Network Project
A Growing Network Model

CID: 01516164
March 2021
Word Count: 2134

Abstract

Network science can be used to study a range of subjects. In this project, we focus on a growing network model based on the citation network introduced by Barab´asi and Albert [1]. The network was implemented using an algorithm in python and then our results were compared to theoretical predictions. Three cases are looked at; preferential attachment, random attachment and the mixed case. This project successfully implemented and predicted preferential attachment finding a gradient of 0.54 for the largest degree plotted against N. The theoretical prediction was 0.5. Moreover, we found similar success with the random attachment model data aligning to our predicted results with small deviations. The mixed case was implemented but requires more theoretical analysis.

1 Introduction

Networks science is a rapidly growing area of academic interest. The study of how things connect ranges from biology to the social sciences. Barab´asi and Albert developed their model of a growing network in 1999, and it was an undirected version of Price's citation network. How academics get their papers cited appears to follow a 'rich get richer' phenomena, whereby those with the most citations will receive more citations. This project aimed to study the properties of this network and examine the fat-tailed distribution. We wish to compare the simulated results to theoretical prediction.

1.1 Definition

The Barabasi and Albert model is an example of a growing network. It is used to describe the simplified citation process (the model does not consider directed nodes). The way this model was defined in this project was as a series of steps:

1. Create an initial graph G_0, at time t_0.
2. Increase the time by one.
3. Add one new node.
4. Add m new edges by; (a) connecting one end of the new edge to the new node, and (b) connecting the other end of the new edge to an existing node with a probability Π.
5. Repeat steps 2-5 until a final value of N nodes has been reached [2].

2 Phase 1: Pure Preferential Attachment Π_{pa}

2.1 Implementation

2.1.1 Numerical Implementation

In this phase, there is a probability $\Pi_{pa}(k) \propto k$ of attaching an edge between the new node and an existing node. To implement this, the graph was initially set up with $2m + 1$ nodes at a time $t = 0$. Then, a new node was added to the system. The random.randit function was used to choose an existing edge at random, then one of the two nodes connected to the ends of this edge was chosen at random. An edge was created linking this chosen existing node and the new node we created. This was done for each number of new nodes we wanted to add, m.

This process was then repeated by adding nodes up to a final number of nodes, Nf. This method of choosing an edge randomly and then the node, means that nodes with more connections (degrees k) are more likely to be chose so preferential attachment was fulfilled.

2.1.2 Initial Graph

The initial graph was built of with a number of nodes equal to $2m + 1$ and each node had $2m$ connections. This ensured the condition of the number of edges, E, being equal to mN. Moreover, it means that the number of degrees of each node is always greater than the number of new edges. Without initialising the graph with $2m$ connections for each node, there is a probability that no nodes will have connections so this brought up errors when trying to iterate the process for statistically accuracy.

2.1.3 Type of Graph

The graph is a multigraph and undirected. This is because the code does not provide any direction to the edges nor does it allow for self-loops to be made. It does however, allow multiple edges to be made between the same pair of nodes as the code is based on probability. However, as N increases the chance of this forming decreases and tends to zero for large N.

2.1.4 Working Code

To check the code worked, the code was iterated for only a few time steps and the draw function was used from the networksx package to check the graph was behaving as expected. Moreover, the code was plotted against theoretical distributions to check it matched the predicted results.

2.1.5 Parameters

The parameters are N_F, the final number of nodes to be added to the system (final time-step), and m the number of new edges added at each time-step (for each new node). N was chosen to be much greater than m as mentioned above larger N leads to greater accuracy. Moreover, when $N \approx m$, finite scaling effects can be seen, so a small m is beneficial.

2.2 Preferential Attachment Degree Distribution Theory

2.2.1 Theoretical Derivation

To derive the theoretical probability distribution, $p_\infty(k)$, the master equation was used as a starting point,

$$n(k, t+1) = n(k, t) + m\Pi(k-1, t)n(k-1, t) - m\Pi(k, t)n(k, t) + \delta_{k,t} \tag{1}$$

This describes how the number of nodes n with degree k evolve at each time step, t. $\Pi(k, t)$ is the probability of of an edge forming between a pre-existing node with degree k and the new node. The $\delta_{k,t}$ term represents the addition of a new node with degree m at each time step.

The probability $\Pi(k)$ is proportional to k for the preferential attachment model. The normalisation condition is,

$$\Pi(k) = \frac{k}{\sum_{i=1}^{N} k_i} \tag{2}$$

Using the knowledge that every edge is connected to two nodes, we find $\sum_{i=1}^{N} k_i = 2E(t)$ where E(t) is the total number of edges approximately equal to mN in the long time limit. We also know that n(k,t) can be written as,

$$n(k,t) = N(t)p(k,t) \tag{3}$$

where N(t) is the number of nodes at time t and p(k, t) is the probability that a node has degree k at time t. Then we can substitute Equation (2) and (3) and $E = mN(t)$ into the master equation. This gives us Equation (1) in the form,

$$N(t+1)p(k,t+1) = N(t)p(k,t) + \frac{1}{2}(k-1)p(k-1,t) - \frac{1}{2}kN(t)p(k,t) + \delta_{k,m} \tag{4}$$

We assume that the probability degree distribution is no longer time-dependant when t tends to infinity so, $lim_{t\to\infty} p(k,t) = p_\infty(k)$. We know that $N(t=1) = N(t) + 1$ leading us to re-write Equation (4) into the form,

$$p_\infty(k) = \frac{1}{2}[(k-1)p_\infty(k-1) - kp_\infty(k)] + \delta_{k,m} \tag{5}$$

We take the case where $k > m$ first and consider the Gamma function $\Gamma(z)$ which has the property,

$$\Gamma(z+1) = z\Gamma(z) \tag{6}$$

In this case, Equation (5) can be rearranged as $\delta_{k,m} = 0$. Giving,

$$\frac{p_\infty(k)}{p_\infty(k-1)} = \frac{k-1}{2+k} \tag{7}$$

To find a solution to this, we need to prove that the equation,

$$\frac{f(z)}{f(z-1)} = \frac{z+a}{z+b} \tag{8}$$

has the solution,

$$f(z) = A\frac{\Gamma(z+1+a)}{\Gamma(z+1+b)} \tag{9}$$

This can be done using the form,

$$z + c = \frac{\Gamma(z+c+1)}{\Gamma(z+c)} \tag{10}$$

To get,

$$\frac{f(z)}{f(z-1)} = \frac{\Gamma(z+1+a)}{\Gamma(z+1+b)} \Big/ \frac{\Gamma(z+a)}{\Gamma(z+b)} \tag{11}$$

Then using the property in Equation (6) we can rearrange Equation (11) to come to the solution in Equation (9). Then we can apply this form to Equation (7), where we find a = -1 and b = 2. Using the property of the gamma function again, we get,

$$p_\infty(k) = A\frac{\Gamma(k)}{\Gamma(k+3)} = \frac{A}{k(k+1)(k+2)} \tag{12}$$

where A is a constant. We look at the case where k = m to find the constant A. Substituting Equation (12) into Equation (5) gives us,

$$A = 2m(m+1) \tag{13}$$

Leading to,

$$p_\infty(k) = \frac{2m(m+1)}{k(k+1)(k+2)} \tag{14}$$

which is true for $k \geq m$. For the case $k < m$, $p_\infty(k) = 0$. We also consider a scaling ansatz of the form,

$$p_N(t) = p_\infty F(\frac{k}{k_{max}(N)}) \tag{15}$$

for finite size systems. Here, $F(x)$ is the scaling function and $k_{max}(N)$ is the maximum value of k that we can ignore the finite size effects of the system.

2.2.2 Theoretical Checks

We can check the theoretical degree distribution in Equation (14) by first looking at the case when $k \to \infty$. Here, it is obvious that $p_\infty(k) \to 0$ which implies a node cannot have infinitely many degrees. For large values of k, the probability distribution approximates to $p_\infty(k) \approx k^{-3}$ which matches the scale-free distribution form and shows that the probability of getting larger k decreases as k increases.

Next, we check that the distribution satisfies normalisation,

$$\sum_{k=m}^{\infty} p_\infty(k) = 1 \tag{16}$$

This can be shown by separating the terms into partial fractions,

$$\sum_{k=m}^{\infty} p_\infty(k) - m(m+1)[\sum_{k=m}^{\infty}\frac{1}{k-2}\sum_{k=m}^{\infty}\frac{1}{k+1} + \sum_{k=m}^{\infty}\frac{1}{k+2}] \tag{17}$$

Then we can change the starting point of each sum so that each sum contains $\frac{1}{k}$ and then take out the extra terms so the starting point of each sum is the same. Then some rearranging gives the desired form for normalisation.

$$\sum_{k=m}^{\infty} p_\infty(k) = m(m+1)\frac{1}{m(m+1)} = 1 \tag{18}$$

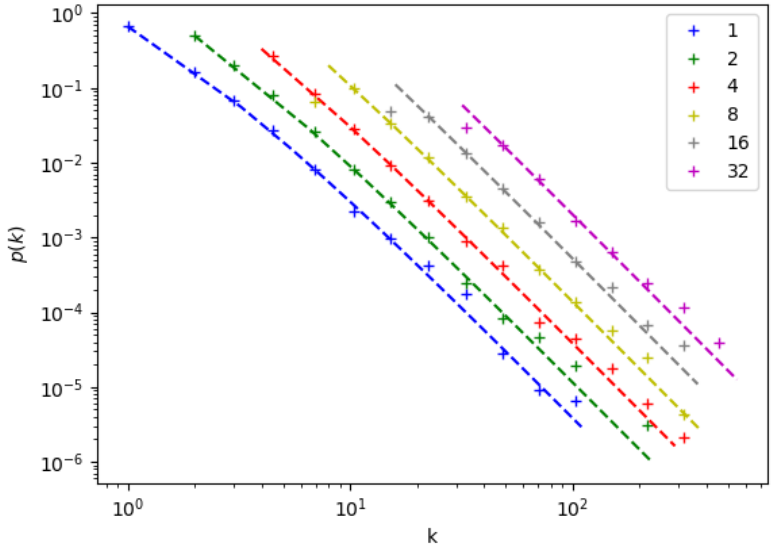

FIGURE 1: The probability of a node having a number of degrees k plotted against k for various values of m shown by the different colours and with $N = 5000$. The system was run 20 times and the results were log-binned to plot this graph. The dashed lines represent the theoretical prediction for each m.

2.3 Preferential Attachment Degree Distribution Numerics

2.3.1 Fat Tails

There are much fewer nodes possessing large values of k in a fat-tailed distribution, which means there is a large amount of statistical noise for large k. Therefore, to reduce this noise the data was log-binned with a scale of 1.45. This was chosen to try and balance the smoothing of data for high k, whilst also not losing too much data at low k.

2.3.2 Numerical Results

To obtain the numerical results, the algorithm was run 20 times with $N = 5000$ nodes added. This was done for values of $m \in [1, 2, 4, 8, 16, 32]$ to get a broad range to show the distribution whilst also achieving an acceptable computational time. The probability of a node having a degree k, $p_N(k)$ was found by log-binning the data and this was plotted against k in Figure 1.

The results followed the theoretical prediction well, but there were some deviations at large k. This implies that there is a $k_{max}(N)$ value as discussed in Equation (15), where the system is no longer scale-free so we cannot ignore the finite-size effects.

To show the results were a good fit, $p_N(k)$ was plotted against the theoretical probability distribution in Figure 2. This shows some deviation from the $y = x$ line predominantly for smaller $p_N(k)$. This also implies a cut-off value of k.

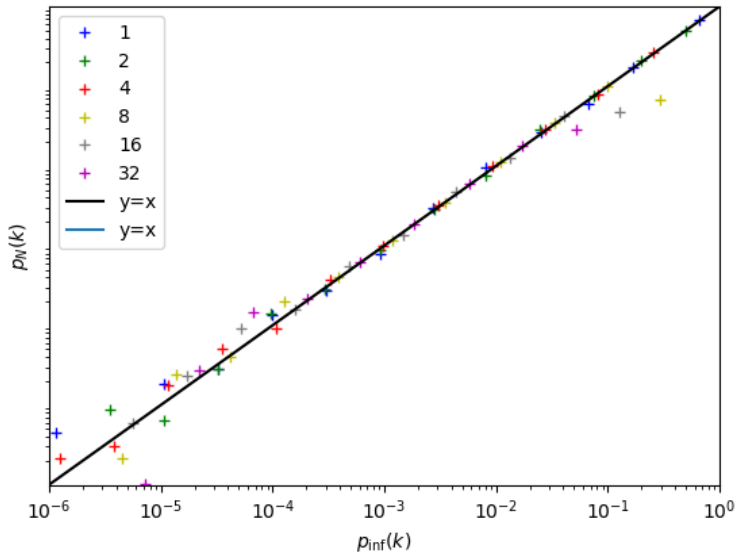

FIGURE 2: The probability of a node having a number of degrees k, $p_N(k)$ plotted against the theoretical probability distribution $p_\infty(k)$ with a line of best fit $y = x$.

2.3.3 Statistics

Both the KS test and Chi squared test were used to look at the accuracy of our algorithm, in Table 1 and 2 respectively. The Chi squared test is favourable when looking at log-binned data, but as sample sizes increase will be less likely to work. Th K-S test is known to work well over large observation, but is used for continuous data. Our data is approximately continuous so the K-S test is arguably better. The K-S data reports larger deviation in our results with the predicted results. The p column in the tables is the probability that an error is made when rejecting the null hypothesis. So a value of $p = 1$ implies that there is a perfect fit which is highly unlikely to happen with real data. This also implies the K-S test is more reliable as all values are close to 1 but none are equal to 1.

m	KS	p
1	0.06666666666666667	0.9999999999999989
2	0.07142857142857142	0.9999999999999868
4	0.07692307692307693	0.99999999999985
8	0.16666666666666666	0.9984852944874484
16	0.18181818181818182	0.9970968144342757
32	0.2	0.9944575548290717

TABLE 1: The K-S test for the difference between the theoretical prediction and the algorithmic results for preferential attachment.

m	Chi Square	p
1	5.6217064986952255e-05	1
2	0.00017878125371852032	1
4	0.000520440300779284	1
8	0.17577629846843026	0.9999999947091983
16	0.04939234369284939	0.9999999989047418
32	0.009359777831747471	0.999999999981795

TABLE 2: The Chi Squared test for the difference between the theoretical prediction and the algorithmic results for preferential attachment

2.4 Preferential Attachment Largest Degree and Data Collapse

2.4.1 Largest Degree Theorem

The largest degree k_1 is the degree that only one node is expected to have a larger degree value, i.e. only one node with $k \geq k_1$. This can be shown with the expectation value,

$$N \sum_{k=k_1}^{\infty} p_\infty = 1 \tag{19}$$

Subbing in Equation (18) gives,

$$N \frac{m(m+1)}{k_1(k_1+1)} = 1 \tag{20}$$

Solving for k_1,

$$k_1 = \frac{-1 + \sqrt{1 + 4Nm(m+1)}}{2} \tag{21}$$

For large N,

$$k_1 \approx \sqrt{mN(m+1)} \tag{22}$$

2.4.2 Numerical Results for Largest Degree

The system was run 20 times with a value of $m = 2$ for $N \in [10, 100, 1000, 10000, 100000]$, m was chosen to be much less that N. The largest degree, k_1 was plotted against N in Figure 3 and the theoretical prediction was plotted. The value of the gradient of the linear fit of our data was 0.54 which deviates slightly from the predicted value of 0.5. To increase the accuracy the program should run over more iterations, though some deviation is expected because Equation (22) is an approximation.

These results dhow that k_1 scales with $\sqrt{N}$, which can be used in the scaling function.

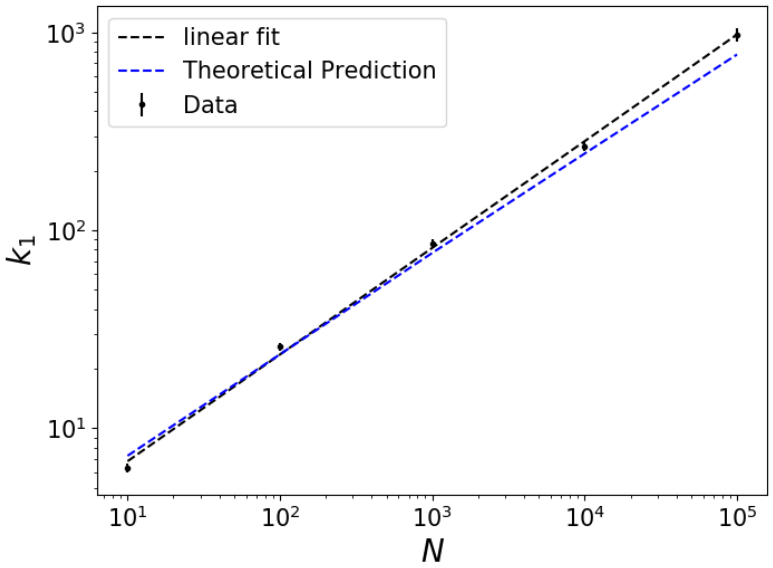

FIGURE 3: The largest degree k_1 plotted against N on a log scale shows a linear fit. The gradient of the algorithms data is compared to the theoretical prediction from Equation (22).

2.4.3 Data Collapse

The finite-size scaling ansatz was used to produce Figure 4. The algorithm was run over 50 iteration for $N \in [10, 100, 1000, 10000]$ and $m = 2$. Fewer values of N were used to ensure there could be greater accuracy in the time we had for computing the results. By looking at the scaling of k_1 and the probability distribution, we could find the scaling function $F(x)$;

$$p_N(k) = p_\infty(k) F(\frac{k}{k_1}) \tag{23}$$

Thus, the data collapse was formed by plotting $p_N(k)/p_\infty(k)$ against $k/\sqrt{N}$.

Figure 4 shows a bump when $k \approx k_1$. This is shows that the data has passed the cut-off value of $k_{max}(N)$ so no longer follows the theoretical prediction and begins to drop-off rapidly.

3 Phase 2: Pure Random Attachment Π_{rnd}

3.1 Random Attachment Theoretical Derivations

3.1.1 Degree Distribution Theory

Starting with the master equation again (Equation (1)), we use the probability, Π_{rnd} which is equal for each node.

$$\Pi_{rnd} = \frac{1}{N(t)} \tag{24}$$

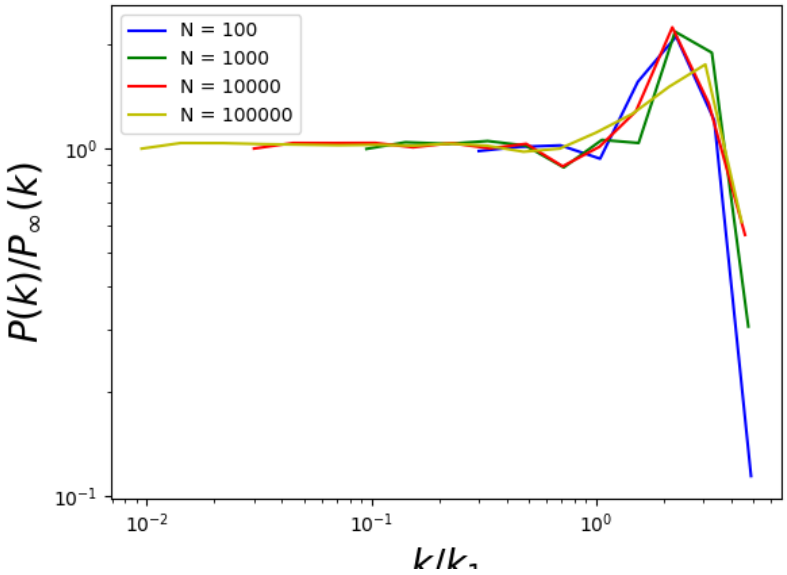

FIGURE 4: The data collapse for a preferential attachment system with $m = 2$ over 50 iterations for various values of N.

Subbing this in as done before gives the form for $t \to \infty$,

$$p_\infty(k) = mp_\infty(k-1) - mp_\infty(k) + \delta_{k,m} \tag{25}$$

For the case $k < m$, $p_\infty(k) = 0$ because each new node has to make m edges. For $k = m$,

$$p_\infty(m) = \frac{1}{m+1} \tag{26}$$

Then we can calculate when $k > m$,

$$p_\infty(m) = \frac{m}{m+1} p_\infty(m-1) \tag{27}$$

This is a geometric series so if $k \geq m$, this can be written as,

$$p_\infty(m+x) = (\frac{m}{m+1})^x p_\infty(m) \tag{28}$$

Thus by subbing $k = m+x$ we get the random attachment theoretical probability distribution,

$$p_\infty(k) = \frac{1}{m+1}(\frac{m}{m+1})^{k-m} \tag{29}$$

To check this solution has the correct properties, we first look at normalisation where we must meet the condition,

$$\sum_{k=m}^{\infty} p_\infty(k) = \frac{1}{m+1} \sum_{k=m}^{\infty} (\frac{m}{m+1})^{k-m} = 1 \tag{30}$$

We can use the standard result from the summation of a geometric series with a multiplier less than one giving,

$$\sum_{k=m}^{\infty} p_\infty(k) = \frac{1}{m+1}\frac{1}{1-\frac{m}{m+1}} = 1 \tag{31}$$

which has met the desired condition. Moreover, when k is large the probability tends to zero which is what we would expect.

3.1.2 Largest Degree Theory

Again we use $N\sum_{k=k_1}^{\infty} p_\infty(k) = 1$ for the largest degree k_1. Substituting this for $p_\infty(k)$ gives,

$$\sum_{k=k_1}^{\infty} (\frac{m}{m+1})^{k-m} = (\frac{m}{m+1})^m (\frac{m+1}{N}) \tag{32}$$

Using $k = k_i + i$,

$$\sum_{i=0}^{\infty} (\frac{m}{m+1})^{k_1+i} = (m+1)(\frac{m}{m+1})^{k_1} \tag{33}$$

Using the standard form of the geometric sum we get,

$$(\frac{m}{m+1})^{k_1} = (\frac{m}{m+1})^m \frac{1}{N} \tag{34}$$

Then taking the log of each side and rearranging gives us,

$$k_1 = \frac{\ln N}{\ln[\frac{(m+1)}{m}]} + m \tag{35}$$

This implies a pure random attachment network is not scale-free and scales when $N >> m$ with $\ln N$.

3.2 Random Attachment Numerical Results

3.2.1 Degree Distribution Numerical Results

The algorithm for random attachment meant that each node in the system had an equal probability of being connected to the new node added to the system . This algorithm was run 20 times with $N = 5000$ for values of $m \in [1, 2, 4, 8, 16, 32]$ to produce Figure 5. The theoretical prediction was also plotted and shows a good alignment with the data.

As before, $p_N(k)$ was plotted against the theoretical probability distribution for Figure 6. This, again, shows some deviation from the $y = x$ line predominantly for smaller $p_N(k)$. This also implies a cut-off value of k.

The statistics to compare the theoretical predictions to computaional results can be seen in Table 3 and 4.

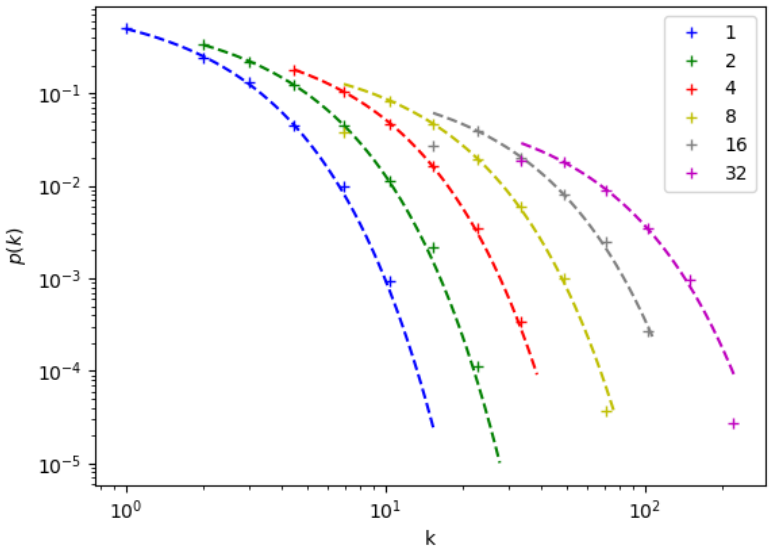

FIGURE 5: The probability of a node having a number of degrees k plotted against k for various values of m shown by the different colours and with $N = 5000$. The system was run 20 times and the results were log-binned to plot this graph. The dashed lines represent the theoretical prediction for each m.

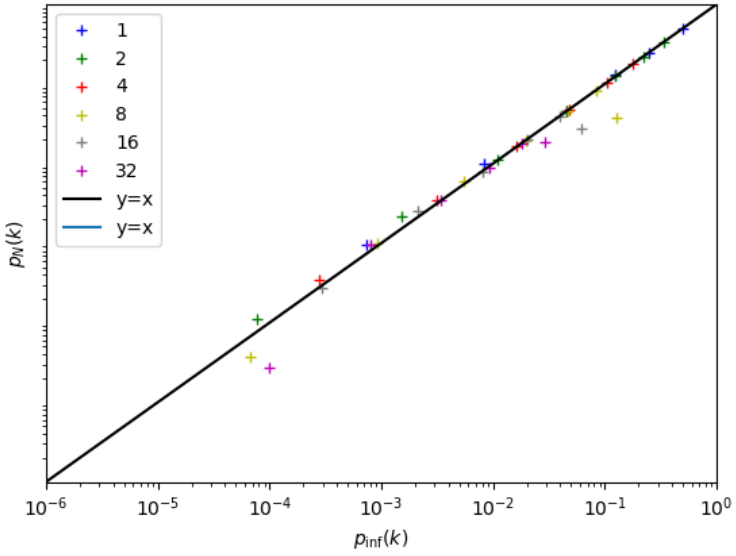

FIGURE 6: The probability of a node having a number of degrees k, $p_N(k)$ plotted against the theoretical probability distribution $p_\infty(k)$ with a line of best fit $y = x$.

m	KS	p
1	0.14285714285714285	0.9999997041284721
2	0.14285714285714285	0.9999997041284721
4	0.14285714285714285	0.9999997041284721
8	0.2857142857142857	0.9627039627039629
16	0.3333333333333333	0.9307359307359307
32	0.16666666666666666	0.9999999999999998

TABLE 3: The K-S test for the difference between the theoretical prediction and the algorithmic results for random attachment.

m	Chi Square	p
1	0.00027048026974607934	0.9999999999995878
2	0.00014453584548491724	0.999999999999937
4	5.5345182377780256e-05	0.9999999999999964
8	0.06266036155293463	0.9999949934268556
16	0.01862936896027887	0.9999974970307963
32	0.0037738497566848477	0.9999999535243995

TABLE 4: The Chi Squared test for the difference between the theoretical prediction and the algorithmic results for random attachment.

3.2.2 Largest Degree Numerical Results

The system was run 10 times with a value $m = 2$ for $N \in [10, 100, 1000, 10000, 100000]$ to produce Figure 7. The linear fit shows that k_1 scales with $\ln N$. The theoretical prediction fits within the standard deviation of the data showing that there is a good fit. The code should run over more iterations to improve the accuracy.

4 Phase 3: Mixed Preferential and Random Attachment

4.1 Mixed Attachment Model Numerical Results

This model was implemented by combing the code for the preferential attachment and random attachment models. We used a probability of q as to which system would be applied to the new added node. For a value of $q = 2/3$, preferential attachment is more likely than random attachment. Figure 8 was plotted for a size $N = 5000$ over a range of values of m for 20 iterations. Unfortunately, the results could not be compared to theoretical derivations due to a lack of time. However, the distribution trend appears to be somewhat

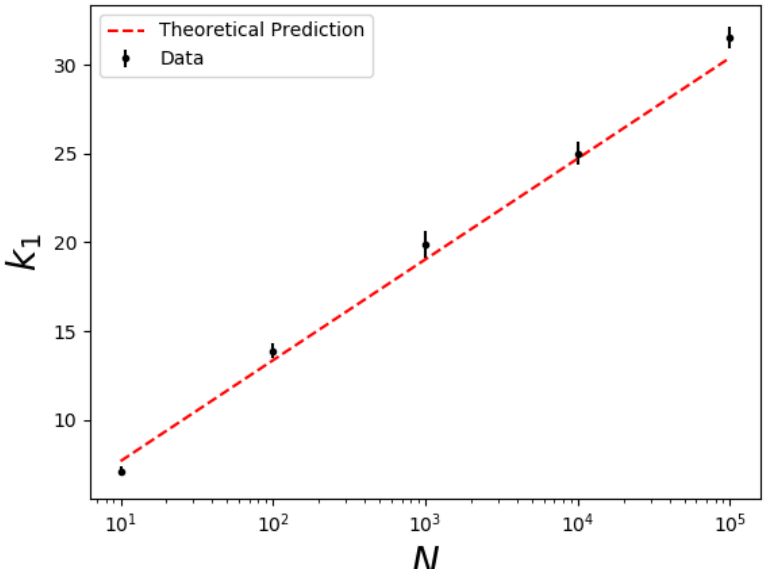

FIGURE 7: The largest degree k_1 plotted against N with a log scale on the y-axis shows a linear fit. The algorithms data is compared to the theoretical prediction from Equation (35).

in the middle of preferential and random as one would expect. Moreover, the trend is more straight that it is curved implying preferential attachment is dominant as expected. Therefore, I feel confident the model was implemented correctly. This mixed model could be the closest representation of the real-life citation network as most people would come across papers through other references but there would be a percentage of papers that are found randomly and end up being cited.

4.2 Conclusion

This investigation has successfully implemented three models of a growing network model; preferential attachment, random attachment and mixed attachment. Theoretical derivations of preferential and random attachment predictions were produced and successfully compared to the simulated results. To improve, a derivation of the mixed attachment method should be produced and compared to the results.

References

[1] A.-L. Barab´asi and R. Albert, Emergence of scaling in random networks Science, 286 173 (1999).

[2] Evans, T. 2021 Network Project Student Notes Brief for the Network Project Report.

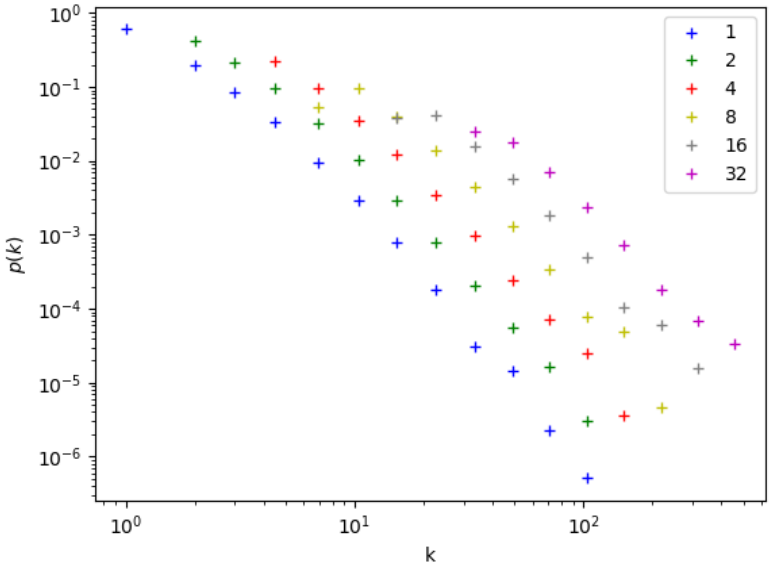

FIGURE 8: The probability of a node having a number of degrees k plotted against k for various values of m shown by the different colours and with $N = 5000$. The system was run 20 times and the results were log-binned to plot this graph.