Network Analysis and Visualization with R and igraph

Georgia Vlassi p2822001 | Social Networking Analysis | 20/05/2021

1. ‘A Song of Ice and Fire’ network

For our analysis and implementations, we should download the dataset from the following link https://github.com/mathbeveridge/asoiaf/tree/master/data/asoiaf-all-edges.csv. Initially, we should load the dataset to a variable named asoiaf. As the necessary columns for our analysis are Source, Target, and Weight, we delete the Type and Id columns by setting them to NULL.

Having prepared the dataset, we create an igraph graph from our data frame. The function used to create the igraph contains two parameters. The first one is the asoiaf data frame, which indicates the edges and the vertices of the graph. The second parameter is a logical scalar, which indicates that our graph is undirected. The graph created above gets assigned to a variable named g.

Without using any parameter, we plot the g graph to take a quick look at the outcome. Observing the graph below, there is no valuable outcome. Further analysis will be presented later.

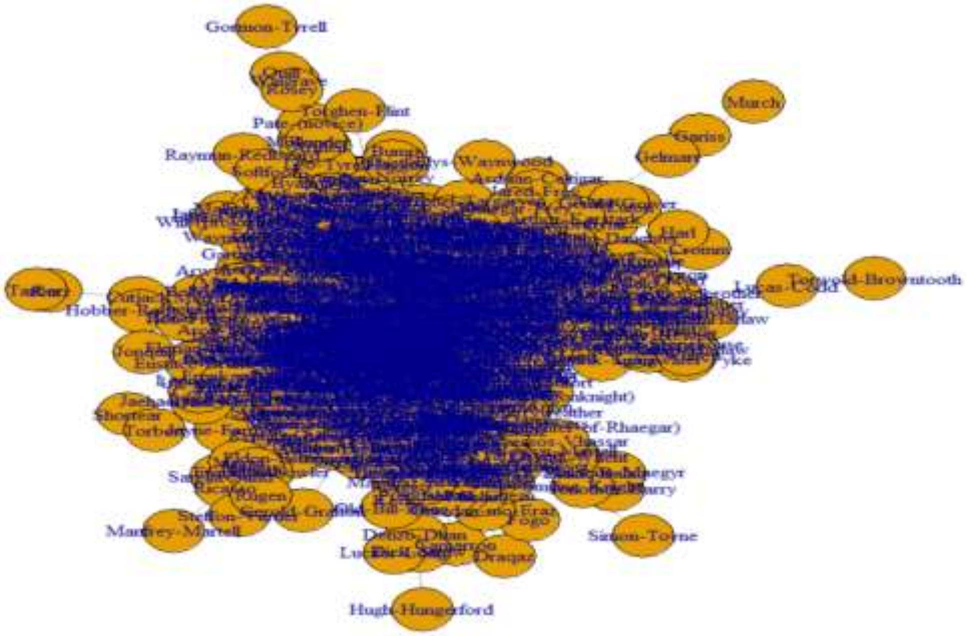

Figure 1 Plain graph

2. Network Properties

Having created our igraph graph, we can explore its basic properties:

- The number of vertices is 796.
- The number of edges is 2823.
- The diameter of the graph is 53.
- To calculate the number of triangles, initially we count the triangles of the graph and then we sum up the all the counts. The outcome is 16965.
- To calculate the top-10 characters of the network as far as their degree is concerned, firstly we should take the degree of all nodes. Then we sort them in descending order and we print the first 10. The output is shown below:

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

- To calculate the top-10 characters of the network as far as their weighted degree is concerned, we should take the weighted degree of all nodes. Then we sort them in descending order and we print the first 10. The output is shown below:

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

3. Subgraph

At Figure 2 we plot the entire network, by using some parameters. We remove the names of the edges, we use the layout of the Fruchterman Reingold and we set specific values at the following parameters edge.arrow.width, the edge.arrow.size and vertex.size to plot the below graph.

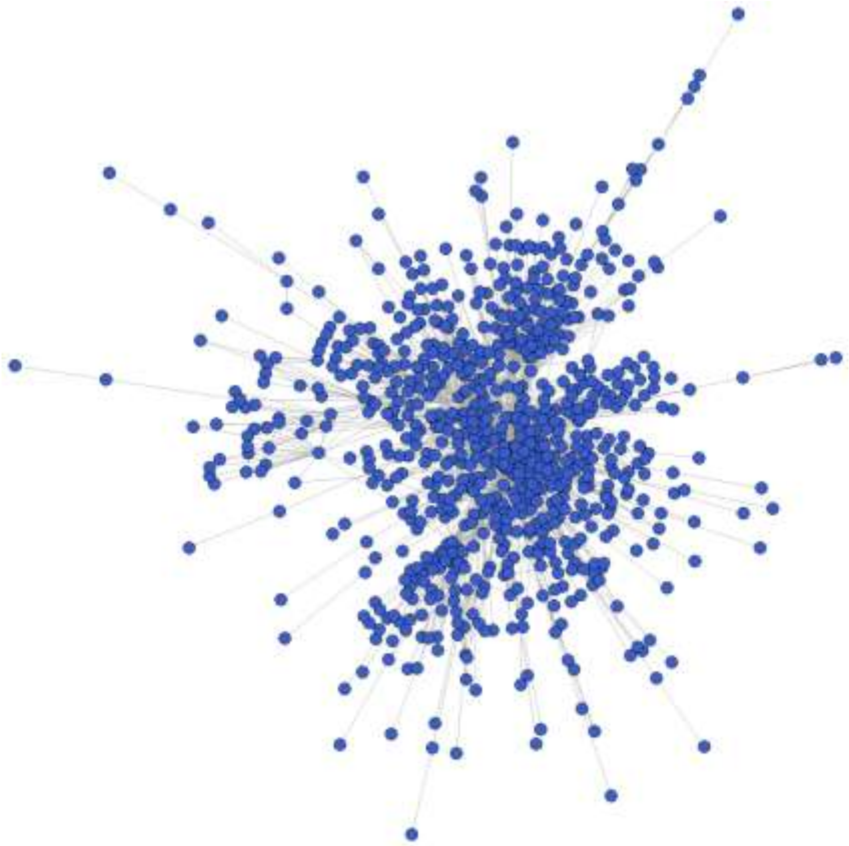

Figure 2 Full graph

At Figure 3 we plot the subgraph network, by using same parameters as the entire network. The subgraph should discard the nodes with less than 10 connections. The degree of the node calculated at the previous question, indicates the number of edges and therefore the connections. We delete the vertices that have degree less than 10.The updated output gets assigned to a new variable named subgraph.

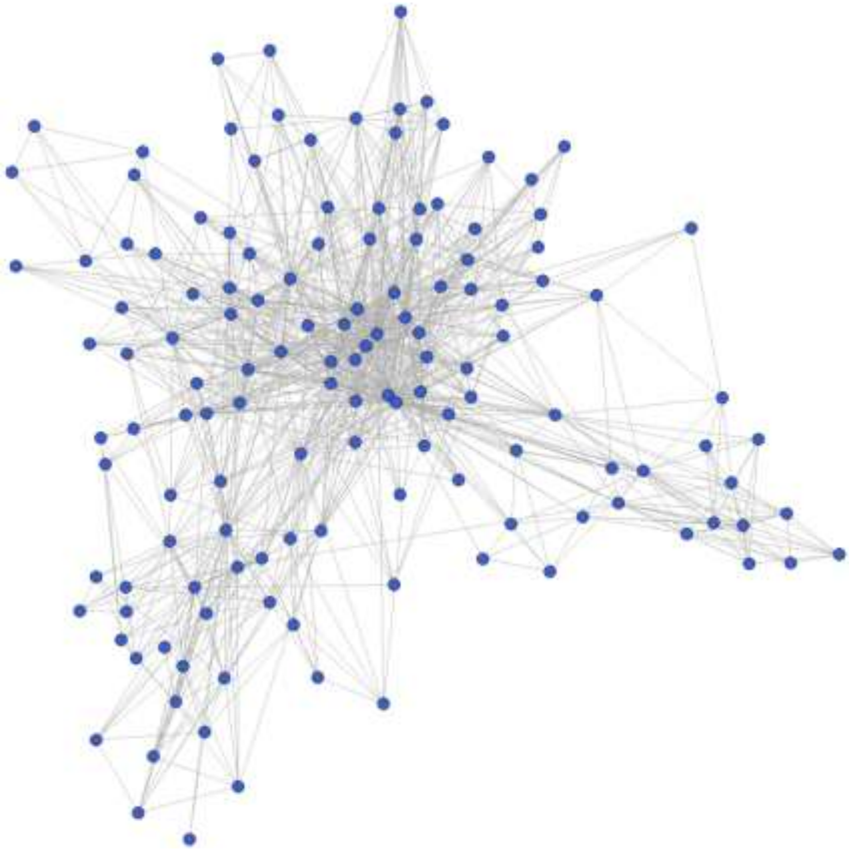

Figure 3 Subgraph

The edge density is used to calculate the edge density of the full graph and subgraph respectively. The edge density is a measure of how many ties between characters exist compared to how many ties between characters are possible. Comparing the results below, we conclude that the subgraph has greater edge density. It can be justified as the number of the edges of the subgraph is closer to its possible maximum edges, than the number of the edges of the full graph to its own possible maximum edges.

full graph	0.008921968
subgraph	0.117003

4. Centrality

In the following table the top 15 nodes are shown based on the closeness centrality. To calculate these nodes, the closeness is used to detect the nodes that are able to spread information very efficiently through a graph. Subsequently, we sort the nodes in descending order and we print the first 15.

Jaime-Lannister	0.0001193460
Robert-Baratheon	0.0001137527
Theon-Greyjoy	0.0001135203
Jory-Cassel	0.0001131734
Stannis-Baratheon	0.0001131606
Tywin-Lannister	0.0001128286
Cersei-Lannister	0.0001116695
Tyrion-Lannister	0.0001114454
Brienne-of-Tarth	0.0001112718
Jon-Snow	0.0001106072
Joffrey-Baratheon	0.0001093733
Rodrik-Cassel	0.0001083658
Doran-Martell	0.0001079098
Eddard-Stark	0.0001073192
Harys-Swyft	0.0001072961

To calculate the top 15 nodes based on the betweenness centrality, the betweenness is used to detect the amount of influence a node has over the flow of information in a graph. Afterwards, we sort the nodes in descending order and we print the first 15. The output is shown in the following table:

Jon-Snow	41698.94
Theon-Greyjoy	38904.51
Jaime-Lannister	36856.35
Daenerys-Targaryen	29728.50
Stannis-Baratheon	29325.18
Robert-Baratheon	29201.60
Tyrion-Lannister	28917.83
Cersei-Lannister	24409.67
Tywin-Lannister	20067.94
Robb-Stark	19870.45
Arya-Stark	19354.54

Barristan-Selmy	17769.29
Eddard-Stark	17555.36
Sansa-Stark	15913.44
Brienne-of-Tarth	15614.41

From the above tables we observe that the character John Snow is ranked on 10th position based on closeness centrality. In comparison with betweenness centrality, where he is ranked on the 1st position. We conclude that he is an important character with a high influence at the other characters and because of him there are lots of connections.

5. Ranking and Visualization

As our scope is to rank the characters, the PageRank in the igraph package is used. The miniCRAN and magrittr packages should be loaded. The first one is used to build a graph of package dependencies and the latter includes the %>% pipes, which are useful to sort the ranking in descending order and to convert it to a matrix.

In the following table we can see the top 10 characters according to the PageRank. As expected from what was presented in the previous question, the character Jon Snow has the biggest ranking value.

Jon-Snow	0.03570539
Tyrion-Lannister	0.03291094
Cersei-Lannister	0.02366461
Daenerys-Targaryen	0.02228040
Jaime-Lannister	0.01979001
Eddard-Stark	0.01896426
Arya-Stark	0.01857171
Stannis-Baratheon	0.01805099
Joffrey-Baratheon	0.01746037
Robb-Stark	0.01736071

As the values are very small, we will multiply them with 1000 to have the higher ranked more evident. Figure 4 shows that the bigger the node the greater the ranking.

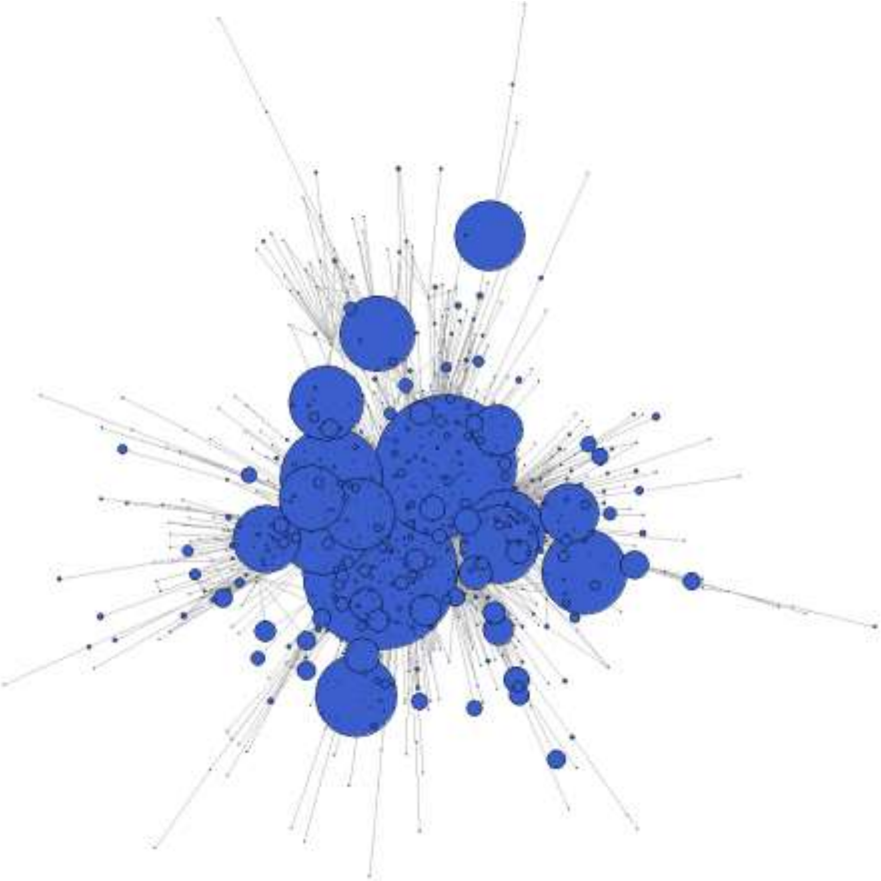

Figure 4 PageRank of the characters