

BUSN 41201 HW5

Group 11: Yu-Ting Weng, Mengdi Hao, Elena Li, Minji Park, Sarah Lee

Question 1

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

```
adj_matrix <- matrix(0, length(V(actnet)), length(V(actnet)),
                      dimnames = list(V(actnet)$name, V(actnet)$name))

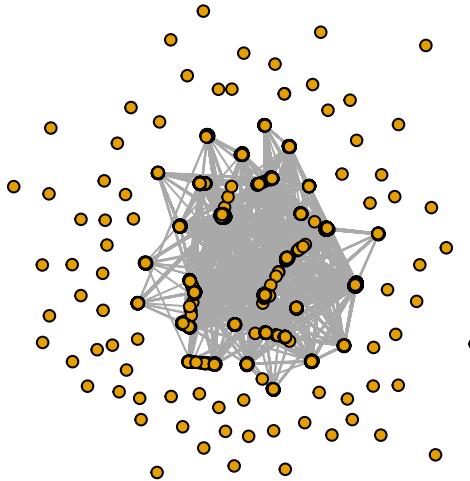
for (cast in casts) {
  if (length(cast) > 1) {
    combos <- combn(cast, 2)
    for (combo in 1:ncol(combos)) {
      pair <- combos[, combo]
      adj_matrix[pair[1], pair[2]] <- 1
      adj_matrix[pair[2], pair[1]] <- 1
    }
  }
}

diag(adj_matrix) <- 0

g <- graph_from_adjacency_matrix(adj_matrix, mode = "undirected", diag = FALSE)

plot(g, vertex.size = 5, vertex.label = NA, edge.arrow.size = 0.5, main = "Actors Network")
```

Actors Network

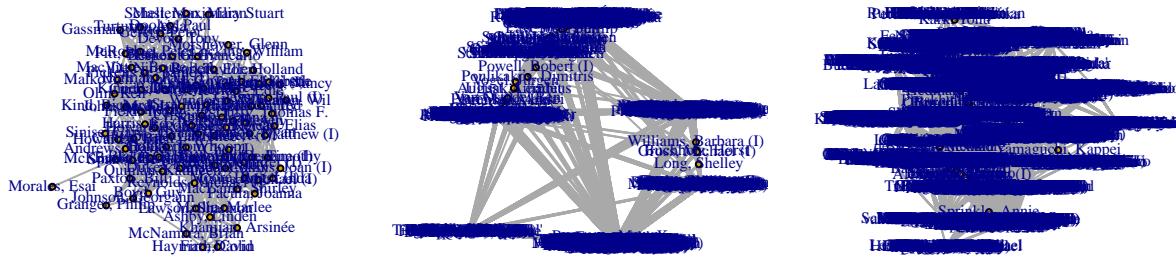


Question 2

- Plot the neighborhoods for “Bacon, Kevin” at orders 1-3.

```
kevin_bacon_id <- which(V(g)$name == "Bacon, Kevin")  
  
par(mfrow=c(1,3))  
for (i in 1:3) {  
  sg <- induced_subgraph(g, unlist(neighborhood(g, order=i, nodes=kevin_bacon_id)))  
  plot(sg, main=paste("Kevin's", i, "-order Neighborhood"), vertex.size=5, vertex.label.cex=0.7)  
}
```

Kevin's 1 –order Neighborhood Kevin's 2 –order Neighborhood Kevin's 3 –order Neighborhood



- How does the size of the network change with order?

Order 1: This includes the immediate neighbors of “Kevin Bacon,” which means all the actors he has directly worked with. **The size of this network is relatively small**, limited to just those actors in the movies that “Kevin Bacon” has appeared in.

Order 2: This includes not only “Kevin Bacon’s” immediate neighbors but also the neighbors of those neighbors. So, it adds actors who have worked with the actors who worked with “Kevin Bacon” but have not necessarily worked with “Kevin Bacon” themselves. **The network grows significantly as it includes one more “degree of separation.”**

Order 3: This further includes the neighbors of the actors in the second-order neighborhood. The network has expanded, including actors three steps removed from “Kevin Bacon.” **The growth from order 2 to order 3 is usually even larger than from order 1 to order 2**, assuming each actor has worked with several other actors.

Question 3

- Who were the most common actors?

```
actor_movie_counts <- colSums(actmat)
most_common_actors <- sort(actor_movie_counts, decreasing = TRUE)

cat("Most common actors (appear in the most movies):\n")

## Most common actors (appear in the most movies):
print(names(most_common_actors)[1:5])
```

```
## [1] "Zivojinovic, Velimir 'Bata'" "Jeremy, Ron"  
## [3] "Doll, Dora"                 "Dobtcheff, Vernon"  
## [5] "Berléand, François"
```

“**Zivojinovic, Velimir ‘Beta’**” is the most common actor.

- Who were most connected?

```
actor_degrees <- degree(actnet)  
most_connected_actors <- sort(actor_degrees, decreasing = TRUE)  
  
cat("Most connected actors (have worked with the most other actors):\n")
```

```
## Most connected actors (have worked with the most other actors):
```

```
print(names(most_connected_actors)[1:5])
```

```
## [1] "Dobtcheff, Vernon"      "Stévenin, Jean-François"  
## [3] "Muel, Jean-Paul"       "Blanche, Roland"  
## [5] "Garrivier, Victor"
```

‘**Dobtcheff, Vernon**’ is the most connected actor.

- Pick a pair of actors and describe the shortest path between them.

```
actor1_name <- "Zivojinovic, Velimir 'Bata'"  
actor2_name <- "Dobtcheff, Vernon"  
  
actor1_id <- which(V(actnet)$name == actor1_name)  
actor2_id <- which(V(actnet)$name == actor2_name)  
  
path <- shortest_paths(actnet, from=actor1_id, to=actor2_id)  
path_vseq <- path$vpath[[1]]  
  
actor_names_in_path <- V(actnet)$name[path_vseq]  
  
cat("The shortest path between", actor1_name, "and", actor2_name, "is:\n")
```

```
## The shortest path between Zivojinovic, Velimir 'Bata' and Dobtcheff, Vernon is:  
cat(paste(actor_names_in_path, collapse=" -> "), "\n")
```

```
## Zivojinovic, Velimir 'Bata' -> Basic, Relja -> Dobtcheff, Vernon
```

This path is the shortest in terms of the number of links (or “degrees of separation”) between the two actors within the movie industry’s network.

Question 4

- Find pairwise actor-cast association rules with at least 0.01% support and 10% confidence.

```
## Apriori  
##  
## Parameter specification:  
##   confidence minval smax arem  aval originalSupport maxtime support minlen  
##           0.1    0.1     1 none FALSE             TRUE      5   1e-04      1  
##   maxlen target  ext  
##           2  rules TRUE  
##  
## Algorithmic control:
```

```
## filter tree heap memopt load sort verbose
##      0.1 TRUE TRUE FALSE TRUE    2     TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[6953 item(s), 14326 transaction(s)] done [0.02s].
## sorting and recoding items ... [6874 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2

## Warning in apriori(casttrans, parameter = list(supp = 1e-04, conf = 0.1, :
## Mining stopped ( maxlen reached). Only patterns up to a length of 2 returned!

## done [0.09s].
## writing ... [92555 rule(s)] done [0.09s].
## creating S4 object ... done [0.01s].
rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)
inspect(head(rules_sorted, 5))
```

```
##      lhs            rhs        support      confidence coverage
## [1] {Anjuman}    => {Rahi, Sultan} 0.0001396063 1.0000000 0.0001396063
## [2] {Rahi, Sultan} => {Anjuman}   0.0001396063 1.0000000 0.0001396063
## [3] {Warrier, Manju} => {Srividya} 0.0001396063 1.0000000 0.0001396063
## [4] {Srividya}      => {Warrier, Manju} 0.0001396063 0.6666667 0.0002094095
## [5] {Nagesh (I)}    => {Rajendraprasad} 0.0001396063 1.0000000 0.0001396063
##      lift      count
## [1] 7163.000 2
## [2] 7163.000 2
## [3] 4775.333 2
## [4] 4775.333 2
## [5] 4775.333 2
```

- Describe What you find.

1. Rule 1 states that when Anjuman appears, Rahi, Sultan also appears, and this happens with 100% confidence, indicating it is always the case in this dataset. The lift is extremely high, suggesting a strong association.
2. Rule 2 is the converse of Rule 1, showing that when Rahi Sultan appears, Anjuman also appears with the same high confidence and lift.
3. Rule 3 involves {Warrier, Manju} leading to {Srividya} with a confidence of 100%, another strong rule.
4. Rule 4 shows {Srividya} leading to {Warrier, Manju} with a lower confidence of about 66.67%, indicating that in about two-thirds of the transactions containing Srividya, Warrier, Manju also appears.
5. Rule 5 has {Nagesh (I)} leading to {Rajendraprasad} with 100% confidence.

The identical high lift values suggest these associations do not occur by chance and are significant within the context of this dataset. It's also important to note that despite the high confidence and lift, the support is very low, meaning that these strong associations are rare within the dataset. The 'count' indicates the number of transactions in which these rules were valid, which is 2 for all these rules, indicating a small dataset or a sparse occurrence of these combinations.

Extra Analysis

What would be a regression based alternative to ARrules? Execute it for a single RHS actor.

```

library(glmnet)

## Loaded glmnet 4.1-8
library(parallel)
library(doParallel)

## Loading required package: foreach
## Loading required package: iterators
no_cores <- detectCores(logical = FALSE)
cl <- makeCluster(no_cores)
registerDoParallel(cl)

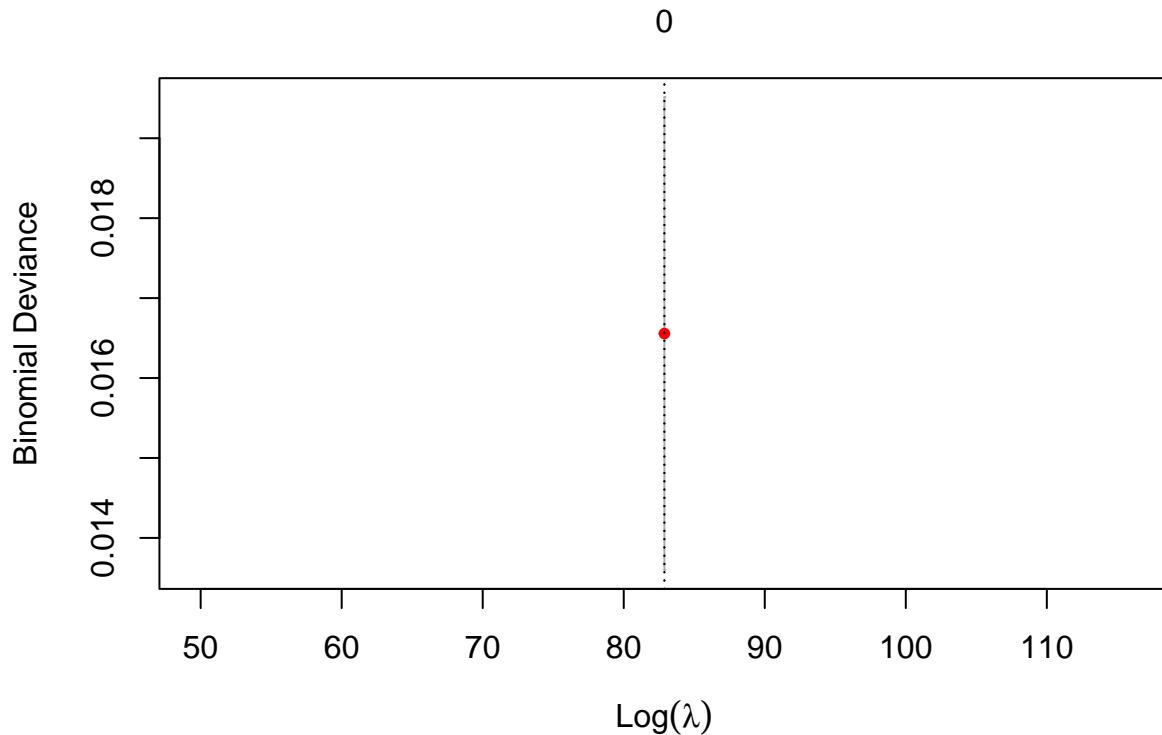
rhs_actor <- "Bacon, Kevin"
response <- as.integer(actmat[, rhs_actor] > 0)
predictors_matrix <- actmat[, colnames(actmat) != rhs_actor]

set.seed(123) # For reproducibility
cv_model <- cv.glmnet(predictors_matrix, response, alpha = 1, family = "binomial", parallel = TRUE)

stopCluster(cl)

plot(cv_model)

```



```

best_coefs <- coef(cv_model, s = "lambda.min")

```

```
print(best_coefs[best_coefs != 0])  
## [1] -6.860734
```