

CSN Lab1 Report

Ali Arabyarmohammadi

September 2022

1 First Problem

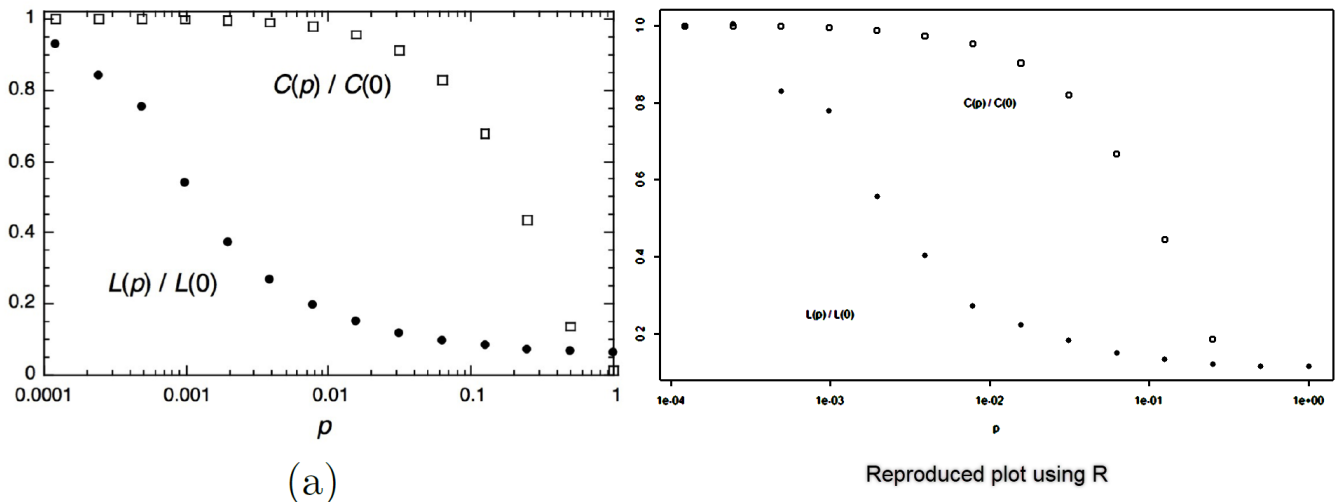
The purpose of the first exercise is to reproduce a given plot and illustrate the clustering coefficient and average shortest path length as a function of the parameter p of the **Watts-Strogatz** model. The clustering coefficient and the average shortest path values are adjusted to be within the range 0 to 1 to include both values in one figure.

In our model, a logarithmic sequence of length 14 was used to generate p values, which ranged from 0.0001 to 1. (Rewiring probability)

- $L(p)$ is a global attribute that calculates the **separation** between two vertices.
- $C(p)$ is a local property that assesses a typical neighborhood's **cliquishness**.

In order to get stable plot for each run of our codes in Rstudio, we had to generate **ten** random graphs with size **500** and then using the mean values of the average path length and clustering coefficient. Then we normalized the values and plotted the value of $C(p)/C(0)$ and $L(p)/L(0)$. In our codes, the only library package we used for this part of the assignment was the **"igraph"**, and by using the function **"sapply"** instead of a **"For-Loop"** we reached a good result in a reasonable execution time by writing a minimum line of codes.

In plot (a), we see the model that we are trying to reproduce, and in the figure next to it, we have shown the result we got in Rstudio.



We quantify the structural characteristics of a graph by its characteristic path length $L(p)$ and clustering coefficient $C(p)$ as p grows from zero to one, as shown in the figure below.

In plotted data, $C(p)/C(0)$ - the clustering coefficient of p over the clustering coefficient of zero - starts at 1, but it's concave down. It goes out and then curves down toward 0.

As we used a logarithmic scale on the x-axis, this dropping down to 0 happens pretty quickly.

On the other hand, the $C(p)/C(0)$ drops down from 1 to 0 much more slowly than $L(p)/L(0)$. So because it looks sort of concave up and dropping from 1 to 0 and the other one concave down and dropping from 1 to 0, where the $L(p)/L(0)$ has gotten close to 0, and the clustering coefficient has remained close to 1, those pieces give us small-world graphs.

So we can see the clustering coefficient remains high even as the path length drops to 0 quickly. So, graphs produced with p in the middle area of the plot are highly clustered and have a short average distance between vertices.

2 Second Problem

In this part, we will plot the average shortest path length as a function of the network size of the Erdos-Renyi model. To make sure that we generate p such that the graph $G(n, p)$ will be connected, we have chosen the value p using the function:

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

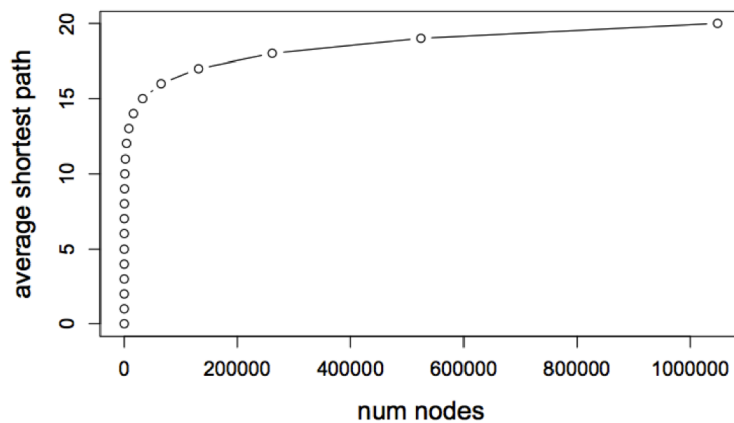
The calculation time to create an Erdos-Renyi graph is too long for larger values of vertices n . As a result, n had to be adjusted. So, for achieving the result in a reasonable time, we used this set of numbers as our nodes: (2, 4, 8, 16, 32 ... 4096, 8192, 16384).

The values of p were calculated for this size n , and the average shortest path length of the graph was determined accordingly.

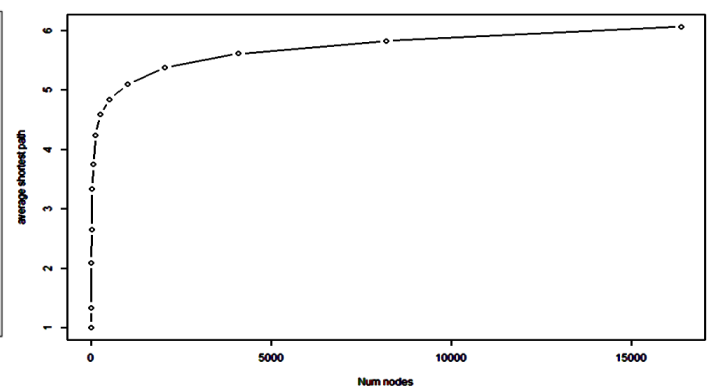
We have shown the model we are attempting to recreate in plot (b) and the outcome we achieved in Rstudio next to it.

For the purpose of reusability and writing a clean code, the function called **"average"** was used in both problems 1 and 2. Like part 1, we just used the package **"igraph"** and implemented the function **"sapply"** to avoid writing a For-Loop. As we see in the following plot, the result we reached resembles the predefined plot.

From the plot can be inferred that average shortest path converges to a fixed value as the number of nodes n increases, given a fixed average degree. That is, the Erdos-Renyi graph can grow very large, yet nodes will still be close.



(b)



Reproduced plot using R