IRRS Lab 5: Network Analysis

Mohana Fathollahi
Kathryn Weissman

January 3, 2023

1 Introduction

For this report, we used python with Google Colab and the package NetworkX[1] to explore different network models. First we demonstrate well-known properties of the randomly generated Watts-Strogatz Model and Erdős–Rényi Model. Then we move on to analyze a real world network. At the end we analyze wiki-vote graph and answer questions related to that.

2 Watts–Strogatz Model

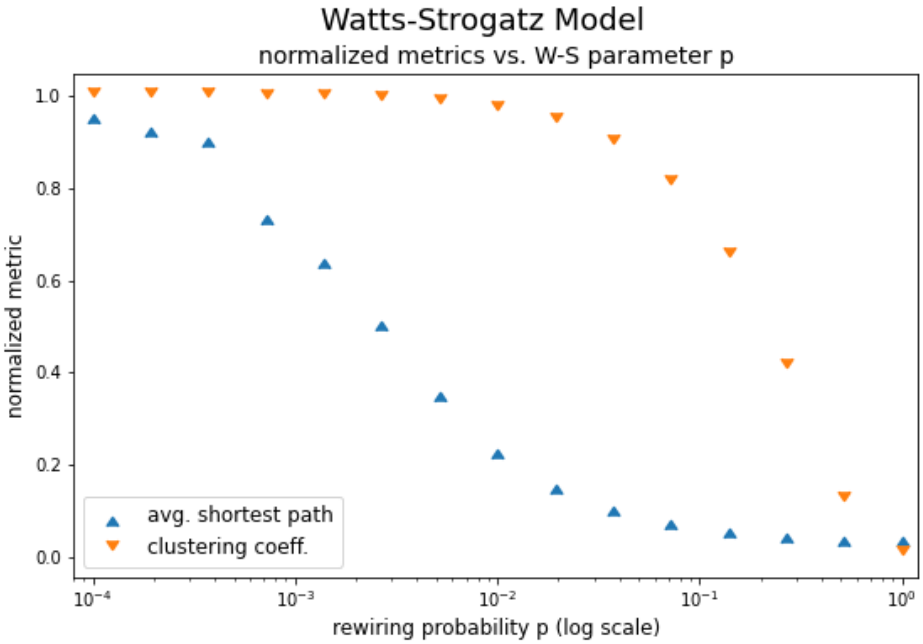

Figure 2.1: The clustering coefficient and average shortest path in the small-world model of Watts and Strogatz as a function of the rewiring probability, p. The values are normalized by dividing by their respective maximum values, which occur when $p = 0$. The largest gap between the two values occurs around $p = .01$ when the clustering coefficient is high and the average shortest path is low.

The small-world effect is used to describe a network property that most pairs of vertices are connected by a short path, and it is true for most networks. Although the theory was introduced earlier, it was

famously demonstrated in experiments by Stanley Milgram in the 1960's where people sent letters with the objective of reaching a specific person. For the letters that reached the target person, on average it took less than six passes, and the phenomenon coined the term "six degrees of separation."[2]

Watts and Strogatz developed the small-world model, which starts with a lattice and then rewires the edges. Each edge is rewired with probability p, where one end is connected to a different lattice node chosen uniformly at random. As the rewiring probability approaches 1, the graph becomes more like a random graph with more edges becoming connected to different random nodes. It can never truly become a random graph because of the limitations proposed for the model that only one end of each chosen edge is rewired and no double-edges or self-edges are created.[2]

As shown in Figure 2.1, the model transitions from large-world at $p = 0$ to small-world as the rewiring probability increases. The values in the plot represent an average of 20 simulations with different random seeds per graph parameter, p. Using Google Colab, it took approximately 8 minutes to run the simulations. The figure also shows the clustering coefficient, or global transitivity, which measures the probability that the adjacent nodes of a node are connected, forming a triangle. It is calculated using an approximation algorithm proposed by Schank et al.[3] [4]

3 Erdős–Rényi Model

The Erdős–Rényi model is a way of generating random graphs with parameters n and p. n represents the number of nodes in the graph, and p is the probability of creating an edge between each node, independent from every other edge.[5] In our experiment, we set p as a function of n and ϵ in order to get connected graphs that allow us to accurately measure the average shortest path. We set $\epsilon = 0.6$ for every graph. We used values of n ranging from 50 to 19,030. Although we hoped to measure the average shortest path of a network with 40,000 nodes, the algorithm did not complete one simulation after running for more than 30 minutes.

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

As shown in Figure 3.1, the experiment demonstrates that as n grows, the average shortest path demonstrates logarithmic growth and begins to stabilize. For a graph of size 19,030 nodes, the average shortest path is less than 4. The values in the plot represent an average of multiple simulations per graph with parameters n and p, except for the largest graph which was only calculated once.

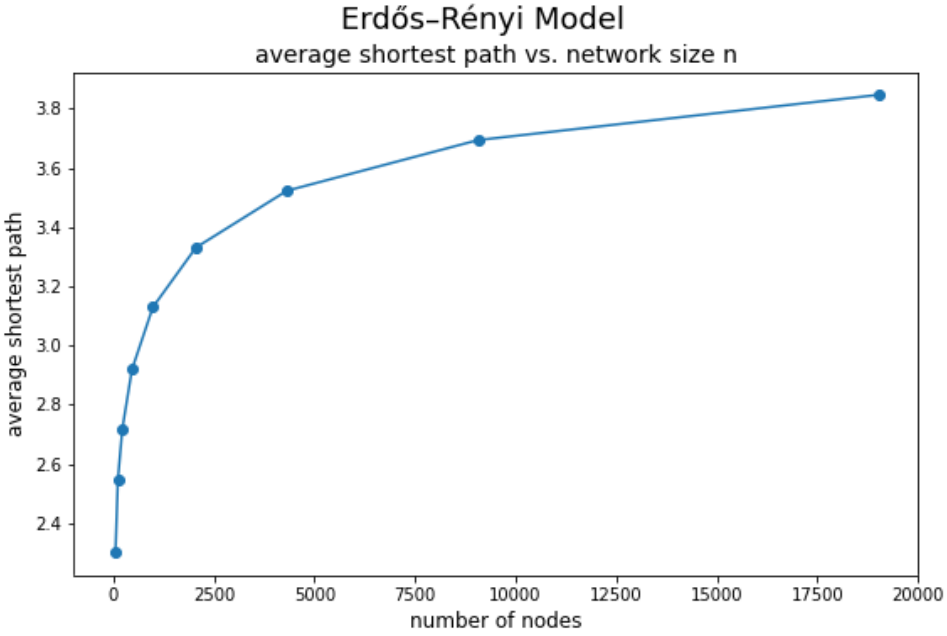

Figure 3.1: The average shortest path in the Erdős–Rényi model as a function of the number of nodes.

4 Analyze wiki-vote network

For this part we used wiki-vote data set.[6] This network contains nodes that represent wikipedia users that participate in the voting. The goal of this voting is selecting an adminship between wikipedia users. Additionally, if user i vote to user j we will have an edge between these two nodes. We provided codes for graph analysis in Graph_wiki_vote.ipynb.

4.1 Size and Diameter in wiki-vote network

In wiki-vote graph we have 7115 nodes and 100762 edges.
To find diameter in our graph we should consider that do we have connect graph or not? We do not have connected graph and instead of diameter for whole graph we can find diameter for each component. In our graph we have 24 components, the first component has 7066 nodes that is giant sub graph and has 99.3% of nodes. Other components have small amount of nodes that represent some users that preferred to vote to not so popular users and popular users do not vote to them.

$$[7066, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$$

In the giant component we need to path seven vertices to find the longest shortest path between users. With considering 7000 nodes in this sub graph we can say that nodes in sub graph are very well connected to each other.

4.2 Transitivity

Transitivity in our graph is 0.125. As we know in numerator of transitivity formula we consider number of closed triples and in denominator we consider connected triples. In our graph we have low value for transitivity that shows that number of closed triples are much lower than connected triples.

4.3 Degree distribution

To analyze the degree distribution of the wiki-vote graph we visualized the degree sequences in a degree rank plot and a degree histogram, shown in the figure 4.1. The degree rank plot sorts the nodes in decreasing order by degree, so the node with rank one has the highest number of connections and the node with rank two has the second highest number of connections, etc. The histogram considers unique degrees in our graph and plots the number of nodes with a given degree. The plots show that there are a large number of nodes with low degree and few nodes with high degree. For example we have just one node with degree higher than 1000, which is node number "2565" and majority of nodes have less than 200 connections. As we can see, a considerable amount of users have degree less than 100. To be more specific 6584 users have less than 100 connections, which in this case are votes.

When analyzing the degree sequence in a normal scale, we suspected that it followed power law distribution which is feature of scale free networks, so we also created plots in log-log scale shown in figure 4.2 in order to check. Since the line is not perfectly straight in log-log scale, it is clear the distribution does not follow one power law, but it may be a combination of two power law functions with different exponents.

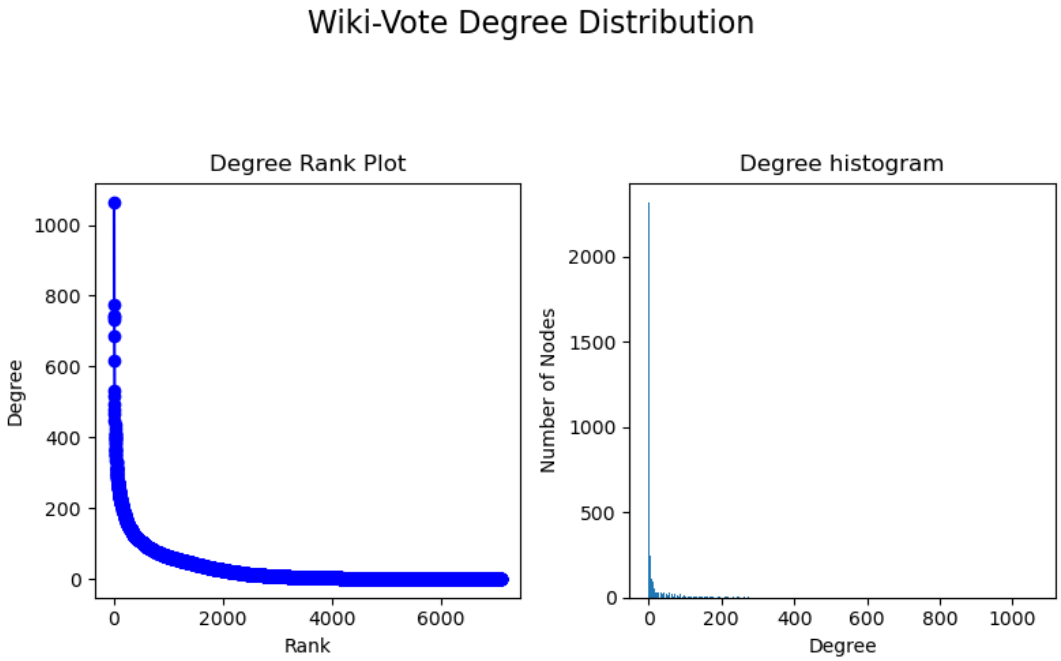

Figure 4.1: Degree Rank plot and Degree histogram of wiki-vote graph

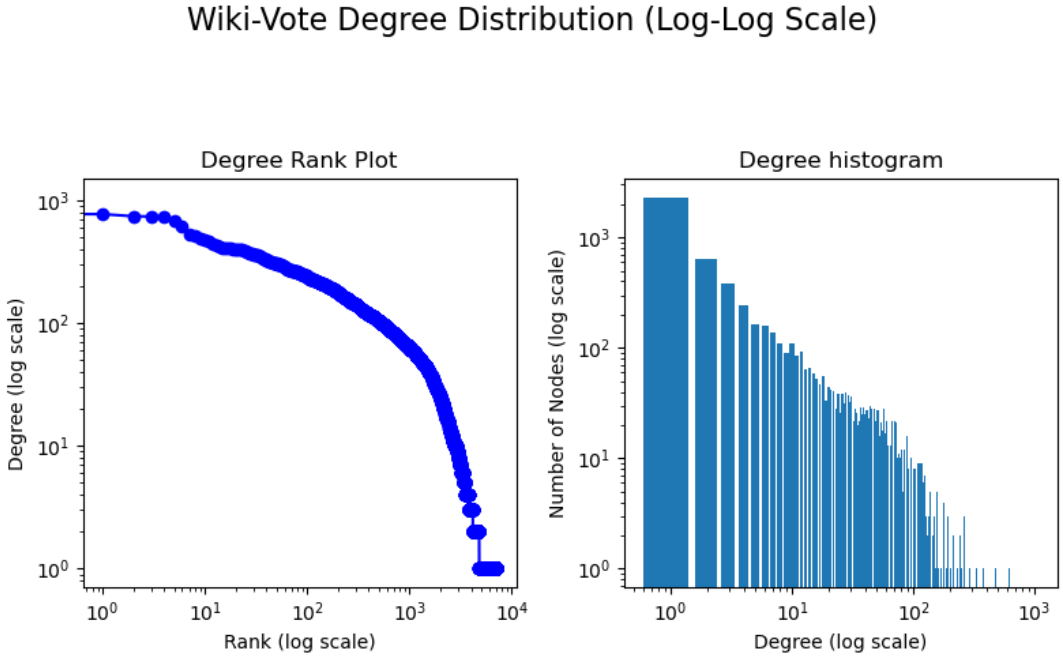

Figure 4.2: Degree Rank plot (log-log scale) and Degree histogram (log-log scale) of wiki-vote graph

4.4 Does Wiki-vote look like a random network?

To answer the question that does the graph look like random network or not, we should remember that in the random network the degree distribution follow normal distribution not power law distribution. Therefore, wiki-vote does not seem random network. Additionally, in the random network probability of having edge between two nodes is equal to clustering coefficient (p=c). In below we calculated this probability. As mentioned before we had 0.125 as transitivity that is much higher than p, therefore we cannot say that wiki-vote is random network.

$$p = \frac{m}{\binom{n}{2}} = \frac{100762}{\binom{7115}{2}} = 0.00398$$

4.5 Page rank

In the next step we will compare page rank for top 20 nodes that have highest degree. The reason for not considering whole graph is that we have more than 7000 nodes and visualising all nodes and considering the size of each node based on page rank value is not possible.
In the plot 4.3 page rank for top 20 nodes have been provided. Highest page rank belong to the node 2565 that previously mentioned as a node with highest degree and lowest page rank belong to node 1608.

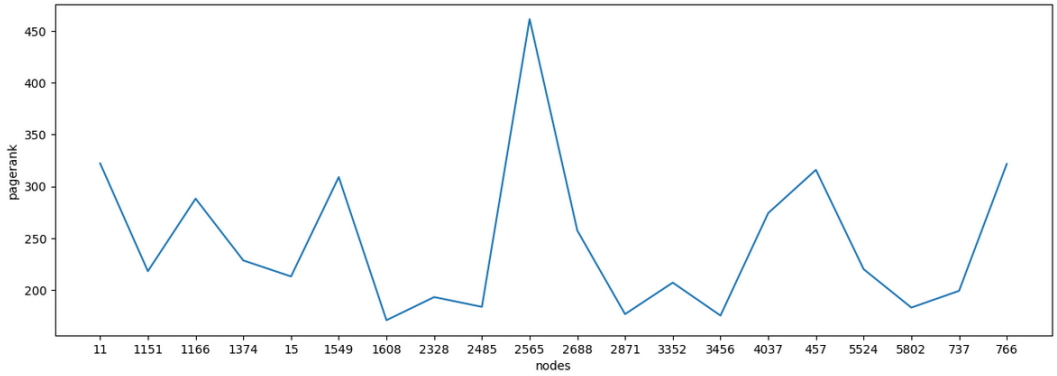

Figure 4.3: Page rank for top 20 nodes. The value is multiplied by 10^4 for visualization purposes.

In the plot 4.4 size of each node is based on its page rank. For example node 2565 has largest size compare to others and node 1608 or 3456 have smallest size.

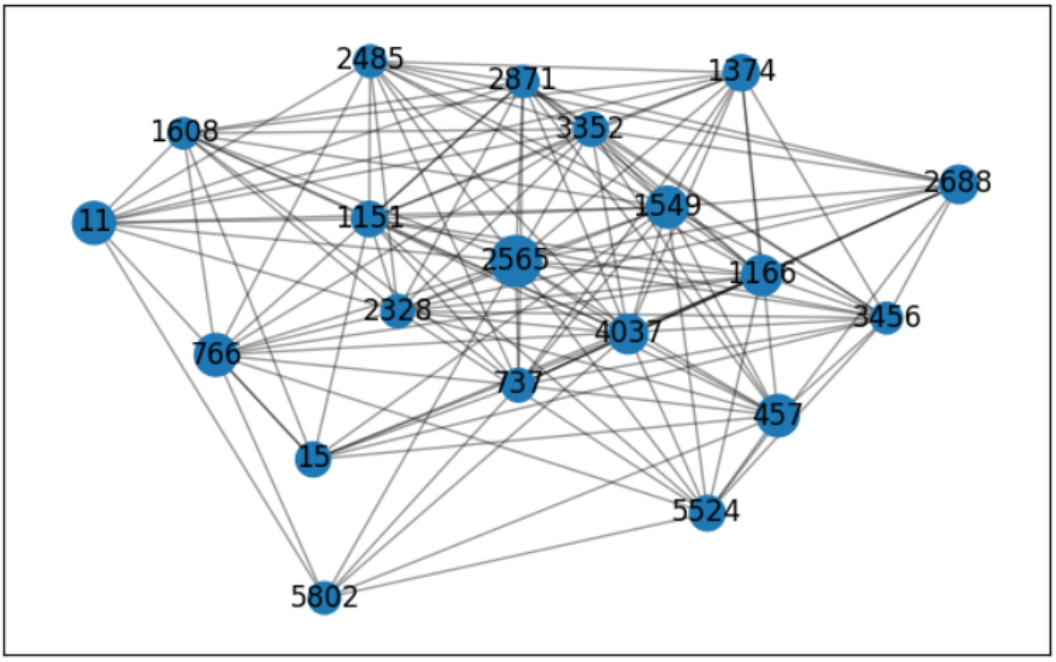

Figure 4.4: Sub-network based on top 20 nodes

4.6 Community detection

We used Girvan-Newman algorithm for community detection. In this algorithm we put our graph as input and it returns the edge that should be removed from the graph in each iteration. As a default it removes the edge with the highest betweenness centrality in each iteration.
We could not run Girvan-Newman algorithm for whole graph in our machine, therefore we used a subset of a graph that consist top 300 nodes that have highest degree in the graph. In the plot 4.5 we can see that we have 2 communities: one of them has 299 nodes that are very well connected to each other and another one has 1 node, node number 8.

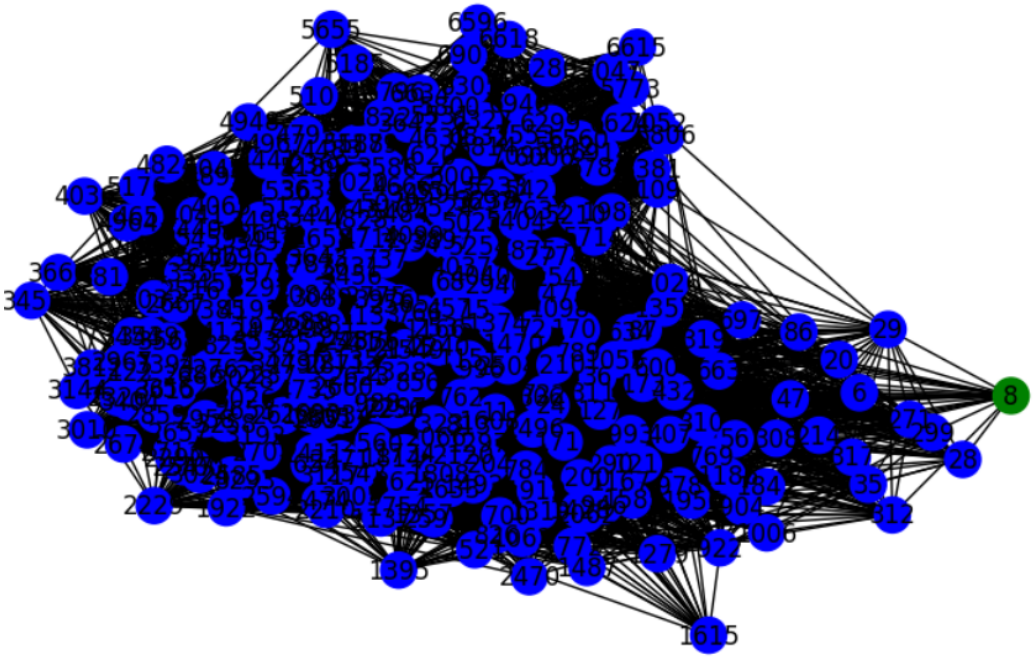

Figure 4.5: Communities based an Girvan-Newman algorithm

References

[1] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.

[2] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, jan 2003.

[3] NetworkX, "average_clustering." `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.approximation.clustering_coefficient.average_clustering.html`, 2022. Last accessed 31 Dec 2022.

[4] T. Schank and D. Wagner, *Approximating clustering-coefficient and transitivity*. No. 9 in Interner Bericht. Fakultät für Informatik, Universität Karlsruhe, Universität Karlsruhe (TH), 2004.

[5] Wikipedia, "Erdős–rényi model." `https://en.wikipedia.org/wiki/Erdos-Renyi_model`, 2022. Last accessed 21 Sep 2022.

[6] "Wiki-vote network." `http://snap.stanford.edu/data/wiki-Vote.html`.