GEORGE WASHINGTON UNIVERSITY
6289,2018 FALL

Report of Homework 3

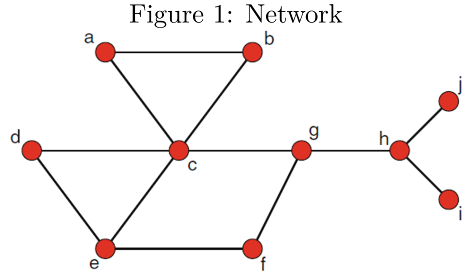
Jiaying Liu

September 30, 2018

1 Question

A network graph is given on the below. For node g, manually validate the degree centrality, closeness centrality and the betweenness centrality. Identify the 2-cores and 3-cores. Use the package modularity and cluster$_walktraptoseperatethesubgroup$.

Figure 1: Network



1.1 Theory

The centrality is usually defines the prominence for undiected network. There are many different measures can be used to evaluate centrality.

- Degree Centrality: The number of edge for a node. This is the simplest measure of centrality.
- Closeness Centrality: This is the measurement focus on how close each node is to every other node in a network. It need to find the shortest path.

$$C_C(v) = (|V(G)| - 1)/sum(d(v, i), iinV(G), i! = v)$$

- Betweenness centrality: it measures the extent that a node sits ”between” pairs of other nodes in the network, such that a path between other nodes has to go through that node.

$$C_B(n_i) = sum(g_{jk}(n_i)/g_{jk})$$

K-Cores. It is a maximal subgraph where each vertex is connected to at least k other vertices in the subgraph.It is a measure of network structure The extent to which nodes show clustering patterns with greater density within each cluster and less density among different clusters In an algorithm, it can be used as a target function for maximization (or other purposes)

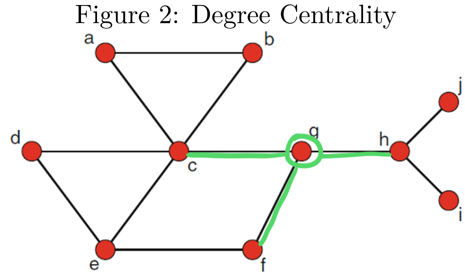
The Modularity is an important network characteristic used in many community detection algorithms

1.2 Result

1.2.1 Degree Centrality

From the figure 2, we can see there are three edge link to the node G. So there are 3 degree centrality

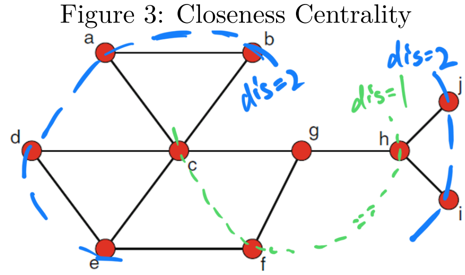
Figure 2: Degree Centrality



1.2.2 Closeness Centrality

From the figure 3, we can see that there are 10 node for total. And for the edge g, the shortest distance between node g and other node are three for 1 distance and six for 2 distance. so the sum distance is 15. According to the equation, the $closenessCentrality = (10 - 1)/(3 * 1 + 6 * 2) = 0.6$

Figure 3: Closeness Centrality



1.2.3 Betweenness Centrality

According to the defination of Betweenness Centrality. Now I need to find out the proportion of all pair shortest path that contain node g. Those shortest path that do not contain node g, the proportion will be 0. From the table 1, we can see that the value of betweenness is $(0.5 * 3 + 1 * 18 = 19.5$

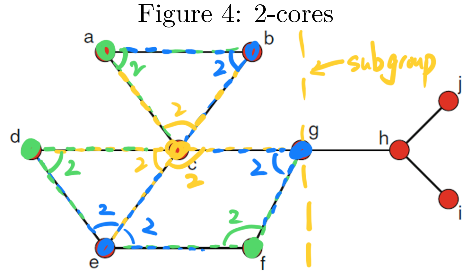
Table 1: All pair of path that contain g

pair	proportion
af	0.5
ah	1
ai	1
aj	1
bf	0.5
bh	1
bi	1
bj	1
cf	0.5
ch	1
ci	1
cj	1
dh	1
di	1
dj	1
eh	1
ei	1
ej	1
fh	1
fi	1
fj	1

1.2.4 K cores

To find out the k core, the first step is to find out the biggest subgroup. And then depend on the definition, we can identify the k core. And from the following figure 4, we can find that there are no 3 core but 10 2 cores.

Figure 4: 2-cores



1.2.5 Modularity

To calculate the modularity, the first thing is to identify. I try four different group and find out the best score is 0.3229. The score of modularity more close to one, the best the group been seprated.

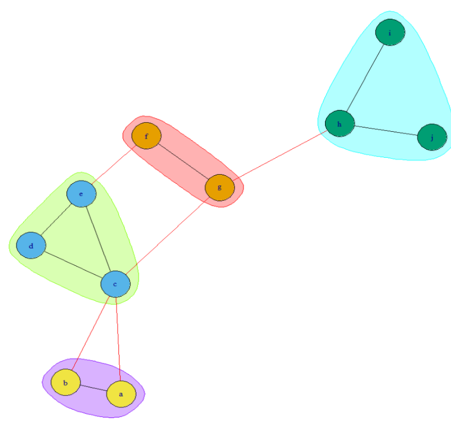
Table 2: The score of modularity

group	modularity
(abcdefg)(hij)	0.2465
(abc)(edfg)(hig)	0.3090
(abc)(de)(fg)(hij)	0.3125
(abcde)(fg)(hij)	0.3229

1.2.6 Cluster Walkstrip

By using the function cluster walkstrip, we can get a anto separate graph. The result was shown below.

Figure 5: Cluster Walkstrip



```
library(statnet)
netmat<- rbind(
  c(1,2),
  c(1,3),
  c(2,3),
  c(3,4),
  c(3,5),
  c(3,7),
  c(4,5),
  c(5,6),
```

```r
  c(6,7),
  c(7,8),
  c(9,8),
  c(10,8)
)
net<-network(netmat,matrix.type='edgelist',direct=F)
network.vertex.names(net)<-c("a","b","c","d","e","f","g","h","i","j")
summary(net)

#detach("package:statnet", unload=TRUE)
#library(igraph)
#library(intergraph)
net1=asIgraph(net)
coreness <- graph.coreness(net1)
table(coreness)
maxCoreness <- max(coreness)
maxCoreness

V(net1)$name <- coreness

net11_3 <- net1
net12_3 <- induced.subgraph(net1,
vids=which(coreness > 1))
net13_3 <- induced.subgraph(net1,
vids=which(coreness > 2))

lay <- layout.fruchterman.reingold(net1)
op <- par(mfrow=c(1,2),mar = c(1,0,2,0))
plot(net11_3,layout=lay,main="All k-cores")
plot(net12_3,layout=lay[
  which(coreness > 1),], main="k-cores 2-2")
par(op)

modularity(net1, coreness)

V(net1)$grp_1 <-c(1,1,1,2,2,3,3,4,4,4)
modularity(net1, V(net1)$grp_1)

V(net1)$grp_2 <- c(1,1,1,1,1,2,2,3,3,3)
modularity(net1, V(net1)$grp_2)

V(net1)$grp_3<- c(1,1,1,2,2,2,2,3,3,3)
modularity(net1, V(net1)$grp_3)
```

V(net1)$name = c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j") cw < -cluster_walktrap(net1) membership(cw) m$