0.1971831  
## [19,] 157 0.01698514 0.0930000  
## [20,] 108 0.02422984 0.1868687

paste("The group with highest density is No.",which.max(density))

## [1] "The group with highest density is No. 5"

paste("The group with highest clustering coefficient is No.",which.max(clustercoef))

## [1] "The group with highest clustering coefficient is No. 12"

2.

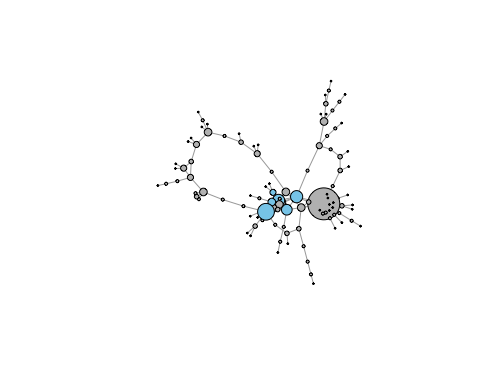
Y=matrix(20:1)  
for(i in 1:20){  
 temp=read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/adoption/sample',i,'.csv',sep = ""))  
 Y[i]=sum(temp[,1])}  
reg=glm(Y~size+density+clustercoef)  
summary(reg)

##   
## Call:  
## glm(formula = Y ~ size + density + clustercoef)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -12.1706 -1.5738 0.8956 2.5702 7.6613   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.03792 3.99126 0.260 0.7981   
## size 0.36194 0.01957 18.498 3.18e-12 \*\*\*  
## density 87.28659 40.96005 2.131 0.0489 \*   
## clustercoef -27.08522 10.07164 -2.689 0.0161 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 25.74763)  
##   
## Null deviance: 11586.00 on 19 degrees of freedom  
## Residual deviance: 411.96 on 16 degrees of freedom  
## AIC: 127.26  
##   
## Number of Fisher Scoring iterations: 2

under the regression adoption respecting to size, density, network clustercoefficent with significant level of 0.05, all the three factors have significant impact on the adoption.

3.

l=0.79  
edge=as.matrix(read.csv('/Users/yangzhenxiong/Documents/R/PS3/adoption/cluster7\_edge.csv',sep=',',header=FALSE))  
g=graph.adjacency(edge,mode="undirected",weighted=NULL)  
ce=centr\_degree(g,mode="total",normalized = T)  
V(g)$color = ifelse(edge[,1]==1,"skyblue","grey")  
plot.igraph(g,vertex.size=((ce$res)\*1.5),vertex.shape="circle",layout=layout.fruchterman.reingold,vertex.label=NA,edge.arrow.size=0.5,arrow.size=1,arrow.width=1,xlim=c(-l,l),ylim=c(-l,l))



4. For this question, we apply the LASSO regression to select variables and remove some the unrealted ones. Then we use the selected variables to do the logit regression.

#find all the possible variables and construct the database.  
  
library(glmnet)

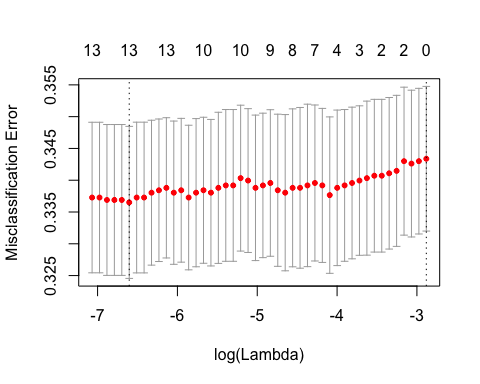
## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.5.2

## Loading required package: foreach

## Loaded glmnet 2.0-16

mydata=data.frame()  
for( i in 1:20){  
 dat=read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/adoption/sample',i,'.csv',sep = ""))  
 edge=as.matrix(read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/adoption/cluster',i,'\_edge.csv',sep = ""),header = FALSE))  
 g=graph\_from\_adjacency\_matrix(edge)  
 deg=degree(g,mode="in")  
 size=nrow(dat)  
 eigen=eigen\_centrality(g,directed = FALSE)  
 cdeg=centr\_degree(g,mode="in")  
 dens=edge\_density(graph,loops = FALSE)  
 g\_trans=transitivity(g,type="global")  
 l\_trans=transitivity(g,type="local")  
 bet <- betweenness(g, directed = TRUE)  
 clo <- closeness(g, mode = "out")  
 apl=average.path.length(g, directed=TRUE, unconnected=TRUE)  
 diam=diameter(g, directed = TRUE)  
 ass=assortativity\_degree(g,directed = TRUE)   
 mdata=cbind(dat,deg,size,cdeg$centralization,dens,g\_trans,l\_trans,bet,clo,apl,ass,diam,eigen$vector)  
 mydata=rbind(mydata,mdata)}  
write.csv(mydata,file="/Users/yangzhenxiong/Documents/R/PS3/adoption/mydata.csv")  
  
#use the cross-validation model to select the factors.  
  
library(glmnet)  
edata=as.matrix(read.csv("/Users/yangzhenxiong/Documents/R/PS3/adoption/mydata.csv"))  
x1=edata[,c(3:17)]  
y1=edata[,2]  
x=apply(x1,2,as.numeric)  
y=as.numeric(y1)  
cvglm=cv.glmnet(x,y,family="binomial",type.measure ="class" )  
plot(cvglm)



#choose the value of lambda that gives the minimum mean of cross-validated error   
  
min=cvglm$lambda.min  
coef(cvglm,s=min)

## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -0.7259009623  
## gender 0.0196613755  
## age 0.0084555719  
## smart -0.2360946406  
## deg 0.0470887721  
## size -0.0007418766  
## cdeg.centralization 0.0429288618  
## dens .   
## g\_trans -0.5613982469  
## l\_trans -0.4585579024  
## bet 0.0001376903  
## clo 5.4190087465  
## apl .   
## ass -0.3652339829  
## diam -0.0058214778  
## eigen.vector -0.9917472233

#use glm to regress the model.  
  
dataf=as.data.frame(read.csv("/Users/yangzhenxiong/Documents/R/PS3/adoption/mydata.csv"))  
model=glm(adoption~age+smart+deg+size+cdeg.centralization+g\_trans+l\_trans+bet+ass+eigen.vector,family = binomial(),data=dataf)  
summary(model)

##   
## Call:  
## glm(formula = adoption ~ age + smart + deg + size + cdeg.centralization +   
## g\_trans + l\_trans + bet + ass + eigen.vector, family = binomial(),   
## data = dataf)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1706 -0.9129 -0.8165 1.3762 1.9460   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.936e-01 3.128e-01 -2.537 0.011181 \*   
## age 9.196e-03 5.667e-03 1.623 0.104638   
## smart -2.500e-01 8.558e-02 -2.922 0.003480 \*\*   
## deg 5.229e-02 1.074e-02 4.869 1.12e-06 \*\*\*  
## size -9.532e-04 6.577e-04 -1.449 0.147276   
## cdeg.centralization 2.325e-01 5.542e-01 0.420 0.674813   
## g\_trans -5.988e-01 5.051e-01 -1.186 0.235805   
## l\_trans -4.660e-01 6.900e-01 -0.675 0.499452   
## bet 1.427e-04 3.837e-05 3.718 0.000201 \*\*\*  
## ass -3.461e-01 4.310e-01 -0.803 0.422019   
## eigen.vector -1.143e+00 3.431e-01 -3.332 0.000862 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3371.9 on 2620 degrees of freedom  
## Residual deviance: 3285.7 on 2610 degrees of freedom  
## AIC: 3307.7  
##   
## Number of Fisher Scoring iterations: 4

Answer: In our result, we find that the variables, including smart,degree,betweeness centrality,eigenvector centrality, have significant impact on the adoption. The result is shown above.

5.

#Use the LASS to predict the original data and find the threholds value.  
  
pre\_original=predict(cvglm,newx=x,type="response",s=min,family=binomial(link = logit))  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

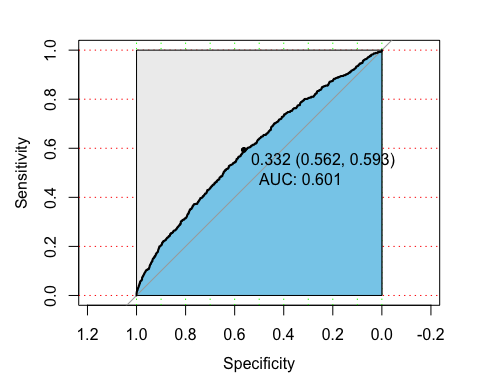
## The following object is masked from 'package:glmnet':  
##   
## auc

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

mroc=roc(mydata$adoption,pre=pre\_original)

## Warning in roc.default(mydata$adoption, pre = pre\_original): Deprecated use  
## a matrix as predictor. Unexpected results may be produced, please pass a  
## numeric vector.

plot(mroc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),  
 grid.col=c("green", "red"), max.auc.polygon=TRUE,  
 auc.polygon.col="skyblue", print.thres=TRUE)



#apply the LASS and threshold to predict the new data.  
  
apre=matrix(20:1)  
for (i in 21:40){  
dat=read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/prediction/prediction',i,'.csv',sep = ""))  
 edge=as.matrix(read.csv(paste('/Users/yangzhenxiong/Documents/R/PS3/prediction/cluster',i,'\_edge.csv',sep = ""),header = FALSE))  
 g=graph\_from\_adjacency\_matrix(edge)  
 deg=degree(g,mode="in")  
 size=nrow(dat)  
 eigen=eigen\_centrality(g,directed = FALSE)  
 cdeg=centr\_degree(g,mode="in")  
 dens=edge\_density(graph,loops = FALSE)  
 g\_trans=transitivity(g,type="global")  
 l\_trans=transitivity(g,type="local")  
 bet <- betweenness(g, directed = TRUE)  
 clo <- closeness(g, mode = "out")  
 apl=average.path.length(g, directed=TRUE, unconnected=TRUE)  
 diam=diameter(g, directed = TRUE)  
 ass=assortativity\_degree(g,directed = TRUE)   
 ndata=as.matrix(cbind(dat,deg,size,cdeg$centralization,dens,g\_trans,l\_trans,bet,clo,apl,ass,diam,eigen$vector))  
 pre=predict(cvglm,newx=ndata,type="response",s=min,family=binomial)  
 adoption=ifelse(pre>=0.332,1,0)  
 apre[i-20]=sum(adoption)  
}  
RANK=matrix()  
RANK=t(rbind(order(apre,decreasing = TRUE)+20,sort(apre,decreasing = TRUE)))  
colnames(RANK)=c("No.","Amount.")  
RANK

## No. Amount.  
## [1,] 28 115  
## [2,] 30 83  
## [3,] 21 80  
## [4,] 34 74  
## [5,] 26 67  
## [6,] 25 62  
## [7,] 32 55  
## [8,] 39 54  
## [9,] 27 52  
## [10,] 37 46  
## [11,] 38 46  
## [12,] 23 44  
## [13,] 36 44  
## [14,] 35 40  
## [15,] 40 40  
## [16,] 22 39  
## [17,] 33 36  
## [18,] 31 33  
## [19,] 29 31  
## [20,] 24 21

RANK[1:10,1:2]

## No. Amount.  
## [1,] 28 115  
## [2,] 30 83  
## [3,] 21 80  
## [4,] 34 74  
## [5,] 26 67  
## [6,] 25 62  
## [7,] 32 55  
## [8,] 39 54  
## [9,] 27 52  
## [10,] 37 46

The rank result is shown above.