Complex Networks

Project 2

Group 5

Alexander Ferraro, Luc Maessen, Stavros Topkas, Myrthe van Leeuwen, Levi van de Pol

Contents

1 Real Networks

1.1 Scale free and not scale free networks

By browsing in the SNAP database, we decided to analyse the following Undirected networks: the Collab network and the Road network of Texas. We expect that the former would be a scale free network (i.e the degree distribution follows a power law [1]), this is because it might present some major hubs; the power law behaviour means that the vast majority of nodes have very few connections , while a few important nodes (Hubs) have a huge number of connections. In scientific collaboration you can expect that some senior professors have more connections, thus creating hubs dominating the network, while there are younger research assistants with very few connections.

On the other hand, by analysing the road network we expect a different degree distribution. Although there are bigger cities than others they will not create a gigantic hubs, this is because if one wants to pass from a big city to another one has to cross multiple small cities and road intersections. Therefore in the end we expect a Poisson distribution or binomial, and also a bigger average path length. The code can be found here

1.2 Network properties

A graph of order N (|V|) and M (|E|) number of edges $G(V, E)$, can be represented by its $N \times N$ the adjacency matrix $A(G)$, whose elements A_{ij} in an undirected graphs are given by:

$$A_{ij} = A_{ji} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Even if is computationally expensive for large networks, from it one can extract important properties such as the mean vertex degree, the clustering coefficient, the characteristic path length.

First from the adjacency matrix we can calculate the vertex degree:

$$k_i = \sum_j A_{ij} \tag{1}$$

and consequently the average degree:

$$\langle k \rangle = \frac{1}{N} \sum_i k_i \tag{2}$$

Another important quantity is the clustering coefficient. We can calculate this coefficient for a particular vertex i from the Adjacency matrix. In particular we can calculate it by taking the third power of A. In fact, the element ii of A^3 indicates the number of different paths of length 3 joining vertex i with itself[1]. Note that a specific edge may belong to a path more than once so we have to take into account this with a prefactor[2]:

$$C_i = \begin{cases} \frac{1}{k_i(k_i-1)} \sum_{j \neq k} A_{ij}A_{jk}A_{ki}, & \text{if } k_i \geq 2 \\ 0, & \text{if } k_i = 0, 1 \end{cases} \tag{3}$$

This expression can be thought also as the number of edges between neighbors of i, i.e "how many neighbors of i are connected between each other". Since there are no self loops the clustering coefficient can be re-written:

$$C_i = \frac{1}{k_i(k_i-1)} A^3_{ii}$$

Then, one can get a macroscopic property of the network the average cluster coefficient is:

$$C = \frac{1}{N} \sum_i C_i \tag{4}$$

We calculated the average cluster coefficient in different ways (sparse multiplication[3] and using external libraries as networkx.py) and compared with the tabulated value in the SNAP library for each considered graph (2).

	C_{Snap}	$C_{networkx}$	C_{adj}
Road Network	0.047004	0.047005	0.047004
Collab Network	0.6324	0.6324	0.6324

Table 1: Average cluster coefficient

1.3 Degree distribution

After extracting the degree for each vertex we computed and studied the degree distribution.

1.3.1 Road Network

In the code we computed the degree distribution by taking all the vertexes with a certain degree in the graph and normalize them by the total number of vertexes. The degree distribution is shown in figure 1. In the same figure 1 is shown the fitted Poisson distribution. For the sake of clarity, in figure 2 we also plotted the degree distribution in a log-log scale. As a first impression we observed that the trend is very similar to a Poisson distribution, nevertheless one need to compute a statistic test. From figure 2 is clear that a power law distribution is not the best choice to fit the degree distribution of the road network, since in the log scale it results as a straight line. Therefore, we assume that the degree distribution is a Poisson distribution, namely :

$$P(k|\lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \tag{5}$$

[1] This can be generalized to the value of the element ij of the n^{th} power of the adjacency matrix is the number of different paths of length n joining vertex i to vertex j.

[2] Note that in the literature there are different definition for the cluster coefficient. In particular two of them are relevant and more accepted than others; one is introduced by Watts and Strotgaz and the other by Newmann

[3] Even if we use sparse matrix multiplication we found this method very time consuming compared to the preexisting libraries

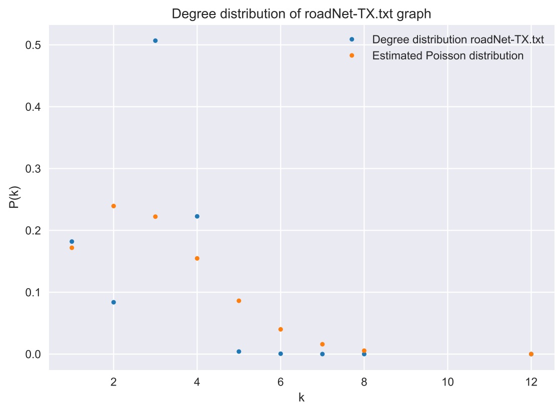


Figure 1: Estimated degree distribution. The x-axis shows the degree. The y-axis shows the percentage of vertices for a given degree. The blue dots show the degree distribution. The orange dots show the fitted Poisson distribution

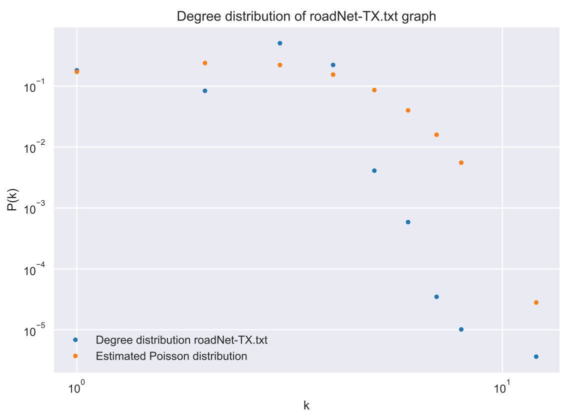


Figure 2: Estimated degree distribution on a log-log scale

This is a reasonable hypothesis since this distribution is encountered in graphs like ER graph. For a random graph Poisson distributed λ is the mean degree, however here we would like to estimate λ with the method of Maximum Likelihood. At first glance one can proceed with the method of the Maximum likelihood to estimate the parameters of the assumed probability distribution, given some observed data. Note that this method assumes that the data are independent and identically distributed. The likelihood function is defined from the experimental data:

$$\mathcal{L}_N(\lambda) = \prod_{i=1}^{N} p(k_i|\lambda) = \prod_{i=1}^{N} \frac{\lambda^{k_i}}{k_i!} e^{-\lambda}$$

The maximum likelihood estimation aims to find the values of the model parameters that maximize the likelihood function over the parameter space:

$$\hat{\Lambda} = \underset{\lambda\in\Lambda}{\arg\max} \mathcal{L}_n(\theta; \mathbf{k}).$$

To simplify the calculations we can take the logarithm of the likelihood and since is a monotonic function doesn't change the position of the maximum. Therefore, we can impose the derivative of $\mathcal{L}_N(\lambda)$ respect λ to find the best parameter:

$$0 = \frac{d}{d\lambda} \ln\left(\mathcal{L}_N(\lambda)\right) = \sum_{i=1}^{N} \frac{d}{d\lambda} \ln\left(\frac{\lambda^{k_i}}{k_i!} e^{-\lambda}\right)$$

$$= \sum_{i=1}^{N} \frac{d}{d\lambda} \left(k_i \ln(\lambda) - \lambda\right)$$

$$= \sum_{i=1}^{N} \left(\frac{k_i}{\lambda} - 1\right)$$

this yields to:

$$\lambda_{ML} = \frac{1}{N} \sum_{i=1}^{N} k_i \tag{6}$$

which is the average degree of the network, see equation (2). At this point we can compute the goodness of fit, or in other words determine how likely our data are distributed according to the model distribution. In statistic there are some tests that can be executed to compute the similarity between two distributions. Here, we performed different statistical tests: the Kullback–Leibler divergence (KL), the Jensen Shannon Divergence (JS), the Kolmogorov-Smirnov test (KS)[4] and the χ^2 test. In particular, when we perform a fit we also aim to minimize the discrepancy between model and data, in fact usually the error can be minimized by knowing the current information loss by measuring the divergence between the two distribution, which will indicate how much information is lost when approximating a distribution for the other. In particular, we

Road Network	KL divergence	JS test	KS test	χ^2
MLE	0.566	0.094	0.207	7.17

Table 2: The table is showing the statistical test results

see that it doesn't fit perfectly this is also because the road network is not completely random since some roads are constructed preferentially to connect a determinated node. In fact, when we are approximating the Road network to be poisson distributed we are losing this information. One way to improve the goodness of fit might be considering a weighted graphs (for example by looking at the traffic in determinated area of Texas or the average emission on the roads) in order to show the importance of every edge and vertex allowing to add a bias on the preferential roads.

At the end, we computed both the average degree and the average degree of the neighbors, respectively: $\langle k \rangle = 2.785$ and $\langle k_{nn} \rangle = 3.129$ experiencing the friendship paradox

1.3.2 Collab network

Analogously, here we calculated and plot the degree distribution. The degree distribution is shown in figure 3. In the same figure 3 is shown the fitted scale free distribution, in this case we used:

$$P_{fit}(k) = c(\alpha, k_{min}) k^{-\alpha},\ k \geq k_{min} \tag{7}$$

c is a normalization constant which depends on α and k_{min}, in this case $k_{min} = 1$ since we exclude isolated vertexes. In order to be a distribution equation 7 has to satisfy:

$$1 = \int_{k_{min}}^{\infty} c(\alpha, k_{min}) k^{-\alpha} dk \tag{8}$$

[4]More about those models in the Appendix

from 8 we get : $c(\alpha, k_{min}) = (\alpha-1)k_{min}^{\alpha-1}$ and in order to converge $\alpha > 1$. This yiels to:

$$P_{fit}(k) = \frac{\alpha-1}{k_{min}}\left(\frac{k}{k_{min}}\right)^{-\alpha} dk \tag{9}$$

also known in the literature as Pareto law, which is a pure power distribution law. For the sake of clarity, in

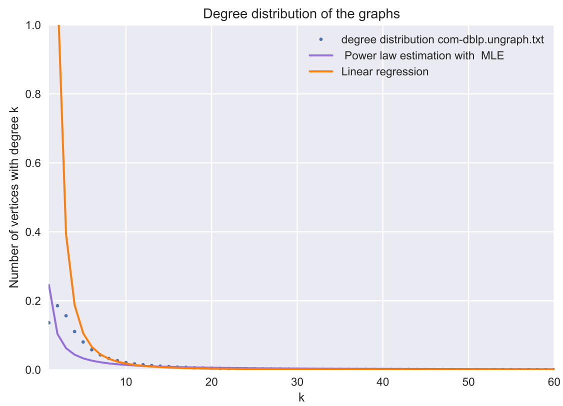


Figure 3: Caption

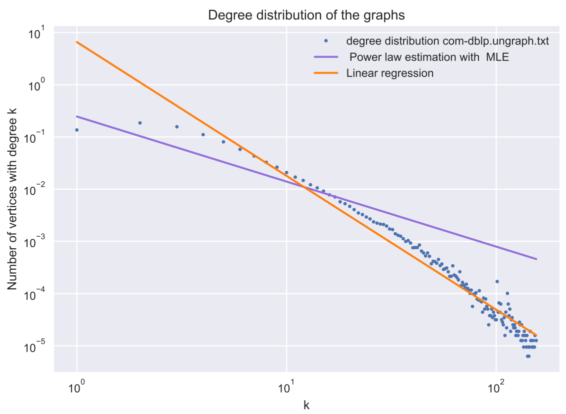


Figure 4: Caption

figure 4 we also plotted the degree distribution in a log-log scale. In this way the fitted function is shown as a nice straight line. Also in this case, at first impression we see that nodes with high degree have a similar trend to the pure scale free (tail of the distribution), on the other hand vertexes with low degree are far from the model, this is because we consider a pure scale free distribution, while a more general definition of "scale free" allows to deviate from a pure power law arbitrarily but without affecting the power-law tail exponent [1]. In first analysis we can proceed with the method of the Maximum likelihood to find the slope and the intercept, this method assumes that the data are independent and identically distributed. The likelihood function is defined then from the experimental data:

$$\mathcal{L}_N(\alpha) = \prod_{i=1}^{N} p(k_i|\alpha) = \prod_{i=1}^{N} (\alpha-1)k^{-\alpha}$$

By using the same procedure as in section 1.3.1:

$$0 = \frac{d}{d\alpha}\ln\left(\mathcal{L}_N(\alpha)\right) = \sum_{i=1}^{N} \frac{d}{d\alpha}\ln\left((\alpha-1)k^{-\alpha}\right)$$

$$= \sum_{i=1}^{N} \frac{d}{d\alpha}(-\alpha\ln(k_i) + \ln(\alpha-1))$$

$$= \sum_{i=1}^{N}\left(-\ln(k_i) + \frac{1}{\alpha-1}\right)$$

this yields to:

$$\alpha_{ML} = 1 + N\left[\sum_{i}^{N}\ln(k_i)\right]^{-1} \tag{10}$$

However, as we can see from figure 4 this is not particularly a good fit, but this does not imply that the distribution is not scale free as intended in [1]. The fit fail mainly because the assumption to perform the maximum likelihood method are no longer valid (k_i are i.i.d. variables).
In this network this might be the case. This could be traced back to the construction of this real network. An heuristic analysis suggest that when high order degree nodes are present by adding a new node this is more likely to be connected to large degree nodes already present in the network, so overall there is a bias in the construction of the network.
In other words, for example in social media, people are more likely to join or connect with people with a big amount of friends, this is also the friendship paradox. In our case, new products are more likely to be suggested with products that have already been suggested with other products.

Moreover the estimated is $\alpha_{ML} = 1.24$, this coefficient rises some problems in the power law distribution 7. In fact, the second order moment diverges. This is also because the maximum likelihood is highly biased from the sampled data and they are non i.i.d. distributed degree in a graph.
Therefore, in order to estimate the right α one have to proceed in different ways. First we can perform a more heuristic fit and then use more complicate objects such as the Hill's estimators.
Assuming the validity of the distribution as the one in equation (7), we can perform a linear fit on the log-log data coordinates.
The statistical tests results are shown in table 3 However, this does not reproduce the small nodes behaviour and the degree distribution at the tail.

This is mainly because the initial distribution is not a pure power law distribution. It presents the characteristic of a power law distribution for high degree vertices but for small finite vertices the distribution can deviate. According to [1] as far as power-law network models are concerned, even the most basic such model, preferential attachment, is known to have a degree distribution with a power-law tail, but the exact expression for the degree distribution in preferential attachment networks is not a pure power law. Thus, to best reproduce our data we might change the starting point and include the class of regularly varying distributions with power-law tail as good representation of scale-free networks:

$$P(k) = \ell(k)k^{-\alpha} \tag{11}$$

In order to estimate the parameter α from the data one can implement the Hill's estimator or the Pickand's tail-index estimator [1]as well[5].
At this point we can compute the goodness of fit

[5] We tried to implemented it but something in the code unfortunately doesn't work

throught the statistical tests as in section 1.3.1.

Collab Network	KL divergence	JS test	KS test	χ^2
MLE	2	0.06	0.277	54
linear fit	27.9	2.21	7.7	53

Table 3: The table is showing the statistical test results

At the end, we computed both the average degree and the average degree of the neighbors, respectively: $\langle k \rangle = 6.62$ and $\langle k_{nn} \rangle = 18.52$ experiencing the friendship paradox.

1.4 Friendship paradox

The friendship paradox states, "on average, the number of friends of a random friend is always greater than the number of friends of a random individual". Formally:

Theorem 1. *Consider an undirected graph $G = (V, E)$. Let v_0 be a node sampled uniformly from V, and v_1 be a node sampled uniformly from the neighbours of a randomly sampled node v. Then,*

$$\mathbb{E}[k_{v_1}] \geq \mathbb{E}[k_{v_0}],$$

where, k_{v_0} and k_{v_1} denote the degrees of v_0 and v_1, respectively.

Proof. Let V be the set of vertices and E the set of edges. Since we sample v randomly from V for the calculation of $\mathbb{E}[k_{v_1}]$, it follows that

$$\mathbb{E}[k_{v_1}] = \frac{1}{|V|} \cdot \sum_{v \in V} \mathbb{E}[k_{v_1} \mid v], \tag{12}$$

where $E[k_{v_1} \mid v]$ means the average number of neighbours of v; so v_1 is sampled among the neighbours of v.

Now this is again equal to

$$\frac{1}{|V|} \sum_{v \in V} \frac{1}{k_v} \sum_{\langle v_1, v \rangle} k_{v_1}$$

Here $\langle v_1, v_0 \rangle$ means that we sum over all neighbours v_1 of v_0.

Note that (v, v_1) is always an edge of the graph. Therefore, we can also view this as a summation over edges in e in E; but we must take into account that an edge (v, v_1) can also occur as (v_1, v). Therefore, the summation equals

$$\frac{1}{|V|} \sum_{e=(v,v_1)} \frac{k_{v_1}}{k_v} + \frac{k_v}{k_{v_1}}.$$

Using the inequality $\frac{1}{\alpha} + \alpha \geq 2$ for all $\alpha > 0$, applied to $\alpha := \frac{k_{v_1}}{k_v}$, we see that this is at least

$$\frac{1}{|V|} \sum_{e \in E} 2 = \frac{2|E|}{|V|}.$$

Now this is equal to

$$\mathbb{E}[k_{v_0}].$$

We conclude that

$$E[k_{v_1}] \geq E[k_{v_0}].$$

□

2 Watts-Strogatz model

2.1 Small world Network

Also, we can immediately see that $p(k) = 0$ for $k = 0, 1, 2, 3, 4$ for the following reason: every vertex is in the beginning the endpoint of 5 edges that start clockwise at that edge. Those may be rewired, but then they still have that vertex as endpoint. Hence, after the rewiring the vertex still has at least 5 neighbors.

2.2 Degree distribution

We have

$$\mathbb{P}_p(K = m) = \sum_{n=0}^{\min(m-r,r)} P(A_i = n) \cdot P(B_i = m - r - n).$$

Therefore, we shall calculate the probabilities for A_i and B_i.

The probability $P(A_i = n)$ is equal to the probability that exactly n of the edges starting at the vertex and going counterclockwise are left unchanged in the rewiring process. This probability is $\binom{r}{n} p^{r-n} (1-p)^n$.

The probability $P(B_i = m - r - n)$ is equal to the probability that exactly $m - r - n$ of the $(N-2)r$ edges which originally do not have the vertex as an endpoint, are rewired to have that vertex as an endpoint. For a specific edge, this probability is $p \cdot \frac{1}{N-2} = \frac{p}{N-2}$. Therefore, the total probability is

$\binom{(N-2)r}{m-r-n} \cdot (\frac{p}{N-2})^{m-r-n} \cdot (1 - \frac{p}{N-2})^{(N-2)r-(m-r-n)}$.

Now by the Poisson approximation of the binomial distribution, this is approximately equal to

$$e^{-E} \frac{E^{m-r-n}}{(m-r-n)!}.$$

Here E is the expected value of B_i. This is equal to the number of potential edges that can be rewired times the probability per edge, so $(N-2)r \cdot \frac{p}{N-2} = pr$. Therefore, we see that

$$P(B_i = m - r - n) \approx e^{-pr} \frac{(pr)^{m-r-n}}{(m-r-n)!}.$$

We conclude that

$$\mathbb{P}_p(K = m) \approx \sum_{n=0}^{\min(m-r,r)} \binom{r}{n} p^{r-n} (1-p)^n \cdot e^{-pr} \frac{(pr)^{m-r-n}}{(m-r-n)!}.$$

Now we consider the model where the rewiring happens with certainty, i.e.

$$\lim_{p \to 1} \mathbb{P}_p = \lim_{p \to 1} \sum_{n=0}^{\min(m-r,r)} \binom{r}{n} p^{r-n} (1-p)^n \cdot e^{-pr} \frac{(pr)^{m-r-n}}{(m-r-n)!}$$

so the only term of the sum that survives is the one with $n = 0$, because of the term $1 - p$, hence we have

$$\begin{aligned} \lim_{p \to 1} \mathbb{P}_p &= \lim_{p \to 1} \binom{r}{0} p^r e^{-pr} \frac{(pr)^{m-r}}{(m-r)!} \\ &= \frac{r^{m-r}}{(m-r)!} e^{-r}. \end{aligned}$$

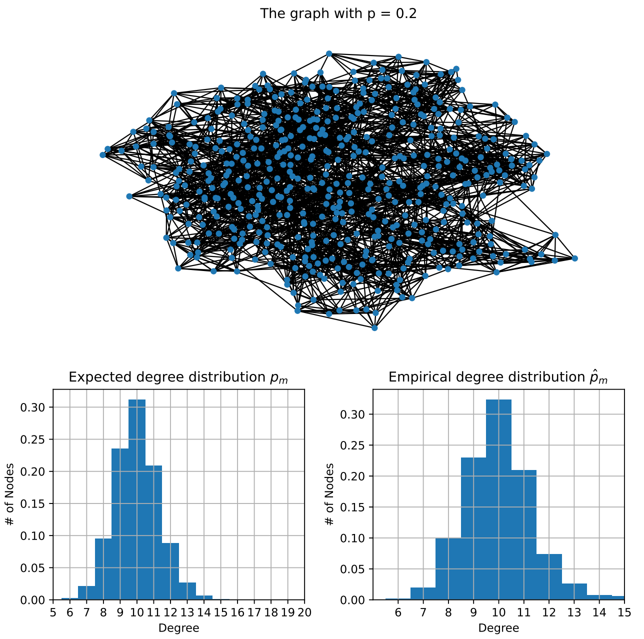


Figure 5: Watts Strogatz network degree distribution with probability $p = 0.2$ of rewiring

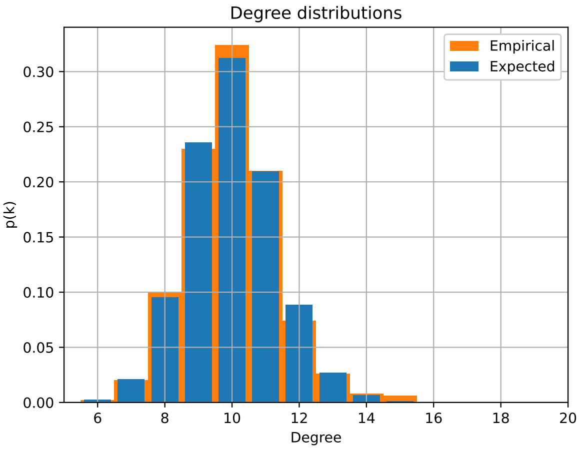


Figure 6: Watts Strogatz network empirical and theoretical degree distribution with probability $p = 0.2$ of rewiring

We shown that in the limit $p = 1$ the degree distribution formula for Watts-Strogatz is

$$\frac{r^{m-r}}{(m-r)!} \cdot e^{-r}.$$

The degree distribution formula for Erdos-Renii with parameters n, p is

$$\frac{(np)^m e^{-np}}{m!}.$$

When we take $np = r$, this becomes

$$\frac{r^m e^{-r}}{m!}.$$

This is exactly the same as for Watts-Strogatz, except that $m-r$ is replaced by m. This means that the distribution is translated to the left. In practice, this means that $0, 1, 2, \ldots, r-1$ neighbours is possible in Erdos-Renii.

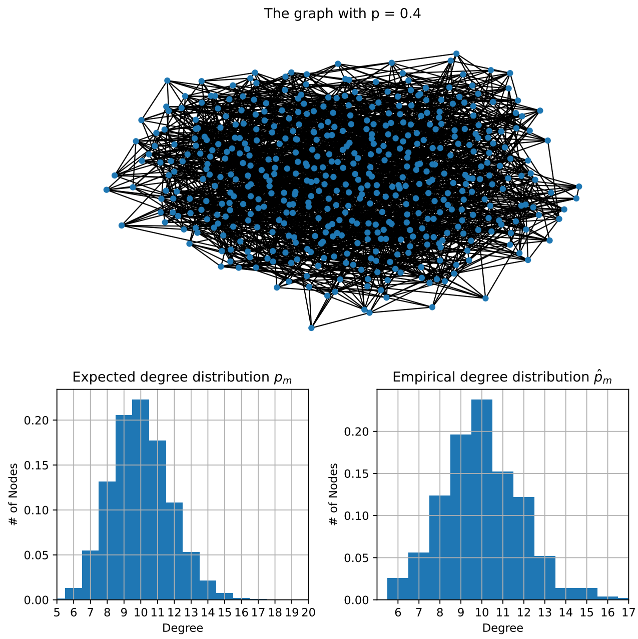


Figure 7: Watts Strogatz network degree distribution with probability $p = 0.4$ of rewiring

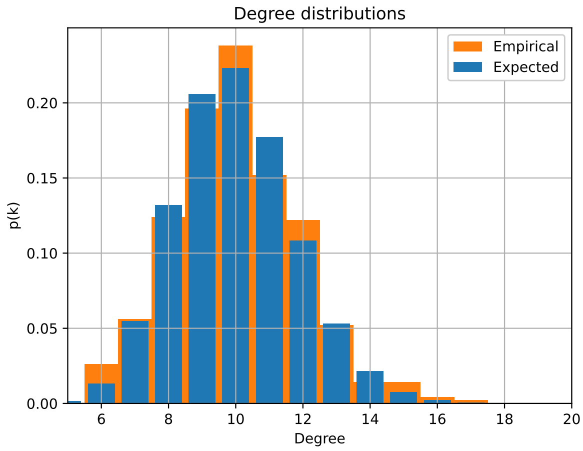


Figure 8: Watts Strogatz network empirical and theoretical degree distribution with probability $p = 0.2$ of rewiring

2.3 Clustering coefficient C_0

For a graph $WS(N, 2r, 0)$ (i.e. the probability of rewiring p is zero), prove that the clustering coefficient C_0 is:

$$C_0 = \frac{3r-3}{4r-2}.$$

We know from lecture 5 that

$$C_0 = \frac{\{\text{\# triangles with vertex in } 0\}}{k_0(k_0-1)/2}.$$

Each vertex had $2r$ neighbours so $k_0 = 2r$. The node 0 has two first neighbours (which are closed to 0), two second neighbours, and so on and two r^{th} neighbours as indicted in the following drawing 2.3.

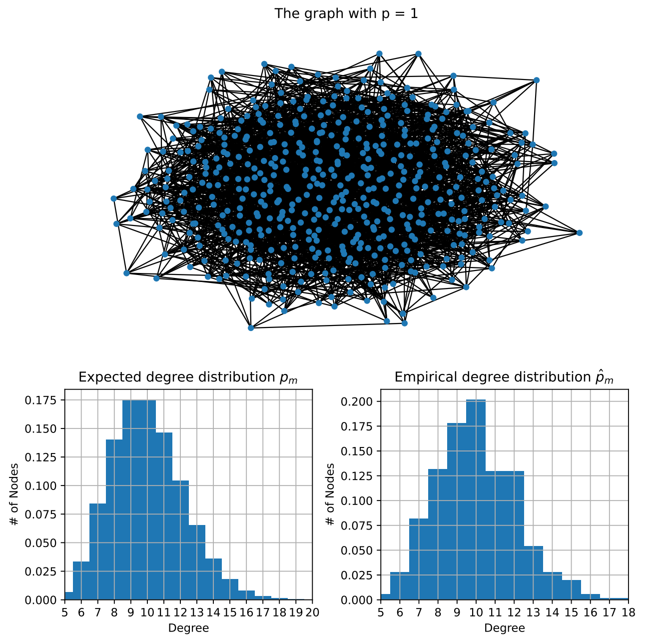


Figure 9: Watts Strogatz network degree distribution with probability $p = 1$ of rewiring

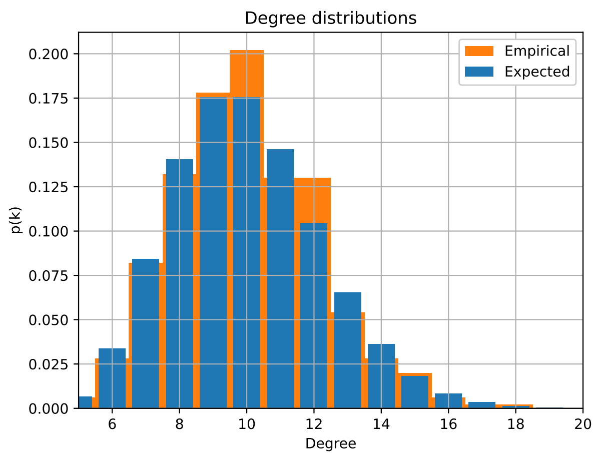
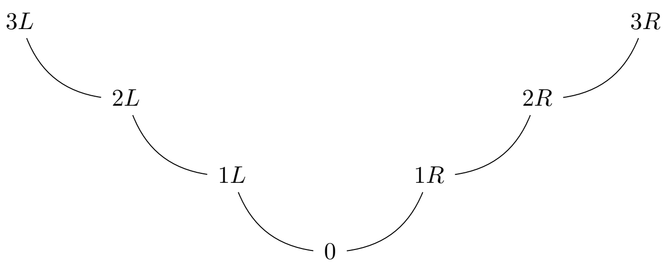


Figure 10: Watts Strogatz network empirical and theoretical degree distribution with probability $p = 1$ of rewiring



Note that we did not put all the connections between the vertices. We note that if a neighbour of 0 is connected to a neighbour of 0 then we have a triangle with vertex in 0. We know that the first neighbours of 0 have $2r-2$ edges which connect to neighbours of 0. This is due to the fact each vertex has $2r$ neighbours and that we don't count the connection of the first neighbour with 0 and the first neighbour is connected to one vertex which is not connected to 0. The second neighbours of 0 have $2r-3$ edges which connect to neighbours of 0. This is due to the fact each vertex has $2r$ neighbours and that we don't count the connection of the second neighbour with 0 and the second neighbour is connected to two vertex which are not connected to 0. The third neighbours of 0 have $2r-4$ edges which connect to neighbours of 0 and so on. Lastly, the r^{th} neighbours of 0 have $2r-r-1$ edges which connect to neighbours of of 0. This is due to the fact each vertex has $2r$ neighbours and that we don't count the connection of the r^{th} neighbour with 0 and the r^{th} neighbour is connected to r vertices which are not connected to 0. Therefore,

$$\begin{aligned}\{\#\text{triangles with vertex in } 0\} &= \frac{2}{2}\sum_{i=1}^{r}(2r-i-1)\\ &= \frac{3}{2}(r-1)r.\end{aligned}$$

The times two is due to the fact that 0 has two i^{th} neighbours for each i with $1 \leq i \leq r$. The divided by two is because we count each link twice. So we conclude

$$C_0 = \frac{\frac{3}{2}(r-1)r}{2r(2r-1)/2} = \frac{3r-3}{4r-2)}.$$

2.4 Clustering coefficient C_p

To compute C_0 in question ii) we used the amount of triangles with vertex in 0 divided by $k_0(k_0-1)/2$. This was in case of no rewiring so $p = 0$. For $p \neq 0$ the probability that all the edges of the triangles we are not rewired is $(1-p)^3$. This is because for each edge the probability of rewiring is p. Therefore, the probability of not rewiring for one edge is $(1-p)$. Therefore a fraction of $(1-p)^3$ of the triangles which we consider remain triangles that contain vertex 0. So $C_p = C_0(1-p)^3$

2.5 Mean shortest path distance

The *mean field approximation* of the Watts-Strogatz model simplifies the *mean shortest path distance* l of a graph, which can be calculated by:

$$l(n,p) \approx \frac{N}{r} f(Nrp), \quad f(x) = \frac{1}{2\sqrt{x^2+2x}}\tanh^{-1}\sqrt{\frac{x}{x+2}}$$

For N sufficiently large the WS graph has *small world properties* i.e. $l \sim \log N$. We start the approximation by noticing that for $x \gg 1$

$$\sqrt{\frac{x}{x+2}} = \left(1 - \frac{2}{x+2}\right)^{\frac{1}{2}} \approx 1 - \frac{1}{x+2} = \frac{x+1}{x+2}.$$

Using the binomial approximation. So with the help of $\tanh^{-1} x = \frac{1}{2}\log\frac{1+x}{1-x}$ we have

$$\tanh^{-1}\sqrt{\frac{x}{x+2}} \approx \tanh^{-1}\frac{x+1}{x+2} = \frac{1}{2}\log\frac{1+\frac{x+1}{x+2}}{1-\frac{x+1}{x+2}} =$$
$$\frac{1}{2}\log(2x+3) = \frac{1}{2}\log\left[2x\left(1+\frac{3}{2x}\right)\right] \approx \frac{1}{2}\log 2x.$$

Also

$$\frac{1}{\sqrt{x^2+2x}} = \frac{1}{x}\frac{1}{\sqrt{1+\frac{2}{x}}} = \frac{1}{x}\left(1+\frac{2}{x}\right)^{-\frac{1}{2}} \approx \frac{1}{x}\left(1-\frac{1}{x}\right) \approx \frac{1}{x}.$$

Hence, in total, the mean shortest distance formula is reduced to

$$l \approx \frac{N}{r}\frac{1}{2}\frac{1}{Nrp}\frac{1}{2}\log(2Nrp) \approx \frac{1}{4r^2p}\log(N). \tag{13}$$

Therefore, to have a logarithmic behavior $p \sim \frac{1}{4r^2}$ or growing $p \sim \frac{1}{N}$.

References

[1] Ivan Voitalov et al. "Scale-free networks well done". In: *Physical Review Research* 1.3 (Oct. 2019). DOI: 10 . 1103 / physrevresearch . 1 . 033034. URL: https : / / doi . org / 10 . 1103 % 5C % 2Fphysrevresearch.1.033034.

[2] Nino Boccara. "Modeling Complex Systems". In: *Book* (Aug. 2004), pp. 1–410. DOI: 10.1007/978-1-4419-6562-2.

[3] Buddhika Nettasinghe and Vikram Krishnamurthy. "Maximum likelihood estimation of power-law degree distributions using friendship paradox based sampling". In: *CoRR* abs/1908.00310 (2019). arXiv: 1908.00310. URL: http://arxiv.org/abs/1908.00310.