Social Network Analysis

A SONG OF ICE AND FIRE

ANDRIANI KARPATHAKI
P2822020 | BUSINESS ANALYTICS MASTER STUDENT| 2020 - 2021

Table of Contents

Figures

Tables

Tasks

1 `A Song of Ice and Fire' network

Your first task is to create an igraph graph using the network of the characters of 'A Song of Ice and Fire' by George R. R. Martin [1]. A .csv file with the list of edges of the network is available online. You should download the file and use columns Source, Target, and Weight to create an undirected weighted graph. For your convenience, you are free to make any transformations you think are appropriate to the file.

Answer

First thing to do was to download the csv files we were interested in, as a zip file. After unzipping the file, we imported the data from files *"asoiaf-all-edges"* and *"asoiaf-all-nodes"* and dropped the columns not useful for the analysis. Lastly, we checked for any missing values before creating the graph.

Code

```
# Read file
edges = read.csv(file.choose(),header=T, sep = ",")
edges <-edges[,c(1,2,5)]

nodes = read.csv(file.choose(),header=T, sep = ",")

# Check For NA values in file
na <- colSums(is.na(edges))
na[na>0]

# Create graph using igraph
g <- graph_from_data_frame(edges, directed=FALSE, vertices=nodes)
```

2 Network Properties

Next, having created an igraph graph, you will explore its basic properties and write code to print:

- Number of vertices
- Number of edges
- Diameter of the graph
- Number of triangles
- The top-10 characters of the network as far as their degree is concerned
- The top-10 characters of the network as far as their weighted degree is concerned

Answer

Having created the graph, we calculated the following measurements:

- **Number of vertices** is equal to **796**
- **Number of edges** is equal to **2823**
- **Diameter of the graph** is equal to **53**
- **Number of triangles** is equal to **5655**
- **The top-10 characters of the network as far as their degree is concerned**

Degree	Nodes
122	Tyrion-Lannister
114	Jon-Snow
101	Jaime-Lannister
97	Cersei-Lannister
89	Stannis-Baratheon
84	Arya-Stark
75	Catelyn-Stark
75	Sansa-Stark
74	Eddard-Stark
74	Robb-Stark

Table 1 - The top-10 characters of the network as far as their degree is concerned.

- **The top-10 characters of the network as far as their weighted degree is concerned.**

Weighted Degree	Nodes
2873	Tyrion-Lannister
2757	Jon-Snow
2232	Cersei-Lannister
1762	Joffrey-Baratheon
1649	Eddard-Stark
1608	Daenerys-Targaryen
1569	Jaime-Lannister
1547	Sansa-Stark
1508	Bran-Stark
1488	Robert-Baratheon

Table 2 - The top-10 characters of the network as far as their weighted degree is concerned

It is worth mentioning that the 10 most important nodes (characters) of the network change when we take into account the weights of edges. However, the top two characters remain the same.

Code

```
# number of vertices
vcount(g)
# number of edges
ecount(g)
# diameter
diameter(g)
# triangles
cl.tri=cliques(g,min=3,max=3)
length(cl.tri)
#alternative
sum(count_triangles(g))/3


# number of degrees of each node
deg <- degree(g, mode="all")
deg_df <- as.data.frame(deg)
```

```
deg_df$nodes <- row.names(deg_df)
row.names(deg_df) <- 1:796
deg_df_order <- deg_df[order(deg_df$deg,decreasing = TRUE),]

# top 10 degrees
head(deg_df_order, 10)

# top 10 degrees with weights
deg1 <- strength(g, vids = V(g), loops = TRUE)
deg1_df <- as.data.frame(deg1)
deg1_df$nodes <- row.names(deg1_df)
row.names(deg1_df) <- 1:796
deg1_df_order <- deg1_df[order(deg1_df$deg1,decreasing = TRUE),]
head(deg1_df_order, 10)
```

3 Subgraph

After that, your task is to plot the network:

You will first plot the entire network. Make sure you set the plot parameters appropriately to obtain an aesthetically pleasing result. For example, you can opt not to show the nodes' labels (vertex.label = NA) and set a custom value for parameters: edge.arrow.width, and vertex.size. Feel free to configure additional parameters that may improve your visualization results. Then, you will create a subgraph of the network, by discarding all vertices that have less than 10 connections in the network and plot the subgraph. In addition to the above plots, you are also asked to write code that calculates the edge density of the entire graph, as well as the aforementioned subgraph, and provide an explanation on the obtained results (a few sentences in your report).

Answer

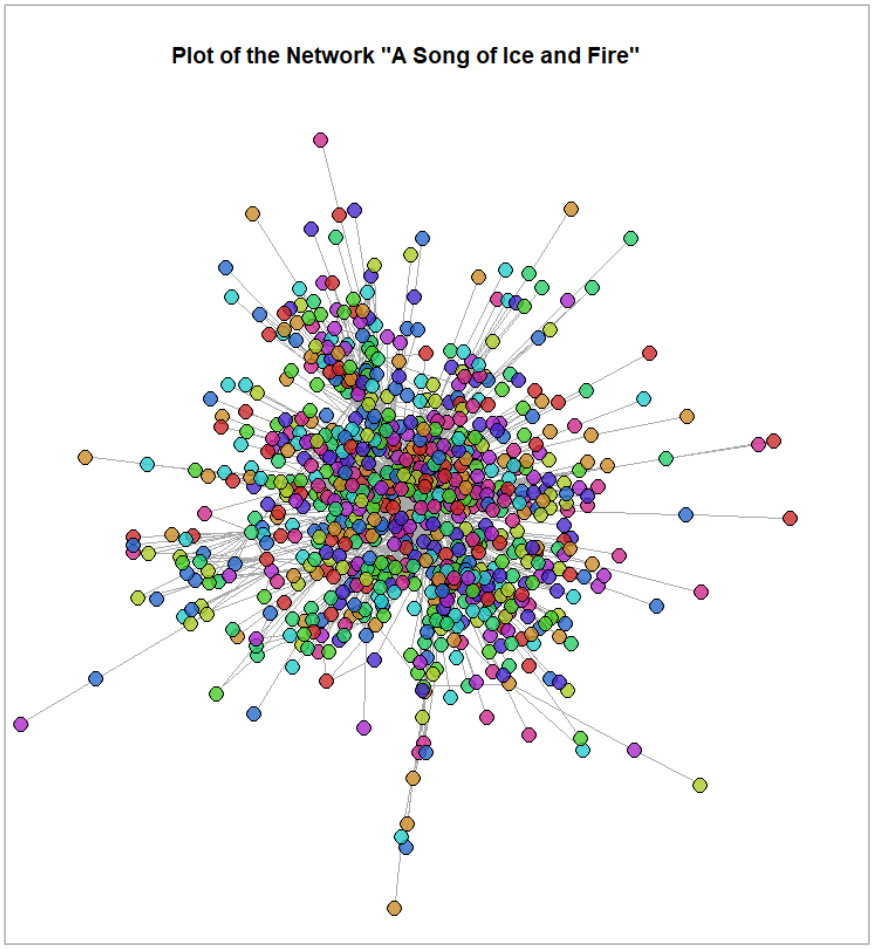

Figure 1 - Plot of full Network

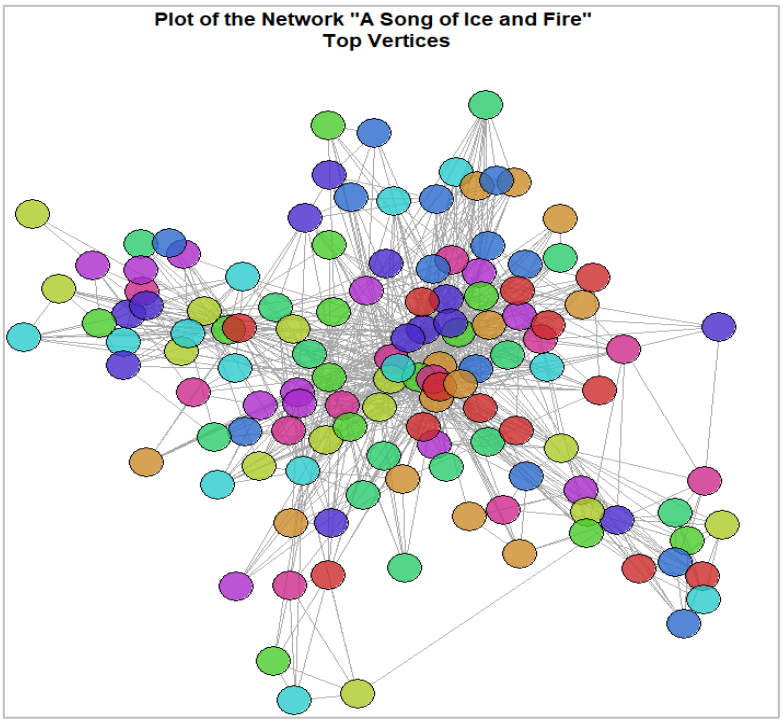

Figure 2 - Plot consisting nodes with more than 10 connections in the Network

As expected, the subgraph is less complicated that the full graph of the Network as only the most important characters have been taken into account (Figures 1-2). To be noted that each bubble in the graph represents a node and grey lines represent the edges between them.

Regarding the edge density which is defined to be the ratio of the number of edges with respect to the maximum possible edges of the graph, we see that for the full graph it is equal to 0.008921968 while for the subgraph it is equal to 0.1258612. This is a rational result given that the subgraph includes the most connected (more than ten connection) characters, thus it is denser.

Code

```
# Full Graph

plot(g, vertex.size=4, vertex.color = rainbow(10, .8, .8, alpha= .8),vertex.label = NA,
edge.arrow.size = 0.3, edge.arrow.width = 4, edge.color = "darkgrey", main = "Plot of
the Network \"A Song of Ice and Fire\"")

# Subgraph consisting nodes with more than 10 connections in the Network

new_graph <- induced.subgraph(g, V(g)[degree(g)>10])

plot(new_graph, vertex.size=10, vertex.color = rainbow(10, .8, .8, alpha=.8),
vertex.label = NA, edge.arrow.size = 0.6, edge.arrow.width = 18, edge.color =
"darkgrey", main = "Plot of the Network \"A Song of Ice and Fire\"
Top Vertices")


# Edge Density

edge_density(g)
edge_density(new_graph)
```

4 Centrality

Next, you will write code to calculate and print the top-15 nodes according to the:

- betweeness centrality
- closeness centrality

In addition, you are asked to find out where the character *Jon Snow* is ranked according to the above two measures and provide an explanation (a few sentences) of the observations you make after examining your results.

Answer

Closeness is as a measure that indicates how long it will take to spread information from a node to all other nodes sequentially while **betweenness** centrality interprets the number of times a node is part of the shortest path between two other nodes. Considering that we shall say that Jon-Snow is not located somewhere very central to the network (10th regarding closeness centrality) in comparison to other 9 nodes but it holds authority over the network, because more information will pass through him (1st regarding betweenness centrality).

Betweenness	nodes
41698.94	Jon-Snow
38904.51	Theon-Greyjoy
36856.35	Jaime-Lannister
29728.50	Daenerys-Targaryen
29325.18	Stannis-Baratheon
29201.60	Robert-Baratheon
28917.83	Tyrion-Lannister
24409.67	Cersei-Lannister
20067.94	Tywin-Lannister
19870.45	Robb-Stark
19354.54	Arya-Stark
17769.29	Barristan-Selmy
17555.36	Eddard-Stark
15913.44	Sansa-Stark
15614.41	Brienne-of-Tarth

Table 3 - Top 15 nodes according betweenness centrality

closeness	nodes
0.0001193602	Jaime-Lannister
0.0001141162	Robert-Baratheon
0.0001134945	Theon-Greyjoy
0.0001133144	Stannis-Baratheon
0.0001131606	Jory-Cassel
0.0001128032	Tywin-Lannister
0.0001116695	Cersei-Lannister
0.0001114330	Tyrion-Lannister
0.0001112842	Brienne-of-Tarth
0.0001106562	Jon-Snow
0.0001093853	Joffrey-Baratheon
0.0001083658	Rodrik-Cassel
0.0001079214	Eddard-Stark
0.0001078516	Doran-Martell
0.0001072271	Harys-Swyft

Table 4 - Top 15 nodes according closeness centrality

5 Ranking and Visualization

In the final step of this homework, you are asked to rank the characters of the network with regard to their PageRank value. You will write code to calculate the PageRank values, and create a plot of the graph that uses these values to appropriately set the nodes' size so that the nodes that are ranked higher are more evident.

Answer

In the final task we are interested in calculating the PageRank value and rank the characters depending on that. Finally, as suggested we created a plot of the graph that uses these values to appropriately set the nodes' size so that the nodes that are ranked higher are more evident (Figure 3).

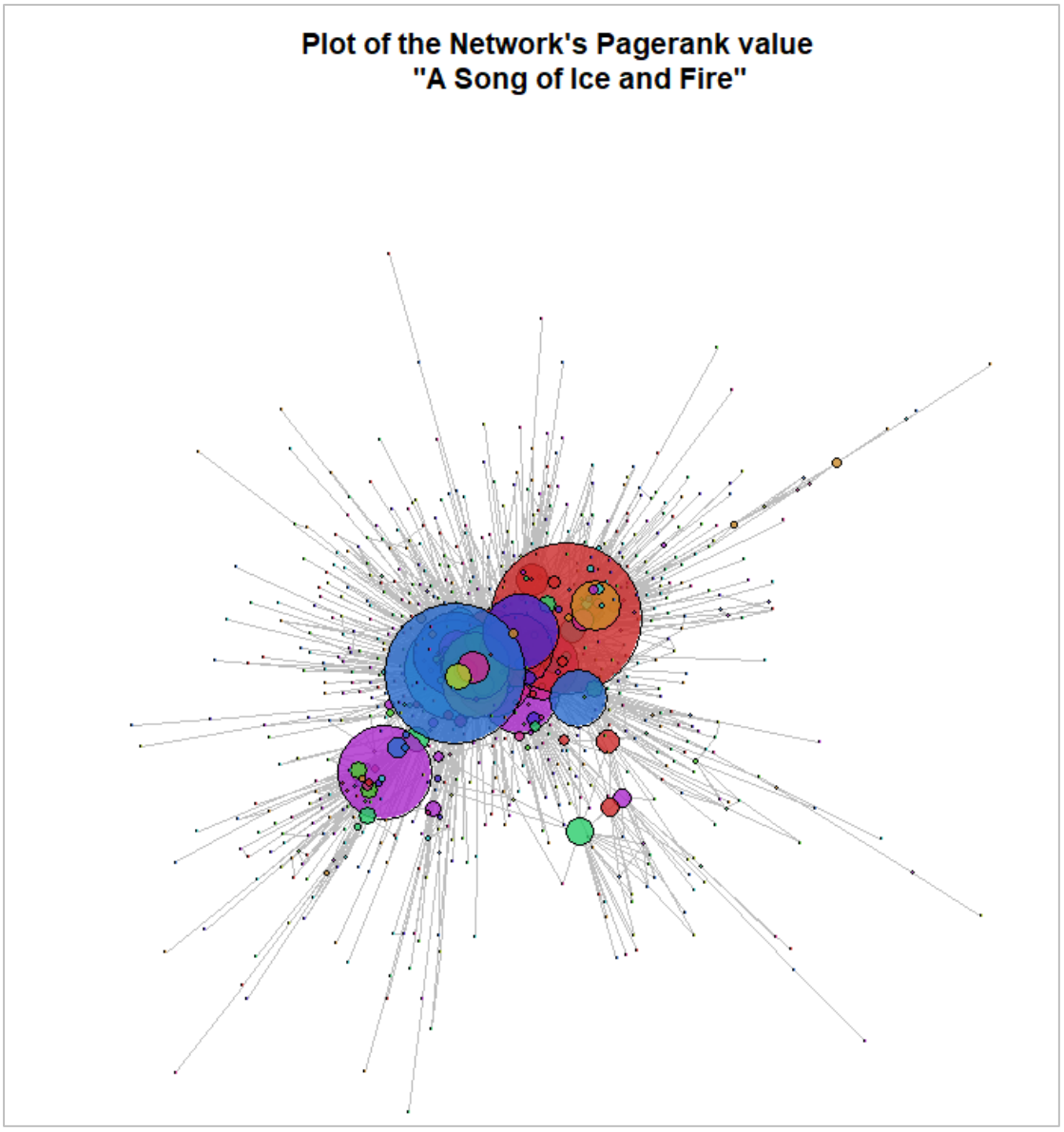

Figure 3 - Graph of the Network depending on the PageRank value

References

- https://igraph.org/r/doc/strength.html
- https://igraph.org/r/doc/closeness.html
- https://igraph.org/r/doc/betweenness.html
- https://en.wikipedia.org/wiki/Betweenness_centrality