Yun-Chu Chen, Dennie Cheng, Mark Planas, Kailun Xu, Prilliana Yanuar

Prof. S. Dewan

BANA 277

November 9, 2018

Social Network Team Assignment

1. *Delete products that are not books from "products" and "copurchase" files.*

```
library(sqldf)
library(dplyr)
library(igraph)
#import data
products <- read.csv(file.choose())
copurchase <- read.csv(file.choose())
attach(products)
#only keep record of product whose type is Book, and clean
     copurchase data to make sure copurchase only has
     source ids and target ids from the Book product.
products_book = filter(products, group == "Book" &
     salesrank <= 150000 & salesrank >= 0)
copurchase <- sqldf("select copurchase.source,
     copurchase.target from copurchase, products_book
                    where copurchase.source in
     (products_book.id)")
copurchase <- sqldf("select copurchase.source,
     copurchase.target from copurchase, products_book where
     copurchase.target in (products_book.id)")
```

2. *Create a variable named in-degree, to show how many "Source" products people who buy "Target" products buy; i.e. how many edges are to the focal product in "co-purchase" network.*

```
in_degree <- degree(net, mode = "in")
```

3. *Create a variable named out-degree, to show how many "Target" products people who buy "Source" product also buy; i.e., how many edges are from the focal product in "co-purchase" network.*

```
out_degree <- degree(net, mode = "out")
```

4. *Pick up one of the products (in case there are multiple) with highest degree (in-degree + out-degree), and find its subcomponent, i.e., all the products that are connected to this focal product. From this point on, you will work only on this subcomponent.*

```
#select out the product id with the highest all-degree and
get the product id = '33' and '4429'
V(net)$name[degree(net)==max(all_degree)]
#we chose '33' to find out all books'ids connecting with
'33' directly and indirectly
subcomponent33 <- subcomponent(net, "33", mode = "all")
```

Book ID 33 is identified as Double Jeopardy (T*Witches), a teen fantasy science fiction book

5. *Visualize the subcomponent using iGraph, trying out different colors, node and edge sizes and layouts, so that the result is most appealing. Find the diameter, and color the nodes along the diameter. Provide your insights from the visualizations.*

```
#convert subcomponent into object as graph
g <- induced_subgraph(net, subcomponent33)
#draw out graph#
#vertex's degree is used to set vertex's size.
V(g)$degree <- degree(g, mode = "all")
#get the first shortest path between two books' ids who
have longest distance.
diam <- get_diameter(g, directed = TRUE)
as.vector(diam)
#set color and size for nodes and edges for graph.
vcol <- rep("gray", vcount(g))
vcol[diam] <- "gold"
ecol <- rep("gray80", ecount(g))
ecol[E(g, path = diam)] <- "orange"
plot(g, edge.arrow.size=.025, edge.color=ecol,
edge.curved=0.2,
     vertex.size=V(g)$degree*0.5,
     vertex.label = ifelse(degree(g) == 53, V(g)$name, NA),
     vertex.label.degree = -pi/2,
     vertex.color=vcol, vertex.size=2,
     vertex.frame.color="gray", vertex.label.color="black",
     vertex.label.cex=1, vertex.label.dist=3)
```

According to the graph, the diameter nodes consist of book IDs 37895, 27936, 21584, 10889, 11080, 14111, 4429, 2501, 3588, and 6676 (Figure 2). From the sources and related targets, it is clear that consumers are purchasing classic, romantic and entertaining books, which are consistently related. Node "4429" reflects the popular interest for the book buyers who are also interested in the books of 1950-2000.

Figure 1:: Network graph highlighting book IDs 33 and 4429, both which have the highest degrees.

Figure 2: Diameter nodes and descriptions. Genre and category columns were added from Amazon.com for analysis.

id	title	genre	category	salesrank	review_cnt	downloads	rating
2501	The Narcissistic Family : Diagnosis and Treatment	Psychology	nonfiction	9727	19	19	5

id	title	genre	category	salesrank	review_cnt	downloads	rating
3588	A Fourth Treasury of Knitting Patterns	Crafts, Hobbies & Home	nonfiction	91126	1	1	5
4429	Harley-Davidson Panheads, 1948-1965/M418	Automotive	nonfiction	147799	3	3	4.5
6676	Song of Eagles	Romance	fiction	130216	1	1	5
10889	Sixpence Bride (Timeswept)	Romance	fiction	96977	16	16	4.5
11080	Counter Intelligence: Where to Eat in the Real Los Angeles	Travel	nonfiction	28673	13	13	5
14111	Memories, Dreams, Reflections (Vintage)	Psychology	nonfiction	4818	38	38	4.5
21584	A Year and a Day	Religion & Spirituality	nonfiction	107460	52	52	4
27936	Numerology For Personal Transformation	Religion & Spirituality	nonfiction	111939	1	1	5
37895	Sons and Lovers (Signet Classics (Paperback))	Classics	fiction	9236	70	70	4

6. *Compute various statistics about this network (i.e., subcomponent), including degree distribution, density, and centrality (degree centrality, closeness centrality and between centrality), hub/authority scores, etc. Interpret your results.*

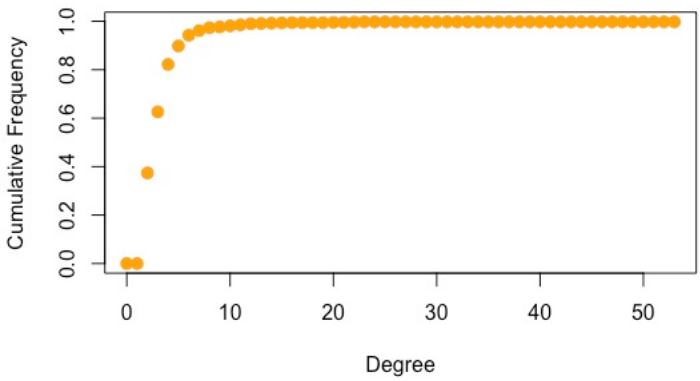
   Degree distribution: from distribution and edge density (0.001436951), we learned that the most books are in the low degree, meaning that most book buyers are looking for particular books and not targeting to further extent.

```
deg.dist <-
     degree_distribution(g,
     cumulative=T, mode="all")

deg <- degree(g, mode = "all")

plot( x=0:max(deg), y=1-
     deg.dist, pch=19, cex=1.2, col="orange",
      xlab="Degree", ylab="Cumulative Frequency")
```

Figure 3: Diameter nodes highlighted coming from book ID 33.




```
edge_density(g, loops=F)
[1] 0.001436951
```

Degree centralization: the difference between centralization 0.02794058 and theoretical max centralization 1630818 is huge. It means most books have few connections with other books, but only a few books have numerous links with other books.

```
centr_degree(g, mode="all", normalized=T)
$res
```

(individual results omitted for brevity)

```
$centralization
[1] 0.02794058


$theoretical_max
[1] 1630818
```

Closeness: according to the closeness function result, the max closeness value is 0.0001612383, which is small. It means there is low efficiency for one node to reach other nodes; books are loosely purchased together.

```
closeness(g, mode="all", weights=NA)
```

(individual results omitted for brevity)

```
centr_clo(g, mode="all", normalized=T)
```

(individual results omitted for brevity)

```
$centralization
[1] 0.1074443


$theoretical_max
[1] 451.2499
```

Betweenness: most of nodes' betweenness are '0'. It means most of books are purchased solely based on the consumers' needs and not related to others.

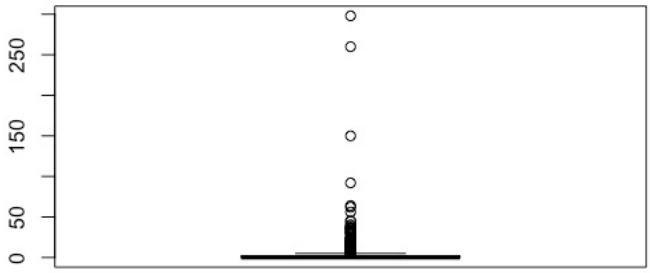


Figure 4: Betweenness centrality boxplot.

```
betwn <- betweenness(g,
     directed=T, weights=NA)

boxplot(betwn)

edge_betweenness(g,
     directed=T, weights=NA)
```

(individual results omitted for brevity)

```
centr_betw(g, directed=T, normalized=T)
```

(individual results omitted for brevity)

```
$centralization
[1] 0.0003616307


$theoretical_max
[1] 735498918
```

Hubs and authorities: according to the hub and authority score's distribution, we found the whole network has loose connections with each other and only very few nodes as hubs to connect to others and authorities as target, like "195144", "33".

```
hubs <- hub_score(g, weights=NA)

boxplot(hubs)

auths <- authority_score(g, weights=NA)

boxplot(auths)

hs <- hub_score(g, weights=NA)$vector

as <- authority_score(g, weights=NA)$vector


plot(g,
     vertex.size=hs*10,
     main = 'Hubs',
     vertex.color = rainbow(52),vertex.label = ifelse(hs ==
     max(hs), V(g)$name, NA),
     edge.arrow.size=0.025,
     layout = layout.kamada.kawai)
```

```
plot(g,
     vertex.size=as*10,
     main = 'Authorities',
     vertex.color = rainbow(52),vertex.label = ifelse(as ==
     max(as), V(g)$name, NA),
     edge.arrow.size=0.025,
     layout = layout.kamada.kawai)
```

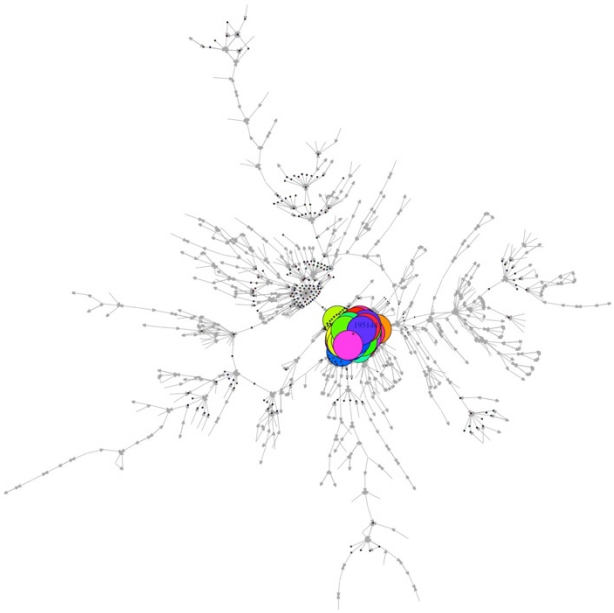


Figure 6; Hubs map with node 195144 highlighted in middle.

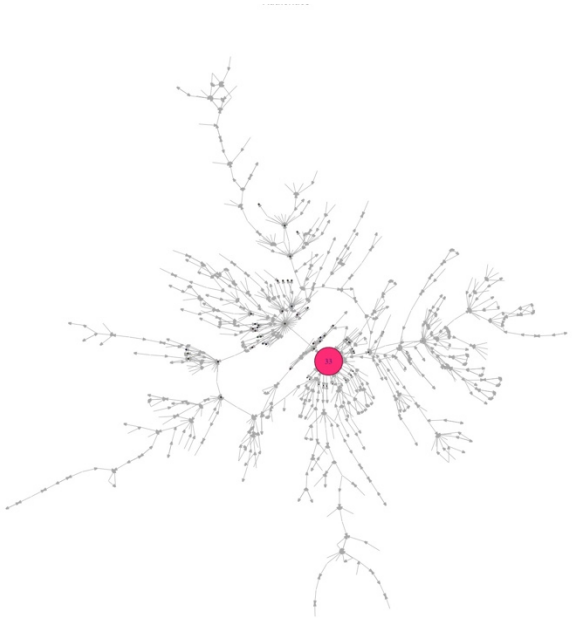


Figure 5: Authorities map with node 33 highlighted in middle.

7. *Create a group of variables containing the information of neighbors that "point to" focal products.*

```
products_book_focal <- products_book
#create table products_book_focal_mean1 to record the means
     of rating, review counts, sales rank of every focal
     product.
products_book_focal_mean <- NULL
products_book_focal_mean1 <- NULL
#figure out all neighbors of each focal product
#define Global variable: vertexid
vertexid <- as_ids(V(g))
```

```
for(i in 1:904){
  neigh_nodes <- neighbors(g, vertexid[i], mode="in")
  neigh_nodes <- as_ids(neigh_nodes)
  neigh_nodes <- as.character(neigh_nodes)
  products_book_focal_temp <- filter(products_book_focal,

     products_book_focal$id %in% neigh_nodes)
  products_book_focal_mean$id <- as.numeric(vertexid[i])
  products_book_focal_mean$nghb_mn_rating <-
     mean(products_book_focal_temp$rating)
  products_book_focal_mean$nghb_mn_salesrank <-
     mean(products_book_focal_temp$salesrank)
  products_book_focal_mean$nghb_mn_review_cnt <-
     mean(products_book_focal_temp$review_cnt)
  products_book_focal_mean1 <-
     rbind(products_book_focal_mean,
     products_book_focal_mean1)
}
products_book_focal_mean1 <-
     as.data.frame(products_book_focal_mean1)
```

8. *Include the variables (taking logs where necessary) created in Parts 2-6 above into the "products" information and fit a Poisson regression to predict salesrank of all the books in this subcomponent using products' own information and their neighbor's information. Provide an interpretation of your results.*

```
#all product info for subcomponent33
product_33 <- merge(products_book_focal,
     products_book_focal_mean1, by.x = "id", by.y = "id",
     all.y=TRUE)
#include Indegree, outdegree, closeness, betweenness into
     the products info of subcomponent 33.
for(i in 1:904){

  in_degree33 <- degree(g, vertexid[i], mode="in")
  out_degree33 <- degree(g, vertexid[i], mode="out")
  closeness33 <- closeness(g, vertexid[i], mode = "all",
     weights=NA)
  betweenness33 <- betweenness(g, vertexid[i], directed=T,
     weights=NA)
```

```
  product_33$in_degree[product_33$id ==
     as.numeric(vertexid[i])] <- in_degree33
  product_33$out_degree[product_33$id ==
     as.numeric(vertexid[i])] <- out_degree33
  product_33$closeness[product_33$id ==
     as.numeric(vertexid[i])] <- closeness33
  product_33$betweenness[product_33$id ==
     as.numeric(vertexid[i])] <- betweenness33

}
#include hub score and authority score
hub_auth_temp <- NULL

hub_auth_temp$id <- as.data.frame(as.numeric(vertexid))
hub_auth_temp$hub_score <- as.data.frame(hs)
hub_auth_temp$auth_score <- as.data.frame(as)
hub_auth_temp <- as.data.frame(hub_auth_temp)
colnames(hub_auth_temp) <- c("id", "hub_score",
     "auth_score")

#merge temporary table containing hub score and authority
     score into product33 table
product_33 <- merge(product_33, hub_auth_temp, by.x = "id",
     by.y = "id", all.x =TRUE)
product_33$nghb_mn_rating <-
     as.numeric(product_33$nghb_mn_rating)
product_33$nghb_mn_salesrank <-
     as.numeric(product_33$nghb_mn_salesrank)
product_33$nghb_mn_review_cnt <-
     as.numeric(product_33$nghb_mn_review_cnt)

#now product_33 contains all network info of books of
     subcomponent "33"
attach(product_33)
#following will do poission regression to check which
     factor impact salesrank most.
fit.salesrank <- glm(salesrank ~ review_cnt + downloads +
     rating + nghb_mn_rating + nghb_mn_salesrank +
     nghb_mn_review_cnt + in_degree + out_degree +
     closeness + betweenness + hub_score + auth_score,
                    data = product_33, family = poisson())
summary(fit.salesrank)

Call:
glm(formula = salesrank ~ review_cnt + downloads + rating +
     nghb_mn_rating +
```

```
    nghb_mn_salesrank + nghb_mn_review_cnt + in_degree +
     out_degree +
    closeness + betweenness + hub_score + auth_score,
     family = poisson(),
    data = product_33)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-363.25  -160.45    -7.61   122.01   519.58

Coefficients:
                     Estimate Std. Error   z value Pr(>|z|)
(Intercept)         1.119e+01  1.108e-03 10096.697   <2e-16
     ***
review_cnt         -2.868e-02  1.877e-04  -152.749   <2e-16
     ***
downloads           2.457e-02  1.879e-04   130.759   <2e-16
     ***
rating             -7.061e-03  1.098e-04   -64.314   <2e-16
     ***
nghb_mn_rating     -9.723e-03  1.253e-04   -77.613   <2e-16
     ***
nghb_mn_salesrank   2.057e-07  4.498e-09    45.733   <2e-16
     ***
nghb_mn_review_cnt  7.386e-04  1.969e-06   375.165   <2e-16
     ***
in_degree           2.801e-03  6.819e-05    41.069   <2e-16
     ***
out_degree          5.646e-02  2.057e-04   274.476   <2e-16
     ***
closeness          -1.789e+01  7.874e+00    -2.272   0.0231
     *
betweenness        -7.349e-04  1.111e-05   -66.157   <2e-16
     ***
hub_score           2.452e-01  8.593e-04   285.400   <2e-16
     ***
auth_score          1.895e-01  4.754e-03    39.861   <2e-16
     ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
     ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 16968896  on 517  degrees of freedom
Residual deviance: 15315200  on 505  degrees of freedom
  (386 observations deleted due to missingness)
```

```
AIC: 15321778

Number of Fisher Scoring iterations: 5

#interpret coefficients:

c1 <-coef(fit.salesrank)
c1 <- as.data.frame(c1)
colnames(c1) <- c("coef_value")
attach(c1)
c1$coef_value <- round(coef_value, 4)
c1$exp_value <- round(exp(coef_value), 4)
c1$percentage_change <- (c1$exp_value-1)*100
print(c1)
```

	coef_value	exp_value	percentage_change
(Intercept)	11.1919	72541.5152	7254051.52
review_cnt	-0.0287	0.9717	-2.83
downloads	0.0246	1.0249	2.49
rating	-0.0071	0.9930	-0.70
nghb_mn_rating	-0.0097	0.9903	-0.97
nghb_mn_salesrank	0.0000	1.0000	0.00
nghb_mn_review_cnt	0.0007	1.0007	0.07
in_degree	0.0028	1.0028	0.28
out_degree	0.0565	1.0581	5.81
closeness	-17.8874	0.0000	-100.00
betweenness	-0.0007	0.9993	-0.07
hub_score	0.2452	1.2779	27.79
auth_score	0.1895	1.2087	20.87

According to the regression output, we can explain the coefficients below:

- Due to we are focusing on corelationsihp of books not an isolated book, so "intercept" is not meaningful. All other characters of book are significant to sales rank.
- Product's review counts increase by one, 2.83% decrease on salesrank, means more probability of sales.
- product's downloads impact: downloads increased by one, 2.49% increase on salesrank on average, means less probability of sales.
- Product's rating impact: rating increased by one, 0.7% decrease on salesrank on average, means more probability of sales.
- Neighbors' mean review amount increased by one, 0.07% increase on salesrank on average and less probability of sales.
- Neighbors' mean rating increased by one, 0.97% decrease on salesrank on average, means more probability of sales.

- Neighbors' mean salesrank increased by one, it will not impact much on focal product salesrank. But if neighbors' mean salesrank increased a lot, it will impact focal product's salesrank and sales.
- Product's in-degree increased by one, 0.28% increase on salesrank on average, means less probability of sales.
- Product's out-degree increased by one, 5.81% increase on salesrank on average, means less probability of sales.
- Product's closeness increased by 0.0001, 0.01% decrease on salesrank on average and more sales.
- Product's betweenness increased by one (paths are more shorter to reach others), 0.07% decrease on salesrank on average and more sales.
- Product's hub score increased by 0.1 (more outlinks), 2.779% increase on salesrank on average and less sales.
- Product's authority score increased by 0.1 (more inlinks), 2.087% increase on salesrank on average and less sales.

```
#remove products with no "point-to" neighbors to do poisson
     regression, and get the same result as above.
product_33_haveneigh <-
     product_33[complete.cases(product_33), ]
fit.salesrank_neigh <- glm(salesrank ~ review_cnt +
     downloads + rating + nghb_mn_rating +
     nghb_mn_salesrank + nghb_mn_review_cnt + in_degree +
     out_degree + closeness + betweenness + hub_score +
     auth_score,
                     data = product_33_haveneigh, family =
     poisson())
summary(fit.salesrank_neigh)

Call:
glm(formula = salesrank ~ review_cnt + downloads + rating +
     nghb_mn_rating +
    nghb_mn_salesrank + nghb_mn_review_cnt + in_degree +
     out_degree +
    closeness + betweenness + hub_score + auth_score,
     family = poisson(),
    data = product_33_haveneigh)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-363.25  -160.45    -7.61   122.01   519.58

Coefficients:
                     Estimate Std. Error   z value Pr(>|z|)
(Intercept)         1.119e+01  1.108e-03 10096.697   <2e-16
     ***
```

```
review_cnt          -2.868e-02  1.877e-04  -152.749   <2e-16
     ***
downloads            2.457e-02  1.879e-04   130.759   <2e-16
     ***
rating              -7.061e-03  1.098e-04   -64.314   <2e-16
     ***
nghb_mn_rating      -9.723e-03  1.253e-04   -77.613   <2e-16
     ***
nghb_mn_salesrank    2.057e-07  4.498e-09    45.733   <2e-16
     ***
nghb_mn_review_cnt   7.386e-04  1.969e-06   375.165   <2e-16
     ***
in_degree            2.801e-03  6.819e-05    41.069   <2e-16
     ***
out_degree           5.646e-02  2.057e-04   274.476   <2e-16
     ***
closeness           -1.789e+01  7.874e+00    -2.272   0.0231
     *
betweenness         -7.349e-04  1.111e-05   -66.157   <2e-16
     ***
hub_score            2.452e-01  8.593e-04   285.400   <2e-16
     ***
auth_score           1.895e-01  4.754e-03    39.861   <2e-16
     ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
     ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 16968896  on 517  degrees of freedom
Residual deviance: 15315200  on 505  degrees of freedom
AIC: 15321778

Number of Fisher Scoring iterations: 5
```

```
#predict using mean value of every variables
predict(fit.salesrank,
     data.frame(review_cnt=mean(review_cnt),
     downloads=mean(downloads),
                              rating=mean(rating),
     nghb_mn_rating = mean(nghb_mn_rating),
                                  nghb_mn_salesrank
     =mean(nghb_mn_salesrank),
```

```
                                    nghb_mn_review_cnt =
     mean(nghb_mn_review_cnt),

     in_degree=mean(in_degree), out_degree =
     mean(out_degree),
                                    closeness =
     mean(closeness), betweenness = mean(betweenness),
                                    hub_score =
     mean(hub_score), auth_score = mean(auth_score)),
        type="response")
```

- The predicted result of sales rank is 65799.

```
#two times neighbor average rating and keep others
     constant, predict the salesrank.

predict(fit.salesrank,
     data.frame(review_cnt=mean(review_cnt),
     downloads=mean(downloads),
                                    rating=mean(rating),
     nghb_mn_rating = mean(nghb_mn_rating)*2,
                                    nghb_mn_salesrank
     =mean(nghb_mn_salesrank),
                                    nghb_mn_review_cnt =
     mean(nghb_mn_review_cnt),

     in_degree=mean(in_degree), out_degree =
     mean(out_degree),
                                    closeness =
     mean(closeness), betweenness = mean(betweenness),
                                    hub_score =
     mean(hub_score), auth_score = mean(auth_score)),
        type="response")
```

- The predicted result of sales rank is 62880.

From this assessment we find that the neighbor mean rating decreased the sales rank when `mean(nghb_mn_rating)` is multiplied by two. There is a “negative” effect on the sales rank when neighboring ratings increase.