Networks Project: Growing Network Models

T.D.J du Val de Beaulieu
CID: 00927945

27th March, 2017

Abstract:

Three models were analysed for simple growing networks. The first was the BA model which used preferential attachment to find a random vertex. A theoretical model was derived from the master equation in the limit of infinite vertices. The numerical data was found to fit the data better for a large number of vertices. The theoretical distribution was found to collapse the numerical data until the finite-size cut-off. The maximum degree was found and was used to scale the distribution, collapsing the bumps onto the same line. Furthermore, pure random attachment was also investigated. The assignment of edges was randomly distributed among all vertices. A theoretical model was developed in the infinite limit and a similar analysis was performed as before. The degree distribution was compared between the models. Finally, a random walks model was implemented; this was a continuation of pure random attachment. After finding a random node, a walk was made to neighbouring vertices. The degree distribution was compared, for few and many steps of the walk, to the previous models. The numerical data showed that for a large number of steps the distribution tended to preferential attachment.

Word count: 2495

1 Introduction

The BA model was described by Barabasi and Albert (1999) [1]. It is a simple model for growing networks and is similar to the citation network model of Price (1965). The network grows via preferential attachment and produces the phenomenon of a fat tail. The fat tail network is seen in many areas of research and could relate to the theory that the "rich get richer".

The model is set up in the following way:

1. Set up initial network at time t_0, a graph $\mathcal{G}_0$
2. Increment time $t \to t+1$
3. Add one new vertex
4. Add m edges as follows:
   - Connect one end of the new edge to the new vertex
   - Connect the other end of each new edges to an existing vertex chosen with probability Π
     (a) Preferential attachment: $\Pi_{pa} \propto k$
     (b) Pure random: $\Pi_{rand} \propto \frac{1}{N}$
     (c) Random walks: $\Pi_{rand} \propto \frac{1}{N}$ with a random walk of L steps to a neighbouring vertex
5. Repeat from 2 until reach final number of N vertices in the network

Simple graphs were used throughout the project; therefore, no self-loops, non-weighted and non-directional edges [2]. Also, no repeat edges were allowed. These assumptions have consequences in the maths which will be discussed later. The initial graph ($\mathcal{G}_0$) was complete (each vertex connects to every other vertex). It was initiated with $N(0) = m+1$ nodes, therefore, have the minimum number of degree for each vertex (m) possible to satisfy the boundary conditions of the model ($m \geq k$). There are many ways that the initial graph can be implemented. This method reduced the number of initial nodes allowing the initial graph to be less significant compared to the total number of nodes N.

2 Phase 1: Pure Preferential Attachment

Algorithm:

The algorithm was written using edge list and adjacency list representation. These representations can be stored simultaneously; each has significant benefits for finding different properties of the network. Note, no network package was used because this significantly increases the run time. Also, the model was implemented in C++ to improve speed and,

therefore, provide better statistics. A vertex is chosen proportionately to its degree by randomly selecting an edge from the edge list, then randomly selecting a vertex at one of the ends. Each vertex will be connected to another vertex by an edge. Therefore, each vertex will be in the list k times, where k is the number of times an edge contains that vertex (k is also known as the degree). Because each vertex is in the list k times, the probability of choosing a vertex will be proportionate to the vertex's degree. The adjacency list allows the vertex's degree to be evaluated from the size of the row associated with the vertex.

Various methods were implemented to check the programme conformed to the model:

1. The initial graph contains N vertices, each with $N-1$ degrees.
2. The same vertex cannot be chosen twice when connecting to the new vertex (no repeat edges).
3. The number of edges (excluding the initial graph) will be $N \times m$.
4. Visually checking networks with few N.
5. Comparison to the NetworkX BA Model.
6. The average degree should be approximately $2m$ for large N (every edge has two vertices).

The degrees for each vertex were found using the adjacency list. Each degree was added to a new list. The model was repeated R times, each time appending the list of degrees together. The total length of the degree list was $N \times R$ and, due to their being integer degree numbers, the smallest degree probability was $\frac{1}{NR}$. The parameters required by the programme included the total number of vertices (N), the initial number of vertices (n_0), the number of edges added for a new vertex (m), the number of repeats (R).

Theoretical distribution:

The master equation for the BA model is,

$$n(k,t+1) = n(k,t) + m\Pi(k-1,t)n(k-1,t) - m\Pi(k,t)n(k,t) + \delta_{k,m}. \tag{2.1}$$

This describes the number of nodes with a particular degree at time $t+1$, given the number of nodes at time t.

The theoretical distribution can be found for the degree probability by assuming,

$$p(k,t) = \frac{n(k,t)}{N(t)} \tag{2.2}$$

This can be substituted into 2.1, replacing the number of nodes. The probability that the new edge attaches to a node of with degree k at time t ($\Pi(k,t)$) needs to be replaced by the probability used for the BA model. We assume that the probability of attachment is

proportional to the degree of the node, for this model. Therefore, a new node will more likely attach to a node with a higher degree. The equation is

$$\Pi(k,t) = \frac{k}{2E(t)}. \tag{2.3}$$

This was found by dividing the degree by the sum of all degrees, to find a probability. As every node is connected to another node via an edge, the total number of degrees is twice the number of edges. Although, this only applies for $m = 1$ because we are excluding two edges being the same in the model. For example, the probability of attaching to a node will not be the same for the first attachment to the second, in $m = 2$, because there is one fewer node. Although, as N tends to infinity, 2.3 is a good approximation for the model.

The probability distribution, also requires the assumption,

$$N(t) = \frac{E(t)}{m}. \tag{2.4}$$

This can be proved for large N. Therefore, the number of edges divided by the number of nodes tends to the number of edges added per node. Furthermore, this is independent of the configuration of the initial graph. The initial graph starts with $E(0)$ and $N(0)$. Using the algorithm, the number of edges increases with mt, and the number of nodes increases with t. This can, therefore, be written as

$$\lim_{t\to\infty} \frac{E(t)}{N(t)} = \lim_{t\to\infty} \frac{E(0)+mt}{N(0)+t}. \tag{2.5}$$

After dividing through by t and increasing m to the limit, we find the terms from the initial graph go to zero, leaving m. This is another approximation in the limit of t going to infinity.

Substituting 2.2, 2.3 and 2.4 into the master equation produces,

$$p_\infty(k) = m\frac{k-1}{2E(t)}p_\infty(k-1)\frac{E(t)}{m} - m\frac{k}{2E(t)}p_\infty(k)\frac{E(t)}{m} + \delta_{k,m}. \tag{2.6}$$

This equation can be simplified and rearranged into,

$$p_\infty(k) = \frac{1}{2}[(k-1)p_\infty(k-1) - kp_\infty] + \delta_{k,m}. \tag{2.7}$$

We now take the instance where $k > m$, this can be rearranged into a difference equation,

$$\frac{p_\infty(k)}{p_\infty(k-1)} = \frac{k-1}{k+2}. \tag{2.8}$$

To solve the difference equation, we have to use the properties of the Gamma function.

$$\Gamma(z+1) = z\Gamma(z) \tag{2.9a}$$
$$\Gamma(1) = 1 \tag{2.9b}$$

We need to find the solution to a general difference equation,

$$\frac{f(k)}{f(k-1)} = \frac{k+a}{k+b} \tag{2.10}$$

where the trail solution is,

$$f(z) = A\frac{\Gamma(z+1+a)}{\Gamma(z+1+b)}. \tag{2.11}$$

This equation can be substituted into 2.10,

$$\frac{f(z)}{f(z-1)} = A\frac{\Gamma(z+1+a)}{\Gamma(z+1+b)}\frac{1}{A}\frac{\Gamma(z+b)}{\Gamma(z+a)}. \tag{2.12}$$

Using the properties of a Gamma function (2.9a),

$$\frac{f(z)}{f(z-1)} = \frac{(z+a)}{(z+b)}\frac{\Gamma(z+a)}{\Gamma(z+b)}\frac{\Gamma(z+b)}{\Gamma(z+a)}. \tag{2.13}$$

Canceling the Gamma functions, this shows that the trail solution is a solution to the difference equation proposed,

$$\frac{f(z)}{f(z-1)} = \frac{z+a}{z+b}. \tag{2.14}$$

The solution to the difference equation can be applied to solve 2.8. Comparing 2.14 with 2.8, it is trivial to see that $z = k$, $a = -1$ and $b = 2$. Substituting these values into the solution (2.11),

$$p_\infty(k) = A\frac{\Gamma(k)}{\Gamma(k+3)}. \tag{2.15}$$

Using 2.9a and canceling the Gamma functions,

$$p_\infty(k) = A\frac{\Gamma(k)}{k(k+1)(k+2)\Gamma(k)} \tag{2.16a}$$
$$p_\infty(k) = \frac{A}{k(k+1)(k+2)} \tag{2.16b}$$

The constant A can be found by revisiting an initial assumption, that $k \neq m$. The boundary condition of the model is that $k \geq m$, therefore $p_\infty(k-1) = 0$ when $k = m$. From 2.7, the equation can now be written as,

$$p_\infty(k) = \frac{1}{2}[-mp_\infty] + 1 \tag{2.17a}$$

$$p_\infty(k) = \frac{2}{2+m} \tag{2.17b}$$

Setting 2.16b equal to 2.17b, where $k = m$, we can find A,

$$\frac{A}{m(m+1)(m+2)} = \frac{2}{2+m} \tag{2.18a}$$

$$A = 2m(m+1) \tag{2.18b}$$

The final solution, for the probability distribution of the degree, as $t \to 0$, is

$$p_\infty(k) = \frac{2m(m+1)}{k(k+1)(k+2)}. \tag{2.19}$$

This equation can be checked to adhere to normalisation by integrating the probability distribution from m to infinity; a normalised distribution will integrate to 1. For a discrete probability distribution, the requires a sum.

$$1 = \sum_{k=m}^{\infty} \frac{2m(m+1)}{k(k+1)(k+2)}. \tag{2.20}$$

Using partial fractions,

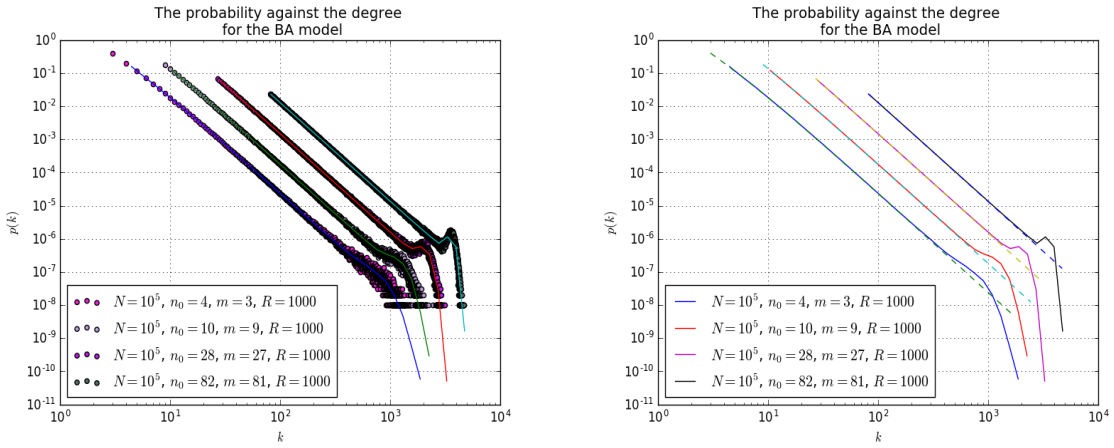
$$1 = 2m(m+1)\left(\sum_{k=m}^{\infty} \frac{1}{2k} + \sum_{k=m}^{\infty} \frac{1}{k+1} + \sum_{k=m}^{\infty} \frac{1}{2(k+2)}\right). \tag{2.21}$$

This results in,

$$1 = \frac{2m(m+1)}{2m(m+1)}. \tag{2.22}$$

This shows that 2.19 is normalised and, therefore, obeys all known constraints on the degree distribution.

Comparison to the numerical simulation:



(a) Raw vs Log binned

(b) Log binned vs Theoretical

Figure 1: The graphs show that the probability of getting a degree decreases linearly on a log-log graph with increasing degree, therefore the probability of finding large measurements is a power law and the distribution is scale-free. The noisy tail was a result of the finite-size of the distribution. As the number of edges added (m) increases, the start of the distribution is higher. This arises from the limitation that $k \geq m$ therefore, there will be a zero probability of k less than m. The distributions are parallel showing that the exponent of the power-law decay is independent of m.

The m values were chosen to give distributions that were separated on a log-log plot. This allowed them to be clearly seen and analysed. The distribution is fat-tailed, where the decay is slower than an exponential. There are a number of issues with these distributions such as fewer data points with larger degrees. This creates poor statistics at the noisy tails, therefore many repeats were performed to improve the number of degrees given to the log binning. The results seem correct because we would expect a finite-size system to have a discrete cut-off for the probability distribution. There will be a finite number of degrees which a vertex can acquire, which is $N - 1$. An expectation number can be calculated for this cut-off which varies with m. The average degree was found to be $2m$ as predicted.

The theoretical distribution is a good fit for small k because small k does not see the finite-size of the system. Large k, has worse statistics and is affected more by the finite-size. The log binned data shows a bump then an exponential decay towards a cut-off. The bump occurs from the initial graph, as these points will have a higher probability for a given degree. The initial vertices are in the network for the longest amount of time and, therefore, will have a greater degree. In am infinite system, the degree of these vertices would increase. The raw data for higher m is much closer to the log binned data. This is because the higher m increases the probability of a vertex getting a higher k. Therefore, there will be a higher probability of getting a larger k value which provides better statistics and reduces the noise.

The KolmogorovSmirnov (KS) test was used to statistically compare the degree distributions. A data set was created using the raw data with the number of points increasing from the minimum k up to a value k. A data set was also generated from the theoretical distribution. The KS statistic describes the confidence with which two data sets can have originated from the same distribution [4]. This is converted into a p value, a measure of the confidence interval. The null hypothesis is that the two samples were drawn from the same distribution. This is rejected if the p value is below a confidence interval.

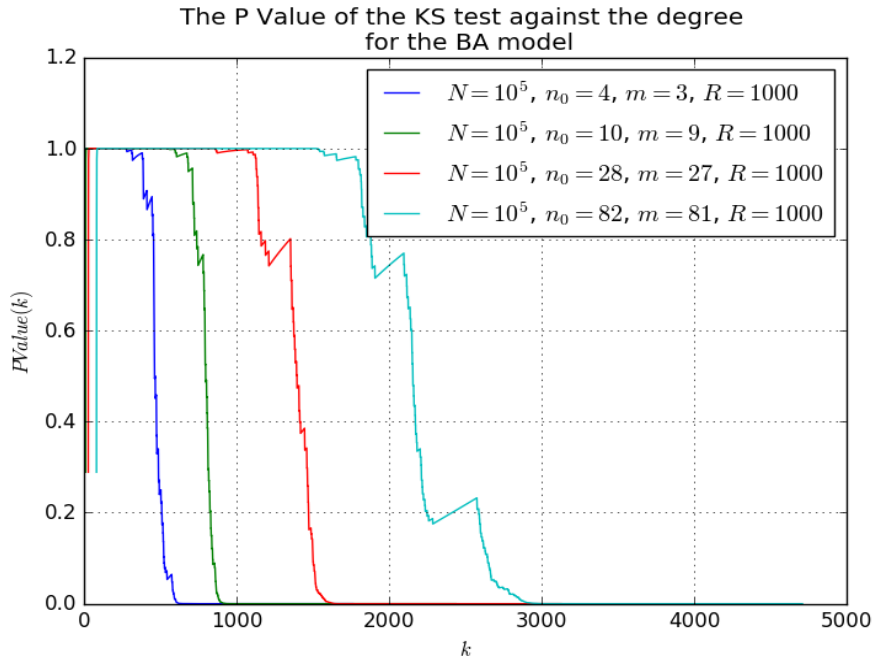


Figure 2: This graphs shows how the p value of the KS test reduces as the number of k values included in the test increases. The KS test compares a data set of the theoretical distribution against the numerical distribution for an increasing number of k. The test shows there is a critical cut-off for a given m where the p value decreases to 0. For k is less than 400, p=1 suggests that we cannot exclude our data sets originating from the same distribution.

Log binning was performed on the degrees. This technique improves the statistic at the end of the distribution where it is noisy [3]. The process creates bins with widths increasing by a factor 'a'. The value of 'a' is chosen to optimise the smoothness of the log binned line while having the most amount of log binned points possible. The bins should never be empty. The bin width 'a' was 1.1 to 1.2, depending on R. There was a limit on the number of values that could be log binned in a reasonable time frame, therefore, this created a significant comprise. Increasing R provided better statistics and smoothed the log binned data much more. Increasing N adhered better to the assumption that N tends to infinity, used for the theoretical results. Visually, large R provided more insight.

Finite size effect:

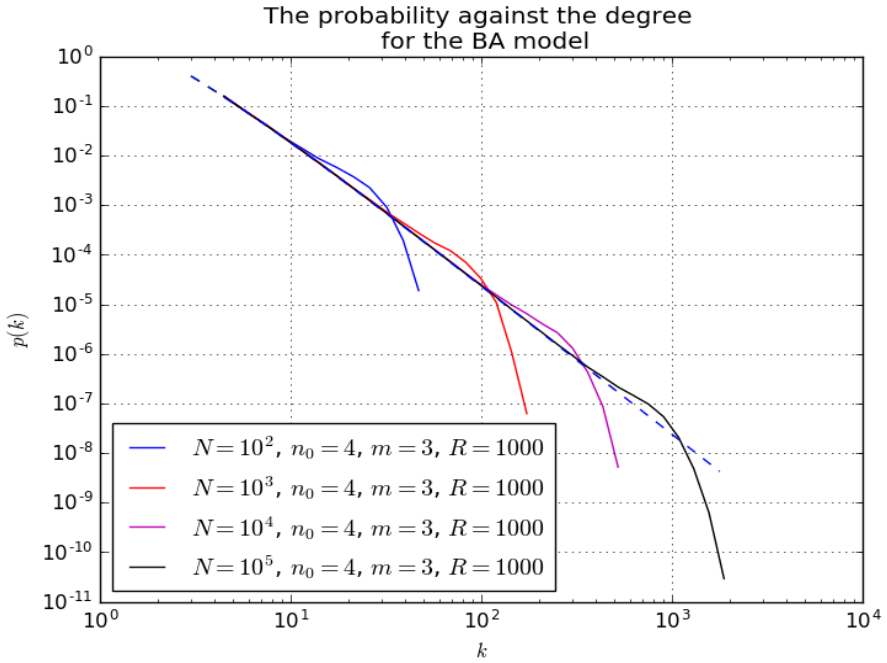


Figure 3: This graphs shows how the degree distribution changes with different N. The degree distribution degrees linearly on

The theoretical expectation of k_1 can be derived from the probability distribution (2.19). This cut-off will be when the distribution expects one node to be found. The expectation is the sum of Np; therefore, summing from k_1 to infinity will equal 1 for a particular value of k_1. The initial equation is,

$$\sum_{k=k_1}^{\infty} Np_{\infty}(k) = 1 \tag{2.23}$$

By substituting 2.19 in and taking N to the other side, this produces,

$$\sum_{k=k_1}^{\infty} \frac{2m(m+1)}{k(k+1)(k+2)} = \frac{1}{N} \tag{2.24}$$

Using partial fractions and algebra, the result of this equation is,

$$\frac{m(m+1)}{k_1(k_1+1)} = \frac{1}{N} \tag{2.25}$$

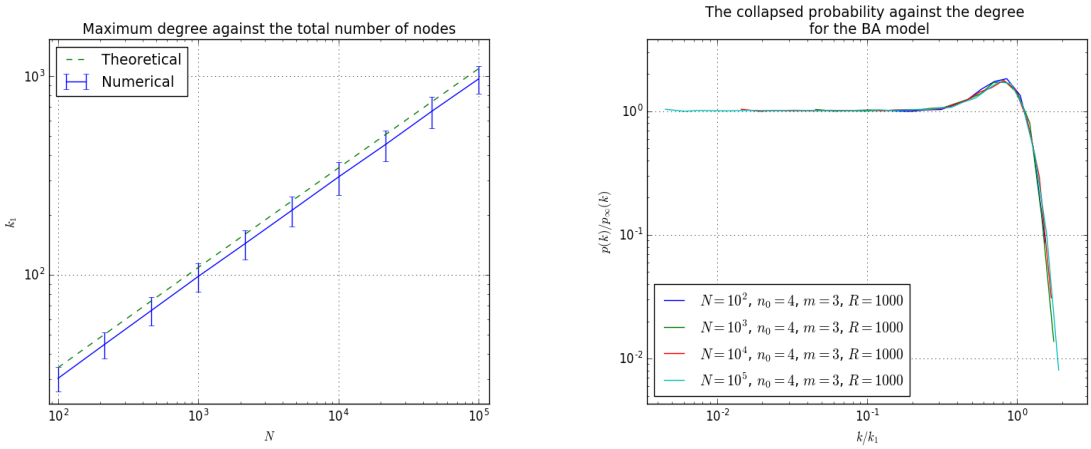
This can be rewritten as,

$$k_1^2 + k_1 - Nm(m+1) = 0, \tag{2.26}$$

which can be solved using a quadratic formula as,

$$k_1 = \frac{-1 \pm \sqrt{1 + 4Nm(m+1)}}{2}. \tag{2.27}$$

The negative answer is unphysical. In the limit of N tending to infinity, k_1 approximately increases as $\sqrt{N}$. Therefore, the exponent of a log(N) against log(k_1) will be approximately 0.5.



(a) k_1 vs N

(b) Data collapse

Figure 4: Graph (a) shows k_1 for varying N. The numerical results were run for 200 repeats with N increasing from 10^2 until 10^5. The last degrees were average to find k_1 and the standard deviation calculated finding σ_{k_1}. The theoretical distribution was also plotted using 2.27. The exponent of the graph was 0.50 ± 0.01. This coincides with the theoretical exponent previously calculated. The maximum error coincided with the theoretical distribution. These errors were the standard deviation not the error on the mean, therefore they are larger than expected for 200 repeats. Graph (b) shows the data collapse using the theoretical distribution and k_1. Each N falls on the same line, therefore, the theoretical distribution and k_1 collapses the numerical data. This provides evidence that the theoretical solution is correct.

Figure 4 shows k_1 for increasing N. The gradient of the log-log plot is 0.50 ± 0.01 which was expected from the theoretical equation (2.27) for large N. The error was found from repeats. The theoretical line is parallel to the numerical data, further showing that they increase with the same exponent. The shifted lines are caused by the tail of the numerical degree distribution falling below the theoretical distribution. This can be seen in Figure 3. The probability is lower than expected at the tail, therefore there will be a lower probability of getting the maximum degree in the numerical data. The degree probability decreases at the same rate, after the bump for each N, thus creating a lower k_1 value. The tails were analysed in detail because this is a property of the finiteness of the system.

In the theoretical distribution, we assume an infinite system, therefore there will be no tail.

The numerical distribution was scaled in the y-axis by the theoretical distribution. Therefore, if the numerical data is the same, there will have a $p(k)/p_\infty(k) = 1$. The numerical data was also scaled in the x-axis by k_1 found previously for each N. Therefore, the cut-off will be $k/k_1 = 1$. The theoretical distribution collapses the data to 1, until the bump, for all N. The bump is caused by the finite-size of the system, thus providing evidence that the theoretical model is the correct model for an infinite sized distribution. Scaling the degrees by k_1 collapses the bumps onto the same point, therefore showing that the finite-size effects are proportionate to the maximum degree found. These are deviations from the infinite time large degree limit of the degree distribution. Higher N collapse slightly closer, showing small corrections to scaling.

3 Phase 2: Pure Random Attachment

Algorithm:

Pure random attachment differs from the BA model by selecting the vertex, that the new edge is attached to, at random. Therefore, each vertex has equal probability of being selected. This was easier to implement by uniformly selecting a random vertex from a list of vertices. Again, no repeat edges could be made, therefore the new edge had to be check if it was already in the edge list.

Theoretical distribution:

The theoretical degree distribution can be derived from the master equation. A different equation needs to be used for $\Pi(k,t)$ because the probability of a new edge connecting with a node is now distributed evenly among each node. Therefore, our new probability is,

$$\Pi(k,t) = \frac{1}{N(t)} \tag{3.1}$$

Again, this is only true for $m = 1$, but we can assume approximately true for all m in the large limit of N. The probability distribution for the degree is now,

$$p_\infty(k)(1+m) = mp_\infty(k-1) + \delta_{k,m} \tag{3.2}$$

Taking the case when $k > m$, the solution can be found by induction (substituting $p_\infty(k)$ back into the equation. This can be repeated until $k = m$.

$$p_\infty(k) = \left(\frac{m}{1+m}\right)^{k-m} p_\infty(m) \tag{3.3}$$

Then $p_\infty(m)$ can be found by setting $k = m$ in 3.2, remembering that the $p_\infty(m-1) = 0$. Therefore, the final equation for the degree distribution is,

$$p_\infty(k) = \frac{1}{m+1}\left(\frac{m}{1+m}\right)^{k-m} \tag{3.4}$$

Normalisation can be check by summing over the the theoretical distribution,

$$1 = \sum_k = m^\infty \frac{1}{m+1}\left(\frac{m}{1+m}\right)^{k-m} \tag{3.5}$$

This simplifies to,

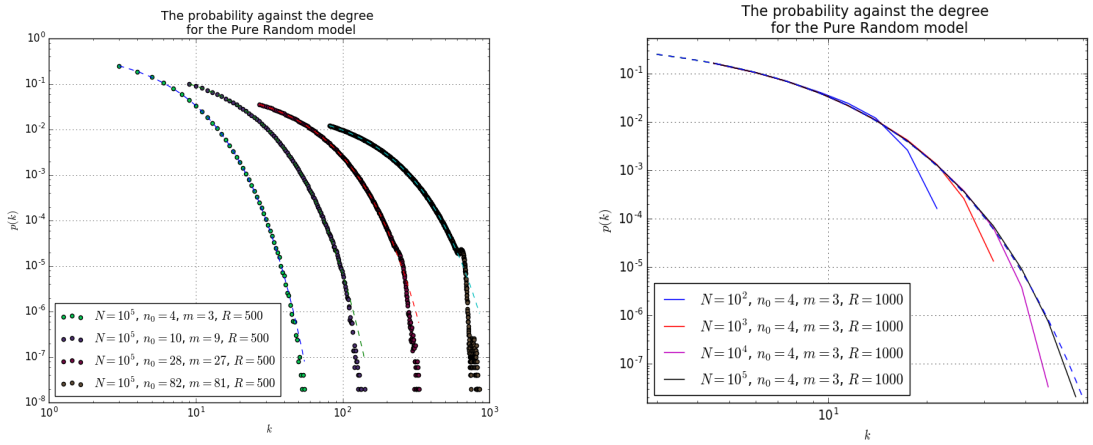
$$1 = \frac{1}{m+1}\left(\frac{m}{1+m}\right)^{-m} \sum_{k=m}^{\infty}\left(\frac{m}{1+m}\right)^{k} \tag{3.6}$$

Using a geometric series, the sum evaluates to,

$$1 = \frac{1}{m+1}\left(\frac{m}{1+m}\right)^{-m} (m+1)\left(\frac{m}{1+m}\right)^{m} \tag{3.7}$$

This shows that the probability distribution is normalised and therefore abides by all the constraints of the distribution.

Comparison to the numerical simulation:



(a) Varying m (b) Varying N

Figure 5: This graph shows the pure random distribution with varying m and N. The degree distribution follows the theoretical distribution with large N following it closer. The distribution is not a fat-tailed distribution, therefore it does not scale with a power law.

Figure 5 shows the numerical data against the theoretical distribution for pure random attachment. The distribution is not fat-tailed. This can be seen from the decay being greater than an exponential. The distribution still contained finite-size effect shown by

small N decaying more rapidly from the theoretical distribution at high k. By inspection, the theoretical distribution was in good agreement with the numerical data for very large N. Increasing, m caused a larger bump for the same reason as preferential attachment. For low k, the theoretical distribution still agrees for different m.

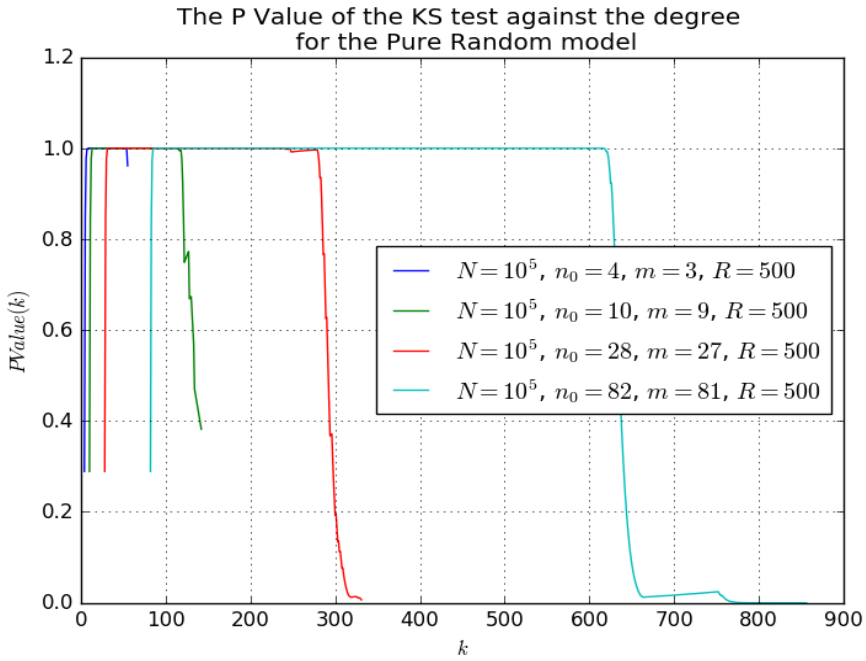


Figure 6: The KS test confirms that the numerical and the theoretical results are a good match. For low m, the test was almost constantly 1, therefore we could not reject the null hypothesis that the data sets are from the same theoretical distribution. This is because the lack of a noisy tail did not skew the data. At high m, the data was noisier therefore the p value decreased to 0 for high k.

Finite size effect:

The expected value for the maximum degree (k_1) can also be calculated for this model. Using the same method used previously, a summation can be made from k_1 to infinity to find the expected value for a vertex. Starting with 2.23 and substituting in 3.7 gives,

$$\frac{1}{m+1}\left(\frac{m}{1+m}\right)^{-m}\sum_{k=k_1}^{\infty}\left(\frac{m}{1+m}\right)^{k}=\frac{1}{N} \tag{3.8}$$

Using a geometric sum, this evaluates to,

$$\left(\frac{m}{1+m}\right)^{k_1}(1+m)=\frac{m+1}{N}\left(\frac{m}{1+m}\right)^{m} \tag{3.9}$$

Canceling the $m+1$, taking the logarithm and rearranging for k_1,

$$k_1 \log\left(\frac{m}{1+m}\right) = -\log N + m \log\left(\frac{m}{1+m}\right) \tag{3.10a}$$

$$k_1 = m - \frac{\log N}{\log m - log(1+m)} \tag{3.10b}$$

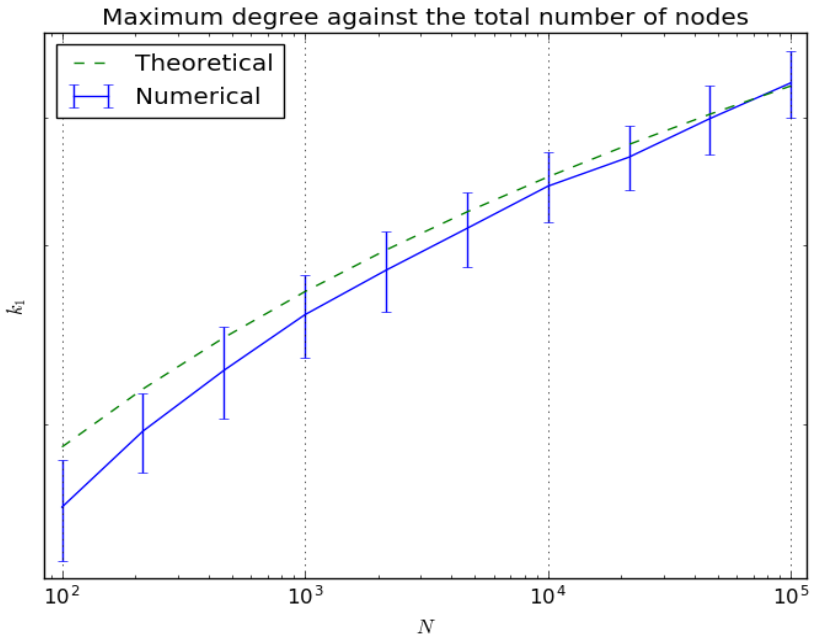


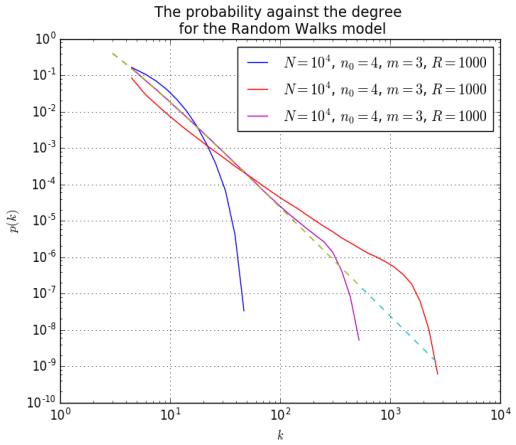
Figure 7: k_1 approaches the theoretical values for large N. This is because the number of the theoretical distribution assumes that the number of vertices is infinite. The end of the probability distribution for the numerical data sticks close to the theoretical distribution for large N. This can be seen in Figure 5. Therefore, the expectation for the cut-off should coincide with the theoretical distribution which it does.

4 Phase 3: Random Walks and Preferential Attachment

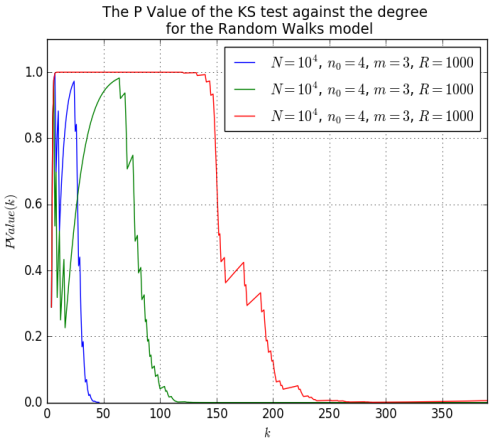
Algorithm:

This model is a continuation from Pure Random. After finding a random vertex, a random walk of L steps is made to neighbouring vertices. The neighbouring vertex can be chosen randomly from the adjacency list of the vertex. This can be repeated L times.

Observations for small and large L:



(a) Varying L



(b) KS test

Figure 8: Graph (a) shows the degree distribution with increasing L. The blue line ($L = 0$) has random attachment because no random walks are taken, therefore the algorithm is the same as before. The red line ($L = 1$) shows preferential attachment shown by an almost power law distribution. The purple line ($L = 100$) approximates the theoretical distribution for pure preferential attachment. Graph (b) shows the KS test for the three different L's. The red line ($L = 100$) clearly shows preferential attachment for $k < 140$. The green line ($L = 1$) is not very consistent with the theoretical distribution for low k. This is shown by the p value not staying at 1 for a large range of k.

The longer the random walk, the more likely you are to end up at a vertex with a higher degree. The model starts by choosing a random vertex with uniform probability. A vertex is then chosen from its adjacency list. The vertices with higher degree are more likely to be connected to a randomly chosen vertex. Therefore, over a large number of steps, there will be a greater probability of increasing the degree of the current vertex. As L tends to infinity, the vertices with the highest k are most probable. Therefore, the vertices chosen will be approximately proportionate to k and the graph will tend to pure preferential attachment. Although, even with $L = 1$, there is some preferential attachment. This is what is seen in Figure 8. At low L, preferential attachment is seen by the graph showing an almost linear gradient on a log-log. At high L, near pure preferential attachment is seen following the preferential theoretical distribution. Therefore, the random walk self-organises to scale-free form [5].

Distributions in real world networks:

This distribution can be interpreted in real-world networks. For example, hyperlinks on a website, such as Wikipedia. A wikipage that is linked to by many other wikipages will have a high degree, where the degree is the number of pages linking to that page. If a wikipage is randomly selected, then a random link is clicked on and a random link is clicked on the next page and so on, the probability of a higher degree will increase with more pages visited. This is because a broader topic will be linked by more wikipages. Therefore, the degree probability distribution of wikipedia pages should follow a fat-tailed distribution. Although, this is not a perfect example because there are both the network is directional

(wikipage pages do not have to link to each other).

5 Conclusion

Three models have been analysed in detail: the Barabasi-Albert model, random attachment and random walk. Each model is an example of a growing network. For the first two models, theoretical distributions have been created for an infinite number of vertices. Both the BA and random model, followed their respective theoretical distributions for large N. The random walks, model showed preferential attachment for $L > 0$. For large L, the numerical distribution tended to the theoretical distribution for pure preferential attachment, derived from the BA model. Statistic analysis was performed on all the distribution using the KS test.

References

[1] T. Evans, *Network Project Student Notes*, Imperial College, London, 2017.

[2] T. Evans, *Networks Lecture Course Notes*, Imperial College, London, 2017.

[3] K. Christensen and N. R. Maloney, *Complexity & Networks: Complexity*, Imperial College Press, London, 2005.

[4] Christopher A. Chung, *Simulation Modeling Handbook: A Practical Approach*, CRC Press, 15 Jul 2003

[5] T. Evans, *Scale Free Networks from Self-Organisation*, Imperial College, London, 2017.