Networks Project: The BA Model (python 3.6.4)

Abstract

Using the Barabasi and Albert model (BA model), theoretical derivations for the degree distribution were compared to numerical simulation data. Three cases of node attachment were explored. First the preferential attachment, then random attachment, and lastly an attachment which allows self-loops and a mixed preferential and random attachment. Visual plots of revealed that the theory matched the numerical data well for all three attachment regimes. Gradients of $m_0 = -2.95 \pm 0.06$ and $m_1 = 0.50 \pm 0.04$ for preferential attachment and $\ln(k_1)$ scaling respectively. Preferential attachment showed finite sized effects and a data collapse was produced which revealed a bump. Chi-squared statistical tests were used after showing that the frequency of k values over multiple runs was Gaussian distributed. P-values of one were obtained which showed that there was a good match between the numerical data and the theoretical data.

1 Introduction

The aim of this project is to investigate the behaviour and properties of growing networks. Using the Barabasi and Albert model (BA model), theoretical derivations for the degree distribution will be compared to numerical simulation data. It is important to know that the numerical results are not exact and theoretical derivations use assumptions such as infinite networks. Visual and statistical methods will be deployed to assess how well the theory matches the numerical data. Three cases of node attachment will be explored. First the preferential attachment, then random attachment, and lastly an attachment which allows self-loops and a mixed preferential and random attachment.

2 Pure Preferential Attachment

2.1 Implementation

2.1.1 Numerical Implementation

The BA model was implemented in python 3.6.4. A simple algorithm was used to produce a growing network. The BA model works by first producing a graph was and then adding nodes as well as edges according to the preferential attachment case. A unit of time was adding one node. This was repeated until the set number of total nodes N were added. More specifically, a list of degrees of the nodes was produced and updated as edges were added. Edges were added by first attaching one end to the new node, and then selecting an existing node, with a probability proportional to the nodes degree, to attach the other end of the edge. This was achieved by randomly selecting an edge from a list of existing edges and then randomly selecting a node from that pair. This works because the nodes with higher degree will appear in more edges and so randomly selecting from this edge list will ensure that it is more likely to select a node with higher degree k. To avoid self-loops, this process was redone if the resulting node was itself until another node was selected. Overflow errors were overcome by using 'float128' variables.

2.1.2 Initial Graph

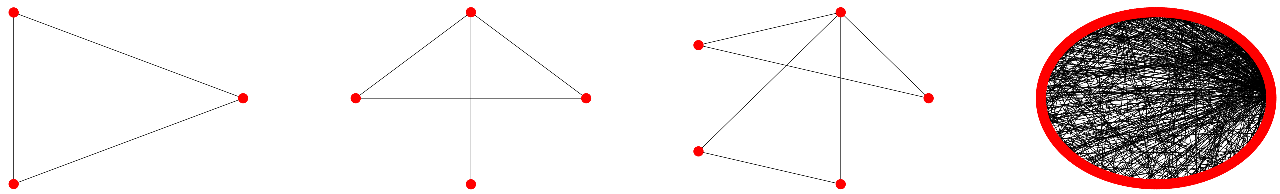
A equally weighted graph with three nodes was used initially. This was done so that there was an equal probability for each node being attached to for the first addition. A small network was selected because it allows to see network growth also relative to the end size of the network, it is a better approximation to the graph growing to infinity which is assumed in theoretical derivations.

2.1.3 Type of Graph

Networks are classified by their edge types. The graph I am creating will have no directed edges and will allow multiple edges between the same pair of nodes. There will be no self-loops for preferential attachment and the number of nodes increases with time. Therefore, the type of network I am creating is a growing weighted network.

2.1.4 Working Code

To check that my programme is working correctly, I check that new nodes and edges are being added by showing a visualisation of the network as in figure (1). Figure (1a,1b,1c) shows the first three time steps with nodes and edges correctly being added, and then figure(1d) shows the final network which was set to 1000 nodes for this test. The final network shows the preferential attachment as there is a higher relative density of edges to the nodes on the centre right. The number of nodes and the number of edges are shown to always be conserved.



(a) (b) (c) (d)

Figure 1: The evolution of the graph is shown. The initial graph (a) has three nodes which all have the same degree equal to two. Figure (b) shows the first node added which connects both its new edges to the same existing node. Figure (c) shows the next node introduced and that it connects it's edges to two different nodes. Figure (d) shows the final network, where it can be seen from the relative density of the edges that the nodes on the right have a higher degree. This shows the preferential attachment.

2.1.5 Parameters

The number of nodes to be added to the network N and the number of edges to add per node m. N was selected to be 10^5 because this gave a large enough network to observe cut-offs and so it shows a better approximation to an infinite network which is what is assumed, and m was varied in powers of two to reveal behaviour at different orders of magnitude.

2.2 Preferential Attachment Degree Distribution Theory

2.2.1 Theoretical derivation

The degree distribution at time t, $p(k,t)$, shows the probability of selecting a node with k edges. Defining the number of nodes with degree k at time t as $N(k,t)$, and the total number of nodes at time t as $N(t) = N(0) + t$, then

$$p(k,t) = \frac{N(k,t)}{N(t)}. \tag{1}$$

As $t \to \infty$, $N(t) \to t$, so that the probability of attaching an edge from a new node to an existing one in long time limit is [1],

$$\Pi(k) = \frac{k}{\sum_i k_i} = \frac{k}{2mt}, \tag{2}$$

where m is the number of edges each new node adds. Adding an edge to a node of degree k will decrease $N(k,t)$, however, adding an edge to a node of degree $k-1$ will increase $N(k,t)$. After he addition of a new node, the expected number of connections to a node with degree k is

$$\Pi(k)N(k,t)p(k,t)m = \frac{p(k,t)k}{2}. \tag{3}$$

This is simply the probability of connecting to a node with degree k multiplied by the number of nodes with degree k. Therefore, the increase in number of nodes with degree $k+1$ is

$$\frac{p(k,t)k}{2}, \tag{4}$$

and the increase in the number of node with degree k is

$$\frac{p(k-1,t)(k-1)}{2}. \tag{5}$$

Using equations (4) and (5) the expected number of nodes with degree $k > m$ after adding a new node is,

$$(N+1)p(k,t+1) = Np(k,t) + \frac{(k-1)p(k-1,t)}{2} - \frac{kp(k,t)}{2}. \tag{6}$$

To include the changes to nodes with degree m another equation needs to be formed. The increase in the number of nodes of degree $m+1$ (which is the decrease in the number of nodes with degree m) is

$$\frac{mp(m,t)}{2}. \tag{7}$$

Also, the increase in the number of nodes with degree m is one. Therefore, the expected number of nodes with degree m after adding a new node is,

$$(N+1)p(m,t+1) = Np(m,t) + 1 - \frac{mp(m,t)}{2}. \tag{8}$$

As $t \to \infty$,

$$(N+1)p(k,t+1) - Np(k,t) \to Np(k,\infty) + p(k,\infty) - Np(k,\infty) = p(k,\infty). \tag{9}$$

From this result, equation (6) becomes,

$$p(k,\infty) = \frac{k-1}{k+2}p(k-1,\infty). \tag{10}$$

For $k < m$, as $t \to \infty$,

$$(N+1)p(m,t+1) - Np(m,t) \to Np(m,\infty) + p(m,\infty) - Np(m,\infty) = p(m,\infty). \tag{11}$$

Then, equation (8) is cast into,

$$p(m,\infty) = \frac{2}{m+2}. \tag{12}$$

Rewriting equation (10) using the substitution $k' = k+1$,

$$p(k+1,\infty) = \frac{k}{k+3}p(k,\infty). \tag{13}$$

Use the recursive formulas to produce the final form of $p(k,\infty)$. The lowest degree distribution is derived using the lowest degree $k = m$ with equation (12) and then the higher degree distributions are worked out using equation (13). First few steps are,

$$\begin{aligned}
p_\infty(m+1) &= \frac{(m)}{(m+3)}p_\infty(m) = \frac{2m}{(m+2)(m+3)}, \\
p_\infty(m+2) &= \frac{(m+1)}{(m+4)}p_\infty(m+1) = \frac{2m(m+1)}{(m+2)(m+3)(m+4)}, \\
p_\infty(m+3) &= \frac{(m+2)}{(m+5)}p_\infty(m+2) = \frac{2m(m+1)}{(m+3)(m+4)(m+5)}.
\end{aligned}$$

Therefore, the terms in denominator will cancel apart from $2m(m+1)$ and setting $k = m+3$,

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

2.2.2 Theoretical Checks

The sum of $p(k \geq m,\infty)$ should be one as this is a probability. As m is a constant the terms in the numerator can be factorised outside the sum,

$$\sum_{k=m}^{\infty} p(k \geq m,\infty) = 2m(m+1)\sum_{k=m}^{\infty}\frac{1}{k(k+1)(k+2)}. \tag{15}$$

To evaluate this sum split the fraction into partial fractions,

$$\frac{1}{k(k+1)(k+2)} = \frac{1}{2k} - \frac{1}{k+1} + \frac{1}{2(k+2)}. \tag{16}$$

Writing out the first few terms of the sum term in equation (15),

$$\begin{aligned}
&\frac{1}{2m} - \frac{1}{m+1} + \frac{1}{2(m+2)} + \frac{1}{2(m+1)} - \frac{1}{m+2} + \frac{1}{2(m+3)} \\
&+\frac{1}{2(m+2)} - \frac{1}{m+3} + \frac{1}{2(m+4)} + \frac{1}{2(m+3)} - \frac{1}{m+4} + \frac{1}{2(m+5)} + \ldots
\end{aligned} \tag{17}$$

From equation (17), can see that the only remaining terms are,

$$\frac{1}{2m} - \frac{1}{m+1} + \frac{1}{2(m+1)} = \frac{1}{2m(m+1)}. \tag{18}$$

Therefore, using this result equation (15) equals one and so the probability is normalised. $p(k < m, \infty) = 0$ as we are in the stable distribution and because m edges are added every time a new node is added. Once the next nodes are inserted they then can also connect their new edge to this existing node, therefore the expected degree of a node is $\langle k \rangle = 2m$. To check that this is the case,

$$\langle k \rangle = \sum_{k=m}^{\infty} kp(k \geq m, \infty) = \sum_{k=m}^{\infty} \frac{2m(m+1)}{(k+1)(k+2)} = 2m(m+1) \sum_{k=m}^{\infty} \left(\frac{1}{(k+1)} - \frac{1}{(k+2)} \right). \tag{19}$$

Writing out the first few terms,

$$\langle k \rangle = 2m(m+1) \left(\frac{1}{(m+1)} - \frac{1}{(m+2)} + \frac{1}{(m+2)} - \frac{1}{(m+3)} + \frac{1}{(m+3)} + ... \right) \tag{20}$$

The only surviving term is $\frac{1}{m+1}$, and so, $\langle k \rangle = 2m$. This matches the expected value and so the theoretical expression is correct.

2.3 Preferential Attachment Degree Distribution Numerics

2.3.1 Fat-Tail

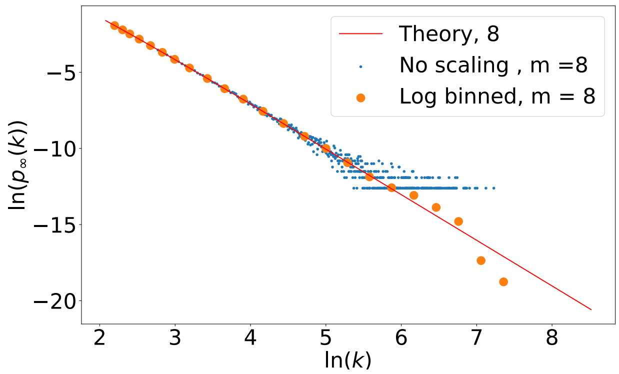
The problem the fat-tail causes is that the pattern can not be inferred in the region after it due to statistical noise. To extract information from the fat-tail I used the log-binning technique. The log-binning produces evenly spread statistics by ensuring that the bin size increases exponentially as the bin number increases. The scaling used for each bin size was 1.35 and so starting from a bin size of one each bin was 1.35 times larger than the previous one. This was done by re-scaling k by the minimum value of k before log-binning. Although log-binning cause some information to be lost, it reduces the level of noise and extracts enough information from the fat-tail to show the finite-cut off effect as in figure (2(a)).

2.3.2 Numerical Results

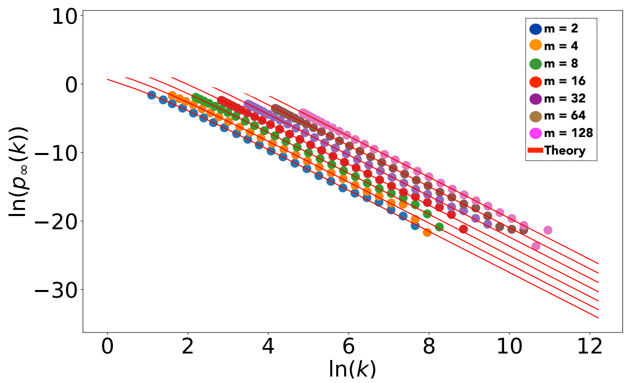
The log-binned points before the cut-off were linearly fitted and the gradients were averaged over m values of 8,16,32,64. The resulting average gradient, m_0, was,

$$m_0 = -2.95 \pm 0.06. \tag{21}$$

This is close to the expected value of $m_o \sim -3$ from the theoretical derivation, and figure (2(b)) shows that the numerical data points lie on the same line as the theoretical data so this is a good fit.



(a) Logarithmic plot of $\ln(p_\infty)$ against $\ln(k)$ is shown for a network size $N = 10^4$ and $m = 8$. This shows the fat-tail from which the underlying pattern is extracted from. The log-binned data shows a good fit visually from initial k to about the region of the fat-tail. After the fat-tail, there is a small bump and then a drop in the $\ln(p_\infty(k))$ value. Theoretically, $\ln(p_\infty(k))$ should continue along the red line but it does not. This is due to the finite nature of this network.



(b) Logarithmic plot of p_∞ against $\ln(k)$ is shown for multiple networks each of size $N = 10^5$ but varying m. The log-binned data shows a good fit visually from initial k to about the region of the cut-off. After the fat-tail, there is a small bump and then a drop in the $\ln(p_\infty(k))$ value. Theoretically, $\ln(p_\infty(k))$ should continue along the red line but it does not. This is due to the finite nature of this network.

Figure 2: Logarithmic plots are shown.

2.3.3 Statistics

To check how well the numerical data matches with the theoretical data, a chi-squared test will be used. The chi-squared test for N data points is given by:

$$\chi^2 = \sum_{i=1}^{N} \frac{(x_i - \mu_i)^2}{\sigma_i^2}, \tag{22}$$

where x_i is the numerical data point, μ_i is the theoretical data point and σ_i is the error on x_i. Here the Poisson error can be used $\sigma_i^2 = \mu_i$ because this we are counting the degrees on each node. To check that this is a valid step a Q-Q plot can be produced for frequencies of a k value over multiple runs. This is shown in figure (3). Figure (3) shows the points lie very close to a straight line which means that the distribution of frequencies of k are Gaussian distributed for a large number of repeats and so this justifies using the chi-squared test. If it is assumed that the theoretical distribution for k is correct, then the p-value from the chi-squared test will give the probability that the value of chi-squared obtained or larger is not by chance.

Numerically, there is only a fixed number of data points for certain k values, hence the KS test will not be used as it only works for continuous distributions. Also, the R^2 test will not be used as it is not good enough to use for comparing data to a theoretical model. The corresponding theoretical values will be calculated using equation (14) and then the chi-squared test will be used to compare answers. Only values of k below the cut-off value will be used because the cut-off is due to finite-sized effects and so does not represent the behaviour of an infinite network.

Table (1) shows the results, the p-values are all one which means that there is a near 100% chance of producing a set of results that deviate away from the theoretical values by at least the deviation of these results. This signifies that there is a very good match between the numerical data and the model.

m	$\chi^2/10^{-5}$	p-value	Critical Value
2	4.9	1.0	30.1
4	6.8	1.0	30.1
8	2.7	1.0	32.7
16	6.9	1.0	35.2
32	2.9	1.0	38.9

Table 1: Data for chi-squared test for preferential attachment.

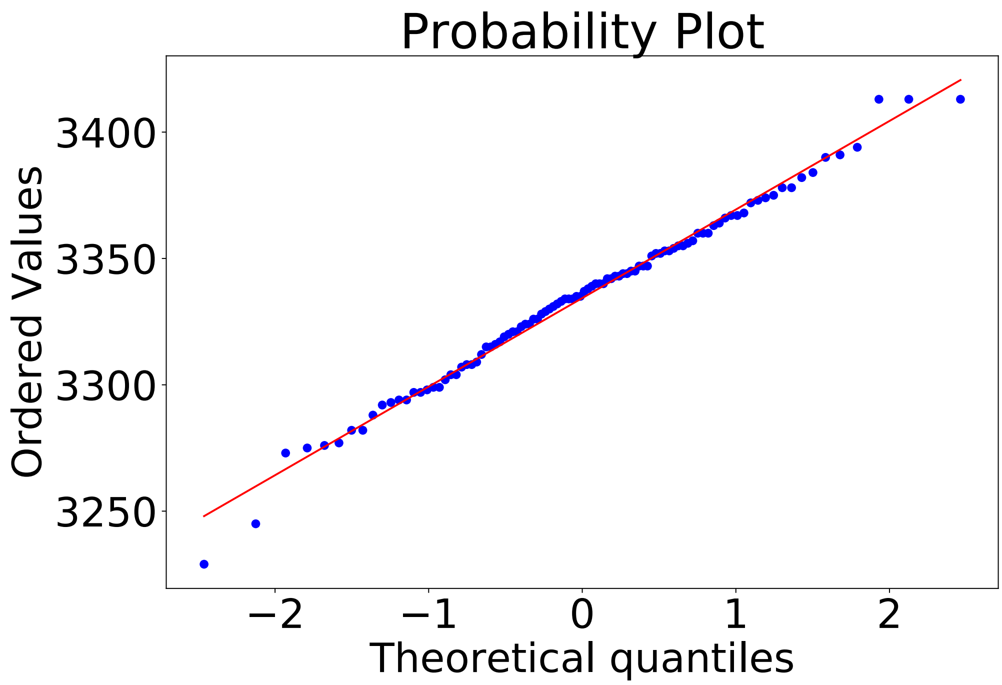


Figure 3: A Q-Q probability plot of the quantiles of the frequencies of $k = 4$ for $m = 4$ over 100 runs against normal distribution quantiles. As the best fit (line in red) is a straight line but not going through $y = x$, the numerical frequency of a k value over multiple runs is Gaussian distributed for a large number of repeats, and so using the chi-squared test is justified.

2.4 Preferential Attachment Largest Degree and Data Collapse

2.4.1 Largest Degree Theory

Defining k_1 to be the largest expected degree, the probability of finding a node with degree greater than k_1 is:

$$p(k \geq k_1) = \frac{1}{N}. \tag{23}$$

As the largest expected degree is an average value and there will only actually be one value which is the largest, the sum of the probability in long time limit of finding a node with degree k from k_1 to ∞ is,

$$\sum_{k=k_1}^{\infty} p_\infty(k) = \frac{1}{N}. \tag{24}$$

The left hand side of equation (24) is the same sum as in equation (15) but from $k = k_1$ instead of $k = m$, so using the result from equation (18) equation (24) becomes,

$$\frac{2m(m+1)}{2k_1(k_1+1)} = \frac{1}{N}. \tag{25}$$

Re-arranging and simplifying this,

$$k_1^2 + k_1 - m(m+1)N = 0. \tag{26}$$

Solving equation (26) using the quadratic formula and discarding the negative solution because it is not possible to have a negative degree in this model,

$$k_1 = \frac{-1 + \sqrt{1 + 4Nm(m+1)}}{2}. \tag{27}$$

2.4.2 Numerical Results for Largest Degree

Fifteen repeats were done for each $N = 10, 100, 1000, 10000, 100000$ for $m = 2, 4, 8$, and then k_1 was averaged over these. To estimate the error on the gradient the standard deviation of data points was used. For large $N \to \infty$,

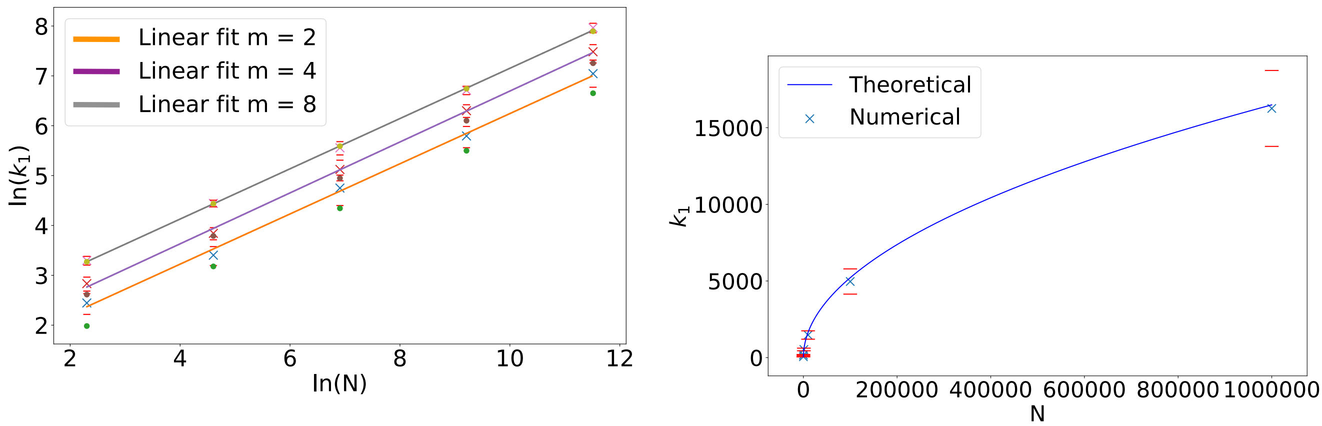
$$\ln(k_1) \sim \ln\left(\frac{\sqrt{4Nm(m+1)}}{2}\right) \sim \frac{1}{2}\ln(N), \tag{28}$$

so the gradient on the logarithmic plot should be around 0.5.

Numerically, a straight line fit will be applied to the logarithmic plots of $\ln(k_1)$ vs $\ln(N)$ to estimate the gradient. This is shown in figure (4(a)). The average gradient, m_1, was found to be,

$$m_1 = 0.50 \pm 0.04 \tag{29}$$

The theoretical value agrees with the numerical value within one standard deviation, also, as shown by figure (4(b)), k_1 grows as $N^{\frac{1}{2}}$ for large N and so this is a good fit.



(a) Logarithmic plots of $\ln(k_1)$ vs $\ln(N)$ for $m = 2, 4, 8$ and are shown with linear fits. The numerical data is marked with an 'x' and theoretical with a dot. This is a straight line in log-space and so there must be a power law. The average gradient was found to be $m_1 = 0.50 \pm 0.04$. Theoretical and numerical match well for higher m values such as for $m = 8$ as can be seen.

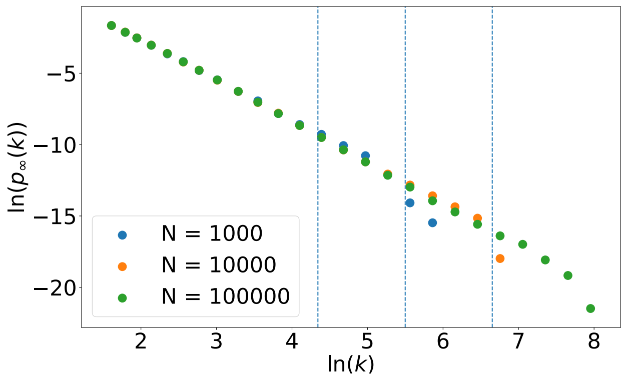
(b) k_1 is plotted against N. The numerical data points follow the same growth as the theoretical data for large N, which is $\sim N^{1/2}$. The error bars increase in size as N increases.

Figure 4: Logarithmic plots are shown.

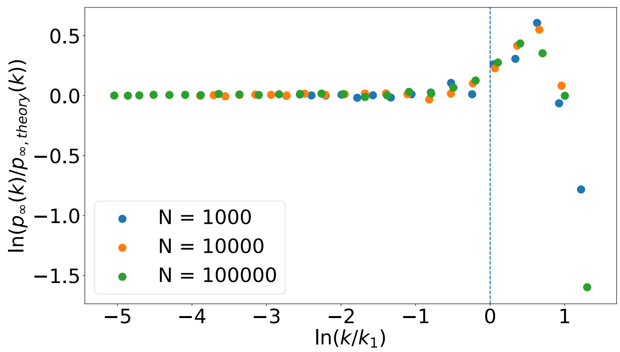
2.4.3 Data Collapse

Keeping $m = 4$ and varying N (from $1000, 10000, 100000$), it can be seen that the cut-off is pushed to higher values of k for larger N and that before the cut-off the same pattern is displayed by the network. Therefore, the value of k_1 will correspond to the cut-off region due to finite sized effects as shown by the blue vertical lines in figure (5(a)).

Using this the x-axis can be transformed by dividing by k_1 and the y-axis can be transformed by dividing by $p_{\infty,theory}(k)$. This will produce a data collapse as it will align the plots as shown in figure (5(b)). Large values of N were used because they are a better approximation to the infinite network and so produce a better data collapse. Figure (5(b)) shows the data collapse is good as the points lie close to each other and trace out a bump. There is a rapid drop after the bump as expected from finite sized effects. The flat end lies around the value of $p_\infty(k)/p_{\infty,theory}(k) = 1$ and the bump begins at a value around $k/k_1 = 1$.



(a) Logarithmic plot of $p_\infty(k)$ vs k is shown for $N = 1000, 10000, 100000$. Vertical blue lines indicate the value of k_1. There is a drop after this value. The same pattern is followed by the network up to the cut-off for all values of N. As N increases the cut-off value of k increases.



(b) Logarithmic plot of $p_\infty(k)/p_{\infty,theory}(k)$ vs k/k_1 is shown for $N = 1000, 10000, 100000$. This is the data collapse for $m = 4$. The points all trace out a similar graph and lie close to each other. There is a flat line initially, leading to a bump. This is followed by a steep drop. The flat end lies around the value of $p_\infty(k)/p_{\infty,theory}(k) = 1$ and the bump begins at a value around $k/k_1 = 1$.

Figure 5: Logarithmic plots are shown.

3 Phase 2: Pure random attachment Π_{rand}

3.1 Random Attachment Theoretical Derivations

3.1.1 Degree Distribution Theory

For random attachment, the probability of attaching to a node of degree k is,

$$\Pi_{rand}(k,t) = \frac{1}{N(t)}, \tag{30}$$

where $N(t)$ is the number of nodes at time t. The original BA model will be altered so that the probability of selecting the node to attach to is given by equation (26). At time t, there are $n_t(k)$ nodes with degree k. The number of nodes with degree $k-1$ that gain an edge to become degree k is $m\Pi_{rand}(k-1,t)n_t(k-1)$, and the number of nodes of degree k which gain an edge to become degree $k+1$ is $m\Pi_{rand}(k,t)n_t(k)$. Then the addition of the new node is simply $\delta_{k,m}$, bringing this altogether the number of nodes with degree k at time $t+1$, $n_{t+1}(k)$, is:

$$n_{t+1}(k) = n_t(k) + m\Pi_{rand}(k-1,t)n_t(k-1) - m\Pi_{rand}(k,t)n_t(k) + \delta_{km}. \tag{31}$$

Substituting in for Π_{rand} ,

$$n_{t+1}(k) = n_t(k) + m\frac{n_t(k-1)}{N(t)} - m\frac{n_t(k)}{N(t)} + \delta_{km}. \tag{32}$$

Then substituting using equation (1),

$$N(t+1)p_\infty(k) = N(t)p_\infty(k) + mp_\infty(k-1) - mp_\infty(k) + \delta_{km} = (N(t)+1)p_\infty(k). \tag{33}$$

Simplifying this,

$$p_\infty(k) = mp_\infty(k-1) - mp_\infty(k) + \delta_{km}. \tag{34}$$

Re-arranging this,

$$p_\infty(k) = \frac{m}{m+1}p_\infty(k-1) + \frac{\delta_{km}}{m+1}. \tag{35}$$

When $k > m$,

$$p_\infty(k) = \left(\frac{m}{m+1}\right)p_\infty(k-1). \tag{36}$$

Then writing out the first few terms,

$$\begin{aligned}
p_\infty(m+1) &= \left(\frac{m}{m+1}\right)p_\infty(m), \\
p_\infty(m+2) &= \left(\frac{m}{m+1}\right)p_\infty(m+1) = \left(\frac{m}{m+1}\right)^2 p_\infty(m), \\
p_\infty(m+3) &= \left(\frac{m}{m+1}\right)p_\infty(m+2) = \left(\frac{m}{m+1}\right)^3 p_\infty(m), \\
p_\infty(m+n) &= \left(\frac{m}{m+1}\right)^n p_\infty(m)
\end{aligned} \tag{37}$$

So from this it can be seen that if we substitute $k = m+n$ then,

$$p_\infty(k) = \left(\frac{m}{m+1}\right)^{k-m} p_\infty(m). \tag{38}$$

When $k = m$, using the fact that the probability of finding a node with degree less than m is zero,

$$p_\infty(m) = \frac{m}{m+1}p_\infty(m-1) + \frac{\delta_{mm}}{m+1} = \frac{1}{m+1}. \tag{39}$$

Therefore, for $k \geq m$,

$$p_\infty(k) = \frac{m^{k-m}}{(m+1)^{k-m+1}}. \tag{40}$$

Theoretical properties to check are that p_∞ sums to one. As summation variable is a dummy variable can substitute $k = j+m$ and then use the geometric sum to get,

$$\sum_{k=m}^{\infty} p_\infty(k) = \frac{1}{m+1}\sum_{j=0}^{\infty}\left(\frac{m}{m+1}\right)^j = \frac{1}{m+1}\left(\frac{1}{1-\left(\frac{1}{m+1}\right)}\right) = 1. \tag{41}$$

Therefore, this is a normalised probability. Now to check the average degree,

$$\langle k \rangle = \sum_{k=m}^{\infty} kp(k \geq m, \infty) = \sum_{k=m}^{\infty} \frac{km^{k-m}}{(m+1)^{k-m+1}} = \frac{(m+1)^{m-1}}{(m)^m} \sum_{k=m}^{\infty} k \left(\frac{m}{m+1} \right)^k. \tag{42}$$

and so using standard summation formulas,

$$\langle k \rangle = 2m. \tag{43}$$

3.1.2 Largest Degree Theory

Defining k_1 to be the largest expected degree, the probability of finding a node with degree greater than k_1 is:

$$p(k \geq k_1) = \frac{1}{N}. \tag{44}$$

As the largest expected degree is an average value and there will only actually be one value which is the largest, the sum of the probability in long time limit of finding a node with degree k from k_1 to ∞ is,

$$\sum_{k=k_1}^{\infty} p_{\infty}(k) = \sum_{k=k_1}^{\infty} \frac{m^{k-m}}{(m+1)^{k-m+1}} = \frac{1}{N}. \tag{45}$$

Evaluating this sum,

$$\sum_{k=k_1}^{\infty} \frac{m^{k-m}}{(m+1)^{k-m+1}} = \frac{(m+1)^m}{(m^m(m+1))} \sum_{k=k_1}^{\infty} \frac{m^k}{(m+1)^k}, \tag{46}$$

Writing out the first few terms of equation (39),

$$\begin{aligned} &\frac{(m+1)^m}{(m^m(m+1))} \left[\left(\frac{m}{m+1} \right)^{k_1} + \left(\frac{m}{m+1} \right)^{k_1+1} + \left(\frac{m}{m+1} \right)^{k_1+2} + ... \right], \\ =&\frac{(m+1)^m}{(m^m(m+1))} \left(\frac{m}{m+1} \right)^{k_1} \left[1 + \left(\frac{m}{m+1} \right)^{1} + \left(\frac{m}{m+1} \right)^{2} + ... \right], \\ =&\frac{(m+1)^m}{(m^m(m+1))} \left(\frac{m}{m+1} \right)^{k_1} \left[\sum_{n=0}^{\infty} \left(\frac{m}{m+1} \right)^{n} \right]. \end{aligned} \tag{47}$$

Then using the geometric sum,

$$\frac{(m+1)^m}{(m^m(m+1))} \left(\frac{m}{m+1} \right)^{k_1} \left[\sum_{n=0}^{\infty} \left(\frac{m}{m+1} \right)^{n} \right] = \frac{(m+1)^m}{(m^m(m+1))} \left(\frac{m}{m+1} \right)^{k_1} (m+1). \tag{48}$$

Therefore,

$$\frac{(m+1)^m}{(m^m)} \left(\frac{m}{m+1} \right)^{k_1} = \frac{1}{N}. \tag{49}$$

Taking the natural logarithm of both sides and re-arranging,

$$k_1 = m - \frac{\ln(N)}{(\ln(m) - \ln(m+1))}. \tag{50}$$

3.2 Random Attachment Numerical Results

3.2.1 Degree Distribution Numerical Results

The same method was used as in section 2.32 to investigate the random attachment model. Visually, the numerical data line up very well with the theoretical data as shown by figure (6). Here there is no finite sized cut-off. The probability drops off smoothly then rapidly. The chi-squared test shows p-values are one which shows that statistically the theory is a good fit for the numerical data.

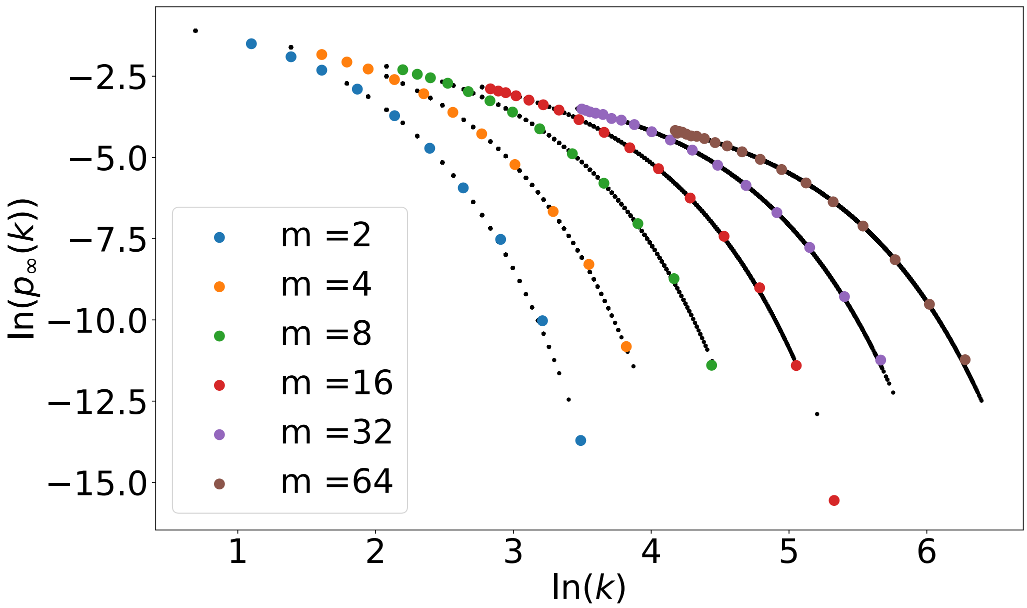


Figure 6: Logarithmic plot of $p_\infty(k)$ vs k for $m = 2, 4, 8, 16, 32, 64$. As m increases the curve shifts to the right. Numerical data lie on same line as theoretical data so visually this is a good fit. Here there is no finite sized cut-off. The probability drops off smoothly then rapidly.

m	$\chi^2/10^{-5}$	p-value	Critical Value
2	6.0	1.0	16
4	2.3	1.0	18
8	3.4	1.0	21
16	1.8	1.0	24
32	1.0	1.0	26

Table 2: Data for chi-squared test for random attachment.

3.2.2 Largest Degree Numerical Results

The same approach was taken as for section 2.4.2, except that k_1 was plotted vs $\ln(N)$ as this will produce a straight line with gradient equal to,

$$m_2 = -\frac{1}{\ln(m) - \ln(m+1)}, \tag{51}$$

and in limit of large N, the $\ln(N)$ term will dominate. The results are shown in table (3), which shows that the corresponding numerical and theoretical gradients are very close to each other. They are all about one standard deviation away from the theoretical values and so this is a good fit.

m	$m_{2,numerical}$	$m_{2,theory}$
2	2.48 ± 0.09	2.47
4	4.53 ± 0.03	4.48
8	8.68 ± 0.09	8.49

Table 3: Data for k_1 for random attachment.

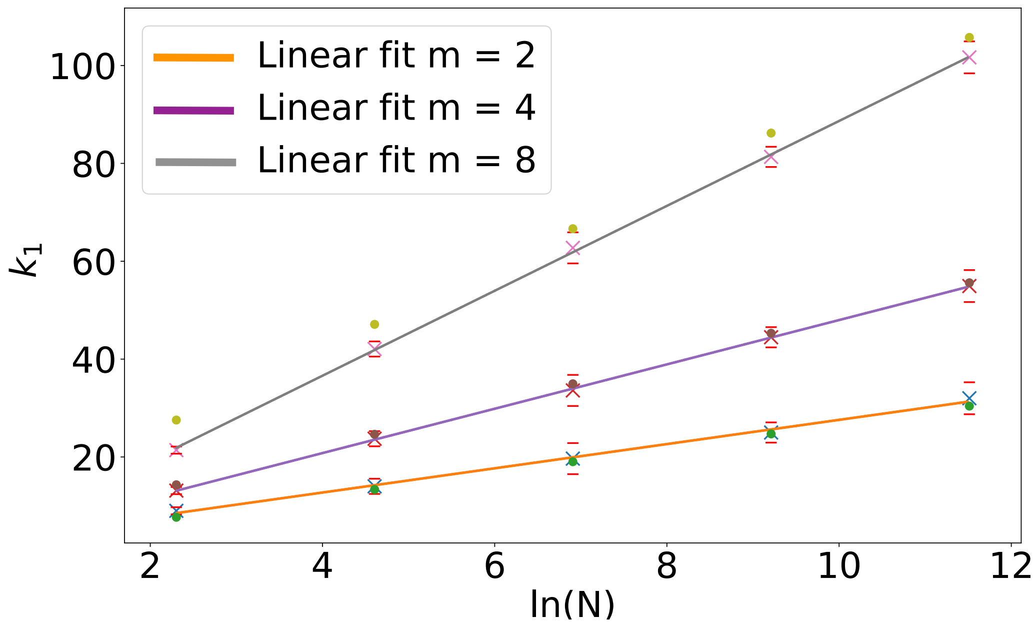


Figure 7: Plots of $k1$ vs $\ln(N)$ for $m = 2, 4, 8$ and are shown with linear fits. The numerical data is marked with an 'x' and theoretical with a dot. The average gradients were 2.48 ± 0.09 for $m = 2$, 4.53 ± 0.03 for $m = 4$ and 8.68 ± 0.09 for $m = 8$. Gradient increases as m increases.

4 Phase 3: Existing vertices model

4.1 Existing Vertices Model Theoretical Derivations

The original BA model will be altered so that sometimes both ends of an edge are added to an existing node. To do this, $r = 0.5m$ of the new edges will be added using the pure preferential attachment Π_{pa}, and the remaining $m - r = 0.5m$ edges will be added with both ends to an existing node with random probability Π_{rand}. However, for self-loops, which are the ones where both ends of one edge are added to an existing node, the node will gain two degrees so $2(m/2) = m$ for the self-loop. Using these facts, the number of nodes with degree k at the next time step can be produced. The number of nodes with degree $k-1$ that gain an edge to become degree k is $0.5m\Pi_{pa}(k-1,t)n_t(k-1) + m\Pi_{rand}(k-1,t)n_t(k-1)$, and the number of nodes of degree k which gain an edge to become degree $k+1$ is $0.5m\Pi_{pa}(k,t)n_t(k) + m\Pi_{rand}(k,t)n_t(k)$. Then the addition of the new node is simply $\delta_{k,0.5m}$, bringing this altogether the number of nodes with degree k at time $t+1$, $n_{t+1}(k)$, is:

$$\begin{aligned} n_{t+1}(k) =& n_t(k) + \frac{m}{2}\Pi_{pa}(k-1,t)n_t(k-1) - \frac{m}{2}\Pi_{pa}(k,t)n_t(k) \\ &+ m\Pi_{rand}(k-1,t)n_t(k-1) - m\Pi_{rand}(k,t)n_t(k) + \delta_{k,m/2}. \end{aligned} \tag{52}$$

Now, $p_\infty(k < 0.5m) = 0$. So for $k > 0.5m$, and using the result obtained in equation (2),

$$\Pi_{pa}(k) = \frac{k}{2mN}, \tag{53}$$

and the result for random attachment from equation (24),

$$\begin{aligned} n(k,t+1) =& n(k,t) + \frac{k-1}{4N(t)} n(k-1,t) - \frac{k}{4N(t)} n(k,t) + \frac{m}{N(t)} n(k-1,t) \\ &- \frac{m}{N(t)} n(k,t) + \delta_{k,m/2}. \end{aligned} \tag{54}$$

Using result form equation (9) and equation (1),

$$n(k,t+1) - n(k,t) = p(k,\infty),$$

in the $t \to \infty$ limit. Equation (54) becomes,

$$p(k,\infty) = \frac{k-1}{4} p(k-1,\infty) - \frac{k}{4} p(k,\infty) + mp(k-1,\infty) - mp(k,\infty) + \delta_{k,m/2}. \tag{55}$$

Factorising for $p(k,\infty)$ and $p(k-1,\infty)$ and re-arranging,

$$\frac{p(k,\infty)}{p(k-1,\infty)} = \frac{k+4m-1}{k+4m+4}. \tag{56}$$

To solve this for $p(k,\infty)$ the Gamma function can be used. The Gamma function defined as:

$$\Gamma(n) = (n-1)! \tag{57}$$

where n is a positive integer which can be complex. One constraint is that the Gamma function must satisfy,

$$\Gamma(z+1) = z\Gamma(z). \tag{58}$$

Using equation (58) a solution can be found for equation (56) in terms of the Gamma function. Propose a solution,

$$p(k,\infty) = C \frac{\Gamma(k+4m)}{\Gamma(k+4m+5)}, \tag{59}$$

where C is a constant to be found. Then,

$$p(k-1,\infty) = C \frac{\Gamma(k+4m-1)}{\Gamma(k+4m+4)}. \tag{60}$$

So,

$$\frac{p(k,\infty)}{p(k-1,\infty)} = \frac{C \frac{\Gamma(k+4m)}{\Gamma(k+4m+5)}}{C \frac{\Gamma(k+4m-1)}{\Gamma(k+4m+4)}}. \tag{61}$$

Now, using equation (58),

$$\Gamma(k+4m-1) = \frac{\Gamma(k+4m)}{(k+4m-1)}, \tag{62}$$

and

$$\Gamma(k+4m+4) = \frac{\Gamma(k+4m+5)}{(k+4m+4)}. \tag{63}$$

Substituting equations (62) and (63) into equation (61),

$$\frac{p(k,\infty)}{p(k-1,\infty)} = \frac{k+4m-1}{k+4m+4}, \tag{64}$$

which is the equation we began with (equation (56)), and so equation (59) is the correct solution. To work out the constant C, set $k = 0.5m$ in equation (55) and as $p(\frac{m}{2}-1,\infty) = 0$,

$$p\left(\frac{m}{2},\infty\right) = 1 - \frac{m}{8}p\left(\frac{m}{2},\infty\right) - mp\left(\frac{m}{2},\infty\right). \tag{65}$$

Re-arranging,

$$p\left(\frac{m}{2},\infty\right) = \frac{8}{9m+8}. \tag{66}$$

Then using equation (59) and inserting the definition of the Gamma function,

$$p\left(\frac{m}{2},\infty\right) = C\frac{(\frac{9m}{2}-1)!}{(\frac{9m}{2}+4)!}. \tag{67}$$

Simplifying this,

$$p\left(\frac{m}{2},\infty\right) = \frac{C}{(\frac{9m}{2}+4)(\frac{9m}{2}+3)(\frac{9m}{2}+2)(\frac{9m}{2})}. \tag{68}$$

Therefore combining equations (67) and (68),

$$\frac{C}{(\frac{9m}{2}+4)(\frac{9m}{2}+3)(\frac{9m}{2}+2)(\frac{9m}{2})} = \frac{8}{9m+8}. \tag{69}$$

Re-arranging this,

$$C = \frac{1}{4(9m+6)(9m+4)(9m+2)(9m)}. \tag{70}$$

So the final form of $p(k,\infty)$ is

$$p(k,\infty) = \frac{(9m+6)(9m+4)(9m+2)(9m)}{4(k+4m+4)(k+4m+3)(k+4m+2)(k+4m+1)(k+4m)}. \tag{71}$$

To check this is correct, evaluate the sum of equation (71) from $k = 0.5m$ to $k = \infty$,

$$\sum_{k=\frac{m}{2}}^{\infty} p(k,\infty) = \sum_{k=\frac{m}{2}}^{\infty} \frac{(9m+6)(9m+4)(9m+2)(9m)}{4(k+4m+4)(k+4m+3)(k+4m+2)(k+4m+1)(k+4m)}. \tag{72}$$

To evaluate this sum, take out the constants and then split the fraction into partial fractions,

$$\frac{1}{24(k+4m+4)} - \frac{1}{6(k+4m+3)} + \frac{1}{4(k+4m+2)} - \frac{1}{6(k+4m+1)} + \frac{1}{24(k+4m)}. \tag{73}$$

Writing out the first few terms from $k = m/2$,

$$\begin{aligned} &\frac{1}{12(9m+8)} - \frac{1}{3(9m+6)} + \frac{1}{2(9m+4)} - \frac{1}{3(9m+2)} + \frac{1}{12(9m)} \\ &\frac{1}{12(9m+10)} - \frac{1}{3(9m+8)} + \frac{1}{2(9m+6)} - \frac{1}{3(9m+4)} + \frac{1}{12(9m+2)} + ... \end{aligned} \tag{74}$$

Summing common denominators beginning with $(9m+8)$,

$$\frac{1}{12(9m+8)} - \frac{1}{3(9m+8)} + \frac{1}{2(9m+8)} - \frac{1}{3(9m+8)} + \frac{1}{12(9m+8)} = 0 \tag{75}$$

Therefore all terms with the term $(9m+n)$ with $n \geq 8$ in the denominator will be zero. This is because they will have the same combination of coefficients, for the same value of $9m+n$ (where n is an integer) in the denominator, which sum to zero,

$$\frac{1}{12} - \frac{1}{3} + \frac{1}{2} - \frac{1}{3} + \frac{1}{12} = 0. \tag{76}$$

Then the surviving terms are,

$$\begin{aligned}
-\frac{4}{12(9m+6)} + \frac{6}{12(9m+6)} - \frac{4}{12(9m+6)} + \frac{1}{12(9m+6)} &= -\frac{1}{12(9m+6)} \\
\frac{1}{12(9m+4)} - \frac{4}{12(9m+4)} + \frac{6}{12(9m+4)} &= +\frac{3}{12(9m+4)} \\
-\frac{4}{12(9m+2)} + \frac{1}{12(9m+2)} &= -\frac{3}{12(9m+2)} \\
\frac{1}{12(9m)} &= +\frac{1}{12(9m)}
\end{aligned} \tag{77}$$

Summing the surviving terms and then using equation (72),

$$\sum_{k=\frac{m}{2}}^{\infty} p(k,\infty) = \frac{4(9m+6)(9m+4)(9m+2)(9m)}{4(9m+6)(9m+4)(9m+2)(9m)} = 1. \tag{78}$$

Therefore this is a normalised probability and the expression is correct.

4.2 Existing Vertices Model Numerical Results

As for previous attachment regimes, the numerical and theoretical $p_\infty(k)$ data was plotted against k for a visual comparison. Visually, figure (8) shows that the numerical data are a good fit to the theoretical data in the middle section. In the beginning, there are numerical fluctuations and towards the end there are finite sized effects. Also, the chi-squared test was used and results shown in table (4). The p-values are 0.99 which is very close to one and so the theoretical model is a good fit for the numerical data.

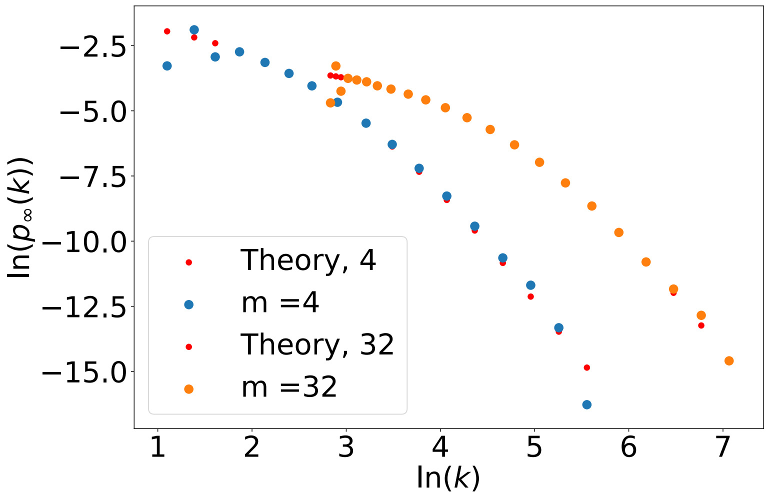


Figure 8: Logarithmic plot of $p_\infty(k)$ vs k for the existing vertices model. Initially, there are fluctuations in the numerical data, then there is a very good visual agreement with the theoretical data toward the middle of the graph. Comparing this to the other attachment regimes it behaves more like the random attachment.

m	χ^2	p-value	Critical Value
2	0.10	0.99	25
4	0.02	0.99	34

Table 4: Data for chi-squared test for existing vertices model.

5 Conclusion

A growing weighted network was simulated for preferential, random and mixed attachment regimes and numerical data was collected. This data was compared to theoretical derivations visually and statistically and a good match was found as well as finite sized effects.

References

[1] http://networksciencebook.com/chapter/5advanced-a-5-13