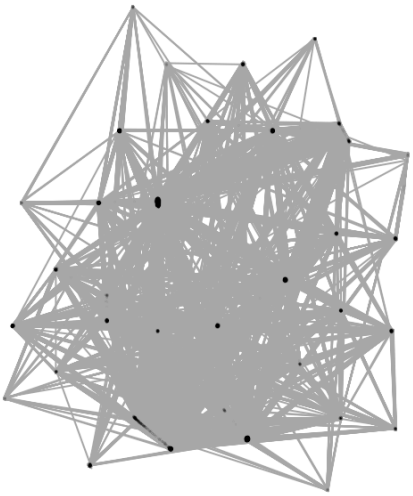
BUS 41201: Big Data

Homework 5

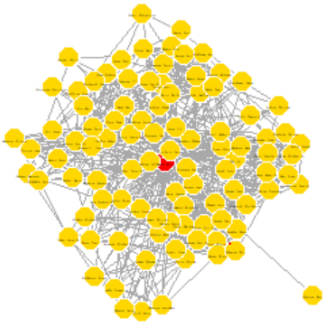
Group 29: Hye-Min Jung, Jayoung Kang, Jeong Lim Kim

[1] The actors network has an edge if the two actors were in the same movie. Plot the entire actors network.

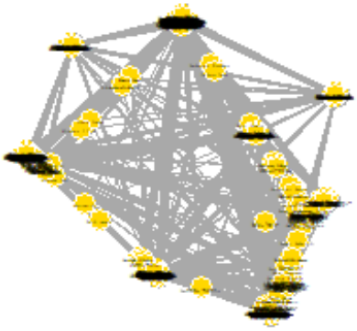


[2] Plot the neighborhoods for “Bacon, Kevin” at orders 1-3. How does the size of the network change with order?

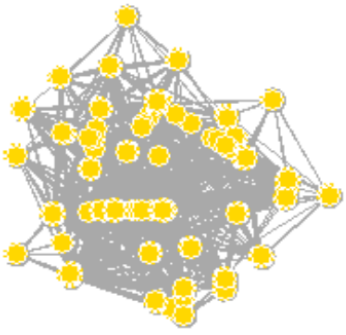
Order = 1



Order = 2



Order = 3



[3] Who were the most common actors? Who were most connected? Pick a pair of actors and describe the shortest path between them.

- Most common: Zivojinovic, Velimir 'Bata'
- Most connected: Dobtcheff, Vernon
- Shortest path between Zivojinovic, Velimir 'Bata' & Dobtcheff, Vernon:
    - "Zivojinovic, Velimir 'Bata'" - "Basic, Relja" - "Dobtcheff, Vernon"

[4] Find pairwise actor-cast association rules with at least 0.01% support and 10% confidence. Describe what you find.

- Under the given conditions there are 92555 rules.
- The range of lift ranges from 25.13333 to 7163.
- The range of confidence ranges from 0.1 to 1
- Interpretation of the highest and lowest lift:
    - The probability of casting Rahi, Sultan is 7163 times higher given that Anjumen is cast in the movie, than when Rahi, Sultan is cast alone.
    - The probability of casting Zivojinovic, Velimir 'Bata' is 25 times higher given that Sokolovic-Bertok, Semka is cast in the movie, than when Zivojinovic, Velimir 'Bata' is cast alone.
- The reason for the high lift range could be due the fact that the association rules limit the confidence and support and also because the movies are from a specific genre during the 80s and 90s. During that time period it is possible that there are fewer prominent actors and actresses and the likelihood that they get cast together is high. Also, we see that the pairwise association rules are usually between actors from the same country so it is possible that we could see communities that represent certain geographic regions if we were to look for them.

Appendix – R Code:

```
## actors network example
setwd("C:/Users/13124/Desktop/Harris/2020-2 Spring/Big Data/HW/hw5/")
library(tidyverse)
library(igraph)

### GRAPH
## read in a graph in the `graphml' format: xml for graphs.
## it warns about pre-specified ids, but we want this here
## (these ids match up with the castlists in movies.txt)
actnet <- read.graph("actors.graphml",format="graphml")

### TRANSACTION
## read in the table of actor ids for movies
## this is a bit complex, because the movie names
## contain all sorts of special characters.
movies <- read.table("movies.txt", sep="\t", row.names=1, as.is=TRUE,
comment.char="", quote="")

## it's a 1 column matrix.  treat it like a vector
movies <- drop(as.matrix(movies))

## each element is a comma-separated set of actor ids.
## use `strsplit' to break these out
movies <- strsplit(movies,",")

## and finally, match ids to names from actnet
casts <- lapply(movies, function(m) V(actnet)$name[match(m,V(actnet)$id)])

## check it
casts['True Romance']

## format as arules transaction baskets
install.packages("arules")
library(arules)
casttrans <- as(casts, "transactions")

## Set up STM information
castsize <- unlist(lapply(casts, function(m) length(m)))

## see ?rep.int: we're just repeating movie names for each cast member
acti <- factor(rep.int(names(casts),times=castsize))

## actors
actj <- factor(unlist(casts), levels=V(actnet)$name)

## format as STM (if you specify without `x', its binary 0/1)
actmat <- sparseMatrix(i=as.numeric(acti),j=as.numeric(actj),
dimnames=list(movie=levels(acti),actor=levels(actj)))

## count the number of appearences by actor
nroles <- colSums(actmat)
names(nroles) <- colnames(actmat)
names(nroles)
```

```
## [1] The actors network has an edge if the two actors were in the
same movie.
## Plot the entire actors network.
actnet <- read.graph("actors.graphml",format="graphml")
actnet_edgelist <- as_edgelist(actnet, names = TRUE)
act_network <- graph.edgelist(actnet_edgelist, directed=FALSE)

faction_vertex_shape = get.vertex.attribute(act_network, "Faction")

act_network_degree = degree(act_network)

plot(act_network,
     edge.arrow.size=.4,
     vertex.shapes = faction_vertex_shape,
     vertex.size = act_network_degree / sum(act_network_degree)*200,
     vertex.label=NA, edge.curved=FALSE)


## [2] Plot the neighborhoods for "Bacon, Kevin" at orders 1-3. How
does the size of the network change with order?
bacon1 <- graph.neighborhood(actnet, order = 1, V(actnet)["Bacon,
Kevin"],mode="in")[[1]]
V(bacon1)$color <- "gold"
V(bacon1)["Bacon, Kevin"]$color <- "red"
plot(bacon1, vertex.label.cex =0.1, vertex.frame.color=0,
edge.arrow.width=.75)

bacon2 <- graph.neighborhood(actnet, order = 2, V(actnet)["Bacon,
Kevin"],mode="in")[[1]]
V(bacon2)$color <- "gold"
V(bacon2)["Bacon, Kevin"]$color <- "red"
plot(bacon2, vertex.label.cex =0.1, vertex.frame.color=0,
edge.arrow.width=.75)

bacon3 <- graph.neighborhood(actnet, order = 3, V(actnet)["Bacon,
Kevin"],mode="in")[[1]]
V(bacon3)$color <- "gold"
V(bacon3)["Bacon, Kevin"]$color <- "red"
plot(bacon3, vertex.label= NA, vertex.frame.color=0,
edge.arrow.width=.75)


## [3] Who were the most common actors? Who were most connected?
## Pick a pair of actors and describe the shortest path between them.
#most common
ordered = order(nroles, decreasing = T)
nroles[ordered[1]]

#most connected
actor_degree = degree(actnet)
order_degree = order(actor_degree, decreasing = T)
actor_degree[order_degree[1]]
```

```
#shortest path
ZtoD <- get.shortest.paths(actnet, from="Zivojinovic, Velimir 'Bata'",
to="Dobtcheff, Vernon")
V(actnet)$name[ZtoD2$vpath[[1]]]


## [4] Find pairwise actor-cast association rules with at least 0.01%
support and 10% confidence.
## Describe what you find.
pairwise <- apriori(casttrans, parameter=list(support=.0001,
confidence=.1, maxlen=2))
inspect(pairwise)

pairs <- labels(pairwise)
pairs <- gsub("\\{|\\}","",pairs)
pairs <- strsplit(pairs," => ")
pairs <- do.call(rbind,pairs)
pairs <- pairs[pairs[,1]!="",] # no lhs
association <- graph.edgelist(pairs)
association <- as.undirected(association)

V(association)$color <- "cyan"
plot(association, vertex.label=NA, vertex.size=3, edge.curved=FALSE)
inspect_movierules <- as.data.frame(inspect(pairwise))
```