HW2: Community Detection

Marwin Carmo

Introduction

Welcome to Homework Assignment No. 2. For this week's assignment, you'll need to visit the course website. There, you'll find a shiny app to generate the graph you'll need for your homework assignment.

Instructions

- Input your first and last name
- Input the due date of the assignment [May 4, 2025]
- Download the `.RDS` file which contains your graph
- Load that `.RDS` file into your `R` environment and begin the assignment
  - This can be done with the command, 'readRDS()"

  Note: you may review the answers to graphs from any other **past** date. So, if you'd like to check if your code is correct, feel free to do so; however, you may not view the answers for any future dates. **Be sure to submit the graph/answers for the due date above!**

```r
library(igraph)
```

```

Attaching package: 'igraph'

The following objects are masked from 'package:stats':

    decompose, spectrum
```

```
The following object is masked from 'package:base':

    union
```

```r
library(ppclust)
```

```r
g <- readRDS("hw2_graph.rds")
```

Below, you will find a template of the questions and fields to provide answers in either R or text format. Please use a mix of code and text to answer each question.

Questions

Question 1. Save the adjacency matrix of your graph object. Refer to W1 for the code. Print the first 3 rows of your matrix using head().

Answer

```r
amat <- as_adjacency_matrix(g, sparse = FALSE)
ng <- graph_from_adjacency_matrix(amat,
                                 mode = "undirected",
                                 diag = FALSE,
                                 weighted = TRUE)
head(amat, 3)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    0    1    1    1    1    1    0    0    0     1     0     0     1     0
[2,]    1    0    1    1    1    1    0    0    0     0     0     1     0     0
[3,]    1    1    0    1    1    1    0    1    1     0     0     1     1     1
     [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
[1,]     0     0     0     0     0     1     0     0     0     1     1     1
[2,]     0     0     0     0     1     1     0     0     0     0     0     0
[3,]     1     1     0     0     1     0     1     0     1     1     0     1
     [,27] [,28] [,29] [,30]
[1,]     0     0     1     1
[2,]     0     0     0     1
[3,]     1     1     1     1
```

Question 2. Apply k-means clustering to your adjacency matrix using a range of k-values and save the results. Print the "withinss" for the first three items in the list (hint: use –> results[[i]][["withinss"]], where i = item index). As k increases, does the within-cluster variation for each cluster increase or decrease?

Answer

As k increases, we see a general reduction in the within-cluster SS.

```r
results.1 <- lapply(1:5, function(k) kmeans(amat,
                                     centers = k,
                                     iter.max = 1000,
                                     nstart = 30))



results.1[[1]][["withinss"]]
```

```
[1] 215
```

```r
results.1[[2]][["withinss"]]
```

```
[1] 123.33333  70.16667
```

```r
results.1[[3]][["withinss"]]
```

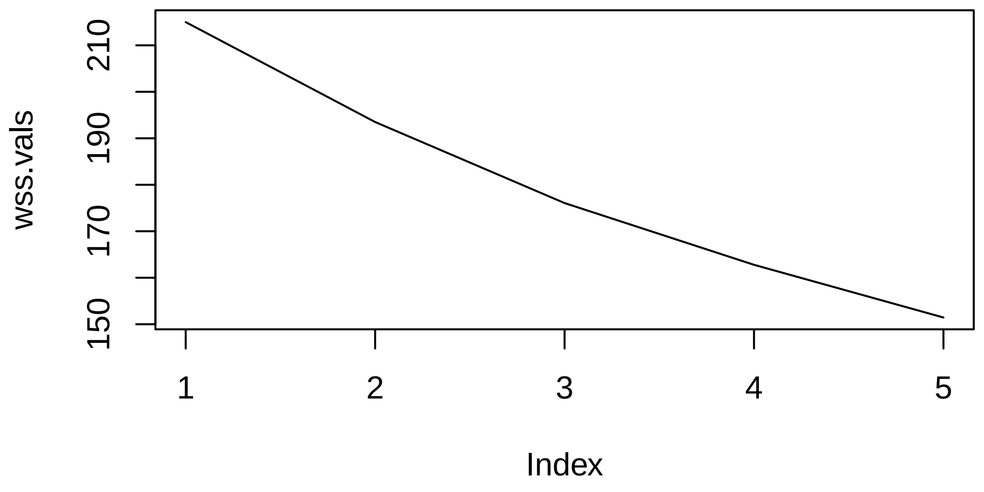
```
[1] 46.25000 62.90909 66.90909
```

Question 3. Plot the withinss values for all list items in your k-means clustering results. Based on the plot, what is the most optimal value of 'k' (i.e., number of clusters in your network)?

Answer

Apparently, there are no clusters (i.e., the nodes form a single community) given that there is no clear 'elbow' on the line. The pattern persists even when increasing to the maximum possible value of k. One could maybe say 2 or 3, but the break in the line is very subtle.

```r
wss.vals <- sapply(results.1, function(res) sum(res$withinss))
plot(wss.vals, type = "l")
```



Question 4. What is Q for the above finding (hint: you will need use results[[k]][[“cluster”]] in the membership argument of modularity())?

Answer

```r
q_kmeans <- modularity(g, membership = results.1[[2]][["cluster"]])
q_kmeans
```

```
[1] 0.110696
```

***Question 5*. Apply fuzzy c-means clustering to your adjacency matrix using a range of k-values and save the results. Using the optimal value of k from Q3, print the first 10 rows for the cluster confidence matrix (u). Which node is most likely to be in Cluster k-1?**

Answer

With a lack of a better option, we can use two as the optimal number of clusters. However, it is clear from these results that all nodes are equally likely to belong in any of the two clusters.

```r
res_fuzzy <- lapply(1:5, function(k) fcm(amat,
                                         centers = k,
                                         m = 2))

head(res_fuzzy[[2]]$u, 10)
```

```
   Cluster 1 Cluster 2
1        0.5       0.5
2        0.5       0.5
3        0.5       0.5
4        0.5       0.5
5        0.5       0.5
6        0.5       0.5
7        0.5       0.5
8        0.5       0.5
9        0.5       0.5
10       0.5       0.5
```

```r
q_fuzzy <- modularity(ng,
           res_fuzzy[[2]]$cluster)
q_fuzzy
```

```
[1] 0.1145482
```

***Question 6*. Apply the Walktrap algorithm to your graph, with t = 4 (default value). Plot the resulting graph and highlight the groups identified in your Walktrap results.**

Answer

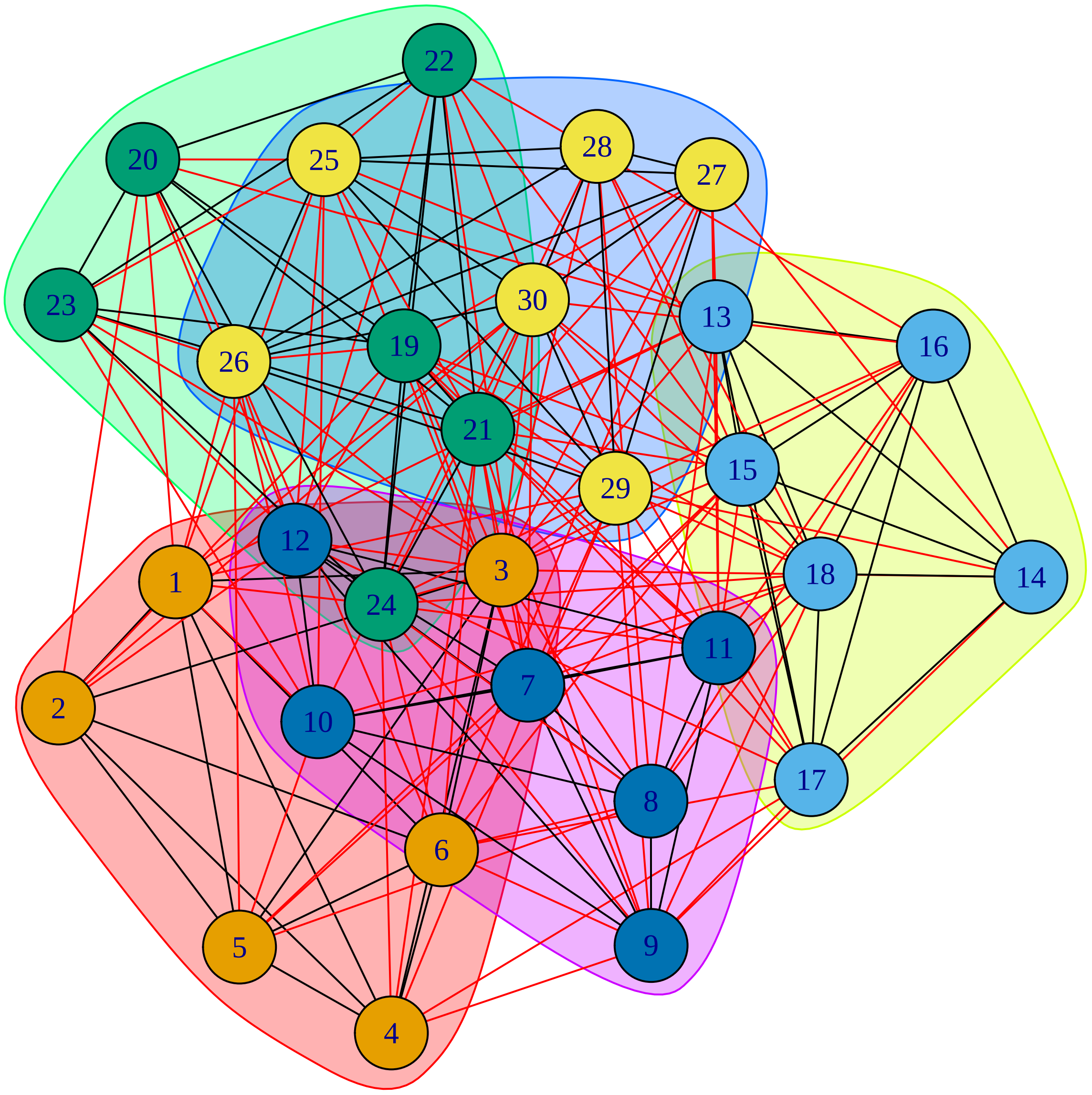
Walktrap suggests four groups. However, this pattern is not visually clear from the resulting plot–the nodes appear “randomly” scattered in the graph. This result adds to the intuition elicited by the k-means analysis about the lack of clear communities within this network.

```r
groups <- cluster_walktrap(g, steps=4)

q_walktrap <- modularity(ng, membership(groups))
q_walktrap
```

```
[1] 0.1804607
```

```r
plot(groups, ng)
```



***Question 7*. Apply spectral clustering to your graph. You will need both your adjacency matrix and graph object to derive your Laplacian matrix. Use the same value of k as in previous questions. Calculate and report modularity, Q for your results.**

Answer

```
D.mat = diag(1/sqrt(degree(ng)), nrow(amat))
identity = diag(1, dim(D.mat))
Laplacian = identity - D.mat  %*% amat %*% D.mat
```

```r
spectrum = eigen(Laplacian)$vectors[,(nrow(amat)-2):(nrow(amat)-1)]

spec.clust = kmeans(spectrum, centers = 2)

q_spec <- modularity(ng,
           spec.clust[["cluster"]])
q_spec
```

```
[1] 0.1058389
```

***Question 8*. Calculate and compare Q (modularity) for each set of results (k-means, c-means, WalkTrap, spectral). Which approach resulted in the highest Q?**

Answer

The highest Q was obtained from the Walktrap algorithm.

```r
q_kmeans
```

```
[1] 0.110696
```

```r
q_fuzzy
```

```
[1] 0.1145482
```

```r
q_walktrap
```

```
[1] 0.1804607
```

```r
q_spec
```

```
[1] 0.1058389
```