

# Week 13 - Social Network Mining

James Quacinella

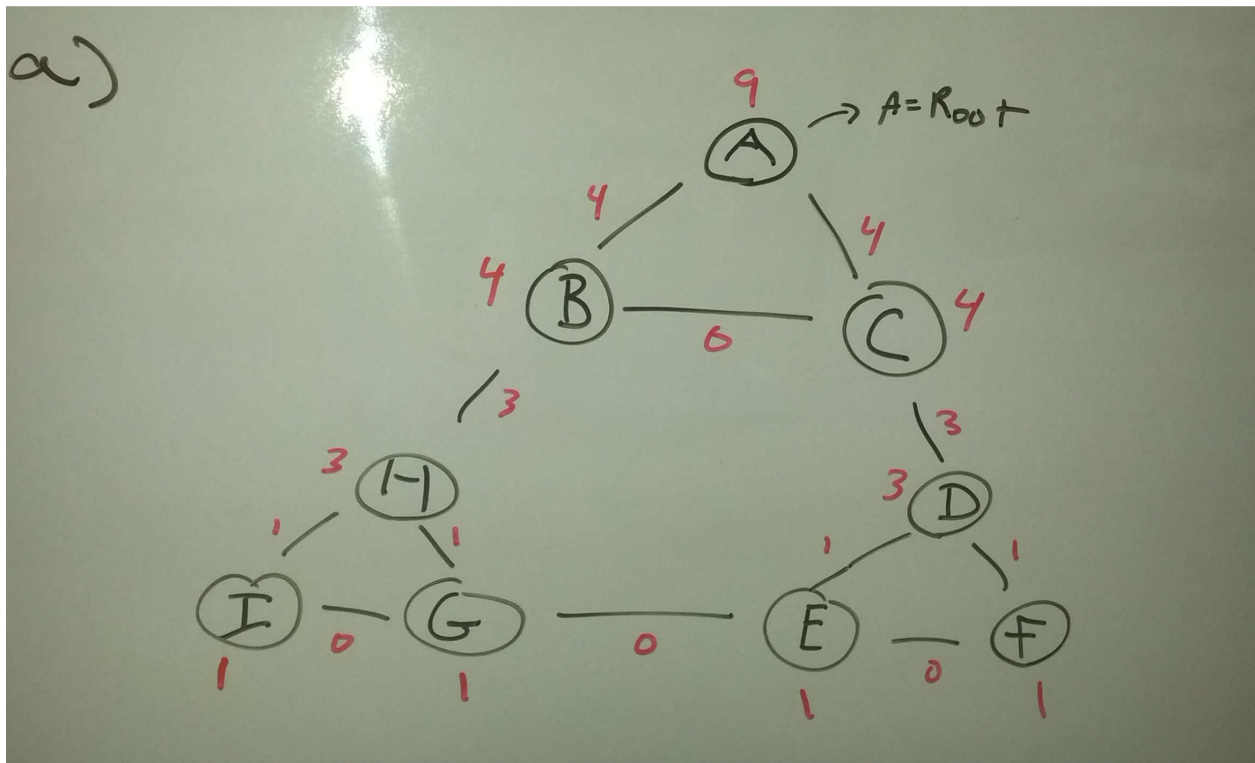
11/14/2015

## 10.2.1

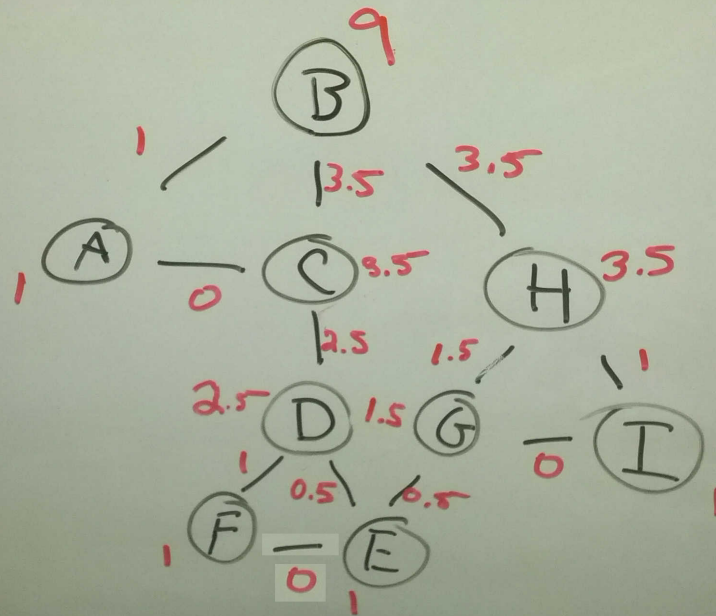
Figure 10.9 is an example of a social-network graph. Use the Girvan-Newman approach to find the number of shortest paths from each of the following nodes that pass through each of the edges. (a) A (b) B.

**Answer**

(a)



(B)



## 10.2.2 (section 10.2.6)

Using symmetry, the calculations of Exercise 10.2.1 are all you need to compute the betweenness of each edge. Do the calculation.

### Answer

Having the diagram start with node A is the same diagram that would result if the root was I or F, with the symmetry being a mirror flip. That means that diagram can be used 3 times in the summation calculation. The other diagram with root node B is the same that would result for the other nodes C, E and H, with the symmetry being a mirror flip. For Nodes D and G, we need to do two flips in order to get the right orientation.

My work is all done in an attached spread sheet. To explain my methodology, I hand calculated the diagram for root node being A and B. In the spreadsheet, for the other nodes, I come up with a node mapping, which maps a node in the original diagram to the node it would match to after the symmetry is applied. For example, to get the numbers for root node I, we can take the root A diagram and flip over F (meaning flip the diagram across an axis starting with F that traverses the diagram ). Using this mapping, I map edges to new edges and copy the calculated value from the diagram for that ‘mapped edged’.

Here are the final results:

```
edges <- c("A => B", "A => C", "B => C", "B => H", "C => D", "D => F",  
          "H => I", "D => E", "G => H", "E => F", "G => I", "E => G")  
scores <- c(4, 4, 6.5, 9.5, 9.5, 4, 4, 6.5, 6.5, 4, 4, 9.5)  
  
library(knitr)  
df <- data.frame(edges=edges, scores=scores)  
kable(df)
```

edges	scores
A => B	4.0
A => C	4.0
B => C	6.5
B => H	9.5
C => D	9.5
D => F	4.0
H => I	4.0
D => E	6.5
G => H	6.5
E => F	4.0
G => I	4.0
E => G	9.5

## Exercise 10.4.1

For the graph of Fig. 10.9, construct:

- (a) The adjacency matrix.
- (b) The degree matrix.
- (c) The Laplacian matrix.

## Answer

```
# Setup the degree matrix
D = matrix(c(2,0,0,0,0,0,0,0,0,
             0,3,0,0,0,0,0,0,0,
             0,0,3,0,0,0,0,0,0,
             0,0,0,3,0,0,0,0,0,
             0,0,0,0,3,0,0,0,0,
             0,0,0,0,0,2,0,0,0,
             0,0,0,0,0,0,3,0,0,
             0,0,0,0,0,0,0,3,0,
             0,0,0,0,0,0,0,0,2), ncol=9)

# Setup col and row names
colnames(D)<-c('A','B','C','D','E','F','G','H','I')
rownames(D)<-c('A','B','C','D','E','F','G','H','I')

# Write up the adjacency matrix
#   A B C D E F G H I
A = matrix(c(0,1,1,0,0,0,0,0,0, # A
             1,0,1,0,0,0,0,1,0, # B
             1,1,0,1,0,0,0,0,0, # C
             0,0,1,0,1,1,0,0,0, # D
             0,0,0,1,0,1,1,0,0, # E
             0,0,0,1,1,0,0,0,0, # F
             0,0,0,0,1,0,0,1,1, # G
             0,1,0,0,0,0,1,0,1, # H
             0,0,0,0,0,0,1,1,0), ncol=9)

# Setup col and row names
colnames(A)<-c('A','B','C','D','E','F','G','H','I')
rownames(A)<-c('A','B','C','D','E','F','G','H','I')

# Calculate Laplacian
L = D - A
```

```
# Print results
```

```
D
```

```
##  A B C D E F G H I
## A 2 0 0 0 0 0 0 0 0
## B 0 3 0 0 0 0 0 0 0
## C 0 0 3 0 0 0 0 0 0
## D 0 0 0 3 0 0 0 0 0
## E 0 0 0 0 3 0 0 0 0
## F 0 0 0 0 0 2 0 0 0
## G 0 0 0 0 0 0 3 0 0
## H 0 0 0 0 0 0 0 3 0
## I 0 0 0 0 0 0 0 0 2
```

```
A
```

```
##  A B C D E F G H I
## A 0 1 1 0 0 0 0 0 0
## B 1 0 1 0 0 0 0 0 1
## C 1 1 0 1 0 0 0 0 0
## D 0 0 1 0 1 1 0 0 0
## E 0 0 0 1 0 1 1 0 0
## F 0 0 0 1 1 0 0 0 0
## G 0 0 0 0 1 0 0 1 1
## H 0 1 0 0 0 0 1 0 1
## I 0 0 0 0 0 0 1 1 0
```

```
L
```

```
##  A B C D E F G H I
## A 2 -1 -1 0 0 0 0 0 0
## B -1 3 -1 0 0 0 0 0 -1
## C -1 -1 3 -1 0 0 0 0 0
## D 0 0 -1 3 -1 -1 0 0 0
## E 0 0 0 -1 3 -1 -1 0 0
## F 0 0 0 -1 -1 2 0 0 0
## G 0 0 0 0 -1 0 3 -1 -1
## H 0 -1 0 0 0 0 -1 3 -1
## I 0 0 0 0 0 0 -1 -1 2
```