Social Network Analysis Home Assignment 3

Ramil Mirzaaghayev

due date - 03.06.2019 23:59

Table of Contents

# Network communities

Please send your reports to [hse.ntwks@gmail.com](mailto:hse.ntwks@gmail.com) with the subject of of the following structure: *[MAGOLEGO SNA 2017] {LastName} {First Name} HW{Number}*

Late submission policy: -1 point per day

Use this file as a template for your report.  
Support your computations with figures and comments. Send ONLY .Rmd versions of your report.

## Warning: package 'igraph' was built under R version 3.5.3

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

## Yahoo Music Network

For this assignment, you have to load [the part of Yahoo Music Network](https://drive.google.com/file/d/0Bwd5a7NScgMtNElfOFdSbE1UYnM/view). Archive contains network in GML format and corresponding list of artists.

## Put your code here  
z = read.graph("music2K.gml", format = "gml")

Edges in this network appear if enough number of users have given ratings to both music bands. Note, that edges are weighted with similarity of the ratings.

### 1. Preprocessing

This network needs to be preprocessed. Do the steps, described on seminar, to make it look properly. How many vertices and edges were deleted?

## Put your code here  
  
# Preprocess the graph  
g = delete.vertices(z, degree(z) == 0)  
simplify(g)

## IGRAPH 4e3041e U-W- 1996 6668 --   
## + attr: id (v/n), label (v/c), weight (e/n)  
## + edges from 4e3041e:  
## [1] 1-- 639 1--1741 2-- 365 2-- 871 2--1403 2--1516 2--1837  
## [8] 2--1881 3-- 324 3-- 827 3--1425 3--1593 4-- 427 4-- 697  
## [15] 4-- 755 4-- 837 4--1629 5-- 102 5-- 272 5-- 358 5-- 723  
## [22] 5-- 759 5--1118 5--1217 5--1386 5--1513 5--1558 5--1892  
## [29] 6-- 760 6-- 870 6--1246 6--1704 7-- 614 7-- 820 7--1010  
## [36] 7--1097 7--1110 7--1408 7--1544 7--1607 7--1877 7--1928  
## [43] 8-- 56 8-- 190 8-- 487 8-- 512 8-- 841 9-- 271 9-- 526  
## [50] 9-- 935 9-- 997 10-- 216 10-- 291 10-- 384 10--1003 10--1292  
## + ... omitted several edges

#plot(g)  
  
# Check the result of preprocessing  
deltaV = length(V(z))-length(V(g))  
deltaE = length(E(z))-length(E(g))  
  
paste("After preprocessing the size of vertices decreased by",deltaV)

## [1] "After preprocessing the size of vertices decreased by 4"

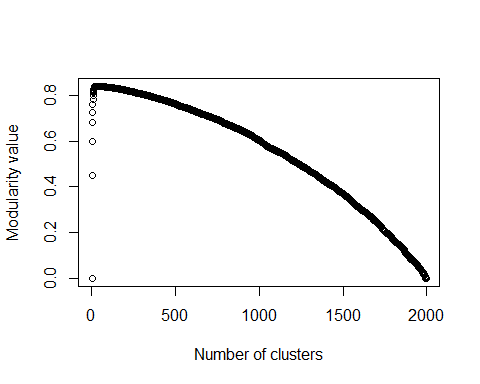
paste("After preprocessing the size of edges decreased by",deltaE)

## [1] "After preprocessing the size of edges decreased by 0"

### 2. Clustering

Define clusters for this networks using one of the algorithms described on lectures and seminars:

## Put your code here  
mm <- fastgreedy.community(g)  
plot(rev(mm$modularity), xlab = 'Number of clusters', ylab = 'Modularity value')



#plot(mm, g)

typeof(mm)

## [1] "list"

Compute density of obtained clusters:

## Put your code here  
#groups(mm)  
#   
density.list = list()  
for (i in seq(length(groups(mm)))){  
 sg = induced\_subgraph(g, mm$membership==i)  
 density.sg = graph.density(sg)  
 density.list[i] = density.sg  
 print(sprintf("Community %d has density = %.4f", i, density.sg))}

## [1] "Community 1 has density = 0.0500"  
## [1] "Community 2 has density = 0.0602"  
## [1] "Community 3 has density = 0.0396"  
## [1] "Community 4 has density = 0.0808"  
## [1] "Community 5 has density = 0.1074"  
## [1] "Community 6 has density = 0.0190"  
## [1] "Community 7 has density = 0.0393"  
## [1] "Community 8 has density = 0.0475"  
## [1] "Community 9 has density = 0.0878"  
## [1] "Community 10 has density = 0.0613"  
## [1] "Community 11 has density = 0.0521"  
## [1] "Community 12 has density = 0.2308"  
## [1] "Community 13 has density = 0.0684"  
## [1] "Community 14 has density = 0.1010"  
## [1] "Community 15 has density = 0.1703"  
## [1] "Community 16 has density = 0.0375"  
## [1] "Community 17 has density = 0.2286"  
## [1] "Community 18 has density = 0.0981"  
## [1] "Community 19 has density = 0.3399"  
## [1] "Community 20 has density = 0.1355"  
## [1] "Community 21 has density = 0.1277"  
## [1] "Community 22 has density = 0.1345"  
## [1] "Community 23 has density = 0.5385"  
## [1] "Community 24 has density = 0.2536"  
## [1] "Community 25 has density = 0.5238"  
## [1] "Community 26 has density = 0.5769"  
## [1] "Community 27 has density = 0.6429"

density.list[[27]]

## [1] 0.6428571

Compute the ratio of inner clusters connections to outer ones:

## Put your code here  
for (i in seq(length(groups(mm)))){  
 sg = induced\_subgraph(g, mm$membership==i)  
 ratio = ecount(sg)/(ecount(g)-ecount(sg))  
 print(sprintf("Community %d has ratio = %.4f", i, ratio))}

## [1] "Community 1 has ratio = 0.0362"  
## [1] "Community 2 has ratio = 0.0393"  
## [1] "Community 3 has ratio = 0.0710"  
## [1] "Community 4 has ratio = 0.0157"  
## [1] "Community 5 has ratio = 0.0227"  
## [1] "Community 6 has ratio = 0.1682"  
## [1] "Community 7 has ratio = 0.0901"  
## [1] "Community 8 has ratio = 0.0507"  
## [1] "Community 9 has ratio = 0.0401"  
## [1] "Community 10 has ratio = 0.0539"  
## [1] "Community 11 has ratio = 0.0623"  
## [1] "Community 12 has ratio = 0.0123"  
## [1] "Community 13 has ratio = 0.0262"  
## [1] "Community 14 has ratio = 0.0230"  
## [1] "Community 15 has ratio = 0.0071"  
## [1] "Community 16 has ratio = 0.0985"  
## [1] "Community 17 has ratio = 0.0073"  
## [1] "Community 18 has ratio = 0.0369"  
## [1] "Community 19 has ratio = 0.0079"  
## [1] "Community 20 has ratio = 0.0115"  
## [1] "Community 21 has ratio = 0.0211"  
## [1] "Community 22 has ratio = 0.0108"  
## [1] "Community 23 has ratio = 0.0063"  
## [1] "Community 24 has ratio = 0.0106"  
## [1] "Community 25 has ratio = 0.0017"  
## [1] "Community 26 has ratio = 0.0068"  
## [1] "Community 27 has ratio = 0.0027"

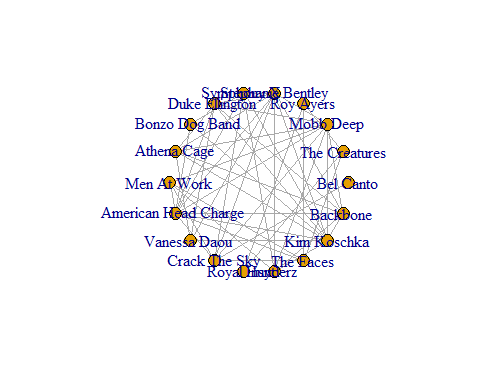
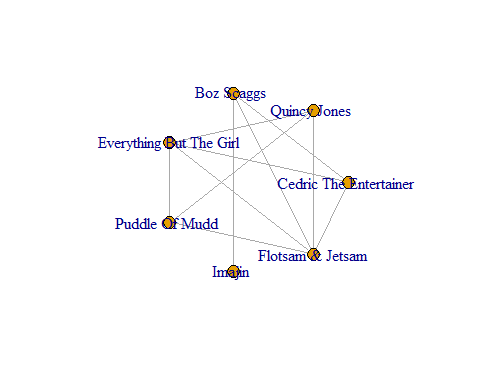
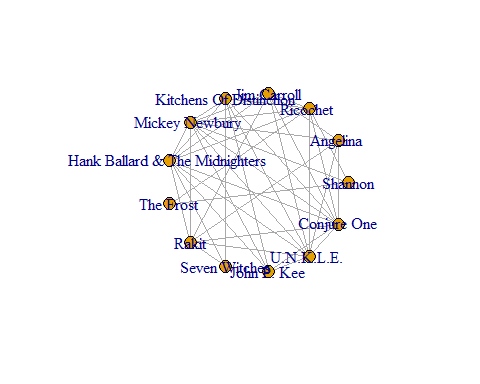
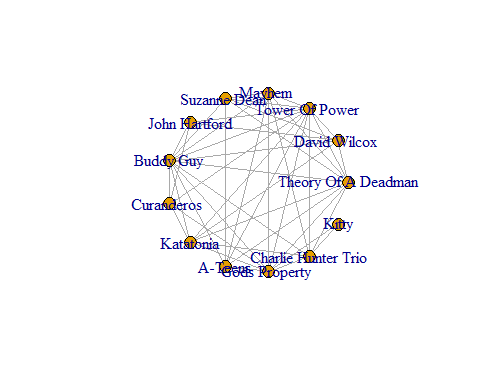
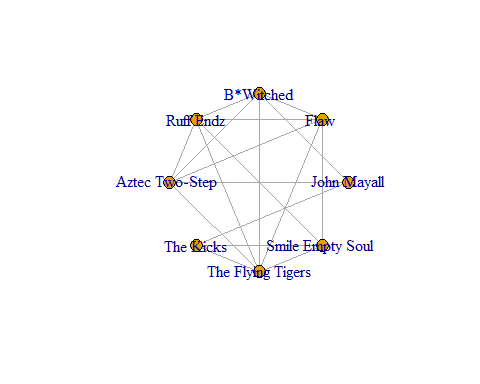
### 3. Visualization & interpretation

Visulize five of the most dense clusters. Use names of artists as node labels on the graph.

## Put your code here  
artists = scan("artists.txt", character(), sep = "\n")  
#artists  
V(g)$label = artists

## Warning in vattrs[[name]][index] <- value: number of items to replace is  
## not a multiple of replacement length

top5 = list()  
for (i in seq(5)){  
   
 i = which.max(density.list)  
 top5=append(top5, density.list[i])  
 density.list[i] = -1  
   
 subgraph = induced.subgraph(g,which(mm$membership == i))  
 plot(subgraph, layout=layout.circle)  
   
}



top5

## [[1]]  
## [1] 0.6428571  
##   
## [[2]]  
## [1] 0.5769231  
##   
## [[3]]  
## [1] 0.5384615  
##   
## [[4]]  
## [1] 0.5238095  
##   
## [[5]]  
## [1] 0.3398693

(Extra task) Try to interpret (using Wikipedia or other resources) obtained clusters.

*Insert comments here*