PHYS 5116 – Network Science I
Assignment 1

Giuliano Porciúncula Guedes

October 10, [Fall] 2025

1. Matrix Formalism (20 points)
Let $\boldsymbol{A}$ be the $N \times N$ adjacency matrix of an undirected unweighted network, without self-loops. Let $\mathbf{1}$ be a column vector of N elements, all equal to 1. In other words, $\mathbf{1} = (1, 1, \cdots, 1)^T$, where T indicates the transpose operation. Use matrix formalism (multiplicative constants, multiplication row by column, matrix operations like transpose and trace, etc., but avoid the sum symbol $\sum$) to write expressions for:

(a) The vector $\boldsymbol{k}$ whose elements are the degrees k_i of all nodes $i = 1, 2, \cdots, N$.

(b) The total number of links, L, in the network.

(c) The number of triangles, n_Δ, present in the network, where a triangle means three nodes, each connected by links to the other two. (Hint: you can use the trace of a matrix.)

(d) The vector $\boldsymbol{k_{nn}}$ whose element i is the sum of the degrees of node i's neighbors.

Solution

We will avoid the sum symbol $\sum$ in our answer, but we will use it to help inform our intuition on how to write it using matrix formalism. In all these answers, we will always keep in mind and talk in the context of the undirected unweighted network.
The degree of a node i can be written as

$$k_i = \sum_{j=1}^{N} \delta_{(i,j)\in\Lambda} \tag{1}$$

Where Λ represents the set of links in our network.
Our adjacency matrix exactly encodes the same information as Λ, but in matrix form. If we look closely, the adjacency matrix has elements $\delta_{(i,j)\in\Lambda}$, i.e., it's 1 if i, j are connected and 0 if they are not. Thus we can use the vector of ones $\mathbf{1}$ to extract the information we want. By multiplying $\boldsymbol{A}$ by $\mathbf{1}$ from the right, we are going to have a new vector where each element is k_i:

$$\boxed{\boldsymbol{k} = \boldsymbol{A}\mathbf{1}} \tag{2}$$

(b) We know that the total number of links is half of the sum of all degrees (half to avoid double counting)

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i \tag{3}$$

Since we have $\boldsymbol{k}$ which each element is the degree, if we multiply it by $\mathbf{1}^T$ from the left we have exactly the sum we want. Thus

$$\boxed{L = \frac{1}{2}\mathbf{1}^T\boldsymbol{A}\mathbf{1}} \tag{4}$$

(c) Since $\boldsymbol{A}$ encodes the information of the links, surely powers of it also encodes some interesting properties. Remember that the elements of $\boldsymbol{A}$ are $\delta_{(i,j)\in\Lambda}$. Look at what happens to the elements of $\boldsymbol{A}^2$:

$$(\boldsymbol{A}^2)_{i,j} = \sum_{\ell=1}^{N} \delta_{(i,\ell)\in\Lambda}\delta_{(\ell,j)\in\Lambda} \tag{5}$$

It is counting how many length 2 paths exist between i and j! The same happens for $\boldsymbol{A}^3$, it gives the number of length 3 paths:

$$(\boldsymbol{A}^3)_{i,j} = \sum_{\ell,\kappa=1}^{N} \delta_{(i,\ell)\in\Lambda}\delta_{(\ell,\kappa)\in\Lambda}\delta_{(\kappa,j)\in\Lambda} \tag{6}$$

Thus, if we want the number of triangles of a node i, we can look at $j = i$, which gives us the diagonal terms $(\boldsymbol{A}^3)_{i,i}$. If we want the total number of triangles in the network, we just sum all of them. That is the trace! But we have to be careful! For each of the three nodes in a triangle, we have two paths that makes the same triangle. Thus we are counting each triangle 6 times by taking the trace of $\boldsymbol{A}^3$ and we have to correct for that. So our final expression is

$$\boxed{n_\Delta = \frac{1}{6}\mathrm{Tr}(\boldsymbol{A}^3)} \tag{7}$$

(d) We have the vector $\boldsymbol{k}$ that gives us the degree of all nodes. We want the sum of the degrees of all the neighbors of a node i. This means we have

$$k_{nn}^{(i)} = \sum_{j=1}^{N} \delta_{(i,j)\in\Lambda} k_j \tag{8}$$

This is just the $i-$th element of the matrix multiplication between $\boldsymbol{A}$ and $\boldsymbol{k}$: $k_{nn}^{(i)} = (\boldsymbol{A}\boldsymbol{k})_i$. Thus we have that

$$\boxed{\boldsymbol{k_{nn}} = \boldsymbol{A}^2\mathbf{1}} \tag{9}$$

■

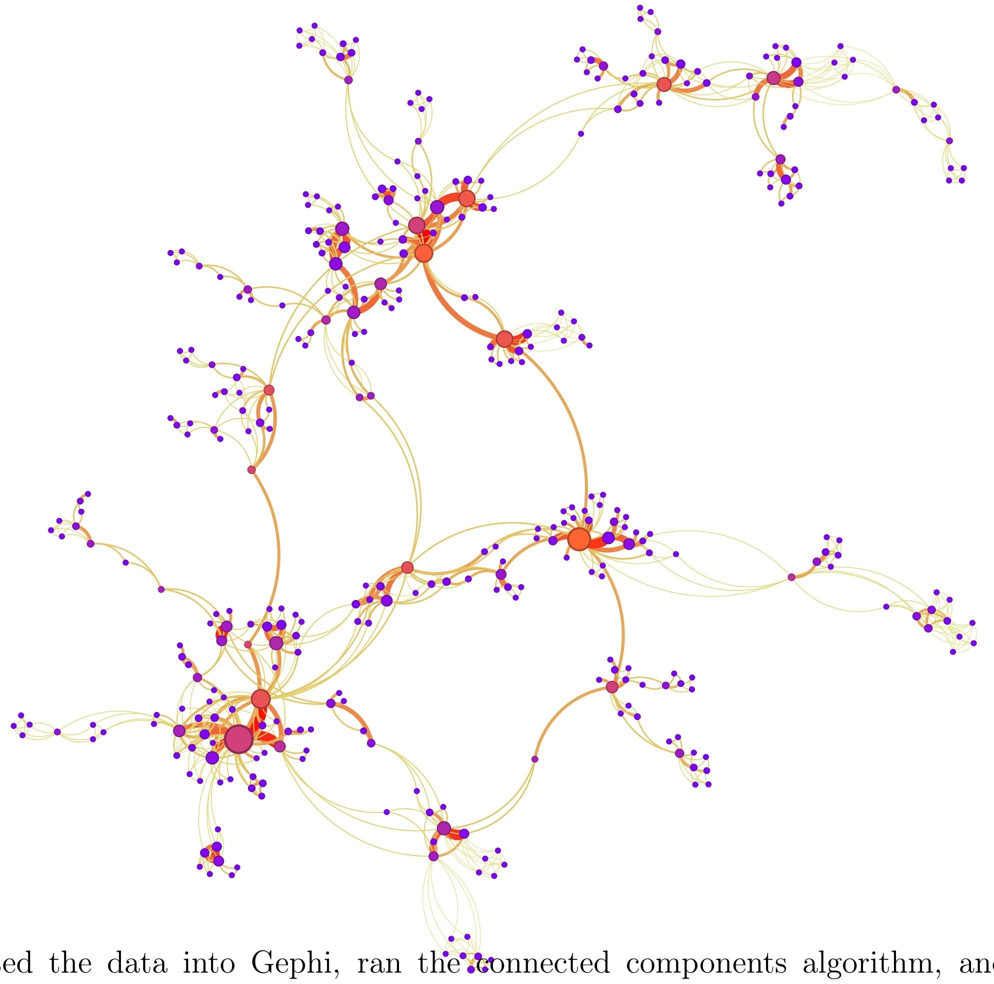
2. Visualization (20 points)

Download the netscience dataset at this URL: https://tinyurl.com/bd6cwsd4. As described in the .txt file, this is a coauthorship network of scientists working on network theory and experiments, compiled by Mark Newman in 2006. Visualize the largest component of the network (Gephi may be the easiest choice of software, but you are free to use your favorite), considering the following:

- Degree or other centrality measures
- Edge properties
- Layout

Make good use of color, size, and layout to create a clear, informative visualization. Describe your approach and comment on your observations (about 5-10 sentences).

Solution



We imported the data into Gephi, ran the connected components algorithm, and retained only the largest component. After computing network statistics, we applied a purple–orange gradient for betweenness centrality (low–high), scaled node size by weighted degree, and colored edges from yellow to red based on edge weight. The visualization reveals 6–10 high–centrality (orange) nodes acting as network bridges. Notably, high collaboration (weighted degree) does not imply high centrality—the most connected node is not the most central. Most nodes have weak collaborations (yellow edges), with only a few strong connections (red edges) between closely linked nodes.

■

3. Erdős–Rényi Networks (20 points)

Consider an Erdős–Rényi network with $N = 6000$ nodes, connected to each other with probability $p = 10^{-4}$.

(a) What is the expected number of links, $\langle L \rangle$?

(b) In which regime is the network?

(c) Calculate the probability p_c such that the network would be at the critical point.

(d) Given the linking probability $p = 10^{-4}$, calculate (or compute) the number of nodes N_{cr} that a given network would need to have in order for all of its nodes to be in a single component. If necessary, round to the nearest integer.

(e) For the network in (d), calculate the average degree $\langle k_{cr} \rangle$ and the average distance between two randomly chosen nodes $\langle d \rangle$.

(f) Calculate the degree distribution p_k of this network (approximate with a Poisson degree distribution). (Use the original network, not the one used in (d) and (e)).

(g) Using the degree distribution in (f), calculate the probability of a node i having degree $k_i = 4$.

Solution

(a) A fully connected network has $N(N-1)$ links (each of the N nodes connect to all other $N-1$ nodes). If each link happens with a probability p, then we have

$$\boxed{\langle L \rangle = N(N-1)p} \tag{10}$$

For $N = 6000$ and $p = 10^{-4}$, we have $\langle L \rangle = 3599.40$.

(c) The Erdős–Rényi network regimes are: subcritical for $p < 1/N$, critical at $p = 1/N$, supercritical at $p > 1/N$ but $p < \ln(N)/N$, and connected at $p \geq \ln(N)/N$. The critical probability is

$$\boxed{p_c = \frac{1}{N}} \tag{11}$$

that is $p_c \approx 1.67 \times 10^{-4}$.

(b) Since $p = 10^{-4}$, the network is in the subcritical regime.

(d) For all nodes to be in a single component, we need to be in the connected regime. So let's call that number of nodes as N_{cr}. This happens, as we wrote above, when we pass the threshold of

$$\boxed{\frac{\ln(N_{cr})}{N_{cr}} = p_{cr}} \tag{12}$$

Using wolfram mathematica to numerically solve for N_{cr}, we find that $N_{cr} \approx 116671$.

(e) $\langle k_{cr} \rangle$ is straightforward. Each node can make a maximum of L links. Since each link has probability p of happening, we have that $\langle k \rangle = (N-1)p$. This gives us

$$\boxed{\langle k_{cr} \rangle = (N_{cr} - 1)p_{cr}} \tag{13}$$

Or calculating it we have $\langle k_{cr} \rangle \approx 11.67$.

For the average distance, we first start by writing the number of nodes at a distance d of the startnig node is:

$$N(d) \approx \sum_{i=1}^{d} \langle k \rangle^{i} = \frac{\langle k \rangle^{d-1} - 1}{\langle k \rangle - 1} \tag{14}$$

Assuming very big $\langle k \rangle \gg 1$ and that $N(\langle d \rangle) \approx N$, we have the following equation

$$\langle k \rangle^{\langle d \rangle} \approx N \tag{15}$$

Thus we write

$$\boxed{\langle d \rangle \approx \frac{\ln(N)}{\ln(\langle k \rangle)}} \tag{16}$$

Calculating from our results, we find $\langle d \rangle \approx 4.75$.

(f) Since each link try is a Bernoulli trial, and we have $n-1$ trials for each node, we have the following binomial distribution:

$$\boxed{p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k}} \tag{17}$$

Approximating the binomial into a poisosn distribution follows. First we approximate the binomial coefficient

$$\binom{N-1}{k} = \frac{N!}{k!(N-1-k)!} = (k!)^{-1} \prod_{i=0}^{k-1} (N-1-i) \tag{18}$$

We assume $N \gg k$ for our approximation. This then makes $(N-1-i) \approx N-1$ since $i < k$. Thus we have our first result

$$\binom{N-1}{k} \approx \frac{(N-1)^k}{k!} \tag{19}$$

Now let's look at $(1-p)^{N-1-k}$. First we'll take the log

$$\ln\left[(1-p)^{N-1-k}\right] = (N-1-k)\ln[1-p] \tag{20}$$

Since $p < 1$ we can use the following expansion log

$$\ln[1-p] = \sum_{i=1}^{\infty} \frac{(-1)^{1+i}}{i} p^i \tag{21}$$

Remember that $p \ll 1$ for our approximation, thus

$$\ln[1-p] \approx -p \tag{22}$$

And so

$$\ln\left[(1-p)^{N-1-k}\right] \approx -(N-1-k)p \tag{23}$$

We know that $\langle k\rangle = (N-1)p$, thus we write $p = \langle k\rangle/(N-1)$ and so

$$\ln\left[(1-p)^{N-1-k}\right] \approx -(N-1-k)\frac{\langle k\rangle}{N-1} = -\langle k\rangle + \frac{k\langle k\rangle}{N-1} \tag{24}$$

Since N is very big, the second term goes to zero

$$\ln\left[(1-p)^{N-1-k}\right] \approx -\langle k\rangle \tag{25}$$

And so by taking the exponential of both sides we have our second result

$$(1-p)^{N-1-k} \approx \mathrm{e}^{-\langle k\rangle} \tag{26}$$

Substituting both results and that $p = \langle k\rangle/(N-1)$ to our probability distribution

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{27}$$

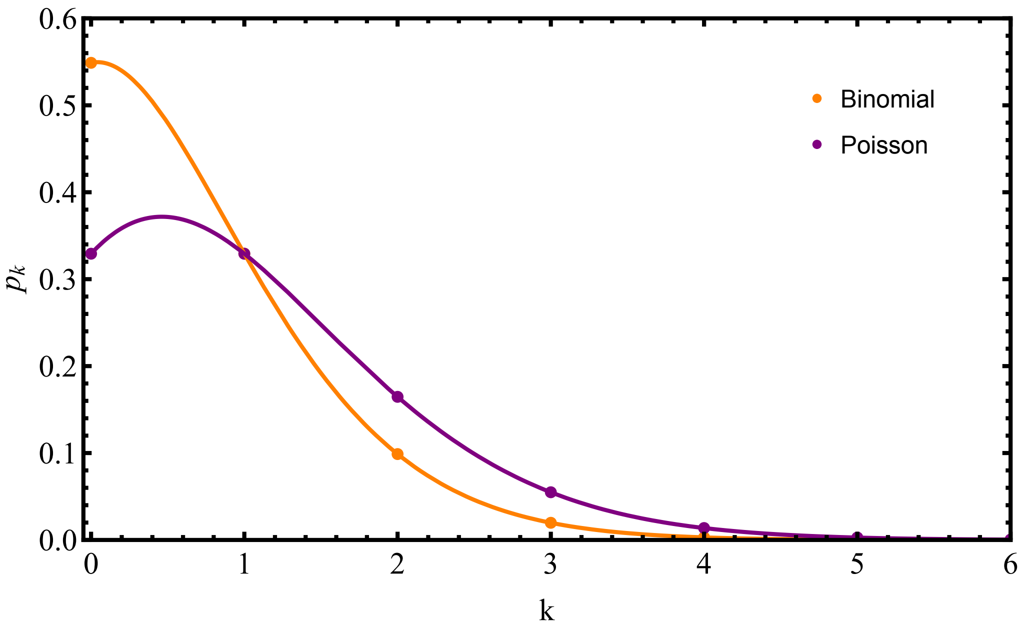
we have that

$$p_k = \frac{(N-1)^k}{k!}\left(\frac{\langle k\rangle}{N-1}\right)^k \mathrm{e}^{-\langle k\rangle} \tag{28}$$

After direct simplification, we have

$$\boxed{p_k = \frac{\langle k\rangle}{k!}\mathrm{e}^{-\langle k\rangle}} \tag{29}$$

Since $p = 10^{-4}$ and $N = 6000$, we have $\langle k\rangle = 0.5999$. Plotting both distributions we have



In my opinion, the poisson distribution was not a good approximation here.

(g) We want p_4, this is $p_4 \approx 2.96 \times 10^{-3}$ for the binomial distribution and $p_4 \approx 13.72 \times 10^{-3}$ for the poisson distribution. This is indeed a very bad approximation since they have a ratio of more than 4.5.

All the numerical part and plotting of this problem was done using Wolfram Mathematica. The notebook, which is called Porciuncula.nb, is available here.

■

4. Randomizing a network (20 points)

Download the airport-network.gml dataset at this URL: https://tinyurl.com/4rfrx4p9. Using the networkx library in Python, load the dataset with the function nx.read_gml() and perform the following analyses.

(a) Make a log-log plot of the degree distribution and compute the global clustering coefficient of this network.

(b) Randomize the network using a "full randomization" (see Fig. 4.17) and recompute (and plot) the degree distribution and report the new global clustering coefficient.

(c) Randomize the network using a "degree-preserving randomization" (see Fig. 4.17) and recompute (and plot) the degree distribution and report the new global clustering coefficient.

(d) Compare and contrast the degree distribution and global clustering coefficient across the different randomizations and original network.

(a)

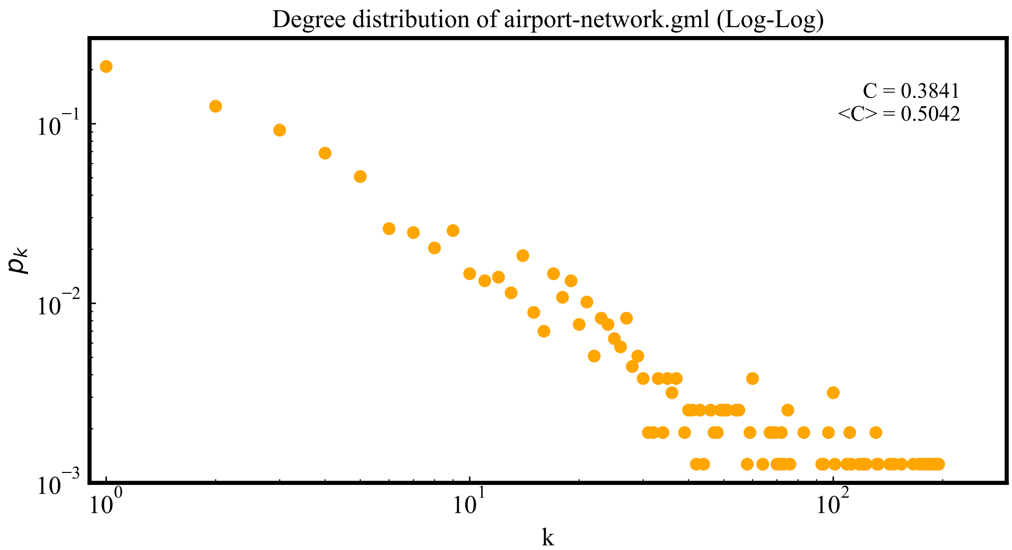


Figure 1: Log-log degree distribution of the airport-network.gml. We a power-law tendency and the presence of hubs.

The global clustering coefficient, also called transitivity, is defined as

$$C = \frac{3n_\Delta}{\text{Number of paths of length 3}} \tag{30}$$

We can, and will, also calculate the average clustering coefficient, which is the average of the clustering coefficient of node i

$$\langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i \qquad\qquad C_i = \frac{\text{Number of links between i's neighbors}}{k_i(k_i-1)} \tag{31}$$

(b)

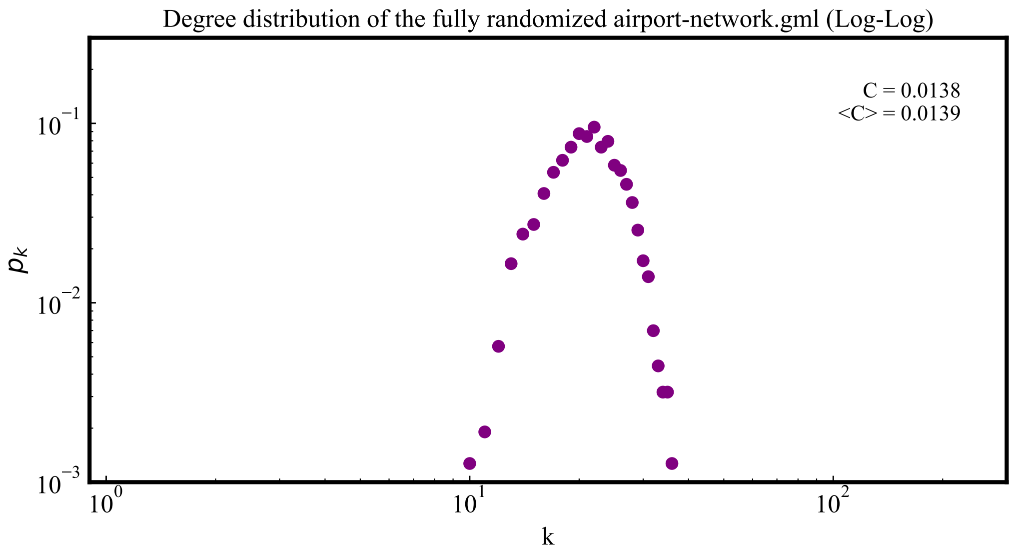


Figure 2: Log-log degree distribution of the fully randomized airport-network.gml. We destroyed the power-law tendency and the presence of hubs. This is a binomial distribution now, as expected by the random graph model. We also destroyed the clustering coefficients. The horizontal and vertical ranges are the same as Figure 1.

(c)

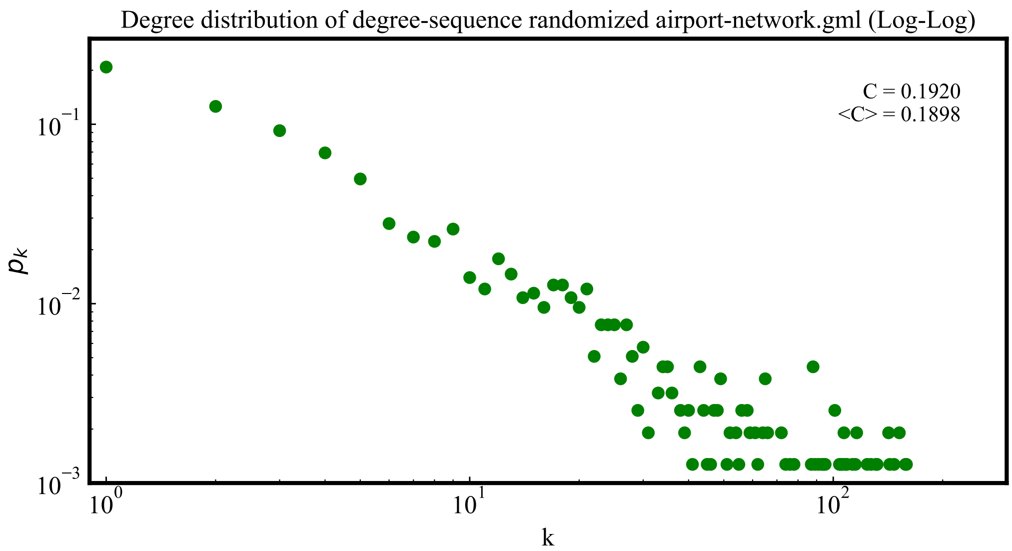


Figure 3: Log-log degree distribution of the degree-sequence randomized airport-network.gml. We can observe the power-law tendency and the presence of hubs. But we see that the clustering coefficients are significantly different than the original network. The horizontal and vertical ranges are the same as Figure 1.

(d) Our first two networks have a total of 1574 nodes and 17215 links, while our degree-preserving randomized network has 14881 links due to the configurational model creating double links and self loops that had to be removed. Our initial network follows a power-law tendency and the presence of hubs. So it's at least scale-free like. We did not characterize it by doing the scale-free tests to obtain quantities such as the exponent. Comparing our unrandomized network to the fully randomized we can see that both the degree distribution and the clustering coefficients are destroyed. In the randomized network, we no longer see the power-law tendency, presence of hubs, and the clustering coefficients are much smaller than the previous ones. Comparing our degree-sequence randomized network to our unrandomized network, we observe that we again see the presence of the power-law tendency is back along with the hubs. On the other hand, we see that the clustering coefficient is now considerably smaller than the unrandomized network. This means that the clustering in the unrandomized network was not by change. It was due to mechanisms that are not described by the degree-sequence. This "hidden" mechanism probably is preferential attachment.

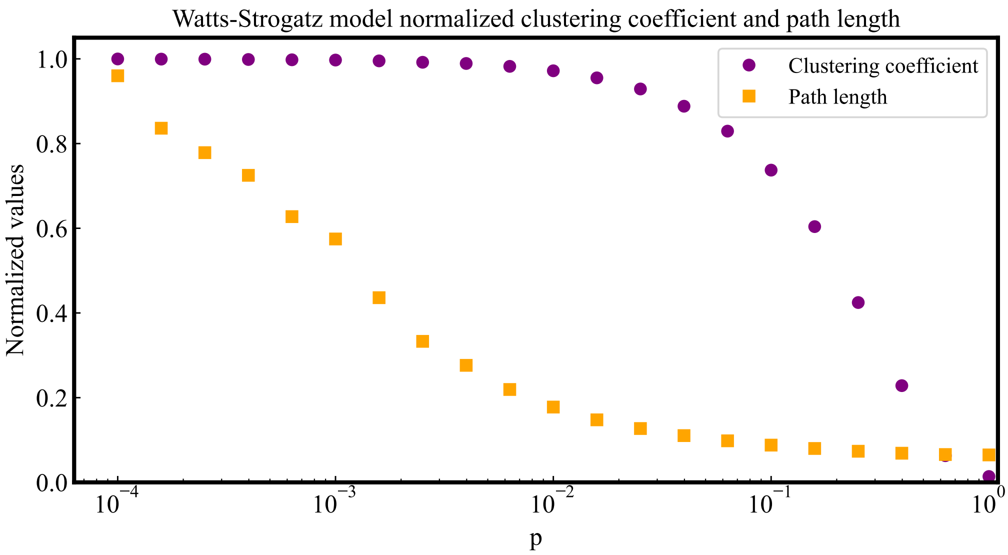
All the numerical part and plotting of this problem was done using a Jupyter Notebook, called Porciuncula.ipynb, that is available here.

■

5. Watts & Strogatz, 1998 (20 points)

The "Watts-Strogatz" model is a simple random network model that produces graphs that have 1) the small world property and 2) high clustering (see Box 3.9). Reproduce the canonical figure from the original 1998 article (also visualized in subplot d of Fig. 3.14 in Box 3.9), and briefly comment on the key insights from the figure.

Solution



The Watts-Strogatz model starts with a regular network, i.e., a node i is connected to the $k/2$ nodes forward and $k/2$ nodes backwards. We have boundary conditions $i + N = i$. Then, with probability p, a link from a node i to j is rewired to a randomly node k, thus i to k. We follow the same parameters as the original paper, it being $N = 1000$ nodes, each node starts with $k = 10$ neighbors, and we do the process 20 times for each point and take the average.

The figure shows us that as we re-wire our network with $p < 0.1$, we have that we keep a very clustered network, i.e., a node's neighbors are still linked. By raising p just a little above 0, we quickly see that the average path length drops drastically (the plot scale is in log-log!). This means that our network creates this long-distance links that enables the nodes in the network to quickly travel to any other node in the network. This is what we call small world property. But as we increase $p > 0.1$, we see that our clustering coefficient plummets, this is a result of turning our network into a random network, and we lose the original triangles.

All the numerical part and plotting of this problem was done using a Jupyter Notebook, called Porciuncula.ipynb, that is available here.