Network analysis

Pierre Achkar

Information retrieval and recommender systems (2022-2023)

Lab Session 5

January 3, 2023

1. Introduction

In this lab session, we will perform network analysis on several synthetic and real networks. We will begin by reproducing well-known facts about network models such as Erdos-Renyi, Watts-Strogatz, and Barabasi-Albert. Specifically, we will plot the clustering coefficient and the average shortest-path as a function of the parameter p in the Watts-Strogatz model, plot the average shortest-path length as a function of the network size in the Erdos-Renyi model, and plot the histogram of the degree distribution in a Barabasi-Albert network.

In the second part of the report, we will build a network where the nodes are movies and edges are added based on the similarity between movies. We will then perform various analyses on this network, including calculating its diameter and transitivity, examining its degree distribution, and applying a PageRank algorithm to the nodes. We will also use a community detection algorithm to identify the communities within the network and perform additional analyses on these communities.

1.1. Random networks:

Random networks are a type of graph characterized by a high degree of randomness. They are typically generated by randomly connecting nodes according to some predetermined probability, resulting in a graph with a structure that is largely unpredictable and free from any predetermined patterns. They are widely used as a baseline for comparing the properties of other types of graphs, such as social networks or biological networks. They are also used to study the behavior of random systems, such as the spread of diseases or the formation of social networks. Despite their randomness, random networks can exhibit a range of interesting properties, such as a power-law degree distribution and small-world behavior. These properties have been widely studied and have helped to shed light on the structure and dynamics of complex systems. In this session, we will look at three types of random graphs:

- **Erdos-Renyi (ER model)**: also known as the Erdos-Renyi random graph model, is a mathematical model of random graphs that was first introduced by Paul Erdos and Alfréd Rényi in their 1959 paper "On the Evolution of Random Graphs" (Erdös & Rényi, 1986). Since its introduction, the ER model has become a fundamental tool in

the study of graph theory and network science, and has been widely used to model and analyze complex systems and networks.

The ER model is defined by two parameters: the number of nodes n and the probability of an edge being present between any two nodes p. Given these two parameters, the ER model generates a graph in which each pair of nodes is independently connected with probability p. This results in a graph with a high degree of randomness, and the structure of the graph can vary widely depending on the values of n and p.

- **Watts-Strogatz model (WS model):** is a class of random graphs that was introduced by Duncan J. Watts and Steven H. Strogatz in 1998 as a model for small-world networks. It is characterized by a high degree of local clustering and a small average shortest path length, which makes it a good model for networks that have both local and global connections, such as social networks.

  The WS model also has two main parameters: the number of nodes n and the probability of rewiring an edge p. The rewiring probability p determines the level of randomness in the graph. When p is low, the resulting graph is likely to be more regular and have a high degree of local clustering. On the other hand, when p is high, the resulting graph is likely to be more random and have a smaller average shortest path length.

  The WS model has been widely studied and has been used to model a range of real-world networks, including social networks, brain networks, and power grids (Watts & Strogatz, 1998).

- **Barabasi-Albert model (BA model):** is a widely used model for generating scale-free networks, which are characterized by a heavy-tailed degree distribution. It was first introduced by Albert-László Barabási and Réka Albert in 1999, and has since become a fundamental tool in the study of network science (Barabási & Albert, 1999).

  The BA model is defined by two parameters: the number of vertices n in the resulting network, and the number of edges m that each new vertex brings to attach itself to existing nodes. These parameters control the growth and evolution of the network, and determine the structure and properties of the resulting graph.

One of the key features of the BA model is its preferential attachment mechanism, which leads to the emergence of a power-law degree distribution and a high degree of heterogeneity in the network. This makes the BA model particularly useful for modeling real-world systems that exhibit scale-free behavior, such as the internet, social networks, and biological networks.

1.2. Network Metrics & Methods

Network metrics are measures or statistics that are used to quantitatively describe the structure and properties of a network. These metrics can provide insights into the characteristics and behaviors of the network, and can be used to compare different networks or to study the evolution of a network over time. In this session, we will work with the following metrics and methods:

- **Average shortest-path:** is a measure of the average distance between two nodes in the graph. It is calculated by finding the shortest path between all pairs of nodes in the graph and taking the average of all of these distances.
  The shortest path between two nodes is the path with the fewest number of edges connecting the two nodes. It can be found using algorithms such as Dijkstra's algorithm or the breadth-first search algorithm. The average shortest path can be useful for understanding the connectivity of a graph or network. For example, a graph with a small average shortest path would be considered more connected, as it would be relatively easy for nodes to reach each other, while a graph with a large average shortest path would be considered less connected, as it would be more difficult for nodes to reach each other.
- **Clustering Coefficient/Transitivity:** is a measure of the degree to which nodes that are connected to a given node are also connected to each other. It is a measure of the "cliquishness" of a graph, or how much the graph tends to form clusters or groups of interconnected nodes. It can be calculated in several ways:
  - Global clustering coefficient:

$$C = \frac{3 * number\ of\ triangles}{number\ of\ connected\ triples}$$

  - Local clustering coefficient:

$$C_i = \frac{nr.\ of\ connections\ between\ i's\ neighbors}{\frac{1}{2} n_i (n_i - 1)}$$

$$\Rightarrow C = \frac{1}{n}\sum_{i} C_i$$

High values of clustering coefficient or transitivity indicate that the nodes in a graph tend to form interconnected clusters or groups, while low values indicate that the nodes in a graph are more dispersed and not as interconnected. This metric can be useful for understanding the structure and organization of a graph and for identifying key nodes or groups of nodes within the graph.

- **Community detection:** is the process of identifying groups or communities of nodes within a graph that are more densely connected to each other than to nodes in other parts of the graph. These groups of nodes can be thought of as subgraphs within the larger graph, and they may correspond to real-world communities or groups with shared characteristics or interests. There are several approaches to community detection in graph theory, including both statistical and algorithmic methods. Some common algorithms used for community detection include modularity optimization, spectral clustering, and the Louvain method.

  One of the main challenges in community detection is defining what constitutes a "community" and how to identify them within a graph. There is no one-size-fits-all definition of a community, and different approaches to community detection may yield different results depending on the specific characteristics of the graph being analyzed.

  Community detection can be useful for a variety of applications, such as identifying groups of individuals with shared interests or characteristics in social networks, identifying patterns of collaboration or communication within organizations, and understanding the structure and organization of complex systems.

- **Degree Distribution:** is a measure of the number of connections or edges that each node has. It describes the distribution of the number of edges that are connected to each node in the graph.

  The degree of a node is the number of edges that are connected to that node. In a directed graph, there are two types of degree: the in-degree, which is the number of edges that point towards the node, and the out-degree, which is the number of edges that point away from the node. In an undirected graph, there is only one type of degree, which is the number of edges that are connected to the node.

The degree distribution of a network can be represented as a histogram, with the x-axis representing the degree of a node and the y-axis representing the number of nodes with that degree. The shape of the degree distribution can provide insights into the structure and organization of the network. For example, a network with a highly skewed degree distribution, where a few nodes have a large number of connections and many nodes have very few connections, may be considered a "scale-free" network.

Understanding the degree distribution of a network can be useful for a variety of applications, such as identifying key nodes or groups of nodes within the network and understanding the role that these nodes play in the network, predicting the spread of diseases or information within the network, and designing efficient routing algorithms for networks.

- **PageRank:** is a measure of the importance of a webpage or node in a network. It was developed by Google as a way to rank web pages in the search results, based on the idea that a webpage is more important if it is linked to by other important web pages.

  It is calculated using a variant of the eigenvector centrality measure, which is a measure of the importance of a node in a network based on the number and importance of its neighbors. In the case of PageRank, the importance of a webpage is based on the number and importance of the other web pages that link to it.

  PageRank is typically calculated using an iterative algorithm, in which the importance of each webpage is initialized to a starting value and then updated based on the importance of the web pages that link to it. The algorithm is repeated until the importance values of the webpages converge to a stable value.

  It has been widely used in the field of information retrieval and is an important component of the Google search ranking algorithm. It is also used in other contexts, such as ranking the importance of nodes in social networks or identifying key nodes in other types of networks.

2. Results & Discussion

In this section, we present the results from both the first and second parts of the lab. Additionally, we provide an overview of the network we designed and implemented in the second part.

2.1. Analyzing network models

The first part of this lab focused on the analysis of some properties of the three random network models already presented (Erdos-Renyi, Watts-Strogatz & Barabasi-Albert model).

For the first task of this section, we reproduced a Watts-Strogatz network using the *sample_smallworld* function of the *igraph* library in *R* using the following configurations:

- *size = 1000 (the size of the lattice along each dimension)*
- *dim = 1 (the dimension of the starting lattice)*
- *nei = 4 (the neighborhood within which the vertices of the lattice will be connected)*
- *i = a variation between 0.0001 and 1*

The graph was reproduced 20 times for each *p* value, and each time the clustering coefficient (*transitivity* function of *igraph*) and the average shortest path were calculated, and then the mean of each of these measures was used to plot the following diagram:

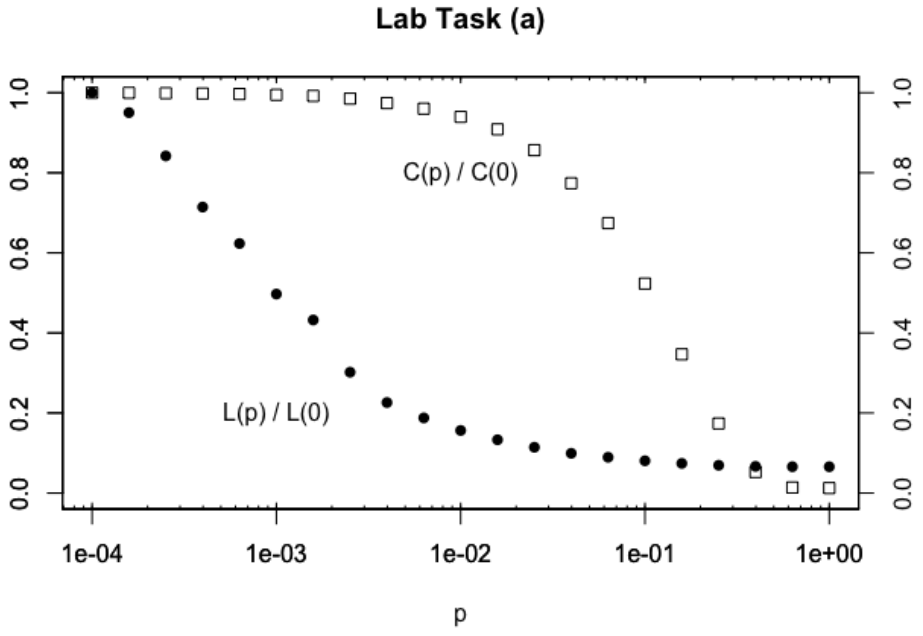

Figure 1: the clustering coefficient and the average shortest path as a function of the parameter p of the WS model

The plot shows the clustering coefficient *C(p)* and the average shortest path *L(p)*, both normalized to their values for the regular network *(C(0), L(0))*. We can see that there is a wide range for the rewiring probability *p*, where the networks have clustering similar to that of the regular network and an average shortest path length similar to that of the random network. Within this range, the networks exhibit small-world attributes. We can also observe that there is a regime in the middle of the plot *(p = 1e-02)* with high clustering but low mean of shortest distance. As *p* increases, both the clustering coefficient and the average shortest path decrease. When *p = 1*, we obtain the lowest values for the clustering coefficient as well as for the average shortest path, and this results in the Erdos-Renyi model.

The next task was to plot the average shortest-path length as a function of the network size of the ER model. For the average shortest path to be defined, the generated graph by the ER model needs to be connected. If the generated graph is not connected, then there exist at least two nodes *i* and *j* that have no path to each other, resulting in an undefined average shortest path value. To ensure, that the ER model is surely connected, *p* needs to be chosen as follows:

$$p > \frac{(1+\epsilon)ln(n)}{n}$$

with $\epsilon > 0$. If *p* is smaller than the specified value above, then the graph will almost surely be disconnected. We reproduced a Erdos-Renyi network using the *sample_gnp* function of the *igraph* library in *R* using the following configurations:

- *n = from 0 to 60.000 (Number of vertices)*
- $p = \frac{(1+\epsilon)ln(n)}{n}$ with $\epsilon = 1e-4$ *(The probability for drawing an edge between two arbitrary vertices)*

When lower values were used for epsilon (e.g. *1e-10*, then graphs with a small node count (i.e. $1 \leq n \leq 10$) were often disconnected and thus, the average shortest path value was undefined. For $\epsilon = 1e-4$ or higher values, this was not the case.
The difficulty of the implementation was the fact that the runtime increased significantly with higher node counts. For this reason it was important to write the R-program efficiently, so that the runtime is acceptable for very high node counts such as *50K* or *60K*.
For each number of nodes, the graph was reproduced *3* times, and each time the average shortest path was computed, and then the mean of the three values was taken to plot the following diagram:

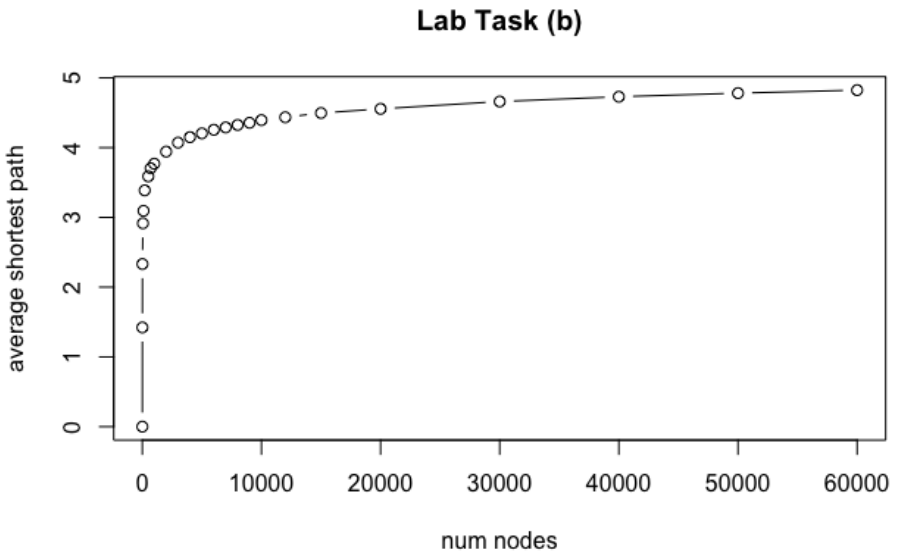

Figure 2: The average shortest-path length as a function of the network size of the ER model

Given a node count *n*, the plot shows the average shortest-path for a graph generated by the ER model. It is visible that the average shortest path rises very quickly for an increasing number of nodes ($0 \leq n \leq 1000$) up to a value of about *3,7*. But for approximately $n \geq 1000$, the curve begins to flatten significantly and the average shortest path value now increases very slowly. Even for very high node counts (e.g. $n = 50K$, $n = 60K$) this is the case. This indicates that the average shortest path converges when the node count n goes towards infinity. From the plot we can extract that the convergence value must be around *5*. As an average path value of *5* can be regarded as a short path length, this plot shows that paths in graphs generated by the ER model are generally short.

The third and final task in this section was to create a Barabasi-Albert Graph and draw a histogram of the degree distribution. For this purpose, we used the *sample_pa* function in the *igraph* library in *R*. The configuration of the models was as follows:

- *n = 10000 (Number of vertices)*
- *m = 3 (the number of edges to add in each time step)*

Figure 3 shows the degree distribution of this graph and Figure 4 shows the same distribution but in log scale.

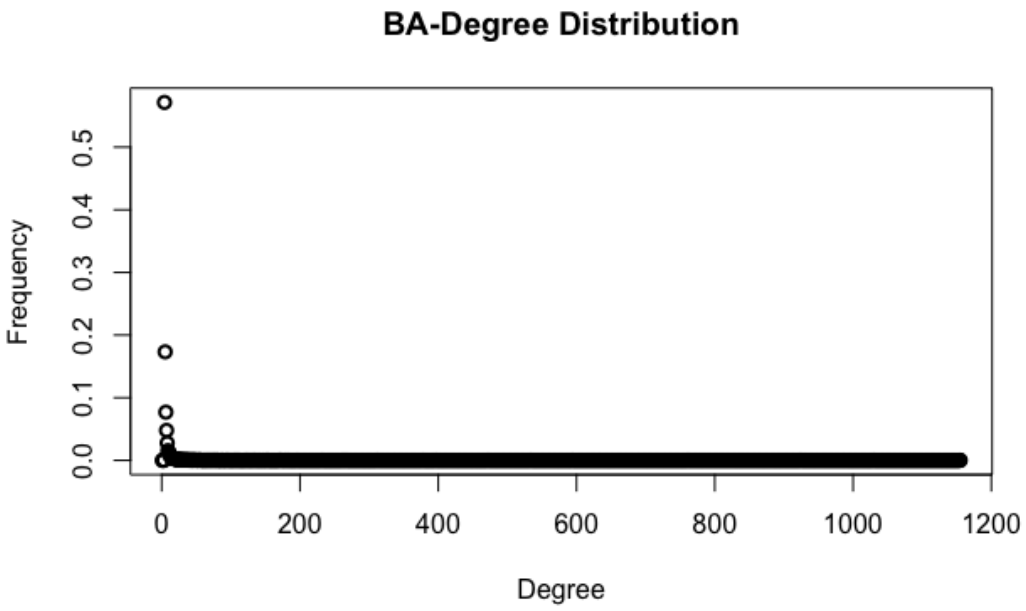

Figure 3: The degree distribution of BA-Graph

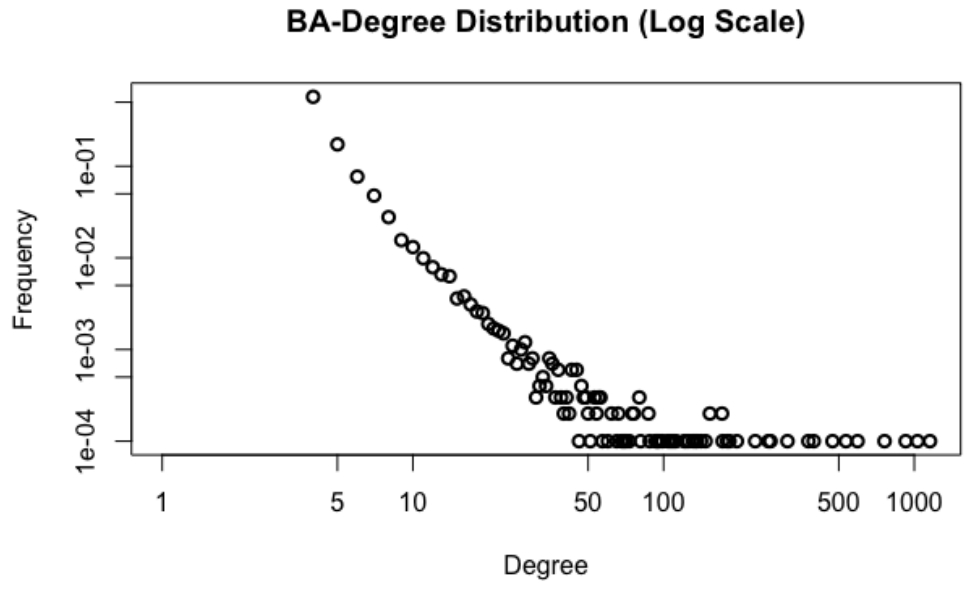

Figure 4: The degree distribution of BA-Graph in log-log scale

From the plot, it can be seen that a small number of nodes have a large number of connections, while the majority of nodes have a small number of connections. This is consistent with the characteristics of the BA model, which is known for its scale-free properties and the presence of highly connected nodes or "hubs". As a conclusion, we

can say that the degree distribution of the BA graph we constructed follows a power law distribution.

2.2. Analyzing and building our own network

For this part of the lab, we had to use a collection of texts to build a network. We chose the "IMDB Dataset of top 1000 movies and tv shows" from Kaggle. This dataset contains information about a variety of movies and television series. It includes information such as the title of the movie or series, the year of release, the content rating, the total running time, the genre, the rating on IMDB, a brief synopsis, the name of the director, the names of the main actors, the total number of votes the movie received on a review website, and the amount it earned at the box office.

2.2.1. Preprocessing

We started by combining important columns into one column for further processing. So we combined the name of the show, the year of release, the genre, the description, the name of the stars and the name of the director in one column, which we called *"Info"*. After that, we applied different techniques (in *Python* using the *nltk* library) that we have already learned in the class to the text that we gathered in the *"Info"* column. These techniques were:

- Lowercasing all worlds
- Tokenizing the text
- Removing English stop words
- Removing of punctuation marks
- Applying "Porter Stemmer" to the tokens

The result was a vector of tokens which we thought was a good representation of each show. Then, we calculated the Jaccard similarity between each 2 shows and removed the results of those shows that had 0 similarity.

2.2.2. Building the Network

After keeping only shows that have at least a Jaccard similarity > 0, we conducted some experiments to define a threshold value at which an edge between two shows would be added.

We obtained very low similarity text scores when analyzing the dataset of movies and series (as shown in Figure 5). We believe this may be due to a few factors. Firstly, the dataset may be relatively small, which could limit the amount of context and information available for accurately measuring the similarity between texts. Additionally, the movies and series in the dataset may not be closely related to each other, which could also affect the similarity scores. Therefore, we decided to use a value of 0.08 as a threshold for adding an edge between two nodes.

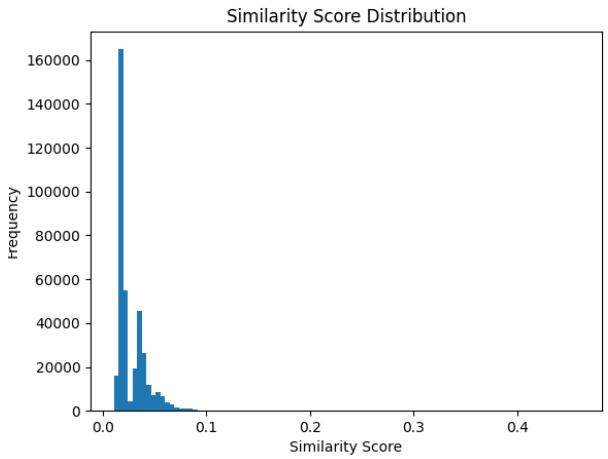

Figure 5: Jaccard similarity between the movies/series

After we built the network, we checked to see if it was connected and found that it was not. There were about 10 nodes that were isolated or only connected to each other. So we decided to remove them and keep the largest connected component. The resulting network has 455 nodes and 665 edges. Figure 6 provides a graphical representation of the network.

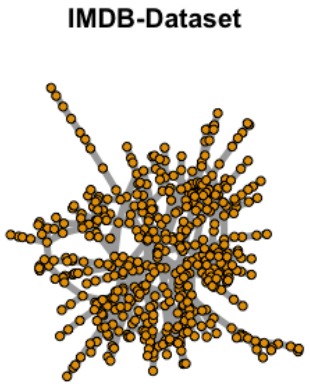

Figure 6: IMDB Dataset Network

By performing some analysis of the network, we have obtained the following measures:

- *diameter: 20*
- *average shortest path length: 7.36302464055768*
- *clustering coefficient: 0.18775025950850124*

We also applied the PageRank algorithm to the data with a damping factor of 0.85. The PageRank values were very low (between 0.008 and 0.0006). Several reasons might explain why this occurred: the network may be small and not well connected, so the PageRank values of the nodes are naturally low, or the network may be structured in such a way that there are few or no high-ranking nodes, resulting in low PageRank values for all nodes.

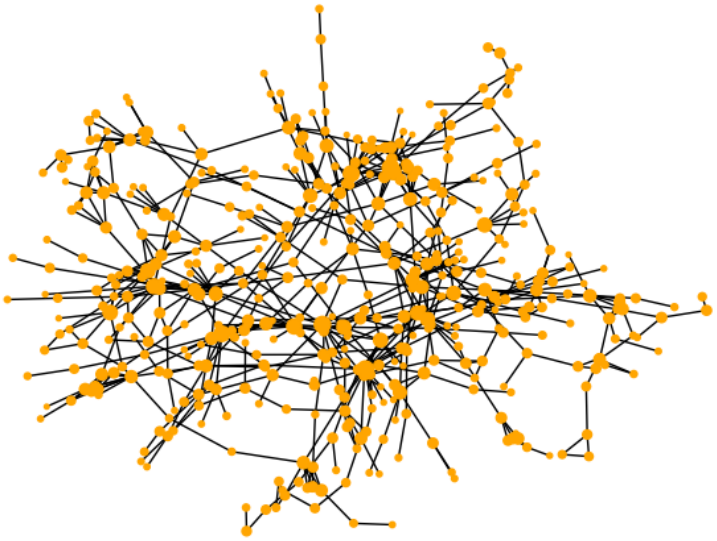

Figure 7: The network with node size proportional to pagerank score

Comparing the clustering coefficient of our network with that of a random network generated by the Erdos-Renyi model was our idea to check whether our network is random or not. If the clustering coefficient for our network is significantly lower than that of the random network, this could indicate that it is more likely to be a random network.
We constructed an ER graph with the following p value:

$$p = \frac{e}{1/2\, n(n-1)}$$

The clustering coefficient of the ER graph was much smaller than our network. Therefore, we concluded that our network cannot be a random graph.

Taking a look at the degree distribution of our network, we can see that it follows the power law distribution.

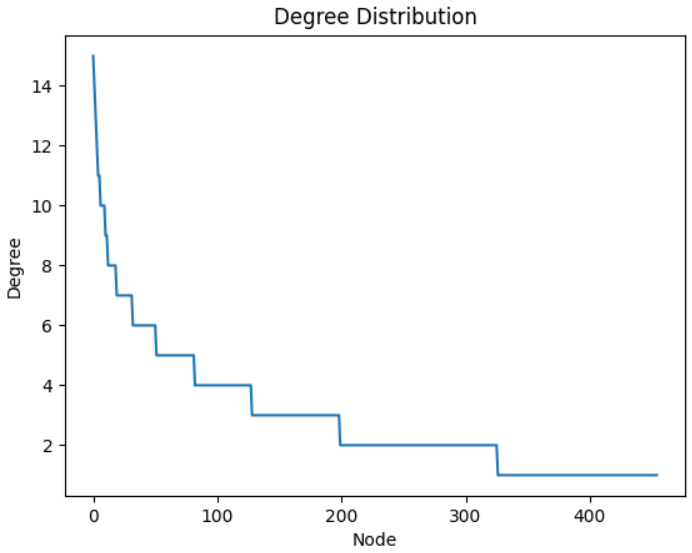

Figure 8: The degree distribution of our network

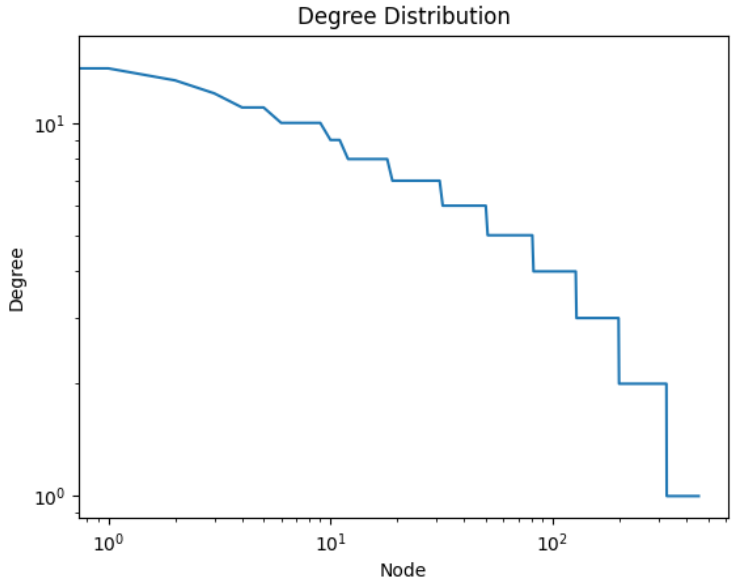

Figure 9: The degree distribution of our network (Log-Log Scale)

It is worth noting that all of these analyses were performed in *Python* using the *networkx* library.

We then exported the edge list of our network and built it in *R* using *igraph* to perform community detection. The reason for this is to use the *ClustAnalytics* library, which performs multiple community detection algorithms on the network and provides all the important

scores to evaluate each algorithm. We have chosen the algorithm with the highest modularity, which is the *Louvin* algorithm. The following figures illustrate our results:

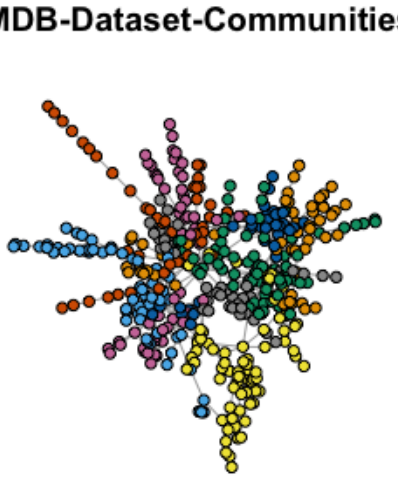

Figure 10: Communities of the network

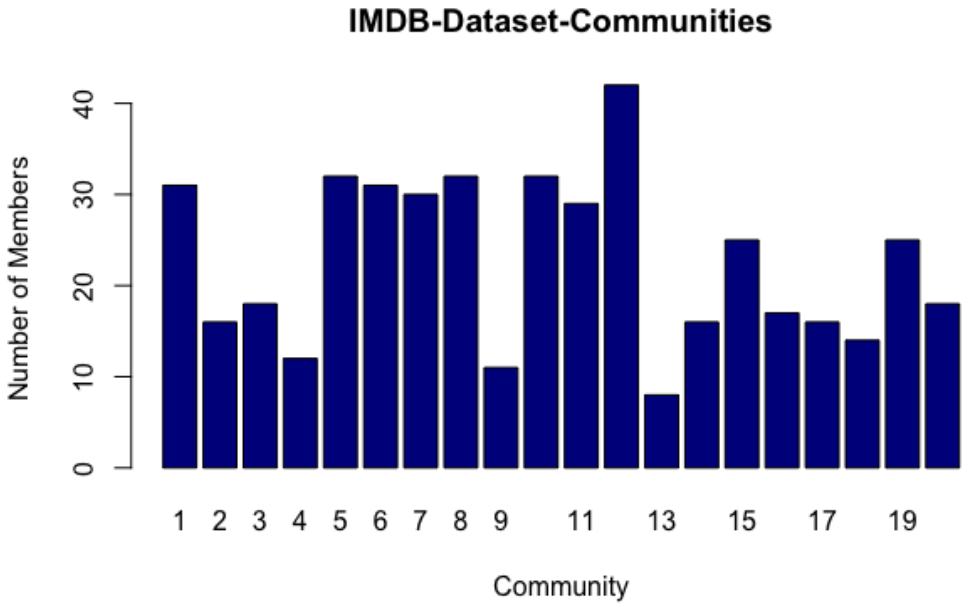

Figure 11: Size of each community

The results we obtained suggest that our network has a relatively small number of nodes and edges, with very low PageRank values and a relatively low clustering coefficient. The

diameter of the network is relatively small, indicating that there are relatively few nodes that are very far apart from each other, and the average shortest path length is also relatively small, suggesting that there are relatively few steps required to reach most nodes in the network. The power-law degree distribution and the presence of communities in the network suggest that there may be some degree of structure in the network.

Overall, this lab demonstrated how useful network analysis is for understanding synthetic and real-world networks. Through the use of various analysis techniques, we have uncovered insights and patterns that would have been difficult to discover without applying these techniques.

References

Barabási, A. L., & Albert, R. (1999). Emergence of Scaling in Random Networks. Science, 286(5439), 509–512.

Erdös, P., & Rényi, A. (1986). The Evolution of Random Graphs. Extremal Graph Theory With Emphasis on Probabilistic Methods, 37–44. https://doi.org/10.1090/cbms/062/06

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. Nature, 393(6684), 440–442. https://doi.org/10.1038/30918